"Federated Learning: Advances, Privacy Mechanisms, and Real-World Deployments"

Prof. Ruksar Fatima Dept. of Computer Science and Engineering Khaja Bandanawaz University	Aliza Mahvash M.tech Student Dept. of Computer Science and Engineering Khaja Bandanawaz University
Ayesha Siddiqua M.tech Student Dept. of Computer Science and Engineering Khaja Bandanawaz University	Syeda Sheeba M.tech Student Dept. of Computer Science and Engineering Khaja Bandanawaz University

ABSTRACT

Federated Learning (FL) has emerged as a powerful paradigm that enables collaborative model training across decentralized clients while preserving data privacy. Instead of aggregating sensitive data in a central server, FL coordinates local training on distributed devices—ranging from smartphones to IoT sensors and institutional servers—and collects only model updates. This design addresses major privacy, ethical, and security concerns associated with centralized data storage. Between 2019 and 2024, extensive research has focused on core FL challenges such as non-IID data distributions, device and system heterogeneity, resource limitations, privacy risks arising from gradient leakage, and practical deployment barriers in fields like healthcare and edge IoT.

In this paper, we review recent advances across four themes: (1) distinctions and best practices for cross-device vs. cross-silo FL; (2) privacy-preserving mechanisms, including differential privacy and secure aggregation; (3) communication- and model-compression techniques for reducing bandwidth usage; and (4) real-world deployments in healthcare and edge-IoT environments. We analyze these works based on efficiency, accuracy, privacy trade-offs, and deployment-level considerations such as resource savings and regulatory alignment.

Our synthesis shows that modern compression techniques—such as quantization, sparsification, and knowledge distillation—can significantly reduce communication costs with minimal accuracy loss, making FL feasible for resource-constrained devices. Privacy mechanisms remain essential for sensitive domains, though they commonly introduce accuracy and utility trade-offs. Cross-silo deployments demonstrate performance close to centralized baselines while maintaining data locality, yet full-scale adoption still depends on standardization, infrastructure readiness, and clearer ROI evidence.

We highlight open gaps such as limited convergence theory for private and compressed FL under non-IID data, lack of unified benchmarks, insufficient empirical ROI studies, and the challenges of scaling FL to large modern models. To support clarity, we provide comparative tables, research-gap matrices, and conceptual diagrams illustrating accuracy-efficiency-privacy trade-offs, followed by prioritized directions for future research.

1. INTRODUCTION

Machine learning (ML) has revolutionized many sectors by leveraging large datasets to train powerful predictive models. Traditional ML workflows typically centralize data from multiple sources into one server or data center for training. However, in domains such as healthcare, finance, mobile devices, and Internet-of-Things (IoT), aggregating large volumes of sensitive or proprietary data at a central location may be infeasible or undesirable due to privacy regulations, institutional policies, or bandwidth limitations [10], [17].

Federated Learning (FL) is an alternative paradigm that addresses these issues by enabling collaborative model training without centralizing raw data. In FL, individual clients hold local data and perform local model updates; a central server aggregates these updates to form a global model, as first formalized in the FedAvg framework [1], [9]. This ensures data remains on-device or within institutional boundaries, reducing privacy risks and legal complexity. FL has since expanded into major application domains including healthcare [18]–[20], mobile devices [5], IoT systems [17], [27], and enterprise cross-silo collaborations [41], [50].

As FL matured between 2019 and 2024, researchers confronted a series of practical challenges that limited naïve FL deployment. Key among these are:

• Data heterogeneity (non-IID data): Client data often follows heterogeneous distributions (e.g., hospitals with different patient demographics, varying sensor behaviors in IoT devices). Non-IID data degrades training stability and model performance, prompting research into specialized optimization techniques and algorithms such as FedProx [21], SCAFFOLD [23], FedBN [36], and theoretical analyses of non-IID impacts [7], [15].

• Resource constraints & communication costs: Many FL clients—mobile devices, low-power IoT sensors, edge devices—have limited computation, memory, and bandwidth. Communication-efficient training methods such as sparsified SGD [8], gradient compression [26], adaptive client participation [60], and lightweight architectures for IoT [55] have been extensively explored.

• Privacy & security risks: Although raw data stays local, gradients and model updates may leak sensitive information, supporting attacks such as membership inference [11], [12] and GAN-based leakage [13]. To mitigate these risks, researchers developed secure aggregation protocols [3], differential privacy approaches [6], [31], [45], homomorphic-encryption-based FL [39], [48], and defenses against poisoning and backdoor attacks [14], [71].

• Real-world deployment challenges: For FL to transition from academic study to practical deployment (e.g., hospital networks, smart cities, industrial IoT), issues such as system architecture, governance, regulatory compliance, personalization, and heterogeneous infrastructure must be addressed. Surveys and system frameworks highlight these deployment considerations [28], [50], [51], [53].

This paper aims to provide a consolidated, up-to-date account of how the FL community tackled these challenges between 2019 and 2024. We review advances in optimization, privacy, compression, communication efficiency, personalization, and system design; summarize real-world applications in healthcare and IoT; analyze trade-offs and gaps; and propose future directions to enable more robust, scalable, and privacy-compliant deployments.

2. BACKGROUND AND FOUNDATIONS

Federated Learning (FL) emerged as a response to the growing inability to centralize sensitive data generated across modern digital ecosystems. Traditional machine learning pipelines assume that data can be freely collected and stored on a central server, but this assumption collapses in real-world domains such as healthcare, finance, mobile devices, and large-scale IoT deployments, where regulations and operational constraints prohibit large-scale data aggregation [10], [17], [19]. FL reverses the centralized learning paradigm by sending the model to the data rather than the data to the model. This decentralized training approach was formalized in early works such as FedAvg [1], [9] and later expanded through large-scale mobile deployments [5], [42], [43].

In this workflow, clients—whether hospitals, banks, smartphones, or sensors—perform local training on private datasets and share only model updates. This enables collaborative learning while maintaining data locality and reducing the risk of exposure [2], [10]. FL commonly operates in two settings: cross-device FL, involving millions of highly resource-constrained and intermittently connected devices [5], [27], and cross-silo FL, involving a small number of reliable organizations with structured governance [41], [50]. Each setting carries distinct assumptions, communication constraints, privacy requirements, and optimization needs that shape system design [28], [51].

Despite its privacy-preserving intent, FL introduces significant technical challenges stemming from real-world data and device heterogeneity. Client datasets are often non-IID, imbalanced, and personalized, causing divergent gradient directions and unstable convergence—phenomena collectively known as client drift [7], [15], [21], [23]. Devices in cross-device FL vary widely in computation, memory, battery life, and network availability, requiring partial participation, asynchronous updates, dynamic sampling, and lightweight model architectures [55], [60]. Communication quickly becomes a bottleneck because FL requires repeated rounds of transmitting model parameters or gradients. Limited uplink bandwidth and large model sizes lead to significant training delays, motivating research into compression strategies such as quantization, sparsification, and pruning [8], [26], [80] as well as adaptive and communication-efficient optimization [30], [35], [56].

To address these constraints, modern FL systems incorporate personalized models [40], [72], hierarchical aggregation structures [29], and efficient client selection mechanisms that improve scalability and robustness [60]. These innovations aim to stabilize convergence under non-IID distributions while reducing both resource consumption and communication overhead.

FL does not inherently guarantee privacy or security; instead, it requires dedicated mechanisms to prevent leakage of sensitive information through transmitted gradients or model deltas. Research has shown that model updates can expose private data through reconstruction, inference, or GAN-based attacks [11]–[13]. Consequently, FL implementations integrate explicit privacy-preserving mechanisms such as differential privacy (DP) [6], [31], [45], secure aggregation [3], homomorphic encryption (HE) [39], [48], and batch-level or task-specific DP techniques for NLP and other domains [32], [68]. Cross-silo deployments further demand strong governance, auditability, and regulatory compliance to build trust between institutions [38], [50], [53], [73]. Beyond privacy, FL systems must defend against a wide range of threats including poisoning attacks, backdoor attacks, Byzantine failures, malicious clients, and colluding adversaries [14], [22], [57], [58], [71]. Addressing these issues requires robust aggregation rules, anomaly detection mechanisms, adversarially resilient optimization, and secure system-level designs [31], [53], [78].

In summary, the foundations of FL rest on three interconnected pillars: distributed optimization, privacy-preserving computation, and system engineering. Each pillar brings its own algorithmic and operational challenges that must be addressed for FL to succeed in large-scale, real-world deployments [2], [28], [51], [54].

Table 1: Comparison of Cross-Device and Cross-Silo Federated Learning

Feature / Dimension	Cross-Device FL	Cross-Silo FL
Number of Clients	Millions of devices (phones, wearables, IoT sensors)	Few institutions (hospitals, banks, companies)
Client Reliability	Low, intermittent, unpredictable	High, stable, synchronized
Compute Capacity	Limited (battery, CPU, memory constraints)	High, institutional-grade servers
Data Characteristics	Highly personalized, extremely non-IID, small datasets	Structured, moderate non-IID, larger datasets
Connectivity	Unstable, variable bandwidth	Stable, high-speed connections
Participation Rate	Very low (1–10% per round)	High (often all participants contribute)
Governance Requirements	Minimal, device-level policies	Strict auditability, compliance, legal regulation
Use Cases	Mobile keyboards, voice assistants, wearables, home IoT	Healthcare imaging, financial fraud detection, enterprise collaboration
System Design Focus	Communication efficiency, resource awareness, sampling	Privacy governance, reliability, stronger security

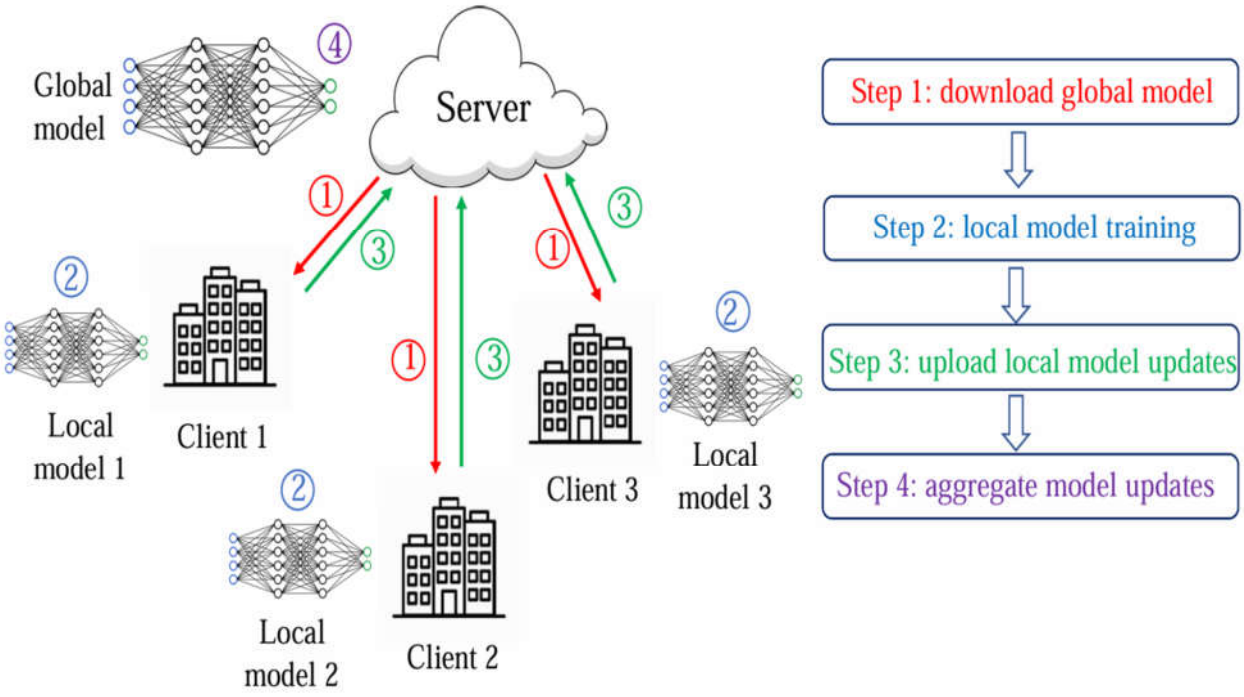

Figure 1: A typical cross-silo FL process

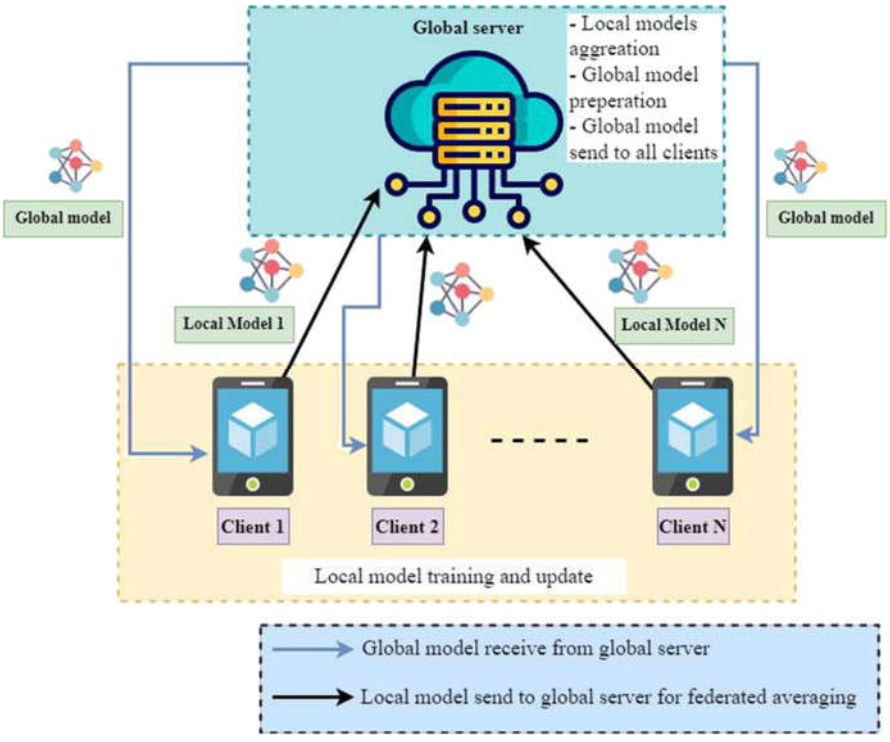

Figure 2: System overview of cross-device federated learning

Table 2: Comparison of federated learning studies

Paper	FL basics	Literature review		Privacy mechanism	Applications			Privacy metrics
		Centralized	Decentralized		NLP	Healthcare	IoT	
Zhang et al. (2021)	✓	✓	✗	✓	✓	✓	✓	✗
Khan et al. (2023)	✓	✗	✗	✗	✓	✓	✓	✗
Li et al. (2023a)	✓	✗	✓	✓	✗	✓	✗	✗
Gabri- elli et al. (2023)	✓	✗	✓	✓	✗	✗	✗	✗
Liu et al. (2024)	✓	✗	✓	✓	✗	✗	✗	✗
Sameera et al. (2024)	✓	✗	✓	✓	✗	✗	✗	✗

3. FEDERATED LEARNING ARCHITECTURES

Federated learning architectures define how distributed clients interact, exchange model updates, and collaboratively converge toward a global objective without sharing raw data. The architectural structure largely determines communication efficiency, model accuracy, scalability, and resilience to failures or adversarial behaviors. Building on the foundational principles of distributed model training, FL architectures are commonly categorized into three primary paradigms: centralized, decentralized, and hierarchical [28], [29], [51]. Each architecture reflects a different coordination topology and is associated with distinct advantages, constraints, and application suitability.

Federated learning enables multiple clients to collaboratively train models using their local data while exchanging only model parameters, as shown in Fig. 1. This paradigm preserves data privacy and supports learning across distributed, heterogeneous, and privacy-sensitive ecosystems [1], [2], [10].

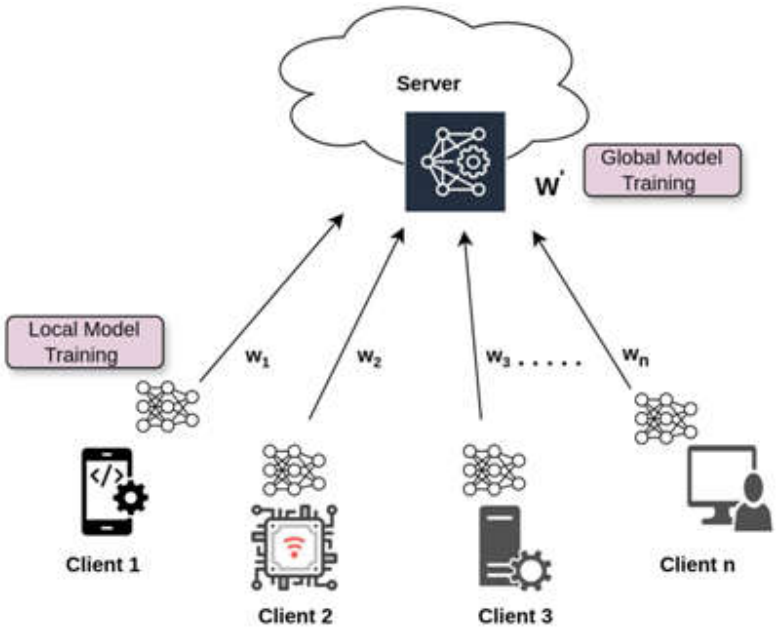

Figure 3: A schematic diagram of federated learning

3.1 Categories of federated machine learning

Figure 2 illustrates the classification of FL based on three key aspects. First, depending on system architecture, FL can be categorized into centralized and decentralized approaches. In centralized FL, a single server orchestrates training using the client–server paradigm, selecting clients, aggregating local updates, and redistributing global model parameters. This is the classic synchronous architecture exemplified by FedAvg [1], [9], and widely deployed in mobile and IoT ecosystems [5], [27]. In decentralized FL, there is no central server. Clients communicate directly with each other, aggregating updates in a P2P fashion using gossip mechanisms or blockchain-assisted coordination [29], [38]. Each client maintains a shared global model and participates in distributed consensus, which eliminates reliance on a central authority.

Second, based on federation scale, FL distinguishes between cross-device and cross-silo settings. Cross-device FL consists of massive numbers of mobile phones, wearables, or IoT devices with highly varied processing capabilities and intermittent connectivity [5], [27], [55]. Cross-silo FL includes a small set of stable, reliable institutions such as hospitals or banks, where devices operate under strong governance and regulatory frameworks [19], [20], [41], [50]. As noted by Gabrielli et al. (2023), cross-device FL must accommodate high device heterogeneity and low participation rates, whereas cross-silo FL benefits from stronger coordination and infrastructure support [51].

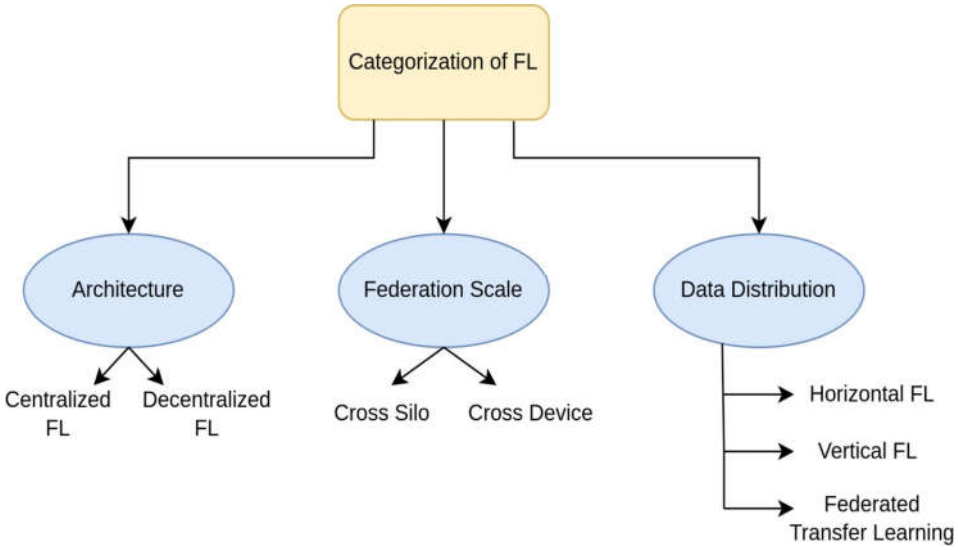

Figure 4: Categories of federated learning

3.2 Centralized Federated Learning Architecture

The centralized architecture is the most widely deployed and conceptually straightforward FL configuration. A single server coordinates client selection, initializes the model, aggregates updates, and distributes global parameters [1], [9], [21]. Clients perform local training and return model updates, which are aggregated typically via weighted averaging (FedAvg). This architecture offers simplicity, predictable synchronization, efficient orchestration, and compatibility with large-scale deployments such as mobile keyboards, wearables, and IoT devices [5], [42], [43].

The central server can also enforce participation rules, validate client updates, maintain metadata, and detect abnormal behavior, providing strong operational control [41], [50]. However, centralized FL suffers from single-point-of-failure risks; the server is a critical component whose failure or compromise disrupts the entire system. Communication bottlenecks arise when many clients attempt to upload updates simultaneously, limiting scalability in large deployments [27], [55].

Moreover, relying on a central aggregator introduces trust and privacy concerns: although raw data remain local, gradient updates can leak sensitive information through reconstruction or inference attacks [11]–[13]. Therefore, centralized FL deployments commonly incorporate differential privacy [6], [31], [45], secure aggregation [3], and homomorphic encryption [39], [48]. To address bandwidth limitations, models may employ compression or sparsification techniques [8], [26], [80].

3.3 Decentralized Federated Learning Architecture

Decentralized FL eliminates reliance on a central server by enabling direct client-to-client communication. Updates are exchanged using peer-to-peer protocols, gossip-based aggregation, or blockchain-based consensus mechanisms [29], [38]. Each client maintains its own version of the global model and synchronizes parameters with selected peers. This architecture improves fault tolerance since no single entity controls or compromises the learning pipeline. It is particularly suitable for edge IoT networks, vehicular systems, sensor grids, and distributed industrial ecosystems where authority is distributed or connectivity is dynamic [27], [52].

Blockchain-enhanced decentralized FL further strengthens transparency and verifiability by maintaining immutable update logs and supporting secure, auditable aggregation [38], [73]. Clustered topologies, peer selection strategies, and cryptographic protocols enhance robustness against adversarial manipulation [14], [31], [53].Nonetheless, fully decentralized FL is more complex to manage. Achieving consensus among highly diverse participants requires sophisticated algorithms that may increase communication cost. The absence of a central coordinator can lead to inconsistent updates if peers exchange parameters asynchronously. Gossip-based exchange reduces communication frequency but introduces slower convergence. In systems with significant data heterogeneity, clients may diverge from the global objective if peer selection is suboptimal. Despite these challenges, decentralized FL remains a powerful architecture for privacy-preserving and resilient learning systems.

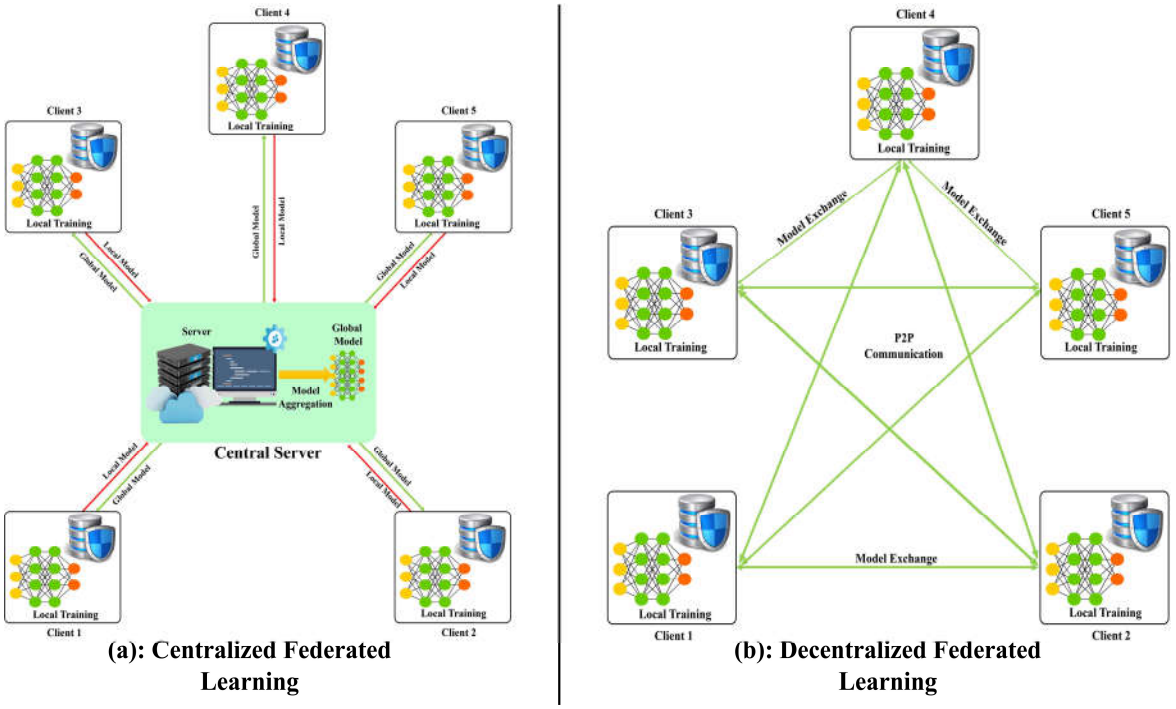

Figure 5: Network Structure of Federated Learning

3.4 Federated Learning Workflow

Federated learning systems typically follow a four-step iterative training cycle [1], [9], [21]:

(1) Client Selection: The server or coordinator identifies eligible clients based on availability, network quality, computational resources, and policy constraints. Cross-device FL typically selects a small subset of clients due to limited device availability, whereas cross-silo FL often involves full participation [41], [50].

(2) Model Broadcasting: The global model is initialized and distributed to clients. Initial weights may be random or pre-trained using public or weakly-private data [10], [42], [43].

(3) Local Training: Clients train the model using their private datasets and compute gradient updates or weight deltas. Local training is affected by data heterogeneity, resource constraints, and personalization needs [7], [15], [72].

(4) Aggregation and Update: The server aggregates client updates—most commonly via weighted averaging, but many adaptive and robust aggregation strategies exist (e.g., FedProx [21], SCAFFOLD [23], adaptive optimizers [30], and robust aggregators [57], [58], [71]). The updated global model is then redistributed to clients for the next round.

4. PRIVACY MECHANISMS IN FEDERATED LEARNING

Privacy protection lies at the core of federated learning because, although raw data remain on-device or within institutional boundaries, model updates can still leak sensitive information. Numerous studies have demonstrated that gradients and shared model parameters can reveal private data through inference, reconstruction, or generative attacks [11]–[13]. As a result, modern FL systems integrate multiple privacy techniques—information obfuscation, encrypted computation, and trustworthy aggregation—to create a layered and defense-in-depth privacy strategy [2], [31], [45], [53].

4.1 Differential Privacy (DP)

Differential Privacy (DP) protects individuals by adding calibrated statistical noise to model updates so that the presence or absence of any specific data point minimally affects the final model output [6]. In FL, DP is most commonly applied on the client side, ensuring that the update is privacy-preserving before it leaves the device. DP optimization must account for privacy budgets, dataset sizes, and training frequency, balancing the trade-off between privacy and model utility [31], [32].

Advanced techniques extend DP for domain-specific tasks (e.g., NLP) or batch-level training to maintain model performance while enhancing anonymity [32], [68]. However, stronger privacy budgets can degrade model accuracy, requiring adaptive noise scaling strategies depending on device constraints and data heterogeneity.

4.2 Secure Aggregation

Secure aggregation ensures that the server cannot see any individual client update; instead, it only receives the aggregated sum of all contributions. The seminal secure aggregation protocol proposed by Bonawitz et al. masks client updates with pairwise random values that cancel out during aggregation, enabling confidentiality even in large-scale deployments [3].

This method is particularly effective in cross-device environments with many participants because individual masked updates reveal no meaningful information even if partially compromised. Secure aggregation provides strong defenses against insider threats on the server, preventing fine-grained analysis or reconstruction of client-specific gradients [31], [53].

4.3 Homomorphic Encryption (HE)

Homomorphic Encryption (HE) enables computation directly on encrypted data. In FL, clients encrypt their updates using HE schemes before transmitting them to the server, which then aggregates encrypted values without learning their contents [39]. Only a designated key holder—or a secure enclave—decrypts the aggregated model.

Although HE offers strong privacy guarantees, it introduces computational overhead, especially in mobile and IoT devices. Research shows that practical FL deployments often rely on optimized or partially homomorphic variants to maintain energy efficiency while preserving security [48].

4.4 Trusted Execution Environments (TEE)

Trusted Execution Environments (TEEs) provide hardware-isolated regions where sensitive computations can occur securely. Instead of encrypting every client update, a TEE allows the aggregation process to take place within a secure hardware enclave, protecting intermediate states from external access [53], [73]. TEEs offer high computational speed compared with heavy cryptographic methods; however, they depend on specialized hardware, raise trust assumptions, and may be vulnerable to hardware-level side-channel attacks.

4.5 Hybrid Techniques

Modern federated systems increasingly adopt hybrid privacy approaches—combining DP, secure aggregation, HE, and TEEs—to balance computational cost with strong privacy guarantees. Studies emphasize that multi-layered mechanisms significantly reduce leakage risks, provide resilience against multiple attack vectors, and maintain acceptable model utility in large-scale deployments [31], [53], [54]. These hybrid designs represent the current trend in privacy-preserving FL, especially in highly regulated domains such as healthcare, smart cities, and finance [18], [19], [49].

5. MODEL COMPRESSION AND COMMUNICATION EFFICIENCY

Federated learning repeatedly exchanges model updates across distributed clients, making communication one of the most expensive components of the entire training pipeline. A large body of work in distributed optimization and FL demonstrates that reducing update size—through sparsification, quantization, pruning, or alternative representations—significantly improves scalability, training speed, and energy efficiency [8], [26], [80]. Communication efficiency is especially critical in cross-device FL, where clients often have limited bandwidth, intermittent connectivity, and resource constraints [27], [55].

5.1 Quantization

Quantization compresses model updates by reducing numerical precision. Transforming 32-bit floating-point gradients into lower-bit representations (e.g., 8-bit or 4-bit) can cut communication cost by more than half with minimal impact on accuracy [26]. Dynamic and adaptive quantization techniques, introduced in communication-efficient FL studies, adjust precision based on gradient importance or update variance to help maintain convergence even under severe non-IID conditions [8], [56].

5.2 Pruning and Structured Compression

Pruning removes parameters that contribute little to model performance.

• Unstructured pruning eliminates individual weights, providing high compression but requiring specialized hardware to efficiently support sparse matrix operations.
• Structured pruning removes entire channels, filters, or blocks, resulting in models that are significantly smaller and optimized for deployment on edge devices and IoT systems [27], [55].

Compression surveys in FL highlight pruning as an essential strategy for reducing both communication overhead and client-side computation, especially when training deep models across resource-constrained devices [80].

5.3 Gradient Sparsification

Gradient sparsification transmits only the most informative updates—such as top-K gradients—or accumulates residuals to be sent later, drastically reducing the number of values communicated per round. Pioneering work in sparsified SGD shows that aggressively dropping gradients maintains accuracy while reducing bandwidth consumption by orders of magnitude [8]. Federated compression surveys confirm that sparsification is one of the most effective communication-effectiveness methods for large-scale FL with thousands of clients [80].

5.4 Knowledge Distillation

Knowledge distillation enables clients to exchange only prediction probabilities or logits instead of full model parameters. In FL, this can significantly shrink communication cost because lightweight student models learn from a shared or centralized teacher model without exchanging high-dimensional gradients. Early FL-specific distillation methods such as FedKD demonstrate competitive model performance while reducing communication size substantially [46]. Distillation is particularly valuable for IoT and edge devices with limited compute and memory resources [27], [55].

5.5 Low-Rank Approximation

Low-rank approximation compresses large matrices—such as weight matrices in fully connected layers—into products of smaller matrices. This reduces the number of parameters transmitted and the computation required per update. Communication-efficient learning studies and FL compression surveys highlight matrix factorization as a powerful technique for reducing model footprint without significantly affecting expressiveness [80]. Low-rank decompositions remain especially beneficial in bandwidth-constrained or latency-sensitive FL environments.

6. REAL-WORLD APPLICATIONS OF FEDERATED LEARNING

Federated learning has progressed from theoretical research to deployment in real-world environments where privacy, distributed data, and regulatory constraints converge. Its ability to train models collaboratively without sharing raw data makes it uniquely suited for domains such as healthcare, IoT, edge computing, transportation, and wearable technologies [17], [19], [27], [33], [49].

6.1 Remote Patient Monitoring and Wearables

Wearable devices continuously capture personal biomedical signals such as heart rate, respiration, stress patterns, and sleep quality. Because this information is deeply sensitive, centralized data aggregation raises significant privacy concerns. Federated learning allows wearables to collaboratively improve models for arrhythmia detection, fall prediction, and personalized health insights while keeping raw sensor data on-device [34], [37], [62], [77]. Studies in FL for mobile and wearable platforms show that on-device training preserves user privacy, reduces network usage, and enhances personalized model performance [5], [34].

6.2 Industrial IoT

Industrial IoT environments rely on large numbers of distributed sensors to monitor machine vibration, temperature, pressure, production metrics, and energy consumption. Sharing this information with cloud servers may leak proprietary manufacturing processes. Federated learning enables local sensors and edge devices to collaboratively train models for predictive maintenance, equipment diagnostics, anomaly detection, and quality assurance without exposing confidential data [17], [27], [52], [63].

Federated approaches in smart factories and industrial automation have been shown to reduce downtime, improve fault detection, and enhance reliability by enabling continuous learning at the edge [24], [47], [69].

6.3 Healthcare and Medical Diagnostics

Healthcare organizations must comply with strict regulatory standards that prohibit the sharing of patient records. However, collaboration across institutions is essential for building accurate and generalizable medical models. FL enables hospitals to jointly train models for medical imaging, disease prediction, and clinical decision support without exchanging sensitive health data [18], [19], [20], [33], [76].

Cross-hospital FL studies demonstrate improvements in diagnostic accuracy, robustness across demographic distributions, and compliance with privacy regulations while maintaining high-quality clinical performance [18], [20], [33].

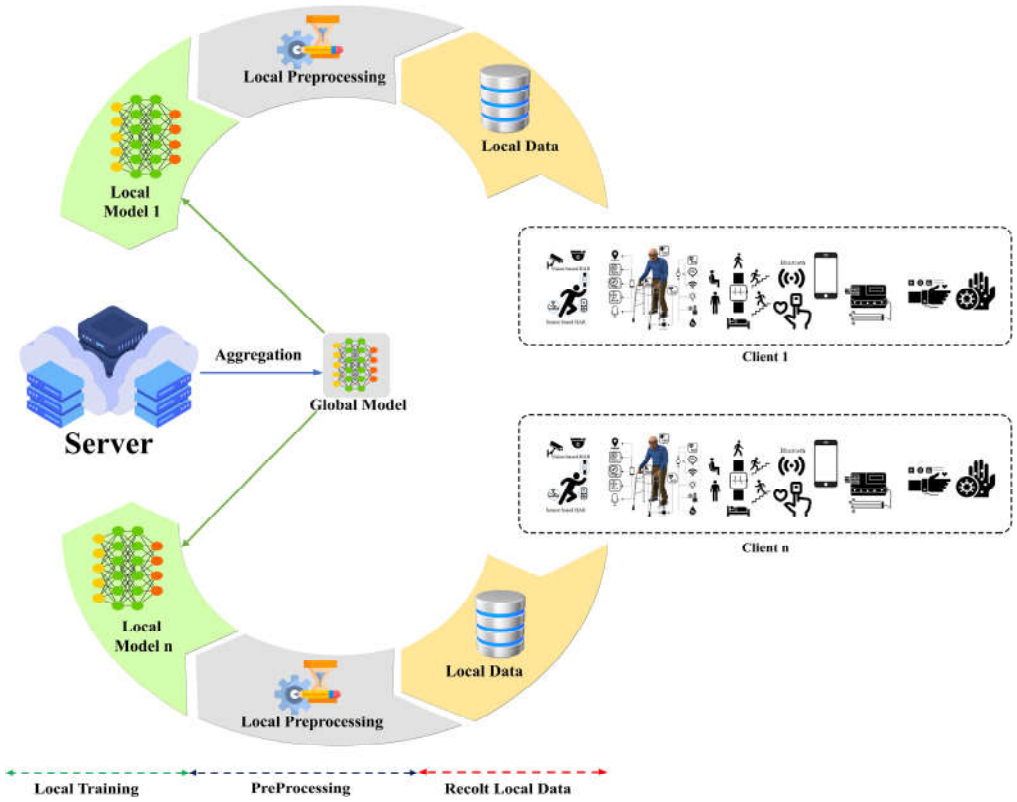

Figure 6: Healthcare application system based on FL

6.4 Edge-Based Transportation Systems

Modern vehicles generate vast amounts of data through cameras, radar, LiDAR, GPS, and onboard diagnostics. Sending raw sensory data to centralized servers raises safety, privacy, and bandwidth concerns. Federated learning allows vehicles and roadside edge units to collaboratively train perception, navigation, and traffic prediction models without exposing raw driving data [25], [70], [75].

FL-based transportation systems support real-time traffic flow prediction, autonomous driving enhancements, and environment understanding, while keeping sensitive driving traces confined to the vehicle [25], [75].

7. CHALLENGES AND OPEN PROBLEMS

Even with major advancements, federated learning (FL) continues to face significant obstacles before it can be reliably deployed at scale. These challenges stem from real-world system constraints, heterogeneous data, security vulnerabilities, and privacy risks that must be addressed to ensure stable and trustworthy FL deployments [2], [28], [51], [53].

7.1 Heterogeneity of Data and System

Federated learning operates across highly diverse devices and organizations that differ in computation, energy capacity, network bandwidth, storage, and availability. This system heterogeneity causes uneven participation, stragglers, variable training speeds, and inconsistent contributions to the global model [5], [27], [41], [55].

Data heterogeneity—also known as statistical heterogeneity—creates even greater challenges. Clients generate data of different sizes, formats, distributions, and contexts. Such non-IID variability leads to divergent gradient updates, training instability, biased global models, and difficulties in convergence [7], [15], [21], [23], [36], [65]. These differences can cause the model to overfit to dominant or frequently participating clients.

7.2 Data Availability

In large-scale FL systems, client participation is voluntary and conditional. Devices must be powered, connected, idle, and willing to contribute. This leads to unpredictable data availability, with many clients frequently offline or inactive [5], [41], [55]. As a result, only a small and potentially non-representative subset of clients participates in each training round, reducing diversity and weakening the generalization ability of the global model. Dynamic client sampling and availability-aware scheduling remain open research challenges [60].

7.3 Data Distribution (Non-IID Data)

FL inherently faces non-IID data distributions, as each client collects data from unique contexts and user behaviors. Labels, features, and sample frequencies differ across clients, causing conflicts between local objectives and the global optimization goal [7], [15], [21], [23], [36].

Consequences include:

- slower convergence,
- unstable model updates,
- client drift,
- biased or unfair global models,
- reduced generalization across user populations.

A wide range of methods—such as FedProx, SCAFFOLD, FedBN, FedAlign, and personalized FL—attempt to mitigate these issues, but no universal solution has emerged [21], [23], [36], [65], [72].

7.4 Communication Overhead

FL requires continuous communication of model parameters or gradients between clients and servers, creating significant bandwidth, energy, and latency constraints. Deep learning models often contain millions of parameters, making communication the primary bottleneck [1], [8], [26], [80].

As the number of clients grows, communication demands escalate rapidly. Low-bandwidth networks, intermittent connectivity, and power-constrained devices worsen the problem. Research in compression (quantization, pruning, sparsification), adaptive communication schedules, and client selection aims to reduce this burden [26], [8], [35], [56], [60].

7.5 Security Issues

Although FL prevents raw data sharing, the system remains vulnerable to a variety of security threats. Because the server cannot easily verify the correctness or intent of client updates, adversaries may perform:

• poisoning attacks, injecting malicious gradients to corrupt or manipulate the global model [14], [57], [58], [71]

• backdoor attacks, embedding hidden behaviors into the model during training [14]

• Byzantine failures, where compromised clients send arbitrary or adversarial updates [57], [58]

• model manipulation attacks, where attackers interfere with the update or aggregation process [31], [53]

In decentralized FL architectures, the attack surface expands further, as there is no central authority to monitor or validate updates [29], [38]. Balancing strong security with computation efficiency remains an open challenge.

7.6 Privacy Protection

Even though data remain local, model updates may leak sensitive information through model inversion, membership inference, or gradient reconstruction attacks [11], [12], [13]. Thus, privacy protection remains essential. Changes in model details might reveal user habits or even allow reconstructing original data through methods like model inversion or membership checks. To guard against this, techniques such as differential privacy, secure multi-party computation, homomorphic encryption, or trusted hardware must be integrated into the system.

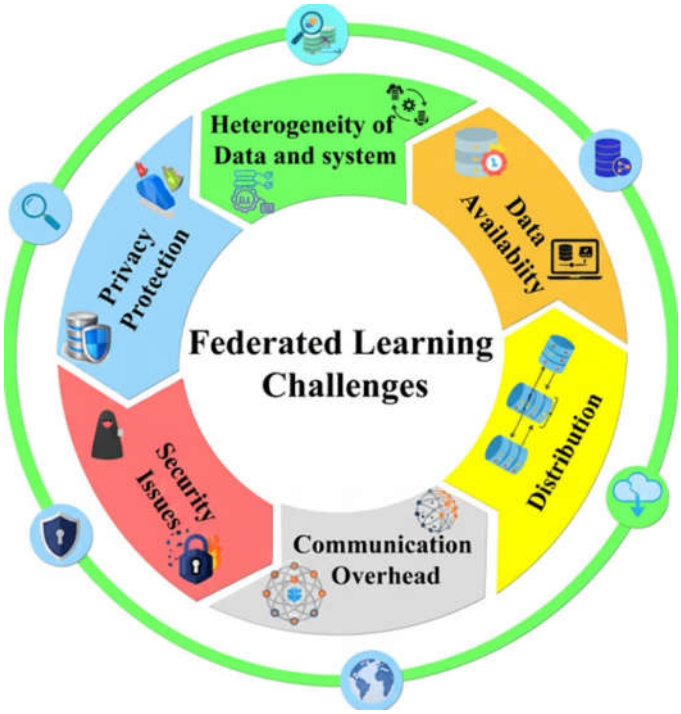

Figure 7: FL Challenges

8. FUTURE DIRECTIONS

Figure 8 illustrates how federated learning can enhance online learning, resource optimization, and privacy-preserving data exchange in smart buildings by enabling localized model training. This highlights FL's ability to manage diverse, continuous IoT data streams efficiently while reducing privacy risks by keeping raw data on-device [17], [27], [49], [52]. FL also aligns naturally with digital twin systems that rely on privacy-sensitive, cross-device collaboration [67]. However, several challenges must be addressed—such as limited communication capacity, the need for differential privacy (DP), encryption, and robustness to varying data distributions—before such systems can scale effectively [26], [31], [45], [47], [55].

Future FL deployments in smart environments will require thoughtful system design to ensure scalability through efficient communication, adaptive learning strategies, and strong privacy-preserving mechanisms.

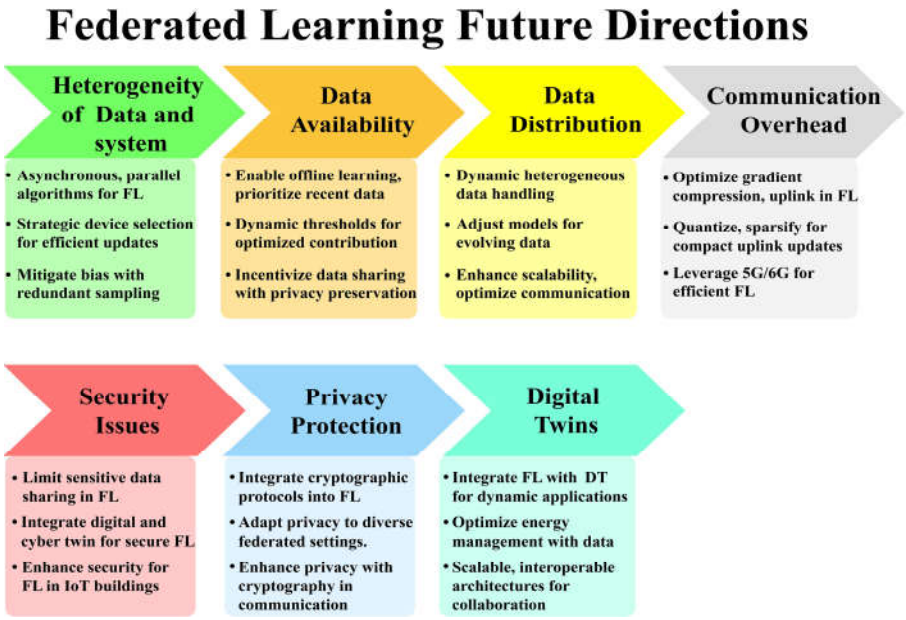

Figure 8: Future FL Directions

8.1 Personalized Federated Learning

A major direction for FL is personalized federated learning, which aims to produce client-specific models rather than a single global model. Personalization techniques include:

- Personalization layers [40]
- Clustered or group-based personalization [44]
- Meta-learning and adaptive optimization [30], [72]

Personalized FL mitigates non-IID challenges by tailoring models to individual user preferences and data distributions, leading to greater accuracy and fairness across clients [7], [23], [36], [65].

8.2 Federated Learning Without Central Servers

Decentralized FL eliminates central servers entirely. Clients exchange updates peer-to-peer or through consensus mechanisms inspired by blockchain technology [29], [38], [73]. This direction aims to: improve resilience to single-point failures, enhance trust in multi-stakeholder environments, enable FL in highly dynamic edge and IoT networks [52] Decentralized architectures will be crucial for applications where no single entity can or should control the entire learning process.

8.3 Unified Privacy-Performance Frameworks

Future FL systems must strike an optimal balance between privacy guarantees and model performance. Research trends point toward unified, layered privacy frameworks that combine:

- Differential Privacy (client- or server-side) [6], [31], [45], [68]
- Secure Aggregation [3]
- Homomorphic Encryption [39], [48]
- Trusted Execution Environments [53], [73]

Hybrid designs that automatically adjust privacy levels based on model sensitivity, device capacity, and task requirements are expected to become standard in large-scale deployments [54].

8.4 Ultra-Light FL Models for Micro-IoT

As federated learning expands into micro-IoT environments—smart homes, environmental sensors, wearables, and embedded medical systems—models must be extremely energy-efficient, low-memory, and communication-aware [55], [62], [77]. Techniques such as: lightweight architectures, aggressive compression [26], [80], distillation-based FL [46], energy-aware learning [47], [69] will enable FL on tiny edge devices with milliwatt-level power budgets.

8.5 FL in 5G/6G Edge Networks

Next-generation networks will dramatically enhance FL's potential. 5G and 6G provide ultra-low latency, high bandwidth, and massive device connectivity—ideal conditions for real-time collaborative learning [24], [70], [79]. These improvements will enable FL in: autonomous driving and intelligent transportation [25], [70], [75], remote and robotic healthcare [18], [62], augmented/virtual reality and edge intelligence [24], [79] Future communication-learning integration will allow FL systems to adapt dynamically to network constraints, enabling continuous, real-time model improvement across billions of devices.

9. REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," Proc. AISTATS, 2017.

[2] P. Kairouz et al., "Advances and open problems in federated learning," Found. Trends Mach. Learn., 2021.

[3] K. Bonawitz et al., "Practical secure aggregation for privacy-preserving machine learning," Proc. ACM CCS, 2017.

[4] T. Li et al., "Federated optimization in heterogeneous networks," MLSys, 2020.

[5] A. Hard et al., "Federated learning for mobile keyboard prediction," arXiv:1811.03604, 2018.

[6] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," NIPS Workshop, 2017.

[7] J. Wang et al., "Addressing non-IID data in federated learning," IEEE TPAMI, 2021.

[8] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning with sparsified SGD," NeurIPS, 2019.

[9] J. Konečný et al., "Federated optimization: Distributed machine learning for on-device intelligence," arXiv:1610.02527, 2016.

[10] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," ACM TIST, 2019.

[11] S. Truex, L. Liu, M. E. Gursoy, L. Yu, and W. Wei, "Demystifying membership inference attacks," IEEE TDSC, 2019.

[12] R. Shokri and V. Shmatikov, "Membership inference attacks against machine learning models," IEEE S&P, 2015.

[13] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the GAN: Information leakage in collaborative learning," ACM CCS, 2017.

[14] E. Bagdasaryan et al., "How to backdoor federated learning," AISTATS, 2020.

[15] Y. Zhao et al., "Federated learning with non-IID data," arXiv:1806.00582, 2018.

[16] V. Smith et al., "Federated multi-task learning," NeurIPS, 2018.

[17] Y. Luo et al., "Survey on federated learning for IoT," IEEE Internet Things J., 2021.

[18] J. Xu et al., "Privacy-preserving federated learning for healthcare," npj Digit. Med., 2022.

[19] N. Rieke et al., “Federated learning in medicine,” npj Digit. Med., 2020.

[20] M. Sheller et al., “Federated learning in medicine: Decentralized training for tumor segmentation,” Med. Image Anal., 2020.

[21] X. Li et al., “FedProx: Mitigating heterogeneity in federated learning,” MLSys, 2020.

[22] A. Reisizadeh et al., “Robust federated learning: Convergence of FedAvg,” IEEE Trans. Signal Process., 2020.

[23] S. P. Karimireddy et al., “SCAFFOLD: Stochastic controlled averaging,” ICML, 2020.

[24] M. Chen et al., “Distributed learning in 5G/6G networks,” IEEE Commun., 2021.

[25] O. Marfoq et al., “Federated learning for autonomous vehicles,” IEEE Trans. Intell. Transp. Syst., 2021.

[26] Y. Lin et al., “Deep gradient compression,” ICLR, 2017.

[27] S. Roy et al., “Federated learning in smart cities and IoT,” IEEE Internet Things J., 2021.

[28] X. Li et al., “A systematic survey of federated learning,” ACM Comput. Surv., 2022.

[29] C. Briggs, Z. Fan, and P. Andras, “Federated learning with hierarchies,” arXiv:2004.11760, 2020.

[30] M. Khodak, M. Balcan, and A. Talwalkar, “Adaptive gradient methods in federated learning,” ICML Workshop, 2019.

[31] L. Lyu et al., “Privacy and robustness in federated learning,” IEEE Security & Privacy, 2020.

[32] Y. Huang et al., “DP-FL for NLP tasks,” ACL, 2020.

[33] S. Raman et al., “Federated learning for medical imaging,” IEEE Trans. Med. Imag., 2021.

[34] S. Ramaswamy et al., “Federated learning for keyword spotting,” InterSpeech, 2019.

[35] X. Chen et al., “Communication-efficient personalized federated learning,” NeurIPS, 2020.

[36] J. Li et al., “FedBN: Tackling feature shift,” arXiv:2102.07623, 2021.

[37] D. Sui et al., “FedED: Federated energy disaggregation,” AAAI, 2020.

[38] C. Thapa et al., “Blockchain-based federated learning: A survey,” IEEE Access, 2021.

[39] M. Kim et al., “Secure FL with homomorphic encryption,” IEEE TDSC, 2021.

[40] K. Wei et al., “Personalization layers in federated learning,” ICML Workshop, 2020.

[41] S. Park et al., “Large-scale cross-silo federated learning,” IEEE BigData, 2021.

[42] Google AI, “Production federated learning at Google,” 2019.

[43] Apple, “Federated analytics: Privacy-preserving telemetry,” Apple ML Journal, 2020.

[44] Y. Chen et al., “Clustered federated learning,” NeurIPS, 2022.

[45] C. Wu et al., “Local differential privacy in federated learning,” IEEE TKDE, 2021.

[46] Z. Xu et al., “FedKD: Knowledge distillation in FL,” arXiv:2006.07529, 2020.

[47] S. Zhao et al., “Energy-efficient FL for edge computing,” IEEE Trans. Mobile Comput., 2022.

[48] X. Luo et al., “HE acceleration for FL,” IEEE Access, 2021.

[49] Z. Jiang et al., “FL for smart healthcare IoT,” Sensors, 2020.

[50] H. Zhang et al., “Cross-silo FL frameworks,” ACM TIST, 2021.

[51] L. Khan et al., “Review of FL architectures,” Future Gener. Comput. Syst., 2023.

[52] R. Li et al., “Federated edge learning in IIoT,” IEEE Trans. Ind. Electron., 2023.

[53] L. Gabrielli et al., “Security and privacy in federated systems,” ACM Comput. Surv., 2023.

[54] H. Liu et al., “Emerging trends in privacy-preserving FL,” IEEE S&P, 2024.

[55] S. Sameera et al., “Lightweight FL for IoT,” IEEE Internet Things J., 2024.

[56] S. J. Reddi et al., “Adaptive federated optimization,” ICLR, 2020.

[57] K. Pillutla et al., “Robust aggregation for FL,” ICML Workshop, 2019.

[58] P. Blanchard et al., “Byzantine-robust learning,” NeurIPS, 2017.

[59] T. Zhou et al., “FedMix: Mixup in federated learning,” arXiv:2103.13652, 2021.

[60] M. Luo et al., “Dynamic client selection,” IEEE TNNLS, 2022.

[61] M. Ye et al., “FL for financial fraud detection,” IEEE BigData, 2022.

[62] S. Sun et al., “FL for remote patient monitoring,” IoT Analytics J., 2023.

[63] H. Yu et al., “FL for smart grids,” IEEE Trans. Smart Grid, 2020.

[64] J. Hamer et al., “FedBoost,” NeurIPS Workshop, 2020.

[65] A. Reisizadeh et al., “FedAlign,” arXiv:2108.11544, 2021.

[66] C. He et al., “FedML research library,” arXiv:2007.13518, 2021.

[67] N. Rodriguez-Barroso et al., “FL for digital twins,” ACM IoT, 2023.

[68] C. Zhang et al., “Batch-level differential privacy in FL,” arXiv:2012.00844, 2020.

[69] J. Kwon et al., “Energy-aware FL for edge,” IEEE Trans. Green Comput., 2022.

[70] T. Yang et al., “Over-the-air federated learning,” Proc. IEEE, 2021.

[71] R. Doku et al., “Robust FL against poisoning attacks,” IEEE TIFS, 2022.

[72] Y. Chen et al., “FedFomo: Personalized FL,” AAAI, 2021.

[73] A. Wainakh et al., “Auditable federated learning,” IEEE S&P, 2022.

[74] X. Ma et al., “FL with graph neural networks,” IEEE TKDE, 2023.

[75] Z. Zhang et al., “FL for autonomous driving,” IEEE Intell. Vehicles, 2022.

[76] A. Jochems et al., “FL for ICU mortality prediction,” Sci. Rep., 2020.

[77] A. Banerjee et al., “Cross-platform FL for wearables,” IEEE Sensors J., 2023.

[78] W. Chai et al., “FedAT: Adversarial training in FL,” IJCAI, 2021.

[79] M. Ferreira et al., “FL in 6G-enabled smart cities,” IEEE Commun., 2024.

[80] J. Duan et al., “Survey of compression in FL,” ACM Comput. Surv., 2022.

[81] D. Shenoy, R. Bhat, and K. Krishna Prakasha, “Exploring privacy mechanisms and metrics in federated learning,” Artif. Intell. Rev., vol. 58, p. 223, 2025.