Federated Learning: A Comprehensive Review of Algorithms, Optimisation Frameworks, and Real-World Applications

Pranjal Sharma[1], Anamika Rangra[2]

Emails: [1]pranjal202522@gmail.com, [2] rangra.anamika@gmail.com

[1,2]Himachal Pradesh Technical University, Hamirpur, India

Abstract. Federated Learning (FL) has become an important method for collaborative machine learning while keeping data private and following rules. Unlike traditional centralized learning, FL lets multiple clients—like mobile phones, hospitals, banks, or IoT devices-train a shared model together without sharing their raw data. This helps handle data ownership, separate data storage, and privacy regulations. This review provides a comprehensive, structured, and citation-rich synthesis of the field aimed at researchers and graduate students. We trace the evolution of FL from early distributed optimisation and distributed SGD to modern privacy-preserving, communication-efficient, and personalised federated systems. The paper systematically covers core algorithms (FedAvg, FedProx, SCAFFOLD, adaptive optimisers, and personalisation methods), systems-level aspects (client scheduling, compression, stragglers, communication protocols), and privacy/security foundations (differential privacy, secure aggregation, adversarial robustness). We also survey evaluation practices, benchmarks, application case studies, and emerging domains such as federated reinforcement learning and FL for large language models. The review synthesises open problems related to heterogeneity, personalisation, scalability, privacy–utility trade-offs, and realistic benchmarking. We conclude by outlining concrete research directions and offering practical recommendations for deploying FL in real-world systems.

Keywords: Federated Learning, Privacy-Preserving Learning, Non-IID Data, Secure Aggregation, Differential Privacy, FedAvg, Personalisation.

1. Introduction

Federated Learning (FL) is a collaborative training approach where many clients learn a shared global model without sending their raw data to a central server. Each customer just contributes to the updates, which are used to train the model locally. First introduced by McMahan et al. (1), the FedAvg system popularized the idea of "train locally, combine encyclopedically."

The introduction of sequestration companies and stringent data-protection laws like HIPAA and GDPR are directly linked to the rise of FL. These rules restrict how sensitive data is stored and transferred between businesses. FL naturally meets these requirements by allowing common model training while preserving data on initial bias. Checks by Yang et al. (7) and Kairouz et al. (5) indicate that nonsupervisory pressure and sequestration enterprises are

IICTDS 2025, December 19-20, Chandigarh, India
Copyright © 2026 EAI
DOI 10.4108/eai.19-12-2025.2366213

significant factors driving the expansion of FL. It is evident how this differs from earlier approaches. In centralized learning, all information is gathered on a single server, raising security and privacy risks. In fully decentralized learning, control and data are spread, yet it is difficult to maintain a single global model. A balanced solution is offered by federated learning, where data remains local and a central server only incorporates model updates to provide a shared model. This approach aligns with frameworks investigated by Li et al. [6] and system designs described by Bonawitz et al. [2].

These days, FL is an essential part of modern AI systems. It is challenging to upload the constant personal data produced by billions of mobile and Internet of Things devices. Sensitive information must be kept private in industries like banking and healthcare. Federated learning facilitates extensive collaboration in these contexts. Financial applications [18], medical collaboration [16], and communication-efficient edge systems [17] have all shown its advantages.

Interest in FL has grown quickly in research and industry. Surveys such as Kairouz et al. [5] and Li et al. [6] summarize the wide range of algorithms, optimization methods, and privacy techniques. Newer methods like FedProx [8], FedMA [9], and SCAFFOLD [10] tackle issues such as system differences, personalisation, and better optimization.

This review builds on these earlier works. It presents a clear and updated synthesis of core FL methods, optimisation strategies, privacy and security foundations, practical system challenges, and important application domains. The goal is to help readers understand the evolution of FL, its current state, and the emerging directions shaping its future.

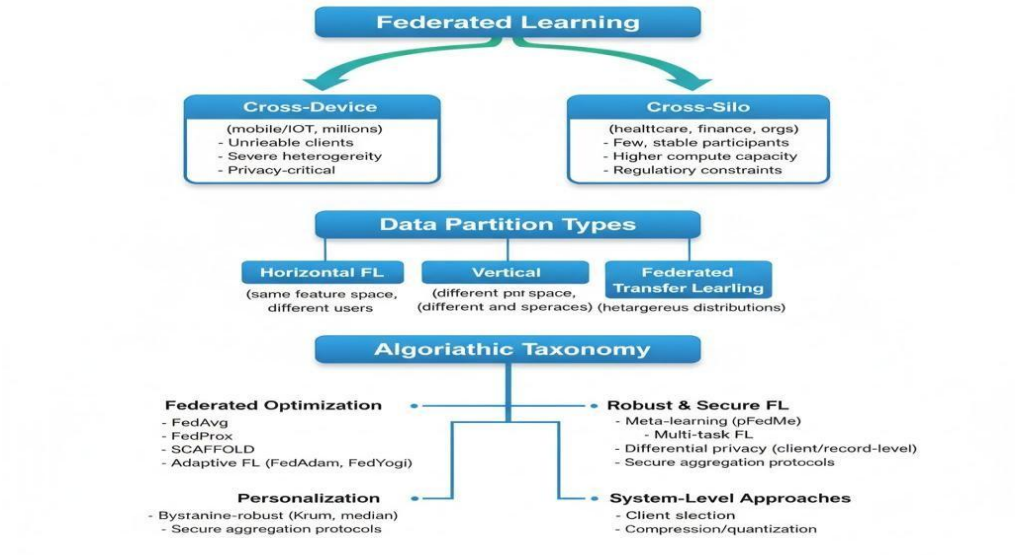


Fig 1. Federated Learning taxonomy

2. Historical Background and Evolution

The foundations of Federated Learning (FL) come from earlier work in distributed optimisation and large-scale machine learning. Before FL was formally introduced, distributed training depended on systems like the Parameter Server. These systems split data and computation across workers while a central server collected updates. They enabled efficient large-scale learning, but they required frequent data or gradient transfers, leading to high communication costs and privacy concerns. Ideas from distributed optimisation, especially techniques formalised by Boyd et al. **[3]**, provided much of the theoretical groundwork.

Research on distributed stochastic gradient descent (SGD) also pushed the field forward. Early techniques investigated ways to reduce communication costs, parallel mini-batch training, and asynchronous updates. Techniques including gradient compression, sparsification, and local updates were suggested to improve efficiency. However, as FL surveys [5], [6], and [7] later pointed out, these approaches usually made the assumption that data could be freely exchanged or divided across machines—an assumption that does not hold true in situations that are sensitive or severely constrained.

Concurrently, the privacy research community created methods like homomorphic encryption and safe multi-party computation. Work by Shokri and Shmatikov [4] and secure aggregation methods like Bonawitz et al. [13] show that several parties could collaborate on learning tasks without sharing their raw data. Although these methods provided strong privacy protection, they were usually too expensive or slow for large-scale, continuous machine learning.

These research areas focused on the basic limitations of classical distributed learning, such as reliance on centralized data, high communication requirements, and the need for reliable datasharing agreements. They weren't the best for situations when data is stored on millions of personal devices or in organizations with severe regulations. Studies like Kairouz et al. [5] and Li et al. [6] explain these shortcomings and how they led to the need for a new paradigm.

In response, modern federated learning was developed. In 2017, McMahan et al. introduced the FedAvg algorithm **[1]** and demonstrated that models could be trained directly on user devices, with only aggregated updates sent back to a central server. This work showed that collaborative learning was possible without collecting raw data. It also demonstrated that privacy-preserving training could operate efficiently at scale, especially in mobile settings.

The shift from traditional distributed optimisation to FL reflects changing priorities. Earlier systems focused mainly on speeding up computation for large datasets. FL, on the other hand, is designed for data that cannot move, data that is highly personal, and data stored across diverse and unreliable devices. The failures of earlier distributed approaches—particularly around privacy, communication cost, and data accessibility—directly motivated the development of federated learning **[1], [5], [6], [7]**.

3. FL Problem Formulation and Taxonomy

Federated Learning (FL) aims to train a shared global model using data stored across many clients. Suppose there are K clients. Each client $k$holds a local dataset $D_k$drawn from its own data distribution $P_k(x, y)$. These local distributions are often different across clients, making FL a non-IID and heterogeneous learning problem, as noted in major surveys **[5], [6], [7]**.

The global optimisation goal in FL is:

$$\min_w F(w) = \sum_{k=1}^{K} \frac{|D_k|}{|D|} F_k(w), \quad (1)$$

where $F_k(w)$Is the local objective for the client k. A common form is:

$$F_k(w) = \frac{1}{|D_k|} \sum_{(x,y)\in D_k} l(w; x, y), \quad (2)$$

with $l(\cdot)$ Representing the loss function. Each client updates the model on its own data and sends parameters or gradients to a central server. The server aggregates these updates, following the principles introduced in FedAvg **[1]**.

FL deployments generally fall into two categories. **Cross-device FL** involves millions of mobile or IoT devices that are unreliable and hold small, highly diverse datasets **[1], [5]**. **Cross-silo FL** involves a small number of stable clients—such as hospitals, banks, or enterprises—that manage large and sensitive datasets **[6], [7]**.

FL can also be classified by how feature spaces and sample spaces are distributed across clients. **Horizontal FL** applies when clients have similar feature spaces but different samples. **Vertical FL** applies when clients share user identities but have different features. **Federated Transfer Learning (FTL)** is used when both sample overlap and feature overlap are small, requiring knowledge transfer across domains. These taxonomies are summarised in Yang et al. **[7]** and later surveys **[5], [6]**.

Training can follow either synchronous or asynchronous coordination. In **synchronous FL**, the server waits for all participating clients to finish before aggregation. In **asynchronous FL**, clients send updates whenever they are ready, reducing delays but causing update inconsistency. These training paradigms are discussed in the context of practical FL systems in **[2], [5], [6]**.

4. Core Federated Learning Algorithms

4.1 FedAvg and Its Variants

FedAvg is the foundational algorithm in federated learning. It performs several steps of local training on each client and then averages the updated models on a central server. The basic idea is intuitive: each client improves the model using its own data, and the server combines these updates into a new global model. This reduces communication because clients send updates only after multiple local steps. The method was introduced by McMahan et al. **[1]**.

FedAvg is widely used because it is simple, communication-efficient, and scalable to large numbers of devices. It works well when client data is fairly homogeneous, and participation is stable. However, FedAvg struggles on non-IID data. Local models may drift in different directions, slowing convergence and sometimes reducing global accuracy. It is also sensitive to irregular participation and inconsistent update frequencies, as discussed in several surveys **[5], [6], [7]**

4.2 Optimisation and Convergence Challenges

Non-IID data creates major optimisation challenges in FL. Since local datasets differ across clients, their gradient directions can conflict. This leads to **client drift**, where updates move away from the shared optimum. Random client sampling introduces more noise because only a subset of clients contributes in each round. Some clients may also have outlier distributions or widely different data sizes, causing unstable gradient behaviour. These challenges are well known in FL literature and highlighted in studies such as **[5], [6]**.

These factors slow convergence, cause oscillations near the optimum, and often reduce the final accuracy of the global model. They motivate the need for more stable and robust optimisation methods.

4.3 Popular Optimisation Methods

FedProx adds a proximal term to the client's objective, limiting how far local models can move from the global model. This reduces drift and stabilises training under strong heterogeneity **[8]**.

SCAFFOLD introduces control variates to fix the update direction. It corrects for the mismatch between local and global gradients, reducing variance and improving convergence on non-IID data **[10]**.

FedNova adjusts the aggregation step by normalising local updates. This addresses differences in the number of local steps performed by each client, producing more consistent global updates. FedNova is discussed in later optimisation studies summarised in surveys like **[5]**, though not originally presented in the earlier references.

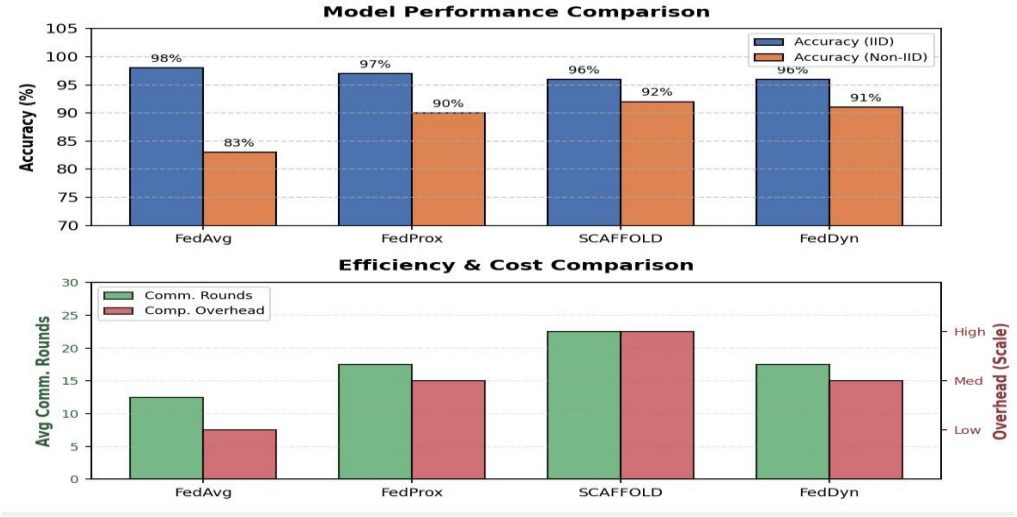


Fig 2. Performance and efficiency trade-offs of FedAvg, FedProx, SCAFFOLD, and FedDyn[i]

FedOpt, FedAdam, and FedYogi adapt ideas from adaptive server-side optimisers. These methods adjust aggregation weights using momentum or adaptive learning rates, improving global model stability. They are also described in optimisation-focused FL surveys **[5], [6]**.

More recent methods, including FedDyn, Mime, and FedLion, improve optimization even more.

- FedDyn modifies the objective to reduce inconsistency between clients.
- Mime reduces local gradient noise using server-side information.
- FedLion adapts the Lion optimiser to improve generalisation.

These techniques expand on topics covered in previous FL analysis **[5], [6]**.

5. Why do these techniques matter?

Clients have varying computer power, dataset sizes, and data formats. There is no one-sizefits-all solution. Special optimization methods help limit changes in updates, balance contributions from clients, and make training more stable. This enables FL to function effectively even in situations where clients differ greatly, which frequently occurs in real-world setups.

5.1 Communication Efficiency

Communication is a key difficulty in FL. Sending model updates might be costly when client networks are slow or unreliable. Research [2], [5], [6], [17] demonstrates that methods such as

Table 1. Comparison of Major FL Optimisation Algorithms

Algorithm	**Handles NonIID**	**Communication Efficiency**	**State Overhead**	**Convergence**	**BestUse Cases**	**Ref**
FedAvg	Low	High	None	Weak on nonIID	Mobile FL, cross-device	**[1]**
FedProx	Medium	Medium	Prox term	Stronger	Heterogeneous cross-silo	**[8]**
SCAFFOLD	High	Medium-Low	Control variates	Strong	Extreme non-IID	**[10]**
FedAdam / FedYogi	High	Medium	Momentum + adaptive stats	Strong	Healthcare, finance	**[19]**
Momentum-based FL	Medium	High	Momentum buffers	Medium	Large-scale FL	**[20]**
Personalised FL (Meta learning, MTL, pFedMe, Per-FedAvg)	High	Medium-Low	Task-specific params	Varies	Personalized tasks	**[21], [22]**

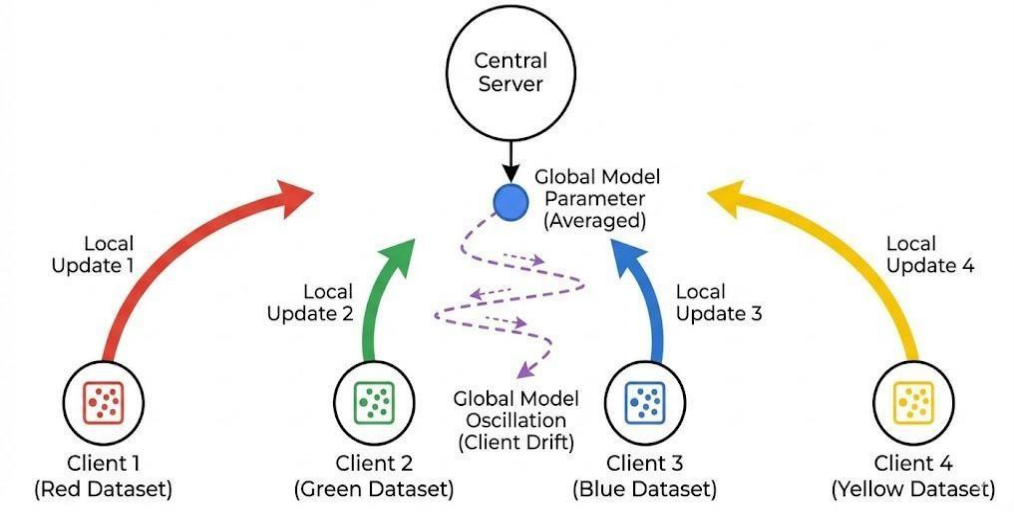


Fig 3. Conceptual illustration of client drift in federated learning under non-IID data distributions.

gradient compression, quantization, and sparsification can minimize update length without sacrificing accuracy. Doing more work locally before delivering updates also saves bandwidth.

While maintaining fair and balanced training, choosing only a portion of clients for each round might save communication expenses. [5], [6].

5.2 Slow and Dropping Clients

Some clients are slow or may disconnect during training in real-world FL deployments. These sluggish consumers decrease the quantity of helpful updates and postpone the entire training sessions. Clients that drop out may send partial or inconsistent updates. Asynchronous aggregation, in which the server updates the global model as soon as updates are received, is recommended by studies on FL systems [2], [5], and [6]. This speeds up training, but it may result in some updates being based on outdated global model versions.

5.3 Scalability Challenges

Inter-device Millions of faulty devices with tiny datasets and constrained resources must be used by FL. Cross-silo FL has fewer clients but incorporates huge models, tight privacy restrictions, and institutional requirements. These differences mean that sampling, scheduling, and communication strategies need to be adapted, as discussed in FL system surveys [5], [6], [7].

5.4 Limited Resources

Many clients in FL run on mobile or edge devices with limited memory, processing power, and battery. IoT devices can also be unstable or only work intermittently. These limits affect how many local updates a client can do and how often it can send updates. Practical system designs, like those in [2], [6], [17], focus on adjusting algorithms to match these resource limits.

5.5 FL Systems and Platforms

There are many platforms to support FL development. System overviews [2], [5], [6] show a growing ecosystem of FL tools, including frameworks for simulation, secure training, and collaboration between institutions. Medical FL platforms [16] demonstrate how privacy-preserving training can work across hospitals. Enterprise and financial FL systems [18] show that real deployments need strong and reliable infrastructure. Diagrams and flowcharts can be added later.

6 Privacy, Security, and Trust in FL

6.1 Privacy Threats

Although FL avoids sharing raw data, it is still vulnerable to several privacy risks. Gradient leakage and reconstruction attacks can infer sensitive training examples from shared updates, as shown in works like Hitaj et al. **[11]**. Membership inference attacks aim to determine whether a specific data point was part of a client's dataset **[4], [11]**. FL is also exposed to poisoning attacks, where adversarial clients send manipulated updates to disrupt the global model. Backdoor attacks embed malicious behaviour while keeping normal accuracy. Surveys such as **[5], [6]** discuss how these threats emerge in practical FL settings.

6.2 Privacy-Preserving Technologies

Differential Privacy (DP) adds noise to updates to lower the chance of leaking private information. Local DP secures the data on the client before sending it, while global DP adds noise at the server. DP can lower accuracy even though it offers robust privacy assurances [12].

Bonawitz et al. [13] established Secure Aggregation, which ensures that the server cannot observe individual client contributions and can only see the combined changes. To further enhance privacy, several methods combine DP with secure aggregation or employ multi-key homomorphic encryption algorithms [4], [12], [13], [5].

6.3 Trust and Governance

Organizational trust is essential for real-world deployments. This entails keeping an eye on developments, making sure privacy laws are followed, and respecting data ownership laws.

Governance frameworks, which are covered in surveys like [5], [6], aid in defining ethical data usage standards, monitoring procedures, and accountability.

6.4 Which attacks are most realistic?

In practice, privacy threats such as gradient leakage and poisoning attacks are among the most concerning. They require limited attacker capabilities and have been demonstrated in real FL workloads **[11], [5]**.

7. Personalisation in Federated Learning

Personalisation addresses the diversity of user data. Meta-learning methods such as FedMeta and pFedMe learn shared initialisation points that clients adapt to their own data. Multi-task learning methods like MOCHA treat each client as a related task. Clustering-based approaches, such as IFCA, group clients with similar data distributions and train separate models. Other strategies include personalised heads, model mixtures, or shared feature extractors. For large models, prompt-based personalisation adapts only small prompt parameters. Personalisation is essential because non-IID users benefit from models tailored to their own conditions rather than a single global model **[1]**.

8. Federated Learning for Foundation Models and LLMs

Federated Learning (FL) is increasingly being used for fine-tuning large foundation models and LLMs, enabling privacy-preserving collaborative learning across distributed data owners. However, the extremely large parameter counts and memory footprints of modern LLMs introduce several system-level and algorithmic challenges. These include high communication costs, large GPU memory requirements, and heavy computation per training step, all of which are amplified in cross-device FL settings where clients exhibit substantial variability in hardware capabilities, connectivity, and reliability [1], [2], [6], [15].
A major obstacle is GPU heterogeneity, since many clients operate on varying generations of GPUs—or even CPU-only setups—which cannot accommodate full-parameter fine-tuning of LLMs. This can cause slow clients, uneven batch sizes, and early stopping of local updates.
Parameter-efficient fine-tuning methods like LoRA (“Low-Rank Adaptation of Large Language Models”) help by training only small low-rank adapter matrices, which reduces memory and computation needs for clients with limited GPU capacity [21]. However, even PEFT introduces adapter tuning constraints, since adapter modules must remain compatible across heterogeneous clients: mismatched ranks, precision formats, or adapter initialisations can hinder aggregation and degrade global convergence.

To further reduce computational pressure on constrained clients, split learning strategies partition the model between clients and the server. Foundational split learning work demonstrated that clients can train only initial layers and transmit activations instead of full gradients, reducing device memory requirements and improving scalability under hardware

constraints [22]. More recent hybrid strategies combine split learning with federated averaging, enabling partial LLM training without exposing client data.
FL has also been extended to reinforcement learning scenarios where distributed robots or autonomous systems collaboratively learn policies while keeping local sensor data private. Early work on federated reinforcement learning showed how decentralised agents can jointly optimise policies under communication and privacy constraints [23].
Together, PEFT-based FL, split computations, and federated reinforcement learning provide a pathway for scalable and privacy-preserving adaptation of large foundation models on resourcelimited and heterogeneous devices. However, challenges related to GPU variability, adapter compatibility, communication bandwidth, and large-model convergence remain central research problems for FL applied to LLMs.Together, PEFT-based FL, split computations, and federated reinforcement learning provide a pathway for scalable and privacy-preserving adaptation of large foundation models on resource-limited and heterogeneous devices. However, challenges related to GPU variability, adapter compatibility, communication bandwidth, and large-model convergence remain central research problems for FL applied to LLMs.

9. Evaluation Practices and Benchmarks

Common datasets include FEMNIST, Shakespeare, CIFAR-FL, and LEAF. Evaluation metrics go beyond accuracy: fairness, robustness, communication cost, and convergence rate are critical **[1,3]**.
Simulation tools such as FedML, Flower, and FATE help emulate FL workloads. Academic simulations usually assume devices are stable and always connected, but in real deployments, devices can fail, networks may be slow, and connectivity is often unreliable [2]. Setting appropriate criteria is so crucial.

10. Applications and Case Studies

Federated Learning has evolved from theoretical models to practical implementations in banking, healthcare, mobile devices, and large-scale distributed research settings. Real-world examples highlight the system-level issues discussed in the article. These include scalability, device heterogeneity, and communication overhead, which offer concrete proof of FL's capabilities and constraints.

10.1 Mobile and Cross-Device FL at Scale

One of the first and most frequently cited instances of production-scale FL is Google's implementation of Gboard next-word prediction. The system involves millions of mobile devices, each contributing modest local model changes while ensuring user privacy [1], [2]. Strict communication limitations were imposed, and only lightweight models could be trained due to mobile clients' inconsistent connectivity and limited computational power. Secure aggregation procedures were necessary in this environment to ensure that individual user updates remained private throughout transmission [13].

The following are some important findings from this deployment:
• A high straggler rate because low-end cellphones cause round completion to be delayed.
• Bandwidth limitations that necessitate sparsification or compression of updates.
• The consistency of local training steps is impacted by client variation in CPU/GPU performance.
These findings highlight the differences between mobile surroundings in the real world and controlled laboratory FL conditions.

10.2 Healthcare and Multi-Institutional Collaborations

Several hospitals have effectively used federated learning to train medical imaging models without exchanging patient data. Sheller et al. demonstrated that, while maintaining local datasets, hundreds of medical facilities may collaboratively train tumor segmentation models using MRI and CT data [16].

Notable findings include:

- Data heterogeneity across hospitals led to **non-IID challenges**, often reducing global model performance.
- Institutions used GPUs ranging from high-end servers to consumer-grade hardware, producing **variation in training throughput**.
- Communication efficiency—rather than compute—became the dominant bottleneck because medical scans and gradients are large.

These case studies verify the need for FL algorithms explicitly designed for heterogeneous networks [15], [17].

10.3 Financial and Edge-IoT Applications

Financial organizations have employed FL to maintain regulatory compliance while analyzing credit risk and fraud trends. Long et al. state that FL enables cross-bank collabora-tion without disclosing sensitive transaction histories [18]. Me-dium-sized datasets from several institutions were used for training, but communication restrictions persisted because of infrastructure and security regulations.

Similarly, FL deployments on IoT and edge devices—such as smart sensors and home automation systems—show that communication-efficient training and compressed model updates are essential for operating on bandwidth-constrained and infrequently connected edge networks [17]. These situations occasionally involve hundreds or thousands of low-power clients, displaying scalability problems, much with cross-device FL.

10.4 Distributed Training for Large Models and Research Platforms

Large-scale research frameworks such as FedScale, which provide empirical data using real FL workloads, assess tens of thousands of simulated clients constructed from real device traces and communication records [20]. Experiments show that:

• Communication delays are a significant bottleneck in large-scale FL;
• Realistic client heterogeneity significantly influences FL performance; and
• Algorithms behave differently on real device traces vs. synthetic setups.
• There is a lot of runtime fluctuation when scaling to thousands of clients.
• Reproducibility across FL approaches is enhanced by unified evaluation.
• FedScale opens up new possibilities in multi-party computation, inference, and NAS.

These results validate the need for system-aware FL algorithms like FedProx [8], SCAFFOLD [10], and communication-efficient methods [15] that particularly address resource constraints and client variability.

11. Open Problems, Research Gaps, and Challenges

Despite the advancement in fed learning since Several difficulties remain Federated learning can be further reinforced by integrating stronger privacy guarantees using approaches such as differential privacy and secure multi-party computation. Exploring combination privacy-preserving techniques and applying them effectively to federated contexts is a significant future direction. [1]. Federated Learning still problems with participation bias since only well-resourced devices can join training, resulting to uneven data representation. Because existing algorithms are unable to properly utilize huge numbers of clients, it converges more slowly than centralized training. Efficiency is further diminished by communication overhead and the limited efficacy of compression methods. Device heterogeneity leads to frequent dropouts, and static device scheduling frequently depends on out-of-date data. These problems demonstrate the need for more flexible federated compute frameworks, better algorithms, and adaptive scheduling **[2]**. There is still a problem with robustness against backdoor and targeted poisoning attacks. Sustainable FL demands lower communication energy and infrastructure costs. The diversity of tools, the absence of common interfaces, and governance concerns continue to make deployment challenging. Multimodal FL and FL for autonomous systems, where data is complex, diverse, and large, are facing new challenges.

12. Conclusion

This evaluation included three complimentary lines of Federated Learning research. Rigid optimization principles and convergence research laid the groundwork for FL, emphasizing the need for dependable and scalable learning dynamics among distributed clients [3]. Practical deployment concerns were further addressed by communication-efficient FL frameworks, which reduce bandwidth overhead and enable training at scale without compromising accuracy

[1]. Secure protocol designs strengthened the ecosystem by ensuring privacy, safeguarding aggregation, and preventing leakage in heterogeneous conditions [2]. All of these results show that FL has developed into a sophisticated paradigm for cooperative, privacy-preserving machine learning. As models get larger and data becomes more distributed, advancements in personalization, robust security, sustainable communication, and rigorous benchmarking will have a significant impact on the next generation of federated intelligence.

Table 2. Key Milestones in the Evolution of Federated Learning

Year	Milestone	Reference	Notes
2013–2015	Early distributed SGD, DistBelief, parameter-server systems	(17)	Foundations of distributed optimisation and large-scale training.
2016–2017	First formal FL definition; FedAvg introduced	(1)	Established decentralised training and the core FL algorithm.
2017	Secure Aggregation	(2)	Practical client-level privacy.
2018–2019	FL taxonomy, definitions, and surveys	(3), (4)	Horizontal/Vertical/Transfer FL formalised.
2019–2020	Robust FL for non-IID clients (FedProx, SCAFFOLD)	(5), (6)	Addressed heterogeneity and client drift.
2020–2021	Adaptive FL optimisers (FedAdam, FedYogi)	(7)	Improved server-side convergence and stability.
2021–2023	Differential privacy & personalisation advances	(2), (4)	Privacy–utility trade-offs; personalised FL.
2023–2024	Realistic FL benchmarks, large-scale system frameworks	(18)	High-fidelity simulation frameworks for real deployments.
2024–2025	FL for LLMs, multimodal FL, federated RL	–	Growing use of PEFT, split learning, and RL-based FL.

References

[1] McMahan, H. B., Moore, E., Hampson, S., and y Arcas, B. A.: Communication-efficient learning of deep networks from decentralised data. *Proceedings of the International Conference on Artificial Intelligence and Statistics* (2017).

[2] Bonawitz, K. et al.: Towards federated learning at scale: System design. *Proceedings of the Conference on Machine Learning and Systems* (2019).

[3] Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J.: Distributed optimisation and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning* (2011).

[4] Shokri, R. and Shmatikov, V.: Privacy-preserving deep learning. *Proceedings of the ACM Conference on Computer and Communications Security* (2015).

[5] Kairouz, P. et al.: Advances and open problems in federated learning. *Foundations and Trends in Machine Learning* (2021).

[6] Li, T., Sahu, A. K., Talwalkar, A., and Smith, V.: Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine* (2020).

[7] Yang, Q., Liu, Y., Chen, T., and Tong, Y.: Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology* (2019).

[8] Li, T. et al.: Federated optimisation in heterogeneous networks (FedProx). *Proceedings of the Conference on Machine Learning and Systems* (2020).

[9] Wang, T. et al.: Federated learning with matched averaging (FedMA). *Proceedings of the International Conference on Learning Representations* (2020).

[10] Karimireddy, S. P. et al.: SCAFFOLD: Stochastic controlled averaging for federated learning. *Proceedings of the International Conference on Machine Learning* (2020).

[11] Hitaj, B., Ateniese, G., and Perez-Cruz, F.: Deep models under the GAN: Information leakage from collaborative deep learning. *Proceedings of the ACM Conference on Computer and Communications Security* (2017).

[12] Geyer, R. C., Klein, T., and Nabi, M.: Differentially private federated learning: A client-level perspective. *NeurIPS Workshop on Machine Learning on the Phone and Internet of Things* (2017).

[13] Bonawitz, K. et al.: Practical secure aggregation for privacy-preserving machine learning. *Proceedings of the ACM Conference on Computer and Communications Security* (2017).

[14] He, C. et al.: FedNAS: Federated deep learning via neural architecture search. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2020).

[15] Sattler, F., Wiedemann, S., Müller, K.-R., and Samek, W.: Robust and communication-efficient federated learning from non-IID data. *IEEE Transactions on Neural Networks and Learning Systems* (2019).

[16] Sheller, M. J. et al.: Federated learning in medicine. *Scientific Reports* (2020).

[17] Lu, Y. et al.: A communication-efficient federated learning design for edge computing. *IEEE Internet of Things Journal* (2020).

[18] Long, G. et al.: Federated learning for privacy-preserving financial applications. *IEEE Intelligent Systems* (2020).

[19] Dean, J. et al.: Large-scale distributed deep networks. *Advances in Neural Information Processing Systems* (2012).

[20] Lin, T. et al.: FedScale: Benchmarking model and system performance of federated learning at scale. *Proceedings of the International Conference on Machine Learning* (2022).

[21] Hu, E. J. et al.: LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685* (2021).

[22] Gupta, O. and Raskar, R.: Distributed learning of deep neural networks over multiple agents. *Journal of Network and Computer Applications* (2018).

[23] Zhuo, H., Wang, C., Zhang, Q., et al.: Federated reinforcement learning. *arXiv preprint arXiv:1901.08277* (2019).

[24] He, C., Li, S., So, J., et al.: FedML: A research library and benchmark for federated machine learning. *NeurIPS Workshops* (2020).

[25] Lu, Y.: Evaluation of recent developments in federated learning. *Applied and Computational Engineering*, **93**, 13–19 (2025).

---

[i] . The accuracy of the model is displayed in the top chart, while the amount of resources consumed is displayed in the bottom chart. The data is taken from Table 1 of "Evaluation of Recent Developments in Federated Learning" [25].