A Survey on Federated Learning Improvement Techniques

Aly Ahmad

***Abstract*—Federated Learning (FL) enables decentralized training of machine learning models across multiple devices, preserving privacy and reducing communication overhead. Despite its benefits, FL faces challenges such as communication inefficiency, data heterogeneity, privacy preservation, resource constraints, and scalability. This survey explores state-of-the-art improvement techniques in FL, including optimization of aggregation frequency, compression methods, clustering algorithms, and model partitioning. Additionally, it examines personalized federated learning approaches and differential privacy mechanisms that enhance security while addressing data non-IID issues. By systematically analyzing these techniques, this work aims to provide insights into advancing Federated Learning systems for robust, efficient, and secure deployment.**

***Index Terms*—Federated Learning, Communication Efficiency, Data Heterogeneity, Privacy Preservation, Personalized Federated Learning, Model Optimization, Differential Privacy.**

I. Introduction

Federated Learning (FL) has emerged as a promising paradigm to address the inherent tension between the need for collaborative model training and data privacy. Unlike traditional centralized machine learning, FL enables multiple clients to jointly train a global model without sharing their raw data, ensuring data remains localized. This is particularly important in fields like healthcare, IoT, and finance, where sensitive data privacy is paramount due to regulations like GDPR and HIPAA.

However, despite its advantages, Federated Learning faces several **key challenges** that hinder its widespread adoption:

- **Communication Efficiency:** Frequent data transfer during global model aggregation introduces significant communication overhead, particularly in resource-constrained environments.
- **Data Heterogeneity:** Non-IID (Independent and Identically Distributed) data across clients can degrade model convergence and accuracy.
- **Privacy Preservation:** While FL mitigates direct data sharing, the transmission of model updates can still lead to privacy breaches.

The existing literature has proposed various **improvement techniques** to address these challenges, ranging from:

- **Adaptive Aggregation Methods** that reduce communication costs.
- **Compression and Quantization Techniques** to optimize data transmission.
- **Clustering Algorithms** to manage data heterogeneity by grouping similar clients.
- **Privacy Mechanisms** like Differential Privacy to enhance security.

This survey aims to analyze and categorize **key techniques** that improve Federated Learning by the problem they are trying to solve.

The rest of this paper is structured as follows: Section II provides the background and foundational concepts of Federated Learning. Section III explores various techniques to address FL challenges. Finally, Section IV concludes the paper.

II. Background

Federated Learning (FL) was first introduced by McMahan *et al.* in their seminal work *"Communication-Efficient Learning of Deep Networks from Decentralized Data"* [1]. The motivation behind FL is to train machine learning models collaboratively without the need to centralize raw data, thereby addressing significant concerns surrounding data privacy, communication efficiency, and scalability.

A. Federated Learning Paradigm

The core of FL involves training a shared global model across multiple clients, each holding its local dataset. Clients compute model updates based on their local data, and only the updates are sent to a central server, where they are aggregated to improve the global model. This iterative process significantly reduces the need for data sharing but introduces unique challenges:

- **Communication Constraints:** Clients often have limited bandwidth, and communication between clients and the central server is costly.
- **Non-IID Data:** Unlike traditional distributed learning, client data in FL is non-IID and unbalanced, as it reflects individual usage patterns.
- **Scalability:** A large number of clients and heterogeneous environments make managing updates and aggregations complex.

To address these challenges, McMahan *et al.* proposed the **FederatedAveraging (FedAvg)** algorithm. This algorithm enhances communication efficiency by allowing clients to perform multiple local updates (using stochastic gradient descent) before sending model updates to the server. The server aggregates these updates to improve the global model, significantly reducing the communication rounds required.

B. Problems in Federated Learning

Section III of this paper will focus on the key problems in FL, which are critical to achieving efficient and effective decentralized learning:

- **Communication Efficiency:** Reducing the communication overhead while maintaining model accuracy.
- **Data Heterogeneity:** Handling non-IID and unbalanced data across clients to ensure robust global models.
- **Privacy Preservation:** Implementing methods like Differential Privacy and Secure Aggregation to protect client data.

III. Federated Learning Improvement Techniques

A. Communication Efficiency

Communication efficiency is one of the most pressing challenges in Federated Learning, especially in environments where devices operate under limited bandwidth and computational resources. Optimizing communication during model training involves minimizing the frequency and size of data transfers without compromising model accuracy.

1) Adaptive Aggregation Frequency: **Wang et al.** propose an *adaptive global aggregation frequency* method to optimize the trade-off between local computation and global communication [2]. Their method adjusts the interval between global aggregations dynamically based on a given resource budget and system characteristics.

The objective is to determine the optimal frequency of aggregation τ, where τ represents the number of local update steps before a global aggregation step:

$$w_i(t) = w_i(t-1) - \eta \nabla F_i(w_i(t-1)), \tag{1}$$

where w_i is the local model parameter for client i, η is the step size, and F_i is the local loss function. Increasing τ reduces communication but may degrade model convergence, especially with non-i.i.d. data.

The proposed control algorithm dynamically adapts τ based on:

$$\min_{\tau} G(\tau) = \max_{m} \frac{c_m \tau + b_m}{R_m \tau}, \tag{2}$$

where c_m is the resource consumption for local updates, b_m is the communication cost per global aggregation, and R_m is the total resource budget. Their experiments demonstrate near-optimal convergence with minimal resource usage.

2) System Design for Communication Efficiency: In large-scale FL deployments, **Bonawitz et al.** address communication efficiency through *system-level optimizations* [3]. Their proposed framework (Illustrated by 1) focuses on:

- **Synchronous Rounds:** Devices perform local updates simultaneously, and the server aggregates updates once enough devices report their results.
- **Client Selection:** A subset of devices is selected to participate in each round, reducing communication overhead.
- **Pipelining:** Overlapping the selection and aggregation phases to minimize idle time.

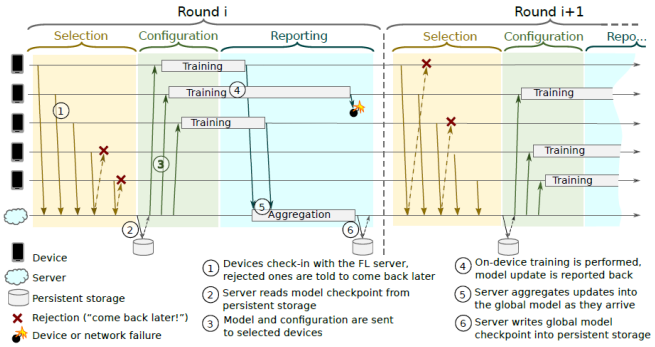

Fig. 1. An Illustration of the operation of the scalable FL system design

- **Secure Aggregation:** Cryptographic masking techniques ensure that individual updates remain private, reducing communication redundancy.

The Federated Averaging algorithm (FedAvg), initially proposed by McMahan *et al.*, is extended in this framework to handle large-scale device participation efficiently. The server orchestrates client participation and dynamically scales to support millions of devices.

3) FedPAQ: Periodic Averaging and Quantization: **Reisizadeh et al.** propose FedPAQ, a communication-efficient FL method that combines three key strategies: periodic averaging, partial device participation, and quantized message-passing [4]. These strategies reduce communication costs while maintaining strong theoretical guarantees.

a) Periodic Averaging: Instead of transmitting updates in every iteration, devices perform multiple local updates before communicating with the server. This reduces the number of communication rounds significantly. Let $x_i(t)$ denote the model at client i:

$$x_i(t+1) = x_i(t) - \eta \nabla F_i(x_i(t)), \tag{3}$$

where t is the local update step. Periodic averaging at the server combines the local models after τ iterations:

$$x_{global} = \frac{1}{n} \sum_{i=1}^{n} x_i(t+\tau). \tag{4}$$

b) Quantized Updates: To reduce communication payloads, FedPAQ employs quantization techniques where each client transmits a compressed version of its local update:

$$Q(x) = \|x\| \cdot \text{sign}(x) \cdot \xi(x, s), \tag{5}$$

where $\xi(x, s)$ represents stochastic quantization with s levels. Quantized updates retain theoretical convergence guarantees while significantly lowering communication costs.

c) Partial Device Participation: FedPAQ also allows a random subset of devices to participate in each communication round. This reduces server load and improves scalability.

The authors show that FedPAQ achieves near-optimal convergence rates for both strongly convex and non-convex loss functions while reducing communication overhead.

4) Comparison and Summary: The combined approaches of adaptive aggregation, system-level optimizations, and periodic averaging with quantization (FedPAQ) ensure that FL systems achieve optimal communication efficiency while maintaining model accuracy and resource constraints. Table **??** summarizes the key methods.

B. Data Heterogeneity

Data heterogeneity, or non-IID (non-Independent and Identically Distributed) data, is a major challenge in Federated Learning (FL). In practice, the local data on each client often varies significantly in distribution, leading to issues such as slow convergence, model drift, and reduced global model accuracy.

1) Impact of Non-IID Data on FL Performance: **Zhao et al.** demonstrated that non-IID data can lead to severe performance degradation in FL systems [5]. They observed that the accuracy of FL models could drop by up to 55% in extreme cases where client devices contain only a single class of data. The authors attribute this accuracy drop to **weight divergence**, which quantifies the difference between local model weights and the global model weights:

$$\text{Weight Divergence} = \frac{\|w_{\text{FedAvg}} - w_{\text{SGD}}\|}{\|w_{\text{SGD}}\|}, \tag{6}$$

where w_{FedAvg} and w_{SGD} represent model weights under FedAvg and centralized SGD, respectively.

The severity of weight divergence is proportional to the Earth Mover's Distance (EMD) between the local data distribution and the global population distribution. A higher EMD indicates more skewed data, which worsens model performance.

2) Iterative Dynamic Clustered Federated Learning: Non-IID and imbalanced data distributions across clients significantly degrade the performance of Federated Learning (FL). To address this, **Gong et al.** propose *Adaptive Client Clustering for Efficient Federated Learning*, a framework that dynamically clusters clients based on their data distributions and updates the clusters during training [6]. This approach reduces communication overhead and improves convergence compared to traditional clustering methods.

a) Key Components of Adaptive Client Clustering: The framework consists of two main components:

1) **Local Training Adjustment:** Adjusts the number of training epochs for each client based on the size of its dataset to balance contributions.
2) **Adaptive Clustering:** Dynamically clusters clients based on data similarity using model similarity metrics.

b) Local Training Adjustment: The number of local training epochs E_i for a client i is adjusted as:

$$E_i = \frac{N_i}{\max_j N_j} \cdot E,$$

where:

- N_i: Number of samples at client i,
- $\max_j N_j$: Maximum number of samples across all clients,
- E: Default number of training epochs.

This ensures that clients with smaller datasets contribute proportionally to the global model.

c) Adaptive Clustering Strategy: Clients are clustered dynamically using a similarity matrix computed from their model updates. The steps include:

1) **Model Updates:** Each client computes its model update Δw_i.
2) **Cosine Similarity Matrix:** The similarity between clients i and j is measured as:
$$S_{i,j} = \frac{\Delta w_i \cdot \Delta w_j}{\|\Delta w_i\| \|\Delta w_j\|}.$$
3) **Clustering:** Clients are grouped into clusters $C_1, C_2, \ldots, C_k$ based on a threshold similarity T. Clients with $S_{i,j} > T$ are grouped into the same cluster.

d) Dynamic Cluster Updating: Clusters are updated iteratively during training as data distributions and model parameters evolve. The algorithm ensures that each cluster aligns with a subset of clients sharing similar data distributions. Let the clustering be represented as $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$, where k is the number of clusters. The global model for each cluster is updated as:

$$w_c^{(t+1)} = w_c^{(t)} - \eta \sum_{i \in C_c} \nabla L_i(w_c^{(t)}),$$

where:

- $w_c^{(t)}$: Model for cluster C_c at iteration t,
- ∇L_i: Gradient of the loss function for client i,
- η: Learning rate.

e) Performance Metrics: The adaptive clustering approach improves both accuracy and communication efficiency, as shown in Table I. The results are reported for the CIFAR-10 dataset under non-IID settings.

TABLE I
PERFORMANCE COMPARISON OF ADAPTIVE CLUSTERING AND FEDAVG [6].

Method	Test Accuracy (%)	Communication Rounds
FedAvg	57.20	120
Static Clustering	64.00	95
Adaptive Clustering	**66.50**	**80**

3) One-shot Clustered Federated Learning: **Dennis et al.** propose a *one-shot clustering technique* in their paper [7]. This method leverages statistical heterogeneity to perform efficient clustering of clients, allowing for separate models to be trained for different client groups. Unlike traditional iterative clustering, the one-shot approach achieves clustering in a single communication round.

a) Key Steps of One-Shot Clustering: The one-shot clustering method consists of two main phases:

1) **Local Clustering at the Client Side:** Each client performs local k-means clustering on its dataset to

compute k cluster centers. Let the dataset at client i be denoted as D_i with n_i samples. The local cluster centers $C_i = \{c_{i1}, c_{i2}, \ldots, c_{ik}\}$ are calculated as:

$$c_{ij} = \frac{1}{|D_{ij}|} \sum_{x \in D_{ij}} x, \quad \forall j \in \{1, \ldots, k\},$$

where D_{ij} represents the subset of data points belonging to the j-th cluster.

2) **Global Aggregation at the Server:** The server collects the cluster centers from all participating clients. These centers are then merged using a weighted aggregation strategy (e.g., distance-based thresholds) to determine the global clusters:

$$C_{\text{global}} = \bigcup_{i=1}^{N} \{c_{ij}\}, \quad \forall i \in \{1, \ldots, N\}, \forall j \in \{1, \ldots, k\}.$$

The one-shot clustering technique exploits statistical heterogeneity to relax cluster separation requirements. Specifically:

- For **active clusters** (present on the same device), the separation condition is stricter:

$$\|\mu_r - \mu_s\|_2 \geq c\sqrt{m_0}(\Delta_r + \Delta_s),$$

where:
  - μ_r, μ_s: Cluster centers.
  - Δ_r, Δ_s: Cluster spread parameters.
  - c, m_0: Constants related to the dataset.
- For **inactive clusters** (absent on the same device), the separation condition is relaxed:

$$\|\mu_r - \mu_s\|_2 \geq 10\sqrt{m_0}k^{1/4}(\lambda_r + \lambda_s),$$

where λ_r and λ_s are additional cluster-specific parameters.

b) Performance Metrics: Table II compares the one-shot clustering method with centralized clustering techniques on Gaussian Mixture Models (GMMs). The results demonstrate that the one-shot approach achieves comparable accuracy with significantly reduced communication overhead.

TABLE II
CLUSTERING ACCURACY OF ONE-SHOT CLUSTERING ON GMMS [**?**].

Parameters	Accuracy (%)
$d = 100, k = 16$	100.00 ± 0.00
$d = 300, k = 64$	98.82 ± 0.70
$d = 300, k = 100$	98.40 ± 0.80

c) Benefits of Heterogeneity in Clustering:: While statistical heterogeneity is often considered a challenge, *Dennis et al.* highlight its potential benefits. Under specific conditions where each device contains data from only a subset of global clusters ($k_1 \leq \sqrt{k}$), heterogeneity relaxes the clustering requirements. Specifically:

- For **active cluster pairs** (present on the same device), a stricter separation condition holds:

$$\|\mu_r - \mu_s\|_2 \geq c\sqrt{m_0}(\Delta_r + \Delta_s), \tag{7}$$

where μ_r and μ_s are cluster centers, and Δ_r and Δ_s represent the cluster spread.
- For **inactive cluster pairs** (absent on the same device), the separation condition is weakened to:

$$\|\mu_r - \mu_s\|_2 \geq 10\sqrt{m_0}k^{1/4}(\lambda_r + \lambda_s), \tag{8}$$

enabling efficient clustering with reduced communication and computational costs.

4) Personalized Federated Learning (PFL) as a Solution: While traditional FL trains a single global model, its performance is often suboptimal under heterogeneous, non-IID data distributions. *Personalized Federated Learning (PFL)* provides tailored models for individual clients, addressing both *client drift* and *solution personalization* challenges [8].

a) Challenges Addressed by PFL::

- **Client Drift:** Local models diverge from the global model due to non-IID data. PFL mitigates this by incorporating client-specific updates.
- **Personalization:** A single global model cannot generalize across diverse client distributions. PFL learns individualized models optimized for each client.

b) PFL Strategies:: PFL can be categorized into the following strategies:

1) **Global Model Personalization:** A global model is trained collaboratively and subsequently personalized for each client. Techniques include:
   - **Regularization-based Methods:** FedProx [9] mitigates model divergence by adding a proximal term to local objectives, while MOON [10] employs contrastive learning to align local and global model representations.
   - **Meta-learning Approaches:** Per-FedAvg [11] uses meta-learning to optimize global models for rapid adaptation. pFedMe [12] further improves adaptation through Moreau envelopes.
2) **Learning Personalized Models:** Individualized models are trained for each client, leveraging their specific distributions:
   - **Parameter Decoupling:** Base layers are shared globally, while upper layers are trained locally for personalization [13].
   - **Clustering-based Methods:** Clients are grouped based on data similarity, and separate models are trained for each group [14].
   - **Weighted Model Aggregation (FedFomo): Zhang et al.** propose *FedFomo* [15], a framework where clients compute personalized weighted combinations of models uploaded by other clients. Instead of relying on a single global average, FedFomo optimizes a client-specific objective using:

$$w_n = \frac{L_i(\theta_i^{t-1}) - L_i(\theta_n^t)}{\|\theta_n^t - \theta_i^{t-1}\|}, \tag{9}$$

where L_i is the validation loss for client i and θ_n^t is the uploaded model of client n. Weights w_n

determine the contribution of each client's model to the local update, favoring models that minimize the validation loss and are closer in parameter space.

c) FedFomo Features and Benefits:: Unlike other approaches, FedFomo offers unique personalization capabilities:

- *Target Distribution Optimization:* FedFomo enables clients to optimize for target distributions that may differ from their local training data, addressing out-of-distribution challenges.
- *Dynamic Model Aggregation:* Clients evaluate all available models and selectively aggregate the most beneficial ones based on validation performance.
- *Scalability and Flexibility:* FedFomo efficiently handles large-scale FL scenarios by reducing communication overhead through a sampling strategy for model downloads.

d) Performance Analysis:: FedFomo consistently outperforms baseline methods like FedAvg and Per-FedAvg in both in-distribution and out-of-distribution settings. Experiments on CIFAR-10 and CIFAR-100 demonstrate significant improvements in accuracy, particularly under extreme non-IID conditions, as shown in Table IV.

5) Summary of PFL Techniques: Table V provides an overview of key PFL techniques and their respective advantages.

C. Privacy in Federated Learning

1) Overview: Privacy is a critical concern in Federated Learning (FL), where sensitive data resides on client devices, but risks of data leakage still exist through model updates. Techniques like Differential Privacy (DP) and Secure Aggregation (SA) have been extensively studied to address these challenges. Additionally, novel methods like client-level differential privacy further refine the protection scope by safeguarding entire datasets rather than individual data points. This section explores these techniques and their roles in privacy-preserving FL.

2) Differential Privacy in FL: Differential Privacy ensures that the inclusion or exclusion of any particular data does not significantly affect the aggregated results, protecting individual contributions. It is formalized as (ϵ, δ)-DP, where ϵ bounds the privacy loss, and δ accounts for the probability of a privacy breach.

3) Noising Before Aggregation in FL (NbAFL): **NbAFL** uses Gaussian noise to ensure differential privacy by perturbing model updates before transmission [16]. This guarantees global DP across uplink and downlink communication channels.

a) Key Features::

- **Privacy-Utility Trade-off:** Larger noise improves privacy but slows convergence.
- **Scalability:** Increasing client numbers enhances model convergence at a fixed privacy level.
- **Optimal Rounds:** The trade-off between privacy and convergence determines optimal communication rounds.

4) Client-Level Differential Privacy: Geyer et al. introduced a mechanism for client-level differential privacy in federated optimization, ensuring that an entire client's dataset is protected from inference attacks [17]. This is achieved by random subsampling and clipping client updates.

a) Algorithm Steps::

1) **Subsampling:** A random subset of m_t clients participates in each round, reducing exposure of any single client.
2) **Clipping:** Updates are scaled to ensure a uniform sensitivity S:
$$\Delta \bar{w}_k = \frac{\Delta w_k}{\max(1, \frac{\|\Delta w_k\|_2}{S})}.$$
3) **Gaussian Mechanism:** Noise calibrated to S is added to the aggregated updates:
$$w_{t+1} = w_t + \frac{1}{m_t}\left(\sum_{k=1}^{m_t} \Delta \bar{w}_k + \mathcal{N}(0, S^2\sigma^2)\right).$$

b) Privacy Budgeting:: A *moments accountant* tracks privacy loss δ, ensuring it stays within a threshold. Training halts if the privacy budget is exhausted, guaranteeing robust client privacy.

5) Secure Aggregation in FL: Secure Aggregation (SA) protects individual updates by ensuring that only the aggregated model is visible to the server. *Bonawitz et al.* developed an efficient SA protocol for federated learning that tolerates client dropouts while maintaining data privacy [18].

a) Protocol Design: SA involves four rounds:

- **Key Exchange:** Clients establish pairwise keys for masking updates.
- **Masked Input Collection:** Clients send masked updates.
- **Consistency Check:** Clients verify the server's integrity.
- **Unmasking:** Remaining clients contribute masks to enable aggregation.

b) Double-Masking Mechanism:: Each client applies two masks—pairwise masks shared with other clients and a self-mask—to prevent information leakage:
$$y_u = x_u + PRG(b_u) + \sum_{v \neq u} PRG(s_{u,v}),$$
where b_u is the self-mask, and $s_{u,v}$ is a shared secret with client v.

c) Performance:: The SA protocol scales efficiently with $O(n^2 + mn)$ complexity for n clients and m-dimensional updates. Robustness to 30% client dropout rates has been demonstrated experimentally.

6) Integrated Privacy Approaches: Combining Differential Privacy and Secure Aggregation strengthens privacy guarantees while improving utility:

- **Enhanced Privacy:** DP ensures robustness against inference attacks, while SA prevents exposure of individual updates.
- **Improved Utility:** SA reduces noise variance in DP by securely aggregating contributions, enhancing model accuracy.

a) Client-Level DP with SA: Client-level differential privacy can be combined with SA to hide client participation while ensuring accurate aggregation:

$$w_{t+1} = w_t + \frac{1}{m_t}\left(\sum_{k=1}^{m_t} \Delta\bar{w}_k + \mathcal{N}(0, \frac{S^2\sigma^2}{m_t})\right).$$

This approach balances privacy protection and model utility, particularly in large-scale federated settings.

D. Conclusion

Differential Privacy, Secure Aggregation, and client-level differential privacy collectively form the backbone of privacy-preserving FL. Each technique addresses unique aspects of privacy, from protecting individual contributions to ensuring robust aggregation in adversarial environments. Their integration enables scalable, secure FL systems with strong privacy guarantees.

IV. Conclusion

Federated Learning has emerged as a transformative paradigm for privacy-preserving collaborative training, addressing key challenges in domains like healthcare, IoT, and finance. This paper surveyed cutting-edge techniques for tackling FL's core challenges, including data heterogeneity, communication efficiency, and privacy preservation. Methods such as clustered FL, personalized FL, Differential Privacy, and Secure Aggregation have been explored in detail, each offering unique solutions to specific issues.

Despite these advancements, significant open research questions remain in scalability, communication efficiency, security, and heterogeneity management. By addressing these challenges, FL can achieve broader adoption, ensuring robust, secure, and efficient learning systems for diverse applications.

TABLE III
Summary of Communication Efficiency Techniques

Method	Key Strategy	Optimization
Adaptive Aggregation	Frequency Control	Dynamic τ
System Design (Bonawitz)	System Optimization	Pipelining
FedPAQ	Periodic Averaging	Quantization

TABLE IV
Performance Comparison of PFL Methods on CIFAR-10 [15].

Method	**Accuracy (In-Distribution)**	**Accuracy**
FedAvg [1]	53.08%	23.11%
FedProx [9]	52.92%	39.79%
Per-FedAvg [11]	72.40%	39.80%
FedFomo (Ours) [15]	**92.10%**	**64.06%**

TABLE V
Summary of Personalized Federated Learning Techniques.

Method	**Key Idea**
Regularization	Penalize local-global divergence
Meta-learning	Optimize global model for local adaptation
Parameter Decoupling	Shared base, local upper layers
Clustering	Group clients based on data similarity
Weighted Aggregation	Optimize client-specific weighted updates

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Singh and J. Zhu, Eds., vol. 54. PMLR, 20–22 Apr 2017, pp. 1273–1282. [Online]. Available: https://proceedings.mlr.press/v54/mcmahan17a.html

[2] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.

[3] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný, S. Mazzocchi, H. B. McMahan, T. V. Overveldt, D. Petrou, D. Ramage, and J. Roselander, "Towards federated learning at scale: System design," 2019. [Online]. Available: https://arxiv.org/abs/1902.01046

[4] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, "Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization," 2020. [Online]. Available: https://arxiv.org/abs/1909.13014

[5] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," 2018. [Online]. Available: https://arxiv.org/abs/1806.00582

[6] B. Gong, T. Xing, Z. Liu, W. Xi, and X. Chen, "Adaptive client clustering for efficient federated learning over non-iid and imbalanced data," *IEEE Transactions on Big Data*, vol. 10, no. 6, pp. 1051–1065, 2024.

[7] D. K. Dennis, T. Li, and V. Smith, "Heterogeneity for the win: One-shot federated clustering," 2021. [Online]. Available: https://arxiv.org/abs/2103.00697

[8] A. Z. Tan, H. Yu, L. Cui, and Q. Yang, "Towards personalized federated learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 12, p. 9587–9603, Dec. 2023. [Online]. Available: http://dx.doi.org/10.1109/TNNLS.2022.3160699

[9] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," 2020. [Online]. Available: https://arxiv.org/abs/1812.06127

[10] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," 2021. [Online]. Available: https://arxiv.org/abs/2103.16257

[11] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning: A meta-learning approach," 2020. [Online]. Available: https://arxiv.org/abs/2002.07948

[12] C. T. Dinh, N. H. Tran, and T. D. Nguyen, "Personalized federated learning with moreau envelopes," 2022. [Online]. Available: https://arxiv.org/abs/2006.08848

[13] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, "Federated learning with personalization layers," 2019. [Online]. Available: https://arxiv.org/abs/1912.00818

[14] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An efficient framework for clustered federated learning," 2021. [Online]. Available: https://arxiv.org/abs/2006.04088

[15] M. Zhang, K. Sapra, S. Fidler, S. Yeung, and J. M. Alvarez, "Personalized federated learning with first order model optimization," 2021. [Online]. Available: https://arxiv.org/abs/2012.08565

[16] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek, and H. Vincent Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[17] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," *ArXiv*, vol. abs/1712.07557, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:3630366

[18] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 1175–1191. [Online]. Available: https://doi.org/10.1145/3133956.3133982