A Theoretical Comparative Study of Privacy Mechanisms and Optimization Techniques in Federated Learning for Edge AI

[1]M L Sharma, [2]Sunil Kumar, [3]Ajay Kumar Garg, [4]Sarvendra mani tripathi , [5]Nipun balhara

[1,2,3]Faculty, Maharaja Agrasen Institute of Technology, Delhi

[4,5]Research Scholar, Maharaja Agrasen Institute of Technology, Delhi
[1]madansharma.20@gmail.com, [2]sunilkumar@mait.ac.in, [3]ajaygargiitr@gmail.com, [3] sarvilnotme@gmail.com, [5]nipunbalhara0025@gmail.com

Abstract

Federated Learning (FL) enables decentralized model training across edge devices while keeping raw data local, thereby enhancing privacy and reducing communication costs. However, theoretical challenges remain: reconciling model utility with formal privacy guarantees, mitigating communication and computation overheads, and ensuring robustness against adversarial participants. This paper presents a conceptual comparative study of prominent privacy mechanisms (differential privacy, secure aggregation, homomorphic encryption, and trusted execution environments) and optimization techniques (client selection, quantization and compression, adaptive learning, and personalization strategies) used in federated settings. Focusing exclusively on theoretical properties, we analyze the assumptions, security models, computational complexity, communication trade-offs, and utility-privacy relationships for each method. The goal is to provide a principled, design-oriented framework guiding researchers and practitioners for deploying FL in resource-constrained and privacy-sensitive environments.

Keywords

Federated Learning, Differential Privacy, Secure Aggregation, Homomorphic Encryption, Edge AI, Communication Efficiency, Theoretical Analysis, Personalization.

1. Introduction

Federated Learning (FL) is a distributed paradigm where multiple clients collaboratively train a global model under the orchestration of a central server, without sharing raw local data. FL is particularly suitable for edge AI applications (mobile devices, IoT sensors, healthcare wearables) where data privacy, bandwidth constraints, and heterogeneity are primary concerns.

Despite wide interest and numerous empirical studies, the theoretical landscape of FL remains nuanced. Implementing privacy mechanisms introduces trade- offs between model utility and formal privacy guarantees; optimizing communication reduces bandwidth use but may affect convergence; personalization improves local performance but complicates aggregation. This paper presents a conceptual comparison of privacy-preserving mechanisms and optimization strategies in FL, emphasizing theoretical properties and trade-offs rather than empirical benchmarks.

2. Literature Review

Federated Learning was popularized by McMahan et al. (2017) with the FederatedAveraging (FedAvg) algorithm, laying the groundwork for collaborative training with periodic aggregation of client-updated model weights. Subsequent work expanded FL to handle system and statistical heterogeneity, communication constraints, and privacy considerations.

Privacy in FL has been addressed via several theoretical frameworks:

- **Differential Privacy (DP):** Originating in the database literature (Dwork et al.), DP provides provable privacy bounds by adding calibrated noise to model updates. Abadi et al. extended DP to deep learning via the moments accountant. In FL, DP can be implemented locally (LDP) or centrally (global DP), each with differing theoretical utility costs.

Global (population-weighted) objective — to reference throughout:

$$F(\theta) = i=1\sum piFi(\theta) , \quad \sum pi=1.$$

(ε, δ)-Differential Privacy definition (use when defining DP formally):

$$Pr[M(D)\in S] \leq e\varepsilon Pr[M(D')\in S] + \delta$$

for all neighbouring datasets D, D′ and measurable sets SSS.

- **Secure Aggregation (SA):** Protocols such as Bonawitz et al. (2017) enable a server to compute the sum of client updates without learning individual contributions, assuming honest-but-curious servers. SA has provable confidentiality properties under certain adversarial models.
- **Homomorphic Encryption (HE):** HE allows computation on encrypted data; in FL, HE can enable encrypted aggregation of client updates with provable cryptographic guarantees but at high computation and communication costs.
- **Trusted Execution Environments (TEEs):** TEEs (e.g., Intel SGX) provide hardware-enforced isolated computation areas. TEEs reduce cryptographic overhead but rely on hardware trust assumptions and may have limited scalability.

Optimization strategies focus on reducing communication overhead and improving convergence:

- **Client Selection and Partial Participation:** Theoretical studies (e.g., Li et al.) model the convergence behavior under partial participation, showing that careful sampling can preserve convergence rates under certain assumptions.
- **Compression and Quantization:** Theoretical analyses of quantized stochastic gradient descent quantify bias and variance introduced by compression schemes, guiding compression ratios that maintain convergence guarantees.

Stochastic compression operator C(·)— bounded variance model (commonly used in theoretical analyses):

$$E[C(g)]=g, \qquad E||C(g)-g^2|| \leq \omega\|g\|^2,$$

where ω≥0 denotes relative variance introduced by compression (e.g., for top-k or randomized quantization).

Effect on convergence (sketch):
If local gradients have variance bounded by σ_g^2 , compressed updates add variance $\omega\|g\|^2$, so an error term appears:

$$\text{Convergence error} \sim O(1/T) + O(\omega/T) + O(\sigma_g^2)$$

- **Adaptive Optimization & Personalization:** Methods like FedProx introduce proximal terms to handle heterogeneity, while theoretical personalization frameworks model multi-task objectives balancing global and local loss.

This literature provides the theoretical building blocks we synthesize below.

3. Theoretical Framework

We establish a conceptual framework for comparing privacy mechanisms and optimization techniques in FL. Consider N clients with local data distributions D_i and local loss functions F_i(\theta). The global objective is to minimize the population-weighted loss F(\theta) = \sum_{i=1}^N p_i F_i(\theta), where p_i corresponds to client weight (e.g., proportional to dataset size)

Key theoretical axes for comparison:

- **Privacy Guarantee Model:** Formal definition (ε,δ)-DP, cryptographic semantic security, or hardware-rooted confidentiality.
- **Threat Model:** Honest-but-curious server, Byzantine (malicious) clients, or colluding clients and server.
- **Utility Impact:** How the method affects convergence rates, bias, variance, and final model accuracy in theory.
- **Communication Complexity:** The additional bits transferred per round due to privacy/optimization mechanism.
- **Computation Overhead:** Local and server-side computational complexity added by privacy/optimization tools.
- **Scalability & Practical Assumptions:** Scalability to large number of clients and realistic hardware assumptions (TEEs available?).

This multi-dimensional framework allows principled discussion without experimental data.

4. Privacy Mechanisms — Theoretical Analysis

4.1 Differential Privacy (DP)

Definition & Types: DP ensures that the inclusion or exclusion of a single data point changes output distributions only slightly. In FL, **local DP (LDP)** adds noise at the client before sending updates; **global DP** adds noise at aggregation time. Differential Privacy provides a *mathematical guarantee* that the output of a mechanism does not significantly change when any single individual's data is modified, added, or removed.
The paper explains this as:

- *"DP ensures that the inclusion or exclusion of a single data point changes output distributions only slightly.*

Local Differential Privacy (LDP):

- Noise is added **on the client device** *before* sending the update.
- This gives the **strongest privacy** because the server never sees the real gradient/local update.

Global (Central) Differential Privacy (GDP):

- Clients send true updates.
- Noise is added **after aggregation** at the server.
- Provides *moderate* privacy and better accuracy than LDP.
- However, requires:
  - a **trusted aggregator** or
  - a privacy-preserving tool like secure aggregation or homomorphic encryption to hide individual updates.

Theoretical Properties:

- **Utility-Privacy Trade-off:** Adding noise increases variance of updates; under convex assumptions, convergence rate may degrade from $O(1/T)$ to $O(1/T) + O(\sigma_g^2)$.

Signal-to-noise ratio (SNR) for DP-noised gradients:

$$SNR=Var(signal)/ \sigma^2$$

where σ^2 is DP noise variance. Low SNR ⇒ learning degrades — useful when discussing when LDP becomes infeasible.

Formal Privacy Guarantee (ε, δ)-DP

DP provides a mathematically quantifiable privacy guarantee.

A mechanism is (ε,δ) -DP if changing one user's data modifies the output distribution only by a bounded factor eε plus a small probability δ.

- **Composition & Amplification:** Iterative updates require careful accounting; privacy loss composes across rounds. Because only a small subset of clients participate each round:

- Each user contributes with probability qqq
- This reduces the effective privacy loss (ε becomes smaller)

This is a powerful theoretical property unique to FL.

- **Threat Model:** DP defends against reconstruction and inference attacks assuming adversary only sees noisy outputs.
- **Limitations:** For stringent ε, noise may render learning ineffective. Local DP is particularly costly in utility. Differential Privacy suffers from several theoretical limitations. Strong privacy budgets require large amounts of noise, substantially degrading model utility—especially in Local DP. Privacy loss accumulates across training rounds, demanding careful composition accounting. The added noise increases gradient variance, slowing convergence and creating a utility floor. DP theory is well-developed for convex problems but remains loose for deep non-convex models, limiting its reliability. Global DP requires trust in the server or additional cryptographic mechanisms such as Secure Aggregation or Homomorphic Encryption. Collectively, these limitations make DP challenging to deploy in resource-constrained, high-accuracy federated learning settings.

4.2 Secure Aggregation (SA)

Definition: Secure Aggregation (SA) is a cryptographic protocol that ensures the server can recover **only the sum of client updates**, while each individual update remains hidden through masking. Under the honest-but-curious server model and non-colluding clients, SA offers **information-theoretic confidentiality** of local updates without adding noise.

Theoretical Properties:

Confidentiality: Under honest-but-curious server model and non-colluding clients, SA ensures information-theoretic secrecy of individual updates. SA guarantees that the server can only learn the **aggregated sum** of client updates and **not** any individual client update.

No Utility Penalty (in principle): SA does not perturb gradients, so convergence guarantees of FedAvg hold if protocol succeeds. Theoretical convergence rate of the underlying optimization algorithm remains the same.

Cryptographic Security Based on Masking:

Clients generate pairwise or group masks such that:

- each client's update is hidden,
- all masks cancel out in the final aggregation.

This results in **information-theoretic security** (for Bonawitz-style protocols) because even an unbounded adversary cannot learn the masked values.

Overheads: Communication rounds increase due to setup and mask exchange. Complexity grows with number of clients; worst-case setup is O(N) rounds and pairwise key exchanges.

Limitations:

1. Vulnerability to Client Dropouts
SA protocols require that all clients participating in a round complete their mask exchanges.
If a client drops out:

- its masks may not cancel,
- the server cannot recover the aggregate sum,
- the entire aggregation round may fail.

Dropout-resilient versions exist but add significant complexity.

2. Collusion Breaks Security Assumptions
SA assumes:

- The server is honest-but-curious.
- Clients do not collude with the server.

If even a few clients collude with the server, they can reveal random masks or partial secrets, allowing the server to infer individual client updates.Thus, SA is weaker than DP with respect to adversarial models.

3. High Communication and Setup Overhead

Implications:

- Mask generation and key exchange introduce multiple setup messages.
- Large numbers of clients → heavy communication cost.
- Pairwise key sharing may scale as $O(N^2)$ making SA difficult for massive FL deployments.

4. Complexity Increases with Number of Clients

SA's computational and communication complexity grows with client count.

Theoretical impact:

- Large-scale federated networks (e.g., 10,000 devices) may experience long delays.
- More clients → more masking keys → more rounds.

This limits scalability.

5. No Protection Against Malicious Server Tampering (Beyond Curious)

The SA threat model assumes an **honest-but-curious** server.

Limitation:
If the server is *malicious*—attempting to tamper with or modify protocol messages—standard SA does not guarantee integrity.

Additional cryptographic tools (signatures, verifiable aggregation) are needed.

6. Not Resilient to Sybil Attacks

Because SA relies on non-collusion:

- An adversary controlling many fake clients (Sybil nodes) can weaken privacy guarantees.
- This is outside the theoretical protection model of SA.

7. Does Not Provide Semantic Privacy Like DP

 SA prevents data visibility **but does not add noise**.

4.3 Homomorphic Encryption (HE)

Definition: Homomorphic Encryption (HE) is a cryptographic technique that allows computations to be performed directly on encrypted data without requiring decryption. In the context of Federated Learning, clients encrypt their local model updates, and the server aggregates these encrypted updates using the homomorphic property. The server obtains an encrypted aggregate, which can be decrypted only with the appropriate key, ensuring that individual client updates remain confidential throughout the process.

Theoretical Properties:

Strong Cryptographic Guarantees: Semantic security ensures that encrypted client updates reveal **no information** about the underlying data—even to a computationally bounded adversary.
The server cannot distinguish between the ciphertexts corresponding to different plaintext updates.

This provides a **much stronger confidentiality guarantee than Secure Aggregation**, because it does not rely on a trust model or mask cancellation.

Arithmetic on Ciphertexts (Homomorphic Property): HE allows operations like addition or multiplication **without decrypting the data**.
For FL, additive HE is typically used:

$$\mathbf{Enc(x1) \oplus Enc(x2) = Enc(x1+x2)}$$

This enables:

- privacy-preserving aggregation
- no information leakage to the server
- compatibility with standard FL workflows

Computation & Communication Cost:

HE requires:

- expensive homomorphic arithmetic (e.g., modular exponentiations, polynomial arithmetic)
- large ciphertext sizes (10× to 100× larger than plaintext)

This leads to:

- **high computational overhead** (especially on edge devices)
- **high communication bandwidth requirements**

This makes HE less practical for real-time or large-scale FL deployments.

Utility: No noise-induced bias (unless combined with DP), so learning utility is preserved if resources permit.

Limitations: HE is often impractical on constrained devices due to heavy computation and bandwidth.

4.4 Trusted Execution Environments (TEEs)

Definition: Hardware enclaves provide isolated execution where the server can safely run aggregation code.

Theoretical Properties:

Performance: TEEs avoid cryptographic overhead, enabling near-native aggregation performance.

Trust Model: Requires trust in hardware vendor and correct enclave attestation; side-channel attacks remain a theoretical risk.

Scalability: TEEs scale well in computation but depend on availability across server infrastructure.

Limitations: Relies on hardware trust and not a formal cryptographic proof; may be unsuitable for fully decentralized scenarios. Trusted Execution Environments provide hardware-isolated computation but come with significant limitations. They rely on trust in the hardware vendor and are susceptible to various microarchitectural side-channel attacks. TEEs lack the cryptographic rigor of HE, making them vulnerable if the hardware enclave is compromised. Their deployment is constrained by hardware availability, limited enclave memory, and scalability challenges in large federated systems. TEEs do not address malicious client behavior and are often unsuitable for decentralized federated learning settings. Supply-chain vulnerabilities further weaken their theoretical security guarantees.

5. Optimization Techniques — Theoretical Analysis

5.1 Client Selection & Partial Participation

In large-scale federated learning systems, it is often impractical for all clients to participate in every training round due to communication constraints, device availability, and energy limitations. **Client selection and partial participation** address this by sampling only a subset of clients in each communication round. This reduces overall bandwidth usage and computational demand while maintaining scalability across large, heterogeneous populations of devices.

From a theoretical standpoint, partial participation influences convergence through the statistical properties of sampling.

Theoretical Principles

1. Unbiased Gradient Estimation

When clients are sampled uniformly or proportional to data size, the aggregated update remains an **unbiased estimator** of the true global gradient. This allows standard convergence analyses to extend to the partial-participation setting.

2. Sampling Variance

Because only a fraction of clients participate, the aggregated gradient has higher variance compared to full participation. This variance decreases as:

- the number of selected clients increases, or
- the number of rounds increases.

Thus, partial participation introduces a stochastic error term that must be balanced with communication efficiency.

3. Scalability Benefits

By selecting only a small subset of devices each round, communication overhead per round becomes proportional to the number of selected clients rather than the total population, enabling federated learning to scale to millions of devices.

4. Importance of Sampling Strategies

Different strategies have theoretical effects:

- **Random selection**: preserves unbiasedness and ensures fairness.
- **Stratified selection**: reduces variance by ensuring representative client subsets.
- **Priority-based selection**: may favor stable or high-quality clients but could introduce bias.

5.2 Compression & Quantization

1. Stochastic Compression Operators

Compression is modeled using a stochastic operator C(g) applied to a gradient or model update ggg.
The operator satisfies:

$E[C(g)] = g$ (unbiased)

and has bounded variance:

$E\|C(g)-g\|^2 \leq \omega\|g\|^2$

where:

- ω measures the distortion caused by compression
- Smaller $\omega \rightarrow$ higher fidelity

- Larger ω → more error, slower convergence

This formulation allows gradient compression to be analyzed using standard stochastic optimization theory.

---

2. Convergence Under Compression

The added noise from compression increases the variance of the update, but convergence can still be guaranteed if:

- the learning rate is sufficiently small, and
- compression is unbiased or variance-controlled.

The impact on convergence typically appears as an additional error term:

$O(1/T) + O(\omega)$

where:

- the $O(1/T)$ term is the standard rate, and
- the $O(\omega)$ term captures compression-induced error.

Thus, compression introduces a **trade-off between communication savings and accuracy**.

---

3. Types of Compression

Common methods include:

• Quantization

Reducing numerical precision (e.g., 8-bit, 4-bit, ternary).
Pros: very low bandwidth usage.
Cons: can introduce bias without proper stochastic rounding.

• Sparsification (Top-k, Random-k)

Sending only the largest-magnitude gradient components.
Pros: large reduction in message size.
Cons: may slow convergence if too few entries are sent.

• Low-rank compression

Approximating updates with low-rank matrices.
Useful in large models with structured gradients.

---

4. Benefits for Federated Learning

Compression provides:

- **Lower per-round communication cost**
- **Scalability to large models**

- **Reduced energy consumption for mobile/edge clients**

These are crucial for real-world FL deployment where bandwidth is limited.

---

5. Trade-offs

Compression improves communication efficiency but:

- increases gradient variance (ω),
- may reduce stability under non-IID data,
- requires careful tuning of step sizes and compression ratios.

Therefore, selecting an appropriate compression method involves balancing accuracy and efficiency.

5.3 Adaptive Local Steps & Proximal Methods

Allowing multiple local SGD steps (FedAvg) reduces communication frequency but may cause divergence under heterogeneous data. Theoretical remedies include FedProx, which adds proximal terms to control drift, leading to improved convergence bounds under bounded heterogeneity. Federated Learning typically relies on local stochastic gradient descent (SGD) performed independently on each client before aggregating updates at the server. Allowing **multiple local updates per round** (as in FedAvg) reduces communication frequency but can introduce instability when client data distributions are heterogeneous (non-IID). **Adaptive local steps** and **proximal methods** aim to control this instability and provide improved convergence guarantees in heterogeneous environments.

1. Local Update Drift

In non-IID settings, each client's update direction deviates from the true global gradient:

$\|\theta_i^{t+1} - \theta^t\| \propto E$

where:

- EEE = number of local SGD steps
- Large EEE increases **client drift**, leading to divergence.

Thus, there is a fundamental trade-off:

- **More local steps → less communication**
- **But larger risk of divergence**

2. Adaptive Local Steps

Instead of fixing the number of local SGD steps, clients may adapt:

- Step count based on local data size
- Step count based on gradient norms
- Step count based on computational constraints or energy availability

Theoretical implication:

- Reducing steps for high-heterogeneity clients improves stability

- Increasing steps for homogeneous clients improves efficiency
- Adaptive strategy balances convergence and communication cost

5.4 Personalization & Multi-Task Theories

Personalization frameworks cast local objectives as multi-task optimization problems. Theoretical models (e.g., meta-learning formulations) trade global generalization for local accuracy, often improving per-client utility but complicating aggregated convergence analysis Personalization and multi-task theories address the limitations of training a single global model in heterogeneous federated learning settings. Clients are treated as distinct but related tasks, each with its own objective function. Methods such as meta-learning, fine-tuning, clustered FL, and model decomposition (shared vs. local parameters) provide personalized updates that better align with local data. Theoretical analyses model these relationships using bounded task divergence, enabling convergence guarantees and improved performance under non-IID distributions. Personalization mechanisms significantly enhance accuracy and stability, especially when global FL solutions fail to generalize across diverse clients.

6. Conceptual Comparative Table

Mechanism / Technique	Privacy Model	Utility Impact (theory)	Comm. Overhead	Comp. Overhead	Scalability / Notes
Local DP	Formal (ε,δ) per-client	High noise → high utility loss for small ε	Low (no crypto)	Low	Good for strong privacy but poor utility
Global DP (server)	Formal after aggregation	Moderate utility loss	Low	Low	Requires trusted aggregator or SA/HE
Secure Aggregation	Cryptographic - sums only	None (if successful)	Moderate (masking setup)	Moderate	Robustness to dropouts needed
Homomorphic Encryption	Cryptographic	None (no noise)	Very High	Very High	Often impractical on MCUs
TEEs	Hardware root of trust	None (no noise)	Low	Moderate	Requires trusted hardware
Compression/ Quantization	—	Adds compression bias/variance	Low (reduced)	Low	Trade-off tunable via \omega
Client Selection	—	Sampling variance affects convergence	Reduced	Low	Good scalability if unbiased
Personalization	—	Often improves local utility	Varies	Low-Moderate	Complex theoretical analysis

(All entries are conceptual; utility impact depends on model class, data heterogeneity, and chosen parameters.)

7. Discussion

From a theoretical standpoint, **no single privacy mechanism is universally optimal**. Local DP provides the strongest client-level privacy but imposes a high utility cost unless datasets are large or noise budgets are relaxed. Secure aggregation preserves learning utility but depends on cryptographic setup complexity and assumptions about participant behavior. HE offers strong cryptographic guarantees without injecting noise yet is computationally heavy and thus impractical on edge devices. TEEs represent a middle ground, delivering high performance under hardware

trust assumptions but are vulnerable to side- channels and supply-chain threats.

Optimization techniques complement privacy mechanisms. Combining compression with SA or global DP can reduce bandwidth while preserving privacy. Proximal methods and careful client selection mitigate the negative effects of heterogeneity. Personalization strategies can absorb some privacy-induced utility loss by focusing on local model adaptation.

Designing an FL system for edge AI requires a principled combination: **choose privacy primitives that match adversarial assumptions**, then apply optimization techniques that preserve convergence guarantees while respecting resource budgets.

8. Challenges and Future Research Directions

Key theoretical and practical challenges include:

Tight Utility-Privacy Bounds for Deep Models: Existing DP analyses are loose for non-convex deep learning; deriving tighter bounds remains open.

Robust Secure Aggregation with Dropouts: Protocols must be robust to high client churn with provable security.

Lightweight Cryptography for Edge Devices: New HE schemes or hybrid cryptographic primitives tailored for constrained hardware are needed.

Unified Theoretical Models for Heterogeneous FL: Convergence analyses that capture system heterogeneity, statistical heterogeneity, and privacy noise in one framework are lacking.

Benchmarking and Standardization: Theoretical comparison requires standardized assumptions and formal benchmarks (e.g., agreed threat models and noise budgets).

Federated On-Device Learning: The theory behind continual, online, and on-device learning with privacy constraints is underdeveloped. Addressing these will deepen the theoretical foundations of FL and improve real-world deployments.

9. Conclusion

This paper provided a theoretical comparative study of core privacy-preserving mechanisms and optimization techniques in federated learning for edge AI. By analyzing privacy models, threat assumptions, utility trade-offs, communication and computation overheads, and scalability, we offer principled guidance for selecting methods under various deployment constraints. Future theoretical work must strive for tighter bounds, robust secure protocols, and cryptographic advances compatible with edge constraints.

References

1. McMahan, H. B., Moore, E., Ramage, D., Hampson, S., & Aguera y Arcas, B. (2017). Communication-Efficient Learning of Deep Networks from Decentralized Data. *AISTATS (Federated Learning workshop)*. arXiv:1602.05629.

2. Dwork, C., & Roth, A. (2014). The Algorithmic Foundations of Differential Privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4), 211–407.

3. Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016). Deep Learning with Differential Privacy. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS)*.

4. Bonawitz, K., Ivanov, V., Kreuter, B., et al. (2017). Practical Secure Aggregation for Privacy-Preserving Machine Learning. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*.

5. Paillier, P. (1999). Public-key Cryptosystems Based on Composite Degree Residue Classes. *Advances in Cryptology — EUROCRYPT '99*, LNCS 1592, 223–238.

6. Gentry, C. (2009). A Fully Homomorphic Encryption Scheme. *Stanford University PhD Thesis*.

7. Bonawitz, K., Ivanov, V., Kreuter, B., et al. (2017). Practical Secure Aggregation for Privacy-Preserving Machine Learning. *arXiv:1611.04482*.

8. Kairouz, P., McMahan, H. B., et al. (2019). Advances and Open Problems in Federated Learning. *arXiv:1912.04977*.

9. Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Processing Magazine*, 37(3), 50–60.

10. Sun, Y., et al. (2020). Federated Learning with Compression and Quantization: Convergence Analysis. *IEEE Transactions on Signal Processing*.

11. Bonawitz, K., Ivanov, V., Kreuter, B., et al. (2019). Practical Secure Aggregation for Privacy-Preserving Machine Learning. *Communications of the ACM*
(overview).

12. Hitaj, B., Ateniese, G., & Perez-Cruz, F. (2017). Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*.

13. McMahan, H. B., et al. (2018). A Survey on Federated Learning: Concepts, Applications and Challenges. *Foundations and Trends in Machine Learning*.

14. Chen, T., et al. (2020). Efficient Federated Learning via Adaptive Client Selection. *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.

15. Konecny, J., McMahan, B., Yu, F. X., Richtárik, P., Suresh, A. T., & Bacon, D. (2016). Federated Learning: Strategies for Improving Communication Efficiency. *arXiv:1610.05492*.