International Journal of Computers and Informatics

Journal Homepage: https://www.ijci.zu.edu.eg

Int. j. Comp. Info. Vol. 11 (2026) 59–73

Paper Type: Original Article

A Systematic Review of Communication-Efficient Federated Learning Through Lossy and Lossless Compression

Aya Hesham [1,2,*], Ahmad Salah [3], Marwa Abdellah [1], and Gamal M. Behery [4]

[1] Faculty of Computers and Informatics, Zagazig University, Zagazig, Egypt; ayahesham21@fci.zu.edu.eg; mmaboelazm@fci.zu.edu.eg;

[2] Computer Science Department, Higher Technology Institute, Tenth of Ramadan, Egypt; aya.hesham@hti.edu.eg;

[3] Faculty of Computing and Information Sciences, Technology and Applied Sciences University, Ibri, Sultanate of Oman; ahmad.salah@utas.edu.om;

[4] Faculty of Computers and Artificial Intelligence, Damietta University, Damietta, Egypt; gbehery@du.edu.eg;

Received: 11 Feb 2026 **Revised**: 25 May 2026 **Accepted**: 26 Jun 2026 **Published**: 28 Jun 2026

Abstract

The shift toward decentralized machine learning has positioned Federated learning (FL) as an appealing solution for privacy-preserving collaborative model training, but its practical implementation is still limited by a fundamental bottleneck: the communication overhead. This expense is more than just a hassle in settings with constrained bandwidth and erratic network conditions; it is a constraint that determines whether FL can work at all. Reducing this cost while maintaining model performance has become a key research challenge in the FL community. This paper reviews compression techniques proposed to address communication overhead in FL, covering 25 studies organized under two compression strategies: lossy compression, encompassing quantization, pruning, sparsification, and knowledge distillation, and lossless compression. Each strategy is analyzed in terms of how it reduces the amount of data transmitted between clients and the server, and what impact it has on model accuracy. The review shows that each technique gives a different perspective of the problem and achieves different levels of communication reduction depending on the model, dataset, and system constraints. This reflects the larger truth that there is no single best solution and the best solution is dependent on the specific requirements of a particular deployment environment. This review aims to provide researchers with a better understanding of the available options, and helps guide more informed decisions in the construction of communication-efficient FL systems.

Keywords: Federated Learning, Communication Efficiency, Quantization, Pruning, Sparsification, Knowledge Distillation, Lossless Compression.

1 | Introduction

1.1 | Background and Motivation

The rapid expansion of connected devices and the increasing sensitivity of data provided by users have fundamentally changed how machine learning systems are built and deployed. In many real-world applications, it is either impractical or restricted to collect raw data into a central server. In particular, in domains such as healthcare, finance and telecommunication, privacy regulations are strict and data is inherently distributed across many locations. FL was proposed as a direct response to this challenge,

Corresponding Author: aya.hesham21@fci.zu.edu.eg

suggesting that instead of moving data to the model, the model should be moved to the data [1]. In this paradigm, participating clients train on their own data locally and only share model updates with a central server, which aggregates the updates to an improved global model without observing the data.

While this design elegantly addresses the data privacy problem, it shifts a significant burden onto the communication infrastructure. Every training round involves sending large model updates from potentially thousands of clients to the server and back, and this cycle must repeat many times before the global model converges. In large neural networks, where the number of trainable parameters can reach into the hundreds of millions, the sheer volume of data exchanged per round is enormous. When this is multiplied across many clients and many rounds, the total communication cost becomes one of the most serious practical obstacles to deploying FL at scale.

What makes this challenge particularly difficult is that it does not have a single clean solution. The communication burden in FL stems from several interacting factors: the size of the model itself, the number of rounds needed for convergence, the heterogeneity of client devices and network conditions, and the additional overhead introduced by privacy mechanisms such as encryption and secure aggregation. Therefore any effective solution must balance between reducing communication cost and maintaining model quality, training stability and data privacy carefully. This tension has led to an explosion of investigations on compression techniques that seek to reduce what is transmitted, how often, and in what form, of communication, making communication efficiency one of the most prominent and significant research directions in the FL literature today.

1.2 | Survey Scope and Research Questions

Recently, the amount of research on Communication-Efficient FL has grown significantly. Many compression strategies with different assumptions, mechanisms, and target environments. Understanding this landscape requires more than simply listing individual approaches — it actually requires a structured lens through which to examine their similarities, differences and trade-offs together. This survey attempts to provide such a lens by organizing the literature around two fundamental questions: what makes communication in FL expensive in the first place, and what compression approaches have been proposed and what they trade off in terms of model accuracy in exchange for reduced communication cost.

In terms of scope, this survey focuses on compression techniques applied specifically within the FL setting, examining how each method reduces communication overhead and what it trades off in terms of model accuracy. The remainder of the paper is organized to reflect this, beginning with FL fundamentals in Section 2, followed by an analysis of communication overhead in Section 3, and a survey of compression techniques covering quantization, pruning, sparsification, knowledge distillation, and lossless compression in Section 4, with a comparative analysis closing the survey.

2 | Federated Learning Fundamentals

To understand the communication challenges in FL we first need to see how the system is structured and how it operates. In this section we discuss three core elements of FL: i) architectural variants, ii) privacy mechanisms, and iii) communication protocols that govern the data exchange between clients and the central server.

2.1 | FL Architectures

FL architectures can be classified according to different perspectives, such as the network structure [2] or the data distribution across the participating clients [3]. Such categorizations aid in understanding the design and implementation of FL systems in various real-world contexts.

- According to network structure, FL architecture can be either classical FL or hierarchical FL. In classical FL [1], a single central server orchestrates the training process by aggregating the model

updates from clients. This architecture is easy to implement, and its design is straightforward, which makes it widely used. However, it could suffer from communication bottlenecks and scalability problems when the number of clients rises significantly. In hierarchical FL [4], an intermediate layer of edge servers is utilized for local model aggregation, after which these edge servers transmit the aggregated models to a central server. The hierarchical structure improves scalability and reduces communication overhead between the clients and the central server, thereby enabling a more efficient use of network resources, especially in large scale environments.

- FL architecture can be categorized into various types depending on the data distribution across clients. In Vertical FL [5], the datasets of different clients have the same sample space but different characteristics. The datasets are complementary and clients collaborate to train a model while jointly preserving those characteristics. Architecture is helpful in contexts where several organizations hold different attributes about the same set of users. In Horizontal FL [6], data are divided horizontally between clients, where each client holds the same feature space but different samples. This is the most common architecture, in which each client trains the model locally using its own data, and the resulting updates are aggregated to improve the global model. Hybrid FL [7] combines horizontal and vertical FL for more flexible and adaptive collaboration among clients and is particularly useful in complex real-world situations arising from data heterogeneity in both samples and features. In Federated Transfer Learning [8], a pre-trained model is shared among clients, and each refines it with local data, making this approach especially helpful when data distributions differ substantially or when clients have limited data. In Decentralized FL [9], clients communicate directly with one another to train the model without the need for a central server; this improves reliability but introduces communication-related challenges. In Cross-Silo FL [10], data is dispersed among different organizations, promoting cooperation while maintaining data separation; well-resourced participants in this setting tend to guarantee reliable communication and effective training. Finally, Edge FL [11] focuses on training models directly on edge devices without sharing local data, which enhances privacy and reduces latency but may encounter challenges related to communication and computation limitations. Figure 1 illustrates a visual summary of these architectural categories.

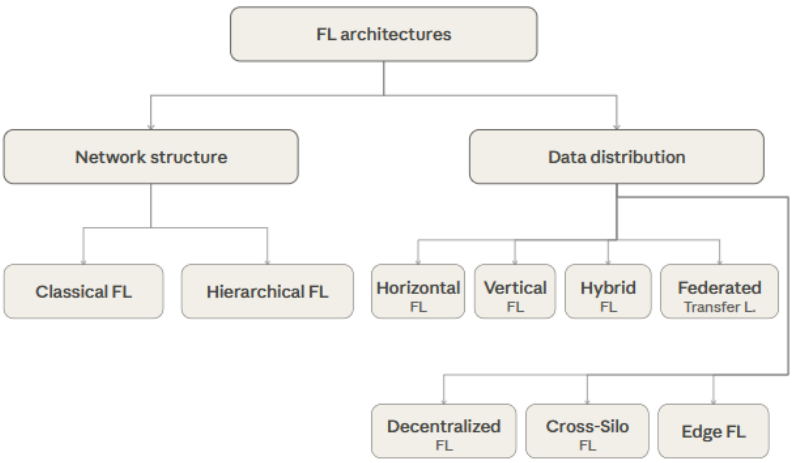

Figure 1. Federated Learning architecture categories.

2.2 | Privacy Aspects

The issue of privacy protection in ML has consistently been a prominent research topic prior to the debut of FL. As researchers have increasingly focused on privacy, FL has emerged as a promising new context in ML, and its research trajectory is strongly tied to earlier privacy protection strategies. FL improves privacy by training models directly on the edge devices without sharing local sensitive data [12]. This reduces the risks of data breaches, unauthorized access and regulatory non-compliance in the industry, especially in sensitive

sectors such as finance, healthcare, mobile applications, e-commerce, telecommunications, autonomous vehicles, cybersecurity and smart cities.

FL is very useful in different sectors because of its decentralized and privacy- preserving nature. This allows hospitals to collaborate on ML models without sharing sensitive patient data [13, 14]. The finance sector applies FL to detect fraud and assess risk while preserving consumers data privacy [15, 16]. FL allows for personalized recommendations in retail and e-commerce without revealing individual user information [17]. In autonomous vehicle systems, FL is applied to improve the driving model across multiple vehicles without sharing the precise location information [18, 19]. Similarly, in Cybersecurity, FL improves threat detection capabilities while keeping the sensitive data confidential [20, 21, 18]. FL is also used in telecommunications and smart cities to optimize networks and public services while maintaining user privacy [22, 23].

However, with the protection of FL security mechanisms, the FL system is still vulnerable to various attacks that would compromise the privacy of the participant data and reliability of the whole system [24]. FL adopts the conventional encryption techniques to secure the communication of parameters [25]. Other popular methods include differential privacy, homomorphic encryption, multiparty computing, secure aggregation, and classical symmetric and asymmetric encryption [26-28].

2.3 | Communication Protocols

Communication protocols are important in FL to make the interaction between edge devices and central server efficient and secure. FL trains ML models across multiple decentralized clients without sharing raw data, which incurs significant communication overhead due to the frequent exchange of model updates, especially in environments with limited bandwidth and unstable network connections. Many FL frameworks are built on Transmission Control Protocol (TCP) [29], which guarantees the full transmission and integrity of model updates. However, TCP, although providing strict guarantees on data integrity, also increases the communication overhead, especially under poor network conditions with packet loss leading to frequent retransmissions. To overcome these problems, researchers have studied different protocols as, for instance, Message Queue Telemetry Transport [30] and the Advanced Message Queuing Protocol [31], which offer lower latency and better adaptation in dynamic or constrained network environments. Ultimately, communication protocols in FL must balance bandwidth efficiency, latency reduction, privacy preservation, and reliability.

3 | Communication overhead in FL

One of the major challenges in FL is the high communication overhead [32]. There are various sources contributing to this overhead. The three most common direct sources are examined first.

Large model size is a primary source of overhead. Throughout each communication round, the central server sends its global model to all participating clients, and clients in turn send their model updates back to the server. In deep neural networks, which may include millions of parameters, this procedure results in the exchange of millions of bytes per client per round. When multiplied across a large number of participating clients, the total communication volume becomes substantial [33].

Frequent communication rounds also contribute significantly. FL depends on iterative training, in which the global model improves gradually with each round. During each round, participating clients perform a number of local training epochs, transmit their updates to the central server, and receive an updated global model. This iterative back-and-forth process, even with small updates, can incur considerable latency and bandwidth consumption over dozens or hundreds of rounds [34].

Privacy-preserving protocols constitute a third direct source. Techniques such as secure aggregation, homomorphic encryption, or differential privacy add extra metadata on top of the raw model updates. These protections inflate message sizes and can require additional retransmissions to ensure the correctness and security of the transmission [26].

Beyond these main sources mentioned above, there are additional factors that contributes to the increase of communication overhead, these include stragglers (i.e., delayed or slower clients), fault-tolerance mechanisms such as retries and redundant client participation, and security mechanisms that add extra computations and data exchanges [1, 35, 27].

The build-up of communication overhead directly affects the system performance in several interconnected ways. Such high communication overhead has a direct impact on system latency and training efficiency. The frequent transmission of the model updates may significantly slow down the training process, and the existence of the stragglers also delays the aggregation process as the server needs to wait for all the selected clients before moving to the next round [1, 35]. Communication overhead also seriously burdens system resources and network bandwidth. The frequent communication of model parameters among a large number of clients consumes a significant fraction of available bandwidth, which may lead to network congestion and reduce the system efficiency. In addition, the involved devices have to allocate additional memory, computational resources, and energy to both communication and local training tasks, and the use of privacy-preserving techniques makes this problem more challenging [26, 27]. In addition to latency and resource constraints, communication overhead may also impact model convergence and accuracy. High overhead often forces to decrease the number of communication rounds and/or to employ partial client participation or model compression strategies. These strategies reduce overhead but can lead to slow convergence or approximation errors that degrade the quality of the learned model. Poor communication can hinder effective aggregation when data are non-IID and heterogeneous [36, 32].

There is a large body of work focusing on different optimization techniques to reduce the communication overhead in FL [37, 38]. One of the most popular ways is to use compression techniques to reduce the size of the transmitted updates. Methods for quantization, sparsification, and gradient compression reduce the amount of data sent between clients and the central server [39-41], while maintaining model performance at a reasonable level by only transmitting the most relevant information or by expressing updates in a more concise manner [42]. Other techniques have been explored such as pruning updates and only transmitting changes between rounds [43-45].

Another research direction is to reduce the communication frequency rather than the size of transmitted data. Clients are allowed to execute several local training iterations before transmitting updates to the central server, which reduce the total number of communication rounds required for convergence [46]. Techniques such as client selection and partial participation reduce the number of participating clients in each round, thus reducing the total communication load [47, 48]. The asynchronous FL has been proposed to avoid synchronization delays by allowing clients to communicate independently without waiting for other clients, which is particularly useful in heterogeneous environments [49].

Hybrid methods that combine compression with dynamic communication scheduling, or adaptive client selection have also been investigated for further overhead reduction [50-52]. Hybrid methods aim at improving the overall efficiency by combining complementary techniques to trade-off system performance with accuracy and resource consumption.

4 |Compression techniques in FL

4.1 |Lossy Compression Methods

Lossy compression techniques in FL context are designed to reduce the size of transmitted model updates by either by approximating less important information or removing them. These methods can lead to higher compression rates by allowing a controlled loss of accuracy, which helps reduce communication overhead. Among lossy compression methods used in FL, quantization, pruning, sparsification and knowledge distillation are the most common techniques. Each one of them has its trade-offs between efficiency and model accuracy.

- Quantization is one of the most popular methods used for compression. In FL, quantization reduces the numbers of bits used to represent model weights or gradients prior to transmission, directly decreasing communication costs. For example, [53] quantizes local updates under privacy constraints, which blends lossy compression and privacy enhancements. In [54], the authors introduce a quantization method for deep networks to reduce floating-point computation without compromising user data security, showing that low-precision representations can also obtain accurate distributed models. The Low Huffman-Coded Delta Quantization (LHDQ) [55] can quantize parameters at an efficient rate of 5.3-bit per parameter. This method reduces the number of transmitted bits and achieves faster convergence, with a minimal loss in accuracy. Work in [56] combines ternary quantization and heuristic sparsification to compress updates while keeping critical information, thereby reducing communication requirements in mobile and IoT environments. More advanced quantization strategies have been proposed to improve performance and adaptability in FL environments. [57] presented a multigrained quantization scheme that accounts for the heterogeneous importance of model parameters, enabling more accurate reconstruction of low-precision weights while preserving convergence guarantees. [58] proposed an adaptive gradient quantization method that adjusts the quantization levels according to the importance of gradients, thus improving communication efficiency while maintaining model accuracy. In [59], a randomized quantization approach is introduced to combine privacy to the quantization process itself, which improves the communication efficiency and data protection compared with traditional sequential methods. [60] presented a layerwise adaptive quantization scheme based on both the dual bit quantization and the differential privacy to significantly reduce the communication overhead while protecting the model performance. These methods are part of a growing trend of adaptive and privacy-aware quantization algorithms that improve the trade-off between compression efficiency, accuracy and security in FL. [58] presents an adaptive gradient quantization method that adjusts the quantization levels according to the importance of gradients, thus improving the communication efficiency while preserving the model accuracy. In [59], a randomized quantization approach is presented to combine the privacy to the quantization process itself, which improves both the communication efficiency and the data protection compared with traditional sequential methods. [60] introduced a layer-wise adaptive quantization scheme based on both the dual bit quantization and the differential privacy to significantly reduce the communication overhead while preserving the model performance. These methods are part of a growing trend of adaptive and privacy-aware quantization algorithms that improve the trade-off between compression efficiency, accuracy and security in FL.

- Pruning is another common lossy compression technique in FL that reduces communication overhead by setting less important model weights to zero or removing them. [43] proposed a layer-adaptive method to assign pruning rates based on layer sensitivity and network depth, achieving up to 68% reduction on the communication cost without much accuracy drop. [61] proposed an adaptive pruning method over heterogeneous clients which is resource-efficient by adaptively adjusting the pruning ratio according to the importance of model weights. [45] obtained sparse weight representations by single-shot pruning to reduce the data transmission without loss of model performance. [62] proposed a combination of adaptive pruning and system-aware optimization for internet of vehicles applications, and show that well-designed pruning can improve convergence and reduce training latency and communication overhead. [44] proposed a reinforcement learning-based federated pruning framework, which can dynamically adjust the pruning rates across heterogeneous devices and utilize sparse aggregation strategies. It can achieve higher model sparsity, accuracy and convergence speed, and lower communication costs.

- Knowledge distillation (KD) is a lossy compression technique that enables a smaller student model to learn from a larger, well-trained teacher model. In FL, KD allows clients to share only the essential knowledge of a model, rather than all of the weights, which reduces the communication overhead

while maintaining the efficacy of learning. Classical KD methods enable clients to share compact representations from a single teacher rather than the whole models [63, 64]. Adaptive and mutual KD frameworks enable clients to collectively learn knowledge from multiple teachers or global models. They can well solve the problems of model divergence, client heterogeneity, and non-IID data by adaptively adjusting the distillation process, which improves the convergence and robustness of the model [65-67].

- Sparsification decreases the communication overhead by only sending the most significant updates to the model, ignoring by that the weights with small magnitudes, thus reducing the volume of data transmitted from clients to the server. A popular method is top-k sparsification where only the most important updates are selected to be transmitted [68]. This approach combines sparsification with the secure aggregation and differential privacy to ensure privacy while maintaining the effectiveness of the FL model. [69] proposed a dynamic threshold-based sparsification method, in which the sparsification threshold is dynamically generated to compress model updates. A compensation mechanism is used to recover missing sparse updates, thus avoiding convergence bias caused by information loss. [70] suggested a Communication-Efficient FL system by combining random sparsification with an error-compensation mechanism, where only a randomly selected subset of the gradient elements is transmitted and only the selected gradient elements are recompensated by locally accumulated errors. [71] proposed a structural sparsification method with learnable thresholds per filter, which enables clients to send a compact threshold vector instead of the whole model parameters to reduce the overhead and allows each client to maintain a model adapted to its local data.

In conclusion, lossy compression techniques can reduce the communication cost significantly and are an essential part of Communication-Efficient FL. Accuracy is compromised but careful design and combination of quantization, pruning, sparsification, and knowledge distillation can yield good results.

4.2 |Lossless Compression Methods

Lossless compression methods aim to reduce communication overhead in FL without any loss of information from communicated model updates. Lossy methods modify or approximate gradients to reach high compression ratios, while lossless methods ensure that the original data is perfectly recoverable at the receiver, maintaining model performance. This property is especially useful when the overhead of the full precision calculation is more important than aggressive reduction.

In practice, lossless compression is commonly applied in FL settings for gradients or parameter updates sent over bandwidth-limited channels, with the aim of eliminating overhead while guaranteeing exact recovery. Techniques based on entropy, such as Huffman coding or arithmetic coding, reduce the communication overhead by assigning shorter codes to more frequent gradient values, and the effectiveness of these techniques depends on the approximation of the distribution of the gradient. [72] showed that the statistical behavior of the gradient entries can be better modeled by a generalized normal distribution, which allows more efficient entropy coding, thus reducing the communication cost while preserving the integrity of the transmitted information.

In general, lossless and lossy compression methods have a basic trade-off between efficiency of communication and preservation of information. Lossy methods allow for much higher compression ratios with controlled approximation errors, while lossless methods allow for exact reconstruction but with lower compression gains. In recent work, lossless methods are rarely used on their own, but rather combined with other compression strategies to improve both communication and model performance simultaneously. [73] employs a combination of bidirectional dynamic quantization and bitmap-based lossless compression, which can achieve compression rates up to 24.21× over FedAvg with a maximum accuracy degradation of less than 1.5%. [74] unified the two-stage lossy and lossless compression in one pipeline, and shows that the

combination can obtain significant reduction in communication cost while maintaining the accuracy loss below 0.5%.

Several important advantages are offered by lossless compression methods in FL systems. They are simple to implement, do not require retraining the model and can provide stable performance for heterogeneous client environments. They are therefore well suited for applications where high reliability and exact reproduction of model updates are needed. However, due to the inherent compression ratio limit, they are less effective in situations with extremely limited bandwidth. Hence, in recent FL systems, lossless methods are mainly employed in hybrid compression frameworks, which combine lossy and lossless techniques to strike a better balance between model performance and communication efficiency.

4.3 | Comparative Summary of Compression Methods

Table 1 present a comparative summary of the compression methods surveyed in this paper, organized by Reference number, Year, Compression Technique, Communication Reduction, Accuracy Impact, Dataset(s), Model Architecture, Privacy Aware, and Baseline Compared to.

Table 1. Comparative Summary of Compression Methods Used in Federated Learning to Reduce Communication Overhead.

Ref	Year	Compression Technique	Communication Reduction	Accuracy Impact	Dataset(s)	Model Architecture	Privacy Aware	Baseline Compared to
Quantization								
[54]	2020	Learnable quantization (OQFL)	4-bit vs 32-bit FL — bit-width reduction	CIFAR-10: ResNet18 91.6% (≈FL 91.73%); MobileNetV2 88.49% ; VGG16 86.96%	CIFAR-10	ResNet18, PreActResNet18, MobileNetV2, VGG16	No	32-bit FedAvg
[53]	2023	Joint vector quantization + (JoPEQ)	Bits per round reduced from ~10^7 to ~1.7×10^5 at R=1	MNIST: 0.70–0.84; CIFAR-10: 0.68–0.71	MNIST, CIFAR-10	Linear regression, MLP, CNN	Yes LDP	FL, FL+SDQ, FL+Lap, FL+Lap+SDQ, MVU
[55]	2024	Delta quantization + Huffman coding (LHDQ)	57.7% less TX time vs 16-bit; 18.5% fewer bits vs 32-bit	LHDQ: 72% vs 32-bit: 74.1%, 16-bit: 73.9% (~2% drop)	CIFAR-10	CNN	No	16-bit & 32-bit FL
[57]	2025	Multi-grained quantization;	85.11% comm. time reduction; 81.71% TX load reduction vs FedAvg	MNIST: 98.21%; CIFAR-10: 44.01%; Fashion-MNIST: 72.89% (non-IID; best vs 7 baselines)	MNIST, CIFAR-10, Fashion-MNIST	CNN, AlexNet, LeNet	No	FedAvg + 7 baselines
[58]	2025	Gradient innovation quantization	faster convergence in fewer rounds no exact reduction in % reported	High accuracy on all setting outperforms baselines ‡	MNIST, CIFAR-10	CNN, ResNet-20	No	FedAvg, FedQSGD, FedProx

[59]	2025	Unified randomized quantization	Outperforms DP-FedPAQ by 0.97%–27.86%; best gains under non-IID	MNIST-CNN: 88.04%; FMNIST-CNN: 80.03%; CIFAR-10: 57.14% (ε=8.0, 4-bit)	MNIST, Fashion-MNIST, CIFAR-10, FEMNIST	CNN, MLP	Yes LDP	DP-FedPAQ, RQM
[60]	2025	Dual-bit deterministic quantization (8-bit local / 2-bit TX)	75% communication overhead reduction; 30–40% fewer rounds; ~70% energy savings vs FedAvg	MNIST: 99.41%; FMNIST: 91.06%; CIFAR-10: 82.94% (ε=2.25)	MNIST, Fashion-MNIST, CIFAR-10	CNN	Yes RDP	FedAvg, SOTA privacy FL
[56]	2025	Compressed sensing + ternary quantization + dual-threshold sparsification	Up to 88.23% upload and 87.26% download reduction vs FedAvg (MNIST); 79.74%/78.08% on CIFAR-10	Outperforms baselines in most settings; slightly lags SignSGD in early non-IID training ‡	MNIST, Fashion-MNIST, CIFAR-10	CNN1, CNN2, LeNet-5+	No implicit encrypt-ion	FedAvg
Pruning								
[45]	2025	One-shot Unstructured Pruning (SNIP) + CSR Sparse Compression	UCI-HAR: 0.97 GB vs FedAvg 4.65 GB (θ=0.9)	Competitive with FedAvg; stable across configs ‡	CIFAR-10, UCI-HAR (non-IID)	AlexNet, Simple CNN	No	FedAvg, FedDLR
[62]	2024	Adaptive pruning + vehicle selection (VFed-AMP)	75% uplink saving vs CVFL; 21.3% vs FedAvg (MNIST); 1.8× faster training (CIFAR-10)	MNIST: 94.85%; BelgiumTS: +13.4% vs baselines; CIFAR-10: 35% at round 90	MNIST, CIFAR-10, BelgiumTS (non-IID)	CNN	No	FedAvg, PruneFL, Random Mask, CVFL
[43]	2026	Layer-adaptive pruning	68.3% communication reduction vs FedAvg	CIFAR-10: 88.7–92.9%; MNIST: 96.4–98.4%; FMNIST: 86.8–90.8%; accuracy loss <2%	CIFAR-10, MNIST, Fashion-MNIST	ResNet-18, CNN	No	FedAvg, FedDST, FedProx
[61]	2025	Dynamic per-layer adaptive pruning based on weight importance	25% avg. reduction; >50% on MNIST vs FedAvg; 63% FLOPs reduction on MNIST	MNIST: 98.33%; CIFAR-10: 90.32%; CIFAR-100: 75.52%; Tiny-ImageNet: 58.33%	MNIST, FMNIST, CIFAR-10, CIFAR-100, SVHN, Tiny-ImageNet	CNN, ResNet-18	No	FedAvg, SOTA

‡ Exact accuracy values not reported by authors; results presented graphically only.

[44]	2024	RL-based dynamic pruning(RBGP)+ sparse aggregation (FedSA)	FedSA improves sparsity by up to 18.9% vs baselines	outperforms baselines in accuracy & convergence speed across all datasets ‡	CIFAR-10, MNIST, Fashion-MNIST	VGG-19, ResNet-18	No	PruneFL, PQSU
Sparsification								
[68]	2025	Top-k gradient sparsification +	4.25×–6.75× overhead reduction vs SecAgg; ~2× less than Top-k sparseSecAgg	Comparable to Top-k sparseSecAgg at ϵ=5; matches Rand-k at ϵ=1 ‡	MNIST, CIFAR-10	CNN	Yes ϵ-DP + SecAgg masking	SecAgg, Top-k sparseSecAgg, Rand-k
[70]	2025	Random shared-index sparsification	80% accuracy at 4% of FedAvg's communication cost	Outperforms all baselines on both IID and non-IID settings ‡	CIFAR-10, GTSRB	ResNet-20	No	FedAvg, FLARE, Top-K, TopRand
[69]	2025	Dynamic threshold sparsification (UD-DTS)	Reaches 70% acc. by round 6 vs 17/15/13 for Top-k/CRM/EAM; fastest convergence	MNIST: 88.13% (FedAvg: 95.51%, Top-k: 85.02%); FMNIST: 71.02% (FedAvg: 80.08%)	MNIST, Fashion-MNIST	MLP, CNN	No	FedAvg, Top-k, EAM, CRM
[71]	2026	Structured sparsification via learnable per-filter threshold gating (TFD)	0.93 MB/round vs FedAvg 270 MB/round; cumulative 194 MB vs 56.6 GB at 200 rounds	Accuracy: 0.918; AUROC: 0.934; F1: 0.896 (vs FedAvg 0.932 / 0.943 /0.911)	FaceForensics++, Celeb-DF (IID)	R(2+1)D-18	No	FedAvg, Local baseline
Knowledge Distillation								
[66]	2022	Adaptive mutual KD (mentor↔mentee) + SVD-based dynamic gradient compression	Up to 94.89% comm. cost reduction vs FedAvg	MIND AUC: 71.0%; ADR F-score: 60.7%; NER CADEC: 67.7%, ADE: 87.4%	MIND, ADR, CADEC, ADE, SMM4H	UniLM-Base (mentor), UniLM (mentee)	No	FedAvg, FetchSGD, FedDropout, FedPAQ
[67]	2023	Historical global model ensemble as KD teacher (FedGKD / FedGKD-VOTE)	Reduces client drift — fewer effective rounds; no extra uplink cost	CIFAR-10: 73.06%; CIFAR-100: 37.29%; Tiny-ImageNet: 35.72%; AG News: 89.60%	CIFAR-10, CIFAR-100, Tiny-ImageNet, AG-News, SST-5	ResNet-8, ResNet-50, DistilBERT	No	FedAvg, FedProx, SCAFFOLD
[65]	2024	Bidirectional KD (global - local); local model as implicit gradient noise	Implicit — no full model weights transmitted upstream	CIFAR-10: 94.2%/93.1%; FMNIST: 96.1%/95.0% (local/global)	CIFAR-10, FMNIST, AG-News, SST2	ResNet-18, VGG-16, MobileNet, DistilBERT	Yes Gradient privacy (implicit)	FedMD, FedAvg, FedProx

[64]	2025	KD transmitting only encrypted soft-labels; SMPC protection	92.4% comm. reduction vs FedAvg	99% accuracy & F1-score (binary & multi-class)	WSN-DS, UNSW-NB15	TinyML student + central teacher	Yes SMPC	FedAvg
[63]	2025	Mutual KD + dynamic local rounds + adaptive distillation weighting; student-only uplink	4.89×–28.45× compression vs FedMD; 4.34× convergence speedup	Fashion-MNIST: 90.95% (−0.47%); CIFAR-100: 69.48–75.13%; Mini-ImageNet: 67.96%	Fashion-MNIST, CIFAR-100, Mini-ImageNet	Custom CNN, ResNet50/101/18	No	FedMD, Fed-MKDFW
Lossless Compression								
[72]	2022	Gradient Quantization + Lossless Compression	Outperforms Norm+Huffman and LZ78 in compression ratio across all layers	No accuracy degradation from lossless step; ResNet50V2 ~0.65–0.75 (CIFAR-10, epoch 100)	CIFAR-10	DenseNet121, ResNet50V2, NASNetMobile	No	Norm+Huffman, LZ78
[74]	2024	SZ2 lossy (weights) + blosc-lz lossless (metadata); EBLC pipeline	5.55–12.61× compression at REL=10^{-2}; 13.26× comm. time reduction at 10Mbps	Within 0.5% of uncompressed at REL=10^{-2}; AlexNet CIFAR-10: 57.90%	CIFAR-10, Fashion-MNIST, Caltech101	AlexNet, MobileNetV2, ResNet50	No	Uncompressed FL
[73]	2025	Bidirectional dynamic quantization + bitmap lossless compression (FedBDQB)	24.21× compression vs FedAvg; 12.87× vs FedDQ; compute time <2× FedAvg	MNIST: 99.14%/97.98%; CIFAR-10: 84.18%/83.57% (IID/Non-IID); drop <1.5%	MNIST, CIFAR-10 (IID & Non-IID)	LeNet-5, ResNet-18	No	FedAvg, FedDQ

Main observation of Table 1 An overview of these researches shows that compression techniques are a most common and varied approach to reduce communication overhead in FL, with different categories addressing the problem from different perspectives. Quantization reduces the number of bits used to represent model updates; sparsification only transmits the most significant gradient values; pruning removes less important weights before transmission; knowledge distillation replaces full weight vectors with compact knowledge representations; and lossless compression encodes updates more efficiently without any information loss. However, despite this variety, there is a consistent trade-off between compression ratio and model accuracy across all categories. Quantization and sparsification methods generally achieve the largest communication reduction. Some methods report more than 80% reduction in communication overhead, but often with a measurable accuracy degradation, particularly in the presence of non-IID data distributions. Although pruning offers a compromise with weight selection based on importance, knowledge distillation adds extra training complexity at the cost of architectural flexibility. Lossless compression achieves more modest gains and is most effective when combined with lossy methods in a hybrid pipeline.

4 | Conclusion

This survey has considered communication overhead as one of the important practical barriers for deploying FL at scale. It has surveyed 25 studies on compression techniques presented to tackle this issue. The survey

first introduced the fundamentals of FL to set the context, then discussed the sources and effects of communication overhead, and finally reviewed the compression techniques under two strategies: lossy compression including quantization, pruning, sparsification, and knowledge distillation, and lossless compression. These approaches together constitute the main methods used so far by the FL community to reduce the amount and rounds of data transmitted between clients and the central server, while carefully balancing the trade-off between communication efficiency and model accuracy.

The review shows that each category offers a different perspective on the problem. Quantization has advanced from fixed low-bit schemes to adaptive, layer-wise and privacy-aware variants. Pruning removes unimportant weights before transmission. Recent approaches combine reinforcement learning and importance-based adaptation to obtain high sparsity with low accuracy degradation. Sparsification only transmits the most important gradient updates and can significantly compress the transmitted data when combined with error-compensation and privacy guarantees. Knowledge distillation transfers compact knowledge representations, instead of the full weight vectors, which makes it more feasible for heterogeneous clients to collaborate efficiently under non-IID data. Lossless compression provides exact reconstruction, but more modest gains, and is becoming more common in hybrid pipelines with lossy methods. The best solutions are those that consider the model, the data and the deployment environment instead of applying a fixed strategy for all. The comparative analysis shows that no one technique can be considered as the best, as the selection depends on the constraints and requirements of the specific system.

Several challenges are still open. Many of the methods reviewed are evaluated under idealized or relatively-controlled conditions, and their performance under highly heterogeneous real-world network environments is less well characterized. The interaction of compression with privacy-preserving mechanisms such as differential privacy and secure aggregation requires further investigation. Aggressive compression may disrupt the noise calibration and security guarantees of these protocols. Moreover, the majority of the surveyed works are dealing with a single compression category separately. Hybrid approaches, combining complementary techniques within a single framework, are still under-explored and constitute a promising way to reach higher communication savings with a limited accuracy loss. It is expected that the systematic organization and the comparative study provided in this survey can serve as a useful reference for the researchers working on the Communication-Efficient FL and the identified gaps can help to direct the future work to more robust, adaptive and practical deployable solutions.

Author Contribution

All authors contributed equally to this work.

Funding

This research received no funding.

Data Availability

The datasets referenced in this study are fully anonymized and publicly accessible. As this work constitutes a systematic literature review, no new data were generated.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017, April). Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics (pp. 1273-1282). Pmlr. url: https://proceedings.mlr.press/v54/mcmahan17a.html.

[2] Al-Quraan, M., Mohjazi, L., Bariah, L., Centeno, A., Zoha, A., Arshad, K., ... & Imran, M. A. (2023). Edge-native intelligence for 6G communications driven by federated learning: A survey of trends and challenges. IEEE Transactions on Emerging Topics in Computational Intelligence, 7(3), 957-979. doi: 10.1109/TETCI.2023.3251404.

[3] Aggarwal, M., Khullar, V., & Goyal, N. (2024). A comprehensive review of federated learning: Methods, applications, and challenges in privacy-preserving collaborative model training. Applied Data Science and Smart Systems, 570-575. doi: 10 . 1201 / 9781003471059-73.

[4] Abad, M. S. H., Ozfatura, E., Gunduz, D., & Ercetin, O. (2020, May). Hierarchical federated learning across heterogeneous cellular networks. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 8866-8870). IEEE.

[5] Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology (TIST), 10(2), 1-19. doi: 10.1145/3298981.

[6] Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., ... & Roselander, J. (2019). Towards federated learning at scale: System design. Proceedings of machine learning and systems, 1, 374-388.

[7] Hao, M., Li, H., Luo, X., Xu, G., Yang, H., & Liu, S. (2019). Efficient and privacy-enhanced federated learning for industrial artificial intelligence. IEEE Transactions on Industrial Informatics, 16(10), 6532-6542. doi: 10.1109/TII.2019.2945367.

[8] Liu, Y., Kang, Y., Xing, C., Chen, T., & Yang, Q. (2020). A secure federated transfer learning framework. IEEE Intelligent Systems, 35(4), 70-82. doi: 10.1109/MIS.2020.2988525.

[9] Li, Y., Chen, C., Liu, N., Huang, H., Zheng, Z., & Yan, Q. (2020). A blockchain-based decentralized federated learning framework with committee consensus. Ieee Network, 35(1), 234-241. doi: 10 . 1109/MNET.011.2000263.

[10] Durrant, A., Markovic, M., Matthews, D., May, D., Enright, J., & Leontidis, G. (2022). The role of cross-silo federated learning in facilitating data sharing in the agri-food sector. Computers and Electronics in Agriculture, 193, 106648. doi: 10.1016/j.compag.2021.106648.

[11] Lim, W. Y. B., Luong, N. C., Hoang, D. T., Jiao, Y., Liang, Y. C., Yang, Q., ... & Miao, C. (2020). Federated learning in mobile edge networks: A comprehensive survey. IEEE communications surveys & tutorials, 22(3), 2031-2063. doi: 10.1109/COMST.2020.2986024.

[12] Li, Q., Wen, Z., Wu, Z., Hu, S., Wang, N., Li, Y., ... & He, B. (2021). A survey on federated learning systems: Vision, hype and reality for data privacy and protection. IEEE Transactions on Knowledge and Data Engineering, 35(4), 3347-3366. doi: 10.1109/TKDE. 2021.3124599.

[13] Guo, Y., Liu, F., Cai, Z., Chen, L., & Xiao, N. (2020, August). FEEL: A federated edge learning system for efficient and privacy-preserving mobile healthcare. In Proceedings of the 49th International Conference on Parallel Processing (pp. 1-11). doi: 10 . 1145/3404397.3404410.

[14] Lee, C. I., Tzeng, C. R., Li, M., Lai, H. H., Chen, C. H., Huang, Y., ... & Liu, M. (2024). Leveraging federated learning for boosting data privacy and performance in IVF embryo selection. Journal of Assisted Reproduction and Genetics, 41(7), 1811-1820. doi: 10.1007/s10815-024-03148-z.

[15] Dasari, S., & Kaluri, R. (2024). 2p3fl: A novel approach for privacy preserving in financial sectors using flower federated learning. Computer Modeling in Engineering & Sciences, 140(2), 2035. doi: 10. 32604/cmes.2024.049152.

[16] Rabbani, H., Shahid, M. F., Khanzada, T. J. S., Siddiqui, S., Jamjoom, M. M., Ashari, R. B., ... & Nooruddin, M. (2024). Enhancing security in financial transactions: a novel blockchain-based federated learning framework for detecting counterfeit data in fintech. PeerJ Computer Science, 10, e2280. doi: 10.7717/peerj- cs.2280.

[17] Li, J., Cui, T., Yang, K., Yuan, R., He, L., & Li, M. (2021). Demand forecasting of e-commerce enterprises based on horizontal federated learning from the perspective of sustainable development. Sustainability, 13(23), 13050. doi: 10.3390/su132313050.

[18] Al-Hawawreh, M., & Hossain, M. S. (2023). Federated learning-assisted distributed intrusion detection using mesh satellite nets for autonomous vehicle protection. IEEE Transactions on Consumer Electronics, 70(1), 854-862. doi: 10.1109/TCE.2023.3318727.

[19] Tian, Y., Wang, J., Wang, Y., Zhao, C., Yao, F., & Wang, X. (2022). Federated vehicular transformers and their federations: Privacy-preserving computing and cooperation for autonomous driving. IEEE Transactions on Intelligent Vehicles, 7(3), 456-465. doi: 10.1109/TIV.2022.3197815.

[20] Folino, F., Folino, G., Pisani, F. S., Sabatino, P., & Pontieri, L. (2024, March). A scalable vertical federated learning framework for analytics in the cybersecurity domain. In 2024 32nd Euromicro international conference on parallel, distributed and network-based processing (PDP) (pp. 245-252). IEEE. doi: 10.1109/PDP62718.2024.00041.

[21] Reddy, D. T., Nandigam, H., Indla, S. C., & Raja, S. P. (2024). Federated Learning in Data Privacy and Security. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal, 13, e31647-e31647. doi: 10.14201/adcaij.31647.

[22] Singh, S., Rathore, S., Alfarraj, O., Tolba, A., & Yoon, B. (2022). A framework for privacy-preservation of IoT healthcare data using Federated Learning and blockchain technology. Future Generation Computer Systems, 129, 380-388. doi: 10.1016/j.future.2021.11.028.

[23] Jiang, J. C., Kantarci, B., Oktug, S., & Soyata, T. (2020). Federated learning in smart city sensing: Challenges and opportunities. Sensors, 20(21), 6230. doi: 10.3390/s20216230.

[24] Cai, X., Geng, S., Zhang, J., Wu, D., Cui, Z., Zhang, W., & Chen, J. (2021). A sharding scheme-based many-objective optimization algorithm for enhancing security in blockchain-enabled industrial internet of things. IEEE Transactions on Industrial Informatics, 17(11), 7650-7658. doi: 10.1109/TII.2021.3054986.

[25] Çavuşoğlu, Ü., & Kökçam, A. H. (2021). A new approach to design S-box generation algorithm based on genetic algorithm. International Journal of Bio-Inspired Computation, 17(1), 52-62. doi: 10.1504/IJBIC.2021. 113872.

[26] Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., ... & Seth, K. (2017, October). Practical secure aggregation for privacy-preserving machine learning. In proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (pp. 1175-1191). doi: 10.1145/3133956.3133982.

[27] Geyer, R. C., Klein, T., & Nabi, M. (2017). Differentially private federated learning: A client level perspective. arXiv preprint arXiv:1712.07557. url: https://doi.org/10.48550/arXiv.1712.07557.

[28] Zhang, X., Fu, A., Wang, H., Zhou, C., & Chen, Z. (2020, June). A privacy-preserving and verifiable federated learning scheme. In ICC 2020-2020 IEEE International Conference on Communications (ICC) (pp. 1-6). IEEE. doi: 10.1109/ICC40277.2020.9149149.

[29] Vineeth, S. (2022). Federated learning over WiFi: Should we use TCP or UDP?. doi: 10.31219/osf.io/tuz6c.

[30] Hillar, G. C. (2017). MQTT essentials-a lightweight IoT protocol: the preferred IoT publish-subscribe lightweight messaging protocol. Packt Publishing Ltd.

[31] Kramer, J. (2009). Advanced message queuing protocol (AMQP). Linux Journal, 2009(187), 3.

[32] Kairouz, P., & McMahan, H. B. (2021). Advances and open problems in federated learning. Foundations and trends in machine learning, 14(1-2), 1-210. doi: 10.48550/arXiv.1912.04977.

[33] Konečný, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., & Bacon, D. (2016). Federated learning: Strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492. url: https://arxiv.org/ abs/1610.05492.

[34] 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.

[35] Zhu, J., Shi, Y., Fu, M., Zhou, Y., Wu, Y., & Fu, L. (2023). Latency minimization for wireless federated learning with heterogeneous local model updates. IEEE Internet of Things Journal, 11(1), 444-461. doi: 10 .1109 /JIOT.2023 .3285937.

[36] Li, X., Huang, K., Yang, W., Wang, S., & Zhang, Z. (2019). On the convergence of fedavg on non-iid data. arXiv preprint arXiv:1907.02189. doi: 10.48550/ arXiv.1907.02189.

[37] Jia, N., Qu, Z., Ye, B., Wang, Y., Hu, S., & Guo, S. (2025). A comprehensive survey on communication-efficient federated learning in mobile edge environments. IEEE Communications Surveys & Tutorials, 27(6), 3710-3741. doi: 10.1109/COMST.2025.3535957.

[38] Zhao, Z., Mao, Y., Liu, Y., Song, L., Ouyang, Y., Chen, X., & Ding, W. (2023). Towards efficient communications in federated learning: A contemporary survey. Journal of the Franklin Institute, 360(12), 8669-8703. doi: 10. 48550/arXiv.2208.01200.

[39] Lu, R., Jiang, Y., Mao, Y., Tang, C., Chen, B., Cui, L., & Wang, Z. (2024). Data-aware gradient compression for fl in communication-constrained mobile computing. IEEE Transactions on Mobile Computing, 24(4), 2755-2768. doi: 10.1109/TMC.2024.3504284.

[40] Xing, L., Luo, Z., Gao, J., Deng, K., Wu, H., & Ma, H. (2025). A survey of federated learning-based gradient compression for internet of vehicles. Engineering Applications of Artificial Intelligence, 159, 111662. doi: 10.1016/j.engappai.2025.111662.

[41] Zhang, C., Zhang, H., Dang, S., Shihada, B., & Alouini, M. S. (2024). Gradient compression and correlation driven federated learning for wireless traffic prediction. IEEE Transactions on Cognitive Communications and Networking, 11(4), 2246-2258. doi: 10 .1109 / TCCN.2024.3524183.

[42] Villani, M. J., Natale, E., & Mallmann-Trenn, F. (2025). Trading-off accuracy and communication cost in federated learning. arXiv preprint arXiv:2503.14246.

[43] He, W., Cao, H., Zhang, J., & Yang, D. (2026). Efficient Federated Learning Method FedLayerPrune Based on Layer Adaptive Pruning. Electronics, 15(5), 1049. doi: 10.3390/electronics15051049

[44] Zhang, W., Wang, J., Nie, Y., Zhao, H., Liu, Y., Sun, H., ... & Zhang, B. (2024, November). Communication Efficient Reinforcement Learning-Based Federated Pruning. In International Conference on Communications and Networking in China (pp. 3-17). Cham: Springer Nature Switzerland. doi: 10.1007/978-3-032-03215-7_1.

[45] Bustincio, R., de Souza, A. M., da Costa, J. B., Gonzalez, L. F., & Bittencourt, L. F. (2025). Reducing communication overhead through one-shot model pruning in federated learning. Annals of Telecommunications, 80(9), 901-913. doi: 10.1007/s12243-025-01097-x.

[46] Xiao, B., Zhang, J., Ni, W., & Wang, X. (2025, April). Federated Learning With Adjustable Learning Rates for Resource-Constrained Wireless Networks. In 2025 10th International Conference on Computer and Communication System (ICCCS) (pp. 1-6). IEEE. doi: 10 .1109 / ICCCS65393.2025.11069578.

[47] Li, J., Chen, T., & Teng, S. (2024). A comprehensive survey on client selection strategies in federated learning. Computer Networks, 251, 110663. doi: 10.1016/j.comnet.2024.110663.

[48] Sen, M., Aparna, S., Agarwal, R., & Mohan, C. K. (2025). Overcoming Challenges of Partial Client Participation in Federated Learning: A Comprehensive Review. arXiv preprint arXiv:2506.02887.

[49] Guo, J., Xiong, Q., Yang, M., & Zhao, Z. (2023). A double-compensation-based federated learning scheme for data privacy protection in a social IoT scenario. Computers, Materials, & Continua, 76(1), 827. doi: 10.32604/cmc.2023.036450.

[50] Xu, Y., Jiang, Z., Xu, H., Wang, Z., Qian, C., & Qiao, C. (2023). Federated learning with client selection and gradient compression in heterogeneous edge systems. IEEE Transactions on Mobile Computing, 23(5), 5446-5461. doi: 10.1109/TMC.2023.3309497.

[51] Alahmari, S., & Alghamdi, I. (2025). A Comprehensive Survey on Energy-Efficient and Privacy-Preserving Federated Learning for Edge Intelligence and IoT. Results in Engineering, 107849. doi: 10.1016/j.rineng.2025.107849.

[52] Pais, V., Rao, S., & Muniyal, B. (2025). Strategies for reducing the communication and computation costs in cross-silo federated learning: A comprehensive review. IEEE Access. doi: 10.1109/ACCESS.2025.3573933.

[53] Lang, N., Sofer, E., Shaked, T., & Shlezinger, N. (2023). Joint privacy enhancement and quantization in federated learning. IEEE Transactions on Signal Processing, 71, 295-310. doi:10.1109/TSP.2023.3244092

[54] Kim, Y. J., & Hong, C. S. (2020, September). Optimized quantization for convolutional deep neural networks in federated learning. In 2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS) (pp. 150-154). IEEE.

[55] Barhoush, M., Ayad, A., Kohankhaki, M., & Schmeink, A. (2024, May). Communication-efficient decentralised federated learning via low huffman-coded delta quantization scheme. In 2024 International Wireless Communications and Mobile Computing (IWCMC) (pp. 31-36). IEEE. doi: 10.1109/IWCMC61514.2024.10592499.

[56] Zheng, J., & Tang, J. (2025). Communication-efficient federated learning based on compressed sensing and ternary quantization: J. Zheng and J. Tang. Applied Intelligence, 55(2), 100. doi: 10.1007/s10489-024-05979-w

[57] Cao, M., Wang, H., Yuan, Y., Lu, J., Cai, X., Yu, D., & Zhao, M. (2025). FedMQ+: Towards efficient heterogeneous federated learning with multi-grained quantization. Journal of Systems Architecture, 167, 103460. doi: 10.1016/j.sysarc.2025.103460

[58] Tariq, A., Qayyum, T., Serhani, M. A., Sallabi, F. M., Taleb, I., & Barka, E. S. (2025, June). Enhancing communication efficiency in fl with adaptive gradient quantization and communication frequency optimization. In ICC 2025-IEEE International Conference on Communications (pp. 1201-1206). IEEE. doi: 10.1109/ICC52391.2025.11161682.

[59] Feng, C., & Venkitasubramaniam, P. (2025). Randomized quantization for privacy in resource constrained machine learning at-the-edge and federated learning. IEEE Transactions on Machine Learning in Communications and Networking. doi: 10.1109/TMLCN.2025.3550119.

[60] Tariq, O., Dastagir, M. B. A., & Han, D. (2025). ADP-QFed: Privacy-Preserving Quantized Federated Learning for Intelligent Edge Sensing IoT Systems. IEEE Internet of Things Journal. doi: 10.1109/JIOT. 2025.3634475.

[61] Wang, L., Xu, X., & Pei, J. (2025). Communication-efficient federated learning via dynamic sparsity: An adaptive pruning ratio based on weight importance. IEEE Transactions on Cognitive Communications and Networking. doi: 10.1109/TCCN.2025.3577323.

[62] Chang, X., Obaidat, M. S., Ma, J., Xue, X., Yu, Y., & Wu, X. (2024). Efficient federated learning via adaptive model pruning for internet of vehicles with a constrained latency. IEEE Transactions on Sustainable Computing, 10(2), 300-316. doi: 10.1109/TSUSC.2024.3441658.

[63] Liu, S., Shen, H., Law, E. K., & Lam, C. T. (2025). Mutual Knowledge Distillation-Based Communication Optimization Method for Cross-Organizational Federated Learning. Electronics, 14(9), 1784. doi: 10.3390/electronics14091784.

[64] Abou El Houda, Z., Moudoud, H., & Brik, B. (2025). When federated learning meets knowledge distillation to secure consumer edge network. IEEE Transactions on Consumer Electronics. doi: 10.1109/TCE.2025.3559004.

[65] Yu, F., Wang, L., Zeng, B., Zhao, K., & Yu, R. (2024). Personalized and privacy-enhanced federated learning framework via knowledge distillation. Neurocomputing, 575, 127290. doi: 10.1016/j.neucom.2024.127290.

[66] Wu, C., Wu, F., Lyu, L., Huang, Y., & Xie, X. (2022). Communication-efficient federated learning via knowledge distillation. Nature communications, 13(1), 2032. doi: 10.1038/ s41467-022-29763-x.

[67] Yao, D., Pan, W., Dai, Y., Wan, Y., Ding, X., Yu, C., ... & Sun, L. (2023). FedGKD: Toward heterogeneous federated learning via global knowledge distillation. IEEE Transactions on Computers, 73(1), 3-17. doi: 10.1109/TC.2023.3315066.

[68] Li, J., Zhao, Y., Zhao, J., Huang, Y., & Xue, K. (2025, June). Privacy-Preserving and Top-K Sparsified Federated Learning with Low Communication Overhead. In ICC 2025-IEEE International Conference on Communications (pp. 4074-4079). IEEE. doi: 10.1109/ICC52391. 2025.11161487.

[69] Yu, J., Shu, N., Wu, T., Wang, H., Jiang, R., & Chang, C. (2025, July). Unbiased Data-Driven Dynamic Threshold Sparsification for Communication-Efficient Federated Learning. In International Conference on Intelligent Computing (pp. 229-241). Singapore: Springer Nature Singapore. doi: 10.1007/978-981-96- 9805-9_19.

[70] Seo, Y., Lim, H., & Yu, N. Y. (2025). Communication Efficient Over-the-Air Federated Learning With Random FLARE Algorithm. IEEE Signal Processing Letters, 33, 171-175. doi: 10.1109/LSP.2025.3633592.

[71] Al-Fehani, M., Al-Baseer, A., & Al-Kuwari, S. (2026). TFD-Video: threshold-aware federated deepfake detection for video forensics. IEEE Access. doi: 10.1109/ACCESS.2026.3660914.

[72] Chen, Z. J., Hernandez, E. E., Huang, Y. C., & Rini, S. (2022, May). DNN gradient lossless compression: Can GenNorm be the answer?. In ICC 2022-IEEE International conference on communications (pp. 407-412). IEEE. doi: 10.1109/ICC45855.2022.9838754.

[73] Liu, B., Gao, Y., Zhang, C., & Zhong, G. (2025, March). FedBDQB: Communication Efficient Federated Learning via Bidirectional Dynamic Quantization and Bitmap. In 2025 IEEE Wireless Communications and Networking Conference (WCNC) (pp. 1-6). IEEE. doi: 10.1109/WCNC61545.2025.10978684.

[74] Wilkins, G., Di, S., Calhoun, J. C., Li, Z., Kim, K., Underwood, R., ... & Cappello, F. (2024, July). Fedsz: Leveraging error-bounded lossy compression for federated learning communications. In 2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS) (pp. 577-588). IEEE. doi: 10. 1109/ICDCS60910.2024.00060.