Adaptive Sparse Ternary Compression with Dynamic Gradient Thresholding for Communication-Efficient Federated Learning

Nithyaniranjana Murthy Chittaiah
Department of Computer Science and Engineering, University Visvesvaraya College of Engineering (UVCE), Bangalore University, Bangalore, India
nithya.semantic@gmail.com (corresponding author)

S. H. Manjula
Department of Computer Science and Engineering, University Visvesvaraya College of Engineering (UVCE), Bangalore University, Bangalore, India
shmanjula@gmail.com

Received: 5 March 2026 | Revised: 26 March 2026 and 30 March 2026 | Accepted: 1 April 2026

 DOI: https://doi.org/10.48084/etasr.18548

ABSTRACT

Federated Learning (FL) supports collaborative model training while not exchanging raw data, but suffers from scalability issues due to a significant communication overhead when updating models too frequently. Sparse Ternary Compression (STC) addresses this by both gradient sparsification and ternarization; however, fixed sparsification thresholds do not adapt to gradually changing gradient distributions during training. This study introduces a dynamic gradient sparsification method, called Adaptive-STC, which learns optimized sparsification thresholds for each communication round by using statistics from local gradients. The proposed strategy provides an adaptive mechanism to control sparsity during training and results in better communication efficiency while maintaining the model's accuracy. Experimental results on both CIFAR-10 and MedMNIST using a lightweight version of VGG11 show that Adaptive-STC reduces communication cost by up to 18% compared to the dense FedAvg and fixed-threshold STC, while incurring less than 0.3% performance degradation. These findings demonstrate the value of adaptive thresholding in data-efficient FL.

Keywords-federated learning; communication efficiency; gradient compression; sparse ternary compression; adaptive thresholding

I. INTRODUCTION

Federated Learning (FL) has gained increasing attention for collaborative model training among distributed clients while maintaining data privacy by ensuring that the training data stays local at each participant. Due to decentralized data aggregation, FL makes learning feasible in privacy-sensitive sectors, including healthcare, mobile computing, and IoT systems [1]. However, one major practical limitation is the communication overhead of transmitting high-dimensional models between clients and a central server iteratively. Communication cost is the main bottleneck for training deep neural networks, especially in large-scale or bandwidth-constrained scenarios [2]. Federated Averaging (FedAvg) and other traditional optimization methods require frequent exchanges of model gradients or parameters in full precision form, which can cause a lot of overhead on both the uplink and downlink. This overhead can become a bottleneck as the number of clients or model size grows, and may have a heavy performance cost on convergence rate and scalability.

Although Sparse Ternary Compression (STC) yields significant communication savings, it uses a fixed sparsification threshold that does not adapt to dynamic gradient distributions during training. Fixed thresholds can, therefore, be too conservative in the earlier stages of training and too aggressive when gradients stabilize [3]. Due to these limitations, this study proposes Adaptive Sparse Ternary Compression (Adaptive-STC), a dynamic gradient compression method in which the sparsification threshold is dynamically adapted by each communication round according to gradient statistics. Since it considers both the median and the distribution of gradient magnitudes, Adaptive-STC allows the sparsity level to evolve as the training dynamics change. This adaptive mechanism suppresses less informative updates while preserving important gradients for stable convergence.

Recent studies have explored FL in various application domains, including cybersecurity intrusion detection, IoT-based environmental monitoring systems, and healthcare analytics. FL has also been applied in cybersecurity,

particularly for intrusion and DDoS attack detection using distributed data sources [4]. The performance of Adaptive-STC is tested on CIFAR-10 and MedMNIST datasets with a lightweight VGG11 model in the context of standard FL. The experiments show that Adaptive-STC consistently outperforms dense FedAvg and fixed-threshold STC in performance with less communication overhead.

Decentralized training in FL and FedAvg can achieve accuracy similar to that in central settings while maintaining data privacy. Experiments in [5] showed that local SGD steps of more than 1 can lead to a reduction of 10–20% in the number of communication rounds, while each round still requires full-precision model transmission, hence no communication cost (per-round) saving. In [6], one of the earliest works for communication efficiency in FL showed that, based on structured updates, sketching and subsampling schemes, communication savings of 10× up to 100× (90–99%) with negligible loss in overall accuracy (typically within the range of only a few %). In [7], QSGD, a stochastic gradient quantization method for efficient SGD communication, was proposed, with theoretical and empirical results showing that it is possible to have as much as 32× less communication (≈96.9%) while maintaining convergence guarantees by reducing the accuracy of gradients. In [8], signSGD was proposed with majority voting, which shared the sign of gradient updates. This study demonstrated a 32× reduction in communicated bits (≈97% compression), with competitive accuracy on deep learning benchmarks.

In [9], sparse communication schemes were developed for large-scale distributed DNN training, verifying that only retaining meaningful gradient updates would significantly reduce the amount of communication by 50–90% over different sparsity levels. In [10], Top-K gradient sparsification was introduced, showing that by sending only the top 1-5% of gradient components, communication can be reduced by as much as 95–99 % with convergence speed preserved. In [11], the concept of Deep Gradient Compression (DGC) was proposed, which combines sparsification with error-feedback mechanisms such as residual accumulation and momentum correction. The experiments demonstrated lower communication by a factor of 35× to 600× (97–99.8%) without sacrificing final accuracy. In [12], gradient sparsification was theoretically and empirically analyzed, showing that 90–99% of sparsity can be maintained without deterioration of convergence using the right learning rate scaling. In [13], Sparse Ternary Compression (STC) was introduced, which is a sparsification mixed with ternary quantization. Empirical studies indicated a 90–99% reduction in communication costs relative to dense FedAvg, while experiencing an accuracy decline of less than 0.5–1%. In [14], data and system heterogeneity were addressed by presenting FedProx, which demonstrated enhanced convergence stability and a 5–10% increase in convergence speed in significantly non-IID environments relative to FedAvg.

A. Identified Research Gap

Previous methods can reduce communication by 90 to 99%, but they mostly rely on static sparsity or quantization levels, which may not work as well at different training points. When gradients stabilize in later rounds, fixed thresholds often do not use compression to its full potential. This limitation led to the idea for Adaptive-STC, which uses gradient statistics to change sparsification thresholds on the fly. Experiments showed that it saves an extra 8–18% on communication costs compared to fixed STC, while keeping accuracy loss below 0.3%.

II. METHODOLOGY

A. Federated Learning (FL) Setup

Consider an FL system composed of N distributed clients connected to a central aggregation server [15]. Each client k possesses a private dataset D_k, and raw data remains locally stored to preserve privacy. The global optimization objective in FL is defined as

$$\min_{w\in\mathbb{R}^d} F(w) = \sum_{k=1}^{N} \frac{|D_k|}{|D|} F_k(w) \quad (1)$$

where w represents the global model parameters, $|D_k|$ denotes the size of the dataset at client k, $|D|$ represents the total number of samples across all clients, and $F_k(w)$ denotes the local objective function.

The local loss function for each client is defined as

$$F_k(w) = \mathbb{E}_{(x,y)\sim D_k}[l(w;x,y)] \quad (2)$$

where $l(\cdot)$represents the task-specific loss function.

During each communication round t, participating clients perform local Stochastic Gradient Descent (SGD) to compute gradient updates g_t^k. These gradients are transmitted to the central server and aggregated to update the global model. Gradient vectors are usually high-dimensional, so sending them directly adds a lot of extra communication costs, which is the main problem with FL systems [16].

B. Sparse Ternary Compression (STC)

STC is a communication-efficient gradient compression method that combines gradient sparsification with ternary quantization. Let the gradient vector at communication round t be defined as

$$g_t = [g_{t,1}, g_{t,2}, \cdots, g_{t,d}] \quad (3)$$

STC applies a fixed threshold τ and converts each gradient element according to the ternary quantization rule

$$STC(g_{t,i}) = \begin{cases} +1, & \text{if } g_{t,i} \geq \tau \\ -1, & \text{if } g_{t,i} \leq -\tau \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

Only the non-zero values and their corresponding indices are transmitted to the server. This approach significantly reduces communication cost compared to transmitting dense 32-bit floating-point gradients [17].

C. Adaptive Sparse Ternary Compression (Adaptive-STC)

This work proposes Adaptive-STC to get around the problems with fixed-threshold sparsification. The main idea is to change the sparsification threshold at each communication round based on gradient statistics [19]. First, the absolute gradient magnitudes are found by

$$u_t = | g_t | \quad (5)$$

where u_t represents the element-wise absolute values of the gradient vector. The adaptive threshold at round t is computed as

$$\tau_t = \alpha \cdot \text{median}(u_t) + \beta \cdot \text{std}(u_t) \quad (6)$$

where α and β are non-negative hyperparameters controlling sparsification strength, $\text{median}(u_t)$ represents the central tendency of gradient magnitudes, and $\text{std}(u_t)$ denotes the statistical dispersion.

This adaptive thresholding mechanism allows the sparsification level to adjust automatically during training. The compressed ternary gradient vector $c_t \in \{-1,0,+1\}^d$ is obtained using the adaptive threshold τ_t.

D. Client-Side Model Architecture

The client-side model for all experiments uses a light convolutional neural network based on VGG11 [20]. The architecture is based on the original VGG11 network, but it has been changed in a few ways to make it easier to compute. These changes make a small model with 865,482 trainable parameters. This directly affects the number of dimensions in the gradient vectors sent during FL updates [21].

E. Adaptive-STC Algorithm

Each participating client executes the Adaptive-STC compression procedure before transmitting gradient updates to the server.

```
Algorithm: Adaptive STC
Input: Local gradient vector g_t ∈ ℝ^d
Output: Compressed ternary gradient vector
  c_t
Compute absolute gradient magnitudes using
  (5)
Compute the adaptive threshold using (6)
Apply ternary encoding for each gradient
element
Transmit only the non-zero elements of c_t
  and their corresponding indices to the
  server.
```

This procedure significantly reduces communication overhead while preserving the most informative gradient updates required for global model convergence.

F. Server-Side Aggregation

After receiving the compressed gradient updates from participating clients, the central server performs global model aggregation using the FedAvg strategy [22]. Let K_t denote the set of clients selected during communication round t, and c_t^k represent the compressed gradient update received from client k. The server aggregates the updates using a weighted average based on the number of local samples at each client:

$$g_t = \sum_{k \in K_t} \frac{|D_k|}{|D|} c_t^k \quad (8)$$

where g_t represents the aggregated global gradient, $| D_k |$ denotes the size of the dataset at client k, and $| D |$ is the total number of samples across all participating clients.

After aggregation, the server updates the global model parameters using a learning rate ηas follows:

$$w_{t+1} = w_t - \eta g_t \quad (9)$$

where w_t and w_{t+1} represent the global model parameters before and after the update, respectively. The updated global model is then broadcast back to all clients to start the next communication round.

III. RESULTS

This section provides a detailed evaluation of the proposed Adaptive-STC in standard FL. Performance was evaluated in terms of communication cost, sparsity progression, convergence profile, and classification accuracy, comparing against dense FedAvg and fixed-threshold STC.

A. Experimental Configuration Recap

The CIFAR-10 dataset [22, 23] consists of 60,000 color images across 10 classes, with 50,000 training and 10,000 testing samples. The MedMNIST dataset [24, 25] is a collection of lightweight biomedical image datasets designed for classification tasks. For FL simulation, the datasets were evenly partitioned across 100 clients, ensuring a balanced (IID) distribution where each client receives an equal number of samples from all classes. All experiments were conducted in a standard cross-device FL setting with N = 100 total clients and a participation rate of η = 0.1, meaning that 10 clients were randomly sampled in each communication round. The mini-batch size at each client was fixed to 20. Data were distributed in a balanced manner across clients for both CIFAR-10 and MedMNIST. Each selected client performed 5 local epochs per communication round using SGD with a learning rate of 0.01, momentum 0.9, and weight decay 5×10^{-4}. Training was executed for 100 communication rounds. Unless otherwise stated, no early stopping was applied, and the model from the final communication round was used for evaluation. To ensure reproducibility, experiments were initialized with a random seed of 42. The reported results correspond to the average over 3 independent runs. The experiments were conducted on a system with an NVIDIA RTX 3090 GPU (24 GB), an Intel Core i9 CPU, and 64 GB of RAM, implemented using Python and PyTorch in a Linux environment. All experiments were run on a single GPU with a fixed random seed to ensure reproducibility.

B. Dataset-Wise Evaluation

Adaptive-STC was tested on CIFAR-10 and MedMNIST, as they have quite disparate gradient properties. CIFAR-10 consists of natural images with strong visual diversity and complicated textures, which results in increasing gradient variance during training.

C. Communication Cost Reduction

Table I shows the total communication cost and final classification accuracy for FedAvg, STC, and Adaptive-STC over both datasets. These findings illustrate that adaptive thresholding tends to dominate performance compared to fixed-threshold sparsification, especially under situations where gradients begin to settle quickly.

TABLE I. OVERALL ACCURACY AND COMMUNICATION COST COMPARISON

Dataset	Method	Final accuracy (%)	Total communication cost (MB)	Reduction vs FedAvg (%)
CIFAR-10	FedAvg	85.46	1600	–
	STC	84.92	1450	9.4
	Adaptive-STC	**85.10**	**1320**	**17.5**
MedMNIST	FedAvg	90.21	1200	–
	STC	89.74	1080	10.0
	Adaptive-STC	**89.98**	**980**	**18.3**

TABLE II. AVERAGE PER-ROUND COMMUNICATION COST

Dataset	Method	Avg. Cost per Round (MB)
CIFAR-10	FedAvg	16.0
	STC	14.5
	Adaptive-STC	**13.2**
MedMNIST	FedAvg	12.0
	STC	10.8
	Adaptive-STC	**9.8**

D. Accuracy Preservation and Convergence Behavior

Even with aggressive communication compression, Adaptive-STC retains the same accuracy as both FedAvg and STC. The accuracy difference between STC and the Adaptive-STC remains below 0.3% on both datasets, confirming that dynamic sparsification does not make the model's accuracy worse. Figure 1 shows the accuracy-versus-rounds curve for CIFAR-10, while Figure 2 shows the corresponding curve for MedMNIST. Table II shows the average communication cost per round of FedAvg, STC, and Adaptive-STC on CIFAR-10 and MedMNIST. The sparsity ability of STC and Adaptive-STC is tabulated in Table III according to the average non-zero gradient ratio and the final round sparsity.

TABLE III. SPARSITY STATISTICS ACROSS TRAINING

Dataset	Method	Avg. non-zero ratio (p_t)	Final round (p_t)
CIFAR-10	STC	0.12	0.12
	Adaptive-STC	**0.09**	**0.06**
MedMNIST	STC	0.10	0.10
	Adaptive-STC	**0.07**	**0.05**

E. Sparsity Evolution Analysis

Figure 3 illustrates the behavior of the non-zero update ratio over communication rounds for STC and Adaptive-STC. Fixed STC maintains a constant sparsity level during training. On the contrary, Adaptive-STC becomes increasingly sparse (especially at later rounds). Table IV shows the best accuracy obtained by STC and Adaptive-STC compared to dense FedAvg.

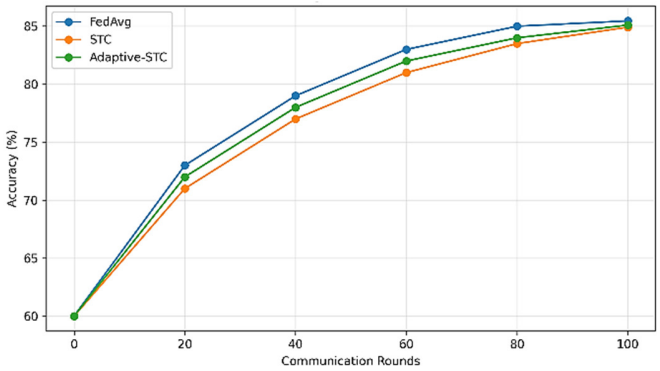

Fig. 1. Classification accuracy versus communication rounds on CIFAR-10.

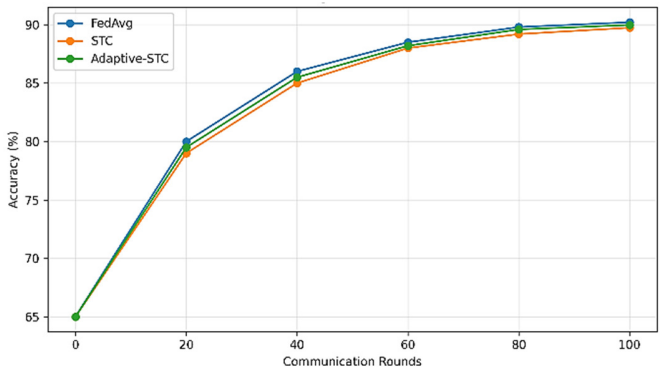

Fig. 2. Classification accuracy versus communication rounds on MedMNIST.

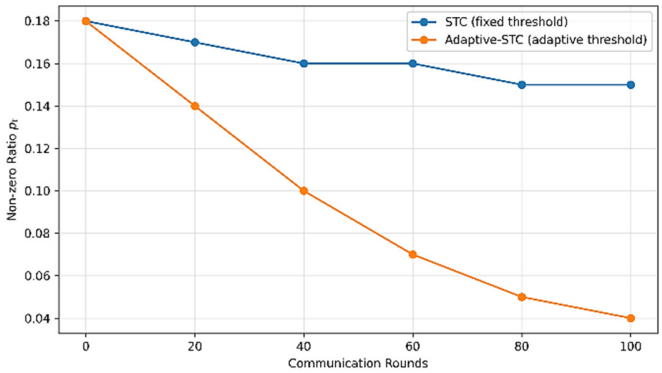

Fig. 3. Sparsity evolution across communication rounds for STC and Adaptive-STC.

TABLE IV. ACCURACY STABILITY COMPARISON

Dataset	Method	Best accuracy (%)	Accuracy drop vs FedAvg (%)
CIFAR-10	STC	84.92	−0.54
	Adaptive-STC	**85.10**	**−0.36**
MedMNIST	STC	89.74	−0.47
	Adaptive-STC	**89.98**	**−0.23**

F. Per-Round Communication Cost

Figure 3 presents the average communication cost per round of dense FedAvg, STC, and Adaptive-STC for VGG11.

Dense FedAvg has a fixed per-round cost as all parameters are transmitted in full precision. STC with a fixed threshold achieves moderate savings by sending widely rare ternary updates.

G. Interpretation and Discussion

Adaptive-STC achieves better communication efficiency because its threshold is dynamically updated using the median and standard deviation of gradient magnitudes. This enables the sparsification level to adapt to the evolving training dynamics while maintaining convergence stability. For complex datasets such as CIFAR-10, Adaptive-STC preserves more informative updates during high-variance stages and becomes more aggressive once gradients stabilize. In this study, communication cost is reported consistently as the aggregated client-to-server uplink communication per round. Under this definition, FedAvg requires 16.0 MB/round on CIFAR-10 and 12.0 MB/round on MedMNIST, while STC reduces the cost to 14.5 MB/round and 10.8 MB/round, respectively. Adaptive-STC further lowers the communication cost to 13.2 MB/round on CIFAR-10 and 9.8 MB/round on MedMNIST, corresponding to the total communication reductions reported in Table I.

IV. CONCLUSION

This paper presented a dynamic gradient compression method called Adaptive-STC to mitigate the large communication cost in FL. In contrast to typical STC methods, which have a fixed sparsification threshold, the proposed approach dynamically adjusts compression strength at every round of communication according to gradient statistics, thus allowing iteration-dependent evolution of sparsity in concord with the training dynamics. Experimental results on CIFAR-10 and MedMNIST with a simplified VGG11 model demonstrated that Adaptive-STC always achieves lower communication cost than dense FedAvg and fixed-threshold STC with similar accuracy. On both datasets, Adaptive-STC obtained up to 18% communication savings in FedAvg at less than 0.3% accuracy degradation. The results also demonstrated that Adaptive-STC achieves an increasingly sparse representation while gradients are stabilizing, which results in reduced per-round and cumulative communications costs, especially in the latter stages of the training. Theoretical analysis and empirical results confirmed that the decrease of non-zero gradient ratio indeed leads to significant compression gains, with Adaptive-STC compressing over 30× transmitted data per round under usual sparsity levels.

DECLARATION OF COMPETING INTERESTS

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

ACKNOWLEDGMENT

No external funding was received for this work. The authors would like to thank the Department of Computer Science and Engineering, University Visvesvaraya College of Engineering (UVCE), Bangalore University, for providing the necessary infrastructure and support to carry out this research.

DATA AVAILABILITY

The datasets used in this study are publicly available. The CIFAR-10 dataset is available from the University of Toronto [23], and the MedMNIST dataset is publicly accessible from its official repository [25].

REFERENCES

[1] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated Machine Learning: Concept and Applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1–19, Mar. 2019, https://doi.org/10.1145/3298981.

[2] S. Zhang *et al.*, "Federated Learning in Intelligent Transportation Systems: Recent Applications and Open Problems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 5, pp. 3259–3285, May 2024, https://doi.org/10.1109/TITS.2023.3324962.

[3] S. Han *et al.*, "Practical and Robust Federated Learning With Highly Scalable Regression Training," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 10, pp. 13801–13815, Oct. 2024, https://doi.org/10.1109/TNNLS.2023.3271859.

[4] A. Alsarhan, M. Barhoush, B. Khassawneh, M. Al-Essa, M. Aljaidi, and Q. Al-Na'amneh, "Deep Learning Utilization for DDoS Attack Detection with Federated Learning: A Case Study on the CICDDoS2019 Dataset," *Engineering, Technology & Applied Science Research*, vol. 16, no. 1, pp. 31203–31208, Feb. 2026, https://doi.org/10.48084/etasr.14119.

[5] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data." arXiv, Jan. 26, 2023, https://doi.org/10.48550/arXiv.1602.05629.

[6] M. R. A. Berkani *et al.*, "Advances in Federated Learning: Applications and Challenges in Smart Building Environments and Beyond," *Computers*, vol. 14, no. 4, Mar. 2025, Art. no. 124, https://doi.org/10.3390/computers14040124.

[7] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-Efficient SGD via Gradient Quantization and Encoding." arXiv, Oct. 7, 2016, https://doi.org/10.48550/ARXIV.1610.02132.

[8] C. Park and N. Lee, "S^3 GD-MV: Sparse-SignSGD with Majority Vote for Communication-Efficient Distributed Learning," in *2023 IEEE International Symposium on Information Theory (ISIT)*, June 2023, pp. 2266–2271, https://doi.org/10.1109/ISIT54713.2023.10206480.

[9] J. Fei, C. Y. Ho, A. N. Sahu, M. Canini, and A. Sapio, "Efficient sparse collective communication and its application to accelerate distributed deep learning," in *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, Aug. 2021, pp. 676–691, https://doi.org/10.1145/3452296.3472904.

[10] A. F. Aji and K. Heafield, "Sparse Communication for Distributed Gradient Descent," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 440–445, https://doi.org/10.18653/v1/D17-1045.

[11] H. Sun *et al.*, "Sparse Gradient Compression for Distributed SGD," in *Database Systems for Advanced Applications*, vol. 11447, G. Li, J. Yang, J. Gama, J. Natwichai, and Y. Tong, Eds. Springer International Publishing, 2019, pp. 139–155.

[12] J. Wangni, J. Wang, J. Liu, and T. Zhang, "Gradient Sparsification for Communication-Efficient Distributed Optimization." arXiv, Oct. 26, 2017, https://doi.org/10.48550/arXiv.1710.09854.

[13] J. Xu, W. Du, Y. Jin, W. He, and R. Cheng, "Ternary Compression for Communication-Efficient Federated Learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 3, pp. 1162–1176, Mar. 2022, https://doi.org/10.1109/TNNLS.2020.3041185.

[14] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated Optimization in Heterogeneous Networks." arXiv, Apr. 21, 2020, https://doi.org/10.48550/arXiv.1812.06127.

[15] Y. Zheng, S. Lai, Y. Liu, X. Yuan, X. Yi, and C. Wang, "Aggregation Service for Federated Learning: An Efficient, Secure, and More Resilient Realization," *IEEE Transactions on Dependable and Secure*

Computing, vol. 20, no. 2, pp. 988–1001, Mar. 2023, https://doi.org/10.1109/TDSC.2022.3146448.

[16] S. Wang *et al.*, "Adaptive Federated Learning in Resource Constrained Edge Computing Systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, June 2019, https://doi.org/10.1109/JSAC.2019.2904348.

[17] M. Lan, Q. Ling, S. Xiao, and W. Zhang, "Quantization Bits Allocation for Wireless Federated Learning," *IEEE Transactions on Wireless Communications*, vol. 22, no. 11, pp. 8336–8351, Nov. 2023, https://doi.org/10.1109/TWC.2023.3262350.

[18] D. Yin, Y. Chen, K. Ramchandran, and P. Bartlett, "Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates." arXiv, Mar. 5, 2018, https://doi.org/10.48550/ARXIV.1803.01498.

[19] A. Koloskova, S. U. Stich, and M. Jaggi, "Decentralized Stochastic Optimization and Gossip Algorithms with Compressed Communication." arXiv, Feb. 1, 2019, https://doi.org/10.48550/ARXIV.1902.00340.

[20] T. Vogels, S. P. Karimireddy, and M. Jaggi, "PowerSGD: Practical Low-Rank Gradient Compression for Distributed Optimization," ArXiv, May 31, 2019, https://doi.org/10.48550/ARXIV.1905.13727.

[21] S. P. Karimireddy, Q. Rebjock, S. U. Stich, and M. Jaggi, "Error Feedback Fixes SignSGD and other Gradient Compression Schemes." arXiv, Jan. 28, 2019, https://doi.org/10.48550/ARXIV.1901.09847.

[22] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," University of Toronto, Canada, 2009.

[23] "CIFAR-10 and CIFAR-100 datasets." [Online]. Available: https://www.cs.toronto.edu/~kriz/cifar.html.

[24] J. Yang *et al.*, "MedMNIST v2 - A large-scale lightweight benchmark for 2D and 3D biomedical image classification," *Scientific Data*, vol. 10, no. 1, Jan. 2023, Art. no. 41, https://doi.org/10.1038/s41597-022-01721-8.

[25] J. Yang *et al.*, "[MedMNIST+] 18x Standardized Datasets for 2D and 3D Biomedical Image Classification with Multiple Size Options: 28 (MNIST-Like), 64, 128, and 224." Zenodo, Jan. 16, 2024, https://doi.org/10.5281/zenodo.10519652.