FedBand: Adaptive Federated Learning Under Strict Bandwidth Constraints

Taghreed Alanazi, Abdulrahman Fahim, Muntaka Ibnath, Basak Guler, Amit Roy Chowdhury
Ananthram Swami, Evangelos E. Papalexakis, Srikanth V. Krishnamurthy
[*]*University of California, Riverside, USA*
[**] *DEVCOM Army Research Laboratory, USA*
Email: {talan002@, afahi002@, mibna001@, bguler@ece., amitrc@ece.,
epapalex@cs., krish@cs.}ucr.edu, ananthram.swami.civ@army.mil

***Abstract*—Federated Learning (FL) enables model training across decentralized clients while preserving data privacy. However, bandwidth constraints limit the volume of information exchanged, making communication efficiency a critical challenge. In addition, non-IID data distributions require fairness-aware mechanisms to prevent performance degradation for certain clients. Existing sparsification techniques often apply fixed compression ratios uniformly, ignoring variations in client importance and bandwidth. We propose FedBand, a dynamic bandwidth allocation framework that prioritizes clients based on their contribution to the global model. Unlike conventional approaches, FedBand does not enforce uniform client participation in every communication round. Instead, it allocates more bandwidth to clients whose local updates deviate significantly from the global model, enabling them to transmit a greater number of parameters. Clients with less impactful updates contribute proportionally less or may defer transmission, reducing unnecessary overhead while maintaining generalizability. By optimizing the trade-off between communication efficiency and learning performance, FedBand substantially reduces transmission costs while preserving model accuracy. Experiments on non-IID CIFAR-10 and UTMobileNet2021 datasets, demonstrate that FedBand achieves up to 99.81% bandwidth savings per round while maintaining accuracies close to that of an unsparsified model (80% on CIFAR-10, 95% on UTMobileNet), despite transmitting less than 1% of the model parameters in each round. Moreover, FedBand accelerates convergence by 37.4%, further improving learning efficiency under bandwidth constraints. Mininet emulations further show a 42.6% reduction in communication costs and a 65.57% acceleration in convergence compared to baseline methods, validating its real-world efficiency. These results demonstrate that adaptive bandwidth allocation can significantly enhance the scalability and communication efficiency of federated learning, making it more viable for real-world, bandwidth-constrained networking environments.**

***Index Terms*—Federated Learning (FL), Non-IID Data, Adaptive Sparsification, Dynamic Bandwidth Allocation, Bandwidth-Aware Client Prioritization, Communication-Efficient Learning, Convergence Acceleration, Mininet Emulation.**

I. Introduction

Federated Learning (FL) enables decentralized training of a machine learning (ML) model, by leveraging data distributed across remote clients, such as mobile phones or IoT devices [1]. This distributed paradigm enhances data privacy by ensuring that raw data remains on local devices, eliminating the need to transfer it to a central server. Despite this benefit, FL faces challenges when deployed over real-world networks. Specifically, *bandwidth constraints* and *heterogeneous data distributions (non-IID)* can significantly hinder training performance.

Challenge #1: Bandwidth Constraints. In modern networks, bandwidth is typically partitioned between control-plane data (operational information) and user-plane data (content) [13]. Increased control plane activity reduces the data plane bandwidth, straining the volume of user data that can be carried. FL tasks, such as traffic classification, fall under control plane operations, adding to network overhead. Transmitting full model parameters or gradients is bandwidth-intensive, especially with many clients, risking network overload. However, imposing strict bandwidth limits in FL can delay convergence and degrade performance. The key question is: *How do we ensure rapid and resource-efficient FL training under stringent bandwidth constraints?*

Challenge #2: Non-IID Client Data. FL must cope with data that is not *independent and identically distributed* (non-IID). In such cases, each client holds data that diverges from the global distribution, introducing heterogeneity. For instance, clients in different regions may encounter distinct, skewed traffic patterns (if the model addresses traffic classification). This variability complicates global model development and makes it difficult to ensure consistent performance across all clients. Traditional FL algorithms often suffer from performance degradation in highly skewed scenarios, as they treat all clients' updates equally. Achieving robust global performance (i.e., high average accuracy) while ensuring acceptable performance for underrepresented or outlier clients remains particularly challenging with non-IID data. Consequently, adaptive resource-allocation strategies are needed to ensure both fairness and efficacy. The key question is: *How can we ensure good performance for all clients, especially those with skewed data, while maintaining global accuracy under strict bandwidth constraints?*

Adaptive Resource Allocation and Bandwidth-Aware Client Prioritization. Current FL approaches often apply fixed or uniform sparsification techniques to reduce communication costs, using the same compression level across all clients. However, these methods are suboptimal for non-IID data because they ignore the varying significance of each client's updates. This leads to inefficient bandwidth utilization, degrading both local and global performance. An adaptive framework is needed—one that *(i)* prioritizes clients based on their contribution to the global model, *(ii)* dynamically adjusts bandwidth allocation across clients, and *(iii)* considers the distinct importance of each client's updates in the presence of data heterogeneity.

Proposed Solution: *FedBand*. To tackle these issues, we propose *FedBand* (short for ***Fed**erated Learning with Strict **Band**width Constraints*), a *dynamic sparsification* and *bandwidth-aware allocation* framework for FL. Unlike uniform approaches, FedBand:

- Dynamically adjusts compression ratios based on both available bandwidth and the degree of *non-IID skew* across clients, ensuring adaptive communication of updates.

979-8-3315-0898-2/25/$31.00 ©2025 IEEE

- Allocates bandwidth *proportional to the significance of each client's updates*, prioritizing clients whose local models deviate more from the global model.
- Enables *flexible client participation*, where different clients contribute varying amounts of updates per round, improving fairness and robustness while maintaining strong global performance.

We evaluate FedBand on two datasets: *CIFAR-10* [14] for image classification and *UTMobileNet2021* [5] for network traffic classification. Our experiments show that FedBand achieves 80% accuracy on CIFAR-10 and 95% accuracy on UTMobileNet2021, closely matching the performance of uncompressed models while outperforming uniform sparsification techniques under bandwidth constraints. We point out that while centralized models trained on CIFAR-10 with IID data typically exceed 99% accuracy, such results do not translate to realistic non-IID federated learning scenarios, where data heterogeneity impacts performance. Importantly, FedBand reduces communication costs by an impressive 99.81% to 99.91% compared to an uncompressed model and accelerates convergence by minimizing processing overheads at the protocol level.

- **Development of the FedBand Framework:** We propose FedBand, a federated learning framework designed for scalable and communication-efficient learning under strict bandwidth constraints. FedBand dynamically adjusts client participation and transmission to optimize learning efficiency while ensuring fairness and adaptability to heterogeneous data distributions.
- **Novel Importance-Aware Bandwidth Allocation Strategy:** We introduce a new cost function that adaptively distributes bandwidth by prioritizing clients generating the most impactful updates. Unlike existing approaches, this strategy enables heterogeneity-aware bandwidth allocation, ensuring balanced learning contributions while significantly improving communication efficiency.
- **Extensive Performance Evaluations:** Experiments on CIFAR-10 and UTMobileNet2021 demonstrate that FedBand achieves accuracy comparable to unsparsified models while transmitting less than 1% of model parameters per round, significantly outperforming uniform sparsification approaches in bandwidth-limited environments.
- **Realistic Network Emulations with Mininet:** Using Mininet-based emulations [27] with a full networking stack, we show that FedBand accelerates convergence by 65.57% and reduces bandwidth usage by 42.6% compared to baseline methods, validating its efficiency in real-world network conditions.

II. Related Work

In this section, we summarize key research relevant to our work, highlighting where they fall short in handling both non-IID data and tight bandwidth constraints simultaneously.

Many FL approaches struggle with non-IID client data, which can degrade performance for certain subsets of clients. Personalized FL methods [19], [21] seek to address this heterogeneity by tailoring models for different client distributions, often relying on transfer learning [23] or meta-learning [22]. While these approaches improve local accuracy, they typically assume continuous data availability and do not explicitly address communication efficiency, making them impractical for bandwidth-limited environments.

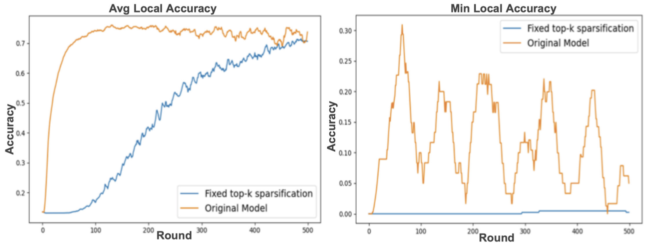


Fig. 1: Average and Minimum Local Accuracies vs. number of rounds for both Original Model (orange) and Fixed top-*k* method (blue) with FedAvg.

Reducing communication overhead has been a major focus in FL to cope with bandwidth scarcity. For instance, Top-*k* sparsification, which transmits the largest *k* gradient magnitudes [2], assumes static compression ratios and overlooks bandwidth variability. CocktailSGD, utilized in large language models, integrates top-K selection, and quantization [4], and FedPM activates sparse subsets of weights with stochastic masks [10]. However, these approaches enforce a fixed sparsification pattern across all clients, failing to incorporate client heterogeneity and adaptive resource allocation.

Aggregation methods like Federated Averaging (FedAvg) optimize global accuracy but overlook minimum local accuracy, which is crucial in non-IID settings. Our CIFAR-10 experiments with 20 users and a VGG19 model highlight this limitation (see Fig. 1). Over 500 communication rounds, a vanilla FL model without compression, using FedAvg for aggregation, (referred to as the "Original Model" or "Full Model") achieves good global accuracy but requires 10,960 MB of transmission per round, with local accuracies frequently dropping to zero. Fixed top-*k* sparsification reduces transmission to 60 MB per round but fails to maintain minimum local accuracy, with many clients experiencing zero accuracy even after multiple rounds.

Several recent methods attempt to improve FL performance in non-IID settings. FedNova [16] and SCAFFOLD [18] enhance performance in non-IID settings by normalizing updates and mitigating client drift but do not account for tight bandwidth constraints. Agnostic Federated Learning (AFL) [3] addresses data heterogeneity by focusing on the worst-performing clients but similarly overlooks bandwidth usage. More recent studies address *resource-limited FL*, such as employing differential privacy in bandwidth- and energy-constrained contexts [6], adapting communication intervals for dynamic bandwidth networks [15], and exploring edge-based FL adaptation [7]. However, these works do not fully address the interplay among *client heterogeneity, parameter importance, compression, and strict bandwidth allocation*.

FedBand addresses these limitations by dynamically adapting compression ratios based on bandwidth availability and client data heterogeneity. Unlike fixed sparsification methods, FedBand prioritizes critical client updates, maintaining high accuracy with significantly reduced communication overhead, making it particularly effective for bandwidth-constrained, non-IID federated environments.

III. Problem Definition

A server coordinates with N clients sharing a fixed total bandwidth B, where B represents the total available bits per communication round. A fraction β (where $0 < \beta \leq 1$) of this bandwidth is allocated for training-related updates. For instance, if $\beta = 0.01$, then 1% of the total bandwidth is used for training communications. Thus, in each round, clients collectively share the bandwidth βB.

Traditional FL approaches, wherein clients transmit their entire local models or gradients, become impractical under realistic bandwidth constraints, leading to increased latency and prolonged convergence times. In such scenarios, allocating bandwidth efficiently among clients is a fundamental challenge. Since all clients share the same communication budget, determining how to distribute this budget in a way that maximizes learning efficiency while ensuring fairness remains an open problem.

Because of *non-IID* data distributions, some clients may have updates that are crucial to the global model, while others may contribute updates with minimal impact. Additionally, allocating bandwidth equally among all clients may lead to inefficient learning, as some clients may require more bandwidth to convey essential updates. This motivates the need for an adaptive bandwidth allocation strategy that can prioritize clients based on their importance to the global model while still maintaining fairness.

To formalize this constraint, the server enforces a global bandwidth budget, requiring that the total transmitted size satisfies $\sum_{i=1}^{N} C_i \leq \beta B$, where C_i is the size of the compressed parameter set sent by client i, and $i \in \{1, \ldots, N\}$. A key fairness consideration in FL is ensuring that no individual client is disproportionately disadvantaged due to its data distribution. To quantify model performance across heterogeneous clients, we measure the fidelity of the global model on each client's data distribution. Specifically, given a global model $g(\cdot)$ and a client's validation dataset D_i^{val}, we compute the empirical accuracy as:

$$\mathbb{E}_{(x,y)\sim D_i}[\mathbf{1}\{g(x) = y\}] = \frac{1}{|D_i^{\text{val}}|} \sum_{(x,y)\in D_i^{\text{val}}} \mathbf{1}\{g(x) = y\} \quad (1)$$

where $\mathbf{1}\{g(x) = y\}$ is an indicator function that returns 1 if the model correctly classifies x, and 0 otherwise. Since FL settings involve limited bandwidth, we must balance model fidelity across clients while adhering to a global bandwidth budget. Thus, the bandwidth allocation must ensure that clients with highly skewed data distributions receive sufficient communication resources to improve their local model performance. Given a total available bandwidth βB, our approach optimizes:

$$\max \min_{i\in\{1,\ldots,N\}} \mathbb{E}_{(x,y)\sim D_i}[\mathbf{1}\{g(x) = y\}] \quad \text{subject to} \quad \sum_{i=1}^{N} C_i \leq \beta B \quad (2)$$

where C_i is the size of the compressed parameter set transmitted by client i, and βB represents the total bandwidth allocated for training updates in each round. This formulation ensures that bandwidth allocation prioritizes fairness by improving the worst-performing clients, thereby enhancing overall model robustness in heterogeneous FL settings.

However, achieving this fairness goal under strict bandwidth constraints is challenging. The key issue is how to dynamically allocate the available bandwidth βB across N clients while accounting for data heterogeneity. Due to the non-IID nature of client data, some clients generate updates that are significantly more impactful to the global model in terms of improving accuracy while maintaining fairness. Consequently, a naive equal bandwidth allocation may lead to suboptimal convergence and fairness. The following section introduces our proposed solution, which adaptively adjusts bandwidth allocation based on client importance, data heterogeneity, and resource availability.

IV. Details of FedBand

FedBand is an FL framework designed to handle both *strict bandwidth constraints* and *non-IID client data*. Its key novelty lies in two intertwined mechanisms:

(i) **Adaptive bandwidth allocation**, where clients receive bandwidth in proportion to their *importance score*, determined via gradient norm or validation loss.

(ii) **Dynamic sparsification**, where each client transmits only its *top-k* most significant parameters, where k is dynamically determined based on the client's allocated bandwidth. This reduces communication overhead while preserving model accuracy.

By first computing each client's importance (either via the L2 norm of its gradient or its validation loss), FedBand dynamically assigns a fraction of the limited bandwidth to prioritize impactful updates. Clients with higher importance scores receive a larger bandwidth allocation and, correspondingly, can transmit more parameters. Figure 2 and Algorithm 1 outline FedBand's round-by-round process, described in detail below.

A. Overview and Key Assumptions

We adopt a synchronous FL setting, where the server coordinates communication rounds across clients—a common approach in FL literature [25]. Each round has a fixed bandwidth pool βB, which is shared among participating clients. While real-world networks employ various resource-sharing mechanisms, we assume that the server can allocate fractional bandwidth B_i to each client i. This assumption allows us to focus on optimizing bandwidth allocation across clients rather than modeling lower-layer wireless constraints such as spectrum contention or interference.

Additionally, we assume that downlink bandwidth is unconstrained during model transmission from the server to clients. In many practical scenarios (e.g., cell towers, satellites), downlink broadcasts naturally incur less overhead than uplink transfers. However, we acknowledge that this assumption may not hold in every real-world setting and leave the study of downlink bandwidth limitations for future work. Our approach nevertheless remains applicable to a wide range of network environments where bandwidth is dynamically shared among users.

B. Max-Min Optimization Approach

Recall from Section III that our objective is to maximize the *minimum local accuracy* while satisfying the constraint $\sum_{i=1}^{N} C_i \leq \beta B$. FedBand achieves this via an *iterative heuristic*: in each round, clients are ranked by an *importance score*, and bandwidth is allocated accordingly to maximize the global model's overall improvement.

Why prioritize outlier (high-loss or high-norm) clients? Allocating additional bandwidth to high-loss or high-norm clients may seem counterintuitive, but our findings indicate that targeting these underperforming clients *early* leads to *faster global convergence*. By prioritizing clients with significant deviations from the global model, FedBand ensures that their updates—which often contain critical information reflecting diverse data distributions—are incorporated promptly. This strategy improves worst-case accuracy, enhances gradient diversity, and accelerates overall training efficiency, leading to better generalization in heterogeneous FL environments.

Algorithm 1 FedBand Algorithm

```
Input: Clients $N = \{1,\dots,N\}$, Total Bandwidth $B$, Fraction $\beta$, Cache Table $CT$, Initial Global Model $\theta_{\text{global}}$
Output: Updated Global Model $\theta_{\text{global}}$
Step 0: Server Initialization
Broadcast initial model $\theta_{\text{global}}$ to all clients
Set equal bandwidth $B_i = \frac{\beta B}{N}$ for all clients in the first round
for each round $r$ do
    for each client $i$ in parallel do
        Evaluate $\theta_{\text{global}}$ on local validation set and compute $loss_i^r$
        Train locally and compute gradients $G_i$
        Compute importance score $S_i = \|\mathbf{g}_i\|$ or $S_i = loss_i^r$
        Compress updates $C_i$ based on allocated bandwidth $B_i$
        Transmit $\{S_i, C_i\}$ to the server
    end for
    Step 1: Server Aggregation and Model Update
    Compute aggregation weights $w_i = \frac{loss_i^r}{\sum_{j \in \mathscr{P}} loss_j^r}$
    Update $\theta_{\text{global}} \leftarrow \theta_{\text{global}} + \sum_{i \in \mathscr{P}} w_i \cdot C_i$
    if $r > 1$ then
        Step 2: Adaptive Bandwidth Allocation
        Compute total importance $S_{\text{total}} = \sum_i S_i$
        Update bandwidth $B_i = \frac{S_i}{S_{\text{total}}} \times \beta B$
    end if
    Broadcast updated model $\theta_{\text{global}}$ and new bandwidth allocations $B_i$ to clients
end for
```

C. Round-by-Round Workflow

Algorithm 1 summarizes FedBand's procedure, which we break down into:

(A) Server Initialization: The model architecture is predetermined and static during training. The server broadcasts an initial, untrained model to all clients. Uplink communication is bandwidth-constrained and carefully managed. In the first round, the server allocates equal bandwidth $B_i = \frac{\beta B}{N}$ to every client and broadcasts this allocation.

(B) Local Model Assessment: Each of the N clients maintains fixed training and validation datasets. In round r, upon receiving the global model, each client evaluates it on its validation set to compute local accuracy (acc_i^r) and local loss ($loss_i^r$). The client then performs local training (e.g., one or more epochs of gradient descent) to calculate gradients G_i. Finally, it transmits $loss_i^r$ to the server, enabling the server to gauge each client's importance in updating the global model.

(C) Computing Importance Scores and Bandwidth Allocation: Initially, the total available bandwidth (βB) is equally distributed among clients. However, in subsequent rounds, FedBand dynamically reallocates bandwidth based on each client's **importance score** S_i, which quantifies its contribution to improving the global model. Each client's importance score S_i is computed using one of two possible metrics:

- **Gradient Magnitude** ($\|\mathbf{g}_i\|$): The L2 norm of the client's gradient updates, given by $\|\mathbf{g}_i\| = \sqrt{\sum_j g_{i,j}^2}$, where $g_{i,j}$ represents the j-th gradient element of client i. A higher norm indicates larger deviations from the global model, suggesting that the client has significant information to contribute.
- **Validation Loss** ($loss_i$): The client's local loss on its validation dataset (distinct from training loss) measured after receiving the global model and before local training. It indicates the global model's generalization to client-specific data and highlights distribution shifts. A higher value signals poorer generalization and higher client uniqueness, guiding the server to allocate more bandwidth to improve the client's contribution.

After computing S_i for all clients, the server normalizes these scores and allocates bandwidth proportionally:

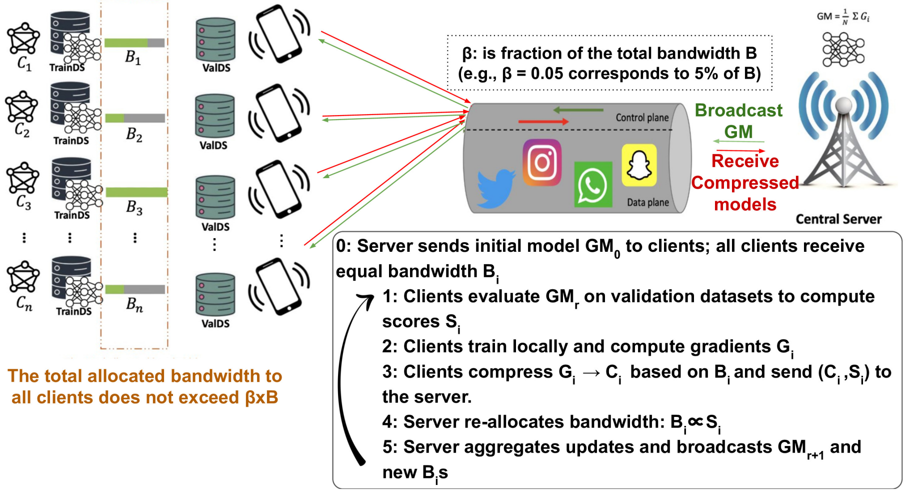
$$B_i = \frac{S_i}{\sum_{j=1}^{N} S_j} \times \beta B. \tag{3}$$

This adaptive allocation ensures that clients contributing the most relevant and diverse updates receive more bandwidth, thereby enhancing the global model's performance. The allocated bandwidth values B_i are broadcast to clients at the start of each round.

(D) Dynamic Compression Ratio Optimization and Gradient Transmission: FedBand dynamically determines how many gradient parameters each client can send in each communication round, ensuring efficient bandwidth utilization. Because full gradient transmission is impractical under strict bandwidth constraints, clients must compress their updates.

FedBand employs sparsification instead of alternative compression techniques such as quantization. Quantization reduces precision across all parameters uniformly, which can degrade important updates [17], while structured pruning removes predefined sets of weights without considering real-time importance. Sparsification, particularly Top-K selection, allows clients to transmit only the most significant updates based on their available bandwidth, maximizing impact while minimizing communication overhead. The process unfolds as follows:

- **Dynamic Bandwidth Allocation:** The total available bandwidth is shared across all clients, and each client is assigned an *individual* bandwidth fraction B_i based on its importance score. This allocation varies from round to round as FedBand dynamically reassigns bandwidth.
- **Bandwidth to Parameter Mapping:** Given its allocated bandwidth B_i, each client determines how many gradient updates it can afford to transmit. The available bandwidth (in MB) is first converted to bits, then divided by the number of bits required per transmitted parameter, which includes both its numerical value and associated metadata (such as indices for sparsification). This ensures that each client efficiently utilizes its bandwidth while transmitting only the most critical updates. Since B_i changes dynamically



β: is fraction of the total bandwidth B (e.g., β = 0.05 corresponds to 5% of B)

The total allocated bandwidth to all clients does not exceed βxB

0: Server sends initial model GM_0 to clients; all clients receive equal bandwidth B_i

1: Clients evaluate GM_r on validation datasets to compute scores S_i

2: Clients train locally and compute gradients G_i

3: Clients compress $G_i \rightarrow C_i$ based on B_i and send (C_i, S_i) to the server.

4: Server re-allocates bandwidth: $B_i \propto S_i$

5: Server aggregates updates and broadcasts GM_{r+1} and new B_is

Fig. 2: Workflow of FedBand.

each round, the number of transmittable parameters and the corresponding compression ratio also adapt accordingly.

- **Compression Ratio Calculation:** Since clients operate under different bandwidth constraints, each client computes its *own* compression ratio—the fraction of model parameters it can transmit in the given round. For example, if a client can send only 10,000 out of 1,000,000 model parameters, its compression ratio is 0.01 (1%). Because bandwidth allocations change every round, these ratios vary dynamically both *across clients* and *over time*.
- **Selecting and Transmitting the Most Significant Parameters:** Instead of selecting parameters arbitrarily, FedBand applies a sparsification strategy (e.g., top-K selection) to prioritize transmitting only the most significant gradient updates (those with the highest magnitudes). Each client determines the exact number of gradients it can send based on its bandwidth B_i, ensuring that communication resources are spent on the most impactful updates. Unlike uniform sparsification, where all clients transmit the same number of gradients, FedBand dynamically adjusts gradient transmission based on importance scores. This dynamic strategy ensures an efficient use of bandwidth for transferring critical updates while suppressing those that contribute little to the training process.
- **Memorization via Caching:** To optimize efficiency and avoid redundant calculations, FedBand employs a *cache-based memorization mechanism*. If a client's allocated bandwidth B_i in a new round is similar to a past value, the previously computed compression ratio is reused instead of being recalculated from scratch. This technique reduces computational overhead and speeds up adaptation to fluctuating bandwidth conditions.

(E) Aggregating Received Parameters: After the clients send their sparsified gradients $\Delta\theta_i$, the server aggregates them to update the global model. Unlike standard FedAvg, which weights updates by sample size, FedBand prioritizes updates from clients with higher validation losses to improve minimum local accuracy, aligning with the objective in (2). The global model update is:

$$\theta_{\text{global}} \leftarrow \theta_{\text{global}} + \sum_{i \in \mathscr{P}} w_i \cdot \Delta\theta_i \tag{4}$$

where, $\Delta\theta_i$ is the sparse gradient from client i, $w_i = \frac{loss_i}{\sum_{j \in \mathscr{P}} loss_j}$, and $\mathscr{P}$ is the set of participating clients in the current round. The weight w_i scales the update based on the client's normalized validation loss. This ensures that underperforming clients contribute more to the global model, enhancing robustness in non-IID environments. Note that if a client cannot send gradients due to bandwidth limits, its gradients are excluded.

V. Experiments and Results

We evaluate FedBand in both **simulated** and **emulated** environments to comprehensively analyze its performance under **realistic bandwidth constraints and non-IID settings**. Our experiments examine **accuracy**, **fairness**, and **convergence efficiency**, comparing FedBand against established baselines.

Simulation Environment: We conduct large-scale simulations using Python and PyTorch on a server with dual AMD EPYC 7543 32-Core Processors and 944GB RAM running Ubuntu 20.04 LTS. The number of clients, communication rounds, and bandwidth constraints are configured to reflect practical FL deployments. Each client locally trains on its dataset and transmits updates per communication round. Performance metrics are recorded over multiple rounds to evaluate **convergence trends, impact of non-IID data, and efficiency under strict bandwidth constraints**, following established FL benchmarking practices [1], [25], [26].

Emulation Environment: To validate our results in a realistic network setting, we used Mininet to emulate FL conditions with variable bandwidth, delays, and packet loss. Bandwidth was managed via Hierarchical Token Bucket (HTB) queuing, and the FL cycle includes GPU-based model training followed by server-side aggregation. CPU resources were carefully controlled to ensure fair allocation throughout the emulation.

In both environments, we set total bandwidth B to match the uncompressed size of client models per round, representing transmission without sparsification. We defined βB as a fraction of B (e.g., $\beta = 0.05$ or $\beta = 0.01$, corresponding to 5% or 1% of B) to simulate realistic constraints, with dynamic allocation based on validation loss or gradient norms (refer to Section IV).

To model realistic client heterogeneity, we introduce non-IID distributions using a Dirichlet distribution with concentration parameter α to control class imbalance across clients [8]. Clients received varying subsets of classes, with random sample sizes scaled to the total dataset size. Minimum thresholds ensured no client was left without data, producing non-IID distributions where both class types and sample counts varied across clients.

Baselines for Comparison: We compared FedBand with five baselines, all of which use weighted FedAvg for aggregation, as detailed in Section IV:

1) **OrgU1 (Original Unsparsified without bandwidth limit):** The full model is transmitted without compression, representing the upper performance bound.
2) **OrgU2 (Original Unsparsified with bandwidth limit):** The full model is transmitted under bandwidth constraints, showing the impact of limited capacity.
3) **Fixed top-k sparsification:** Transmits the largest gradients, with each client sending a fixed number of parameters per round.
4) **FedProx (Full Model Transmission):** A variant of FedAvg that modifies the **local training process** by introducing a proximal term to constrain client updates and improve stability in non-IID settings [9]. We evaluate it with two regularization values ($\mu = 0.1$ and $\mu = 1.0$), where clients send the full model without bandwidth constraints.
5) **Sparse FedProx (Sparsified with bandwidth limit):** A bandwidth-limited version of FedProx, where top-k sparsification is applied while maintaining its proximal term ($\mu = 0.1$ and $\mu = 1.0$).

Our evaluations focus on three metrics: 1) ***Average Local Accuracy***: Mean accuracy across all clients' validation datasets. 2) ***Minimum Local Accuracy***: The lowest accuracy among clients, reflecting fairness and generalizability. 3) ***Time to Convergence***: Time required for the model to converge, indicating training efficiency.

Why Local Accuracies? In FL, especially with non-IID data, client updates can vary significantly, leading to performance disparities. Evaluating both average and minimum local accuracy across all clients provides a balanced view: average accuracy measures overall model effectiveness, while minimum accuracy highlights fairness by ensuring no client is severely underperforming. This captures the trade-off between maximizing accuracy and maintaining equitable performance.

Research Question:. We explore the following: *How does FedBand's performance—measured by average and minimum accuracy—compare with other baselines, especially under varying data skew and realistic strict bandwidth constraints?*

Datasets. We use the ***CIFAR-10*** for image classification and ***UTMobileNet2021*** for traffic classification. Each experiment is repeated five times with different random seeds, and results are averaged for statistical robustness.

The **CIFAR-10** dataset [14] contains 60,000 images across ten classes, with 50,000 for training and 10,000 for testing.

The **UTMobileNet2021** dataset [5] includes traffic data from 14 applications, exhibiting class imbalance (e.g., 391 samples for YouTube vs. 21,004 for Facebook), leading to skewed representation across applications. To mitigate this imbalance, we apply Synthetic Minority Over-sampling Technique (SMOTE) [11] for underrepresented classes and Random Undersampling (RUS) [12] for overrepresented ones. The sampling threshold is set near the average class size to balance representation while avoiding excessive training overhead.

A. Results with the CIFAR-10 Dataset

Our experiments with the DenseNet169 model [20] on 100 simulated clients evaluate FedBand's performance under varying data skew and bandwidth constraints. Each client performs 1,000 iterations over 3 epochs. The total available bandwidth per round is set to $B = 5,400$MB, which corresponds to the full size of the model if transmitted without compression in each communication round. For FedBand, bandwidth is allocated dynamically based on gradient norms, which serve as importance scores to prioritize clients with more significant updates. *Note:* Due to space limitations, not all tables and figures are reported here, but the trends consistently align with the reported results.

1) Impact of Data Heterogeneity and Bandwidth Constraints: Table I presents the accuracy results across different levels of data heterogeneity (Dirichlet parameter α) and bandwidth constraints (β). The results demonstrate that FedBand consistently outperforms sparsified baselines while closely matching fully unsparsified baselines (OrgU1 and FedProx with full model transmission), demonstrating its ability to maintain high accuracy while significantly reducing communication overhead.

Comparison with Fully Unsparsified Models (OrgU1, FedProx): FedBand achieves accuracy comparable to OrgU1 and FedProx (Full Model) while transmitting only 5MB per round under stricter bandwidth constraints and 10MB per round under more relaxed conditions, significantly less than the 5,400MB required by unsparsified models (Table I). For instance, at $\alpha = 20.0$, FedBand attains 80% average accuracy, matching OrgU1 and FedProx ($\mu = 1.0, 0.1$), while using 99% less communication bandwidth (see Fig. 3). Even at high data skew ($\alpha = 0.5$), FedBand achieves 79% average accuracy, closely matching the performance of unsparsified baselines. These results underscore FedBand's capability to maintain high accuracy under extreme bandwidth constraints.

Comparison with Sparsification-based Baselines (Fixed Sparsification, Sparse FedProx): FedBand consistently outperforms Fixed Sparsification and Sparsified FedProx ($\mu = 1.0, 0.1$) across all evaluated conditions, with the gap widening as data becomes increasingly non-IID. At $\beta = 0.0018, \alpha = 20.0$, FedBand surpasses Sparse FedProx ($\mu = 1.0$) by 12% in average accuracy (82% vs. 70%) and 20% in minimum accuracy (74% vs. 54%). Under even tighter constraints ($\beta = 0.0009, \alpha = 0.5$), FedBand maintains a substantial advantage, reaching 77% average accuracy, significantly outperforming Sparse FedProx's 60%, and achieves 60% minimum accuracy compared to Sparse FedProx's 48% (As shown in Fig. 4). While FedProx attempts to address heterogeneity through proximal regularization, this approach alone is insufficient under strict bandwidth constraints

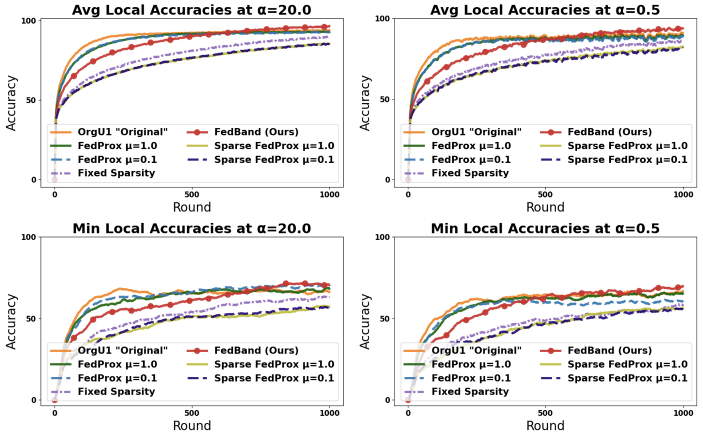


Fig. 3: CIFAR-10: Avg. and min. accuracies, $\beta = 0.0018$, 10MB per round for Fixed sparsification, Sparse FedProx, and FedBand, while OrgU1 ”Original” and FedProx send the full model 5,400MB with different α values.

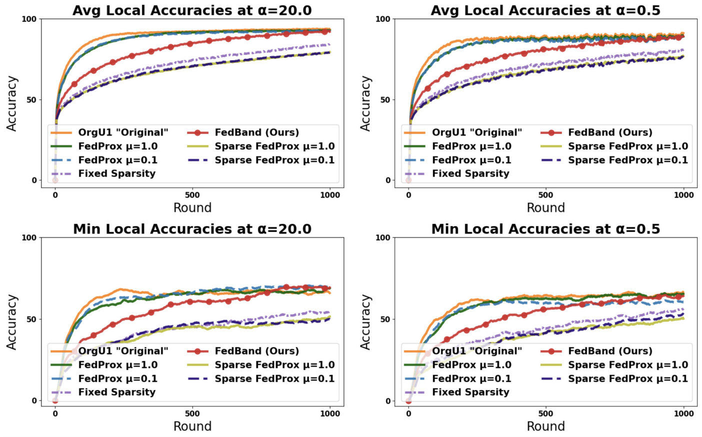


Fig. 4: CIFAR-10: Avg. and min. accuracies, $\beta = 0.0009$, 5MB per round for Fixed sparsification, Sparse FedProx, and FedBand, while OrgU1 ”Original” and FedProx send the full model 5,400MB with different α values.

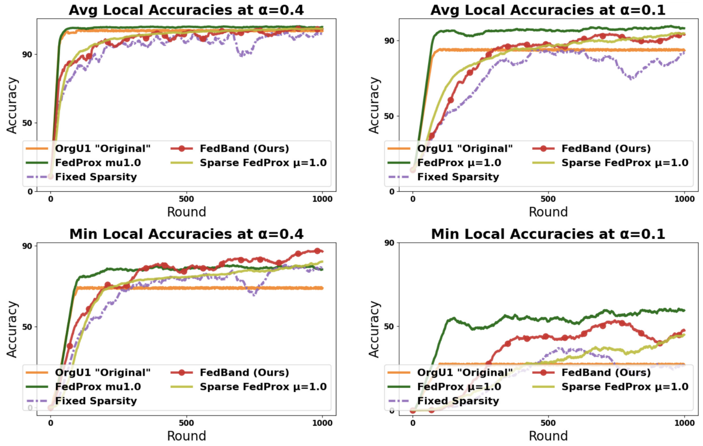


Fig. 5: UTMobileNet: Avg. and min. accuracies, $\beta = 0.0008$, 5MB per round for Fixed sparsification, Sparse FedProx, and FedBand, while OrgU1 ”Original” and FedProx send the full model 5,664MB with different α values.

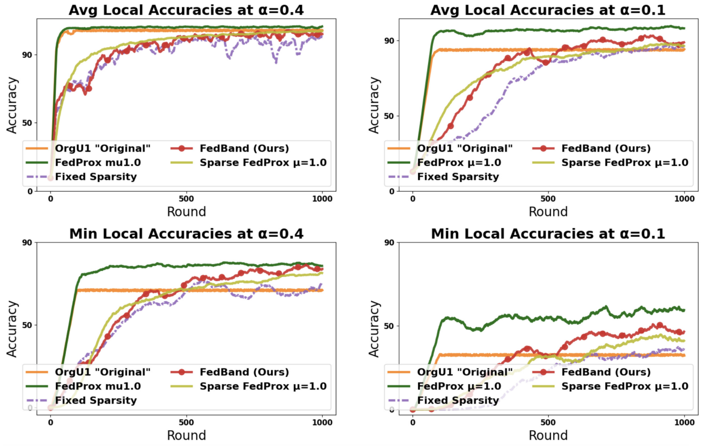


Fig. 6: UTMobileNet: Avg. and min. accuracies, $\beta = 0.0005$, 3MB per round for Fixed sparsification, Sparse FedProx, and FedBand, while OrgU1 ”Original” and FedProx send the full model 5,664MB with different α values.

due to its uniform sparsification, highlighting FedBand's superior adaptability.

2) Efficiency of Communication and Training Convergence: We evaluate convergence efficiency by comparing accuracy versus training time under identical bandwidth constraints. Our analysis reveals that while full-model transmission methods (OrgU2 and FedProx Full) incur significant communication overhead, leading to slower convergence, FedBand achieves comparable accuracy with drastically reduced transmission costs. At 30,000 seconds, with $\beta = 0.0009$ and $\alpha = 0.5$, FedBand reaches a minimum accuracy of 60%, significantly surpassing Fixed Sparsification's 42% and OrgU2's lower accuracy (32%) under the same constraints. This demonstrates FedBand's capability to enhance convergence by selectively transmitting the most impactful updates, thereby effectively managing communication overhead. Additionally, FedBand exhibits superior training efficiency compared to FedProx. Specifically, FedProx requires nearly **twice the computational time per round** due to its proximal regularization term, whereas FedBand completes each training round approximately 1.5× faster. These results indicate that FedBand's dynamic bandwidth allocation not only reduces communication overhead but also accelerates training convergence by prioritizing updates critical to model improvement.

3) Dynamic Compression Ratio Adaptability and Scalability: A key advantage of FedBand is its dynamic adjustment of compression ratios (CR) according to both bandwidth constraints (β) and data heterogeneity (α). As illustrated in Fig. 7, at $\beta = 0.0018$, FedBand's CR ranges from 0.0006–0.0012 ($\alpha = 20.0$) but dips to 0.0002–0.0011 ($\alpha = 0.5$), reflecting selective bandwidth allocation to more impactful clients under higher skew. Under stricter constraints ($\beta = 0.0009$), CR narrows to 0.0003–0.0006 ($\alpha = 20.0$) and can reach 0.0 when $\alpha = 0.5$. This indicates that in extreme cases where both bandwidth is highly limited and data is significantly skewed, FedBand may allocate no bandwidth to clients whose updates provide minimal or no contribution to the global model. Instead of wasting bandwidth on uninformative updates, FedBand prioritizes transmission from clients with more critical updates, ensuring efficient use of available resources. In contrast, static methods (Fixed, Sparse FedProx) apply fixed CRs (0.00104 at $\beta = 0.0018$, 0.00052 at $\beta = 0.0009$) regardless of skew, often leading to suboptimal performance in heterogeneous conditions. By continuously adjusting CR per client and round, FedBand balances accuracy, fairness, and communication overhead while scaling effectively as the number of clients grows, thus avoiding the inefficiencies of uniform allocation.

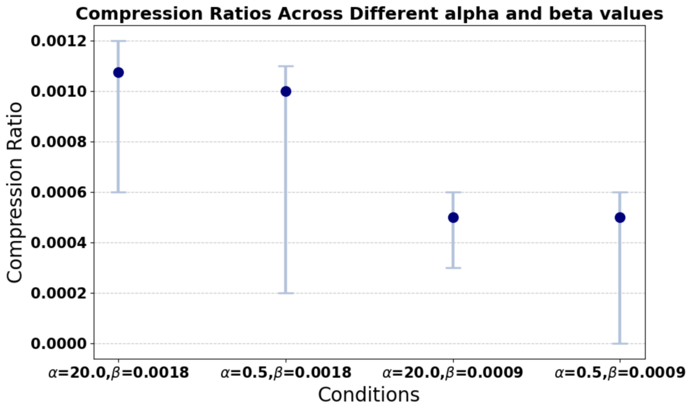


Fig. 7: Sample illustration of dynamic compression ratio variations on CIFAR-10 under different α (data skew) and β (bandwidth) conditions.

B. Results with the UTMobileNet2021 Dataset

Our experiments with the UTMobileNet2021 dataset involved 45 participants, each performing 1000 training iterations over 3 epochs using a custom ResNet50 model [24] for one-dimensional traffic data. The total bandwidth B was set to 5,664MB per round, corresponding to the full size of the model if transmitted without compression in each round. Bandwidth was allocated based on clients' validation loss as importance score, prioritizing critical updates. Again, due to space constraints, only selected tables and figures are included, with omitted results aligning with reported trends.

1) Impact of Data Heterogeneity and Bandwidth Constraints: Table I summarizes accuracy under varying data skew (α) and bandwidth constraints (β). FedBand consistently matches or slightly exceeds full-model baselines (OrgU1, FedProx) and Sparse FedProx in average accuracy, and clearly outperforms Fixed sparsification in both average and minimum accuracy.

Comparison with Fully Unsparsified Models (OrgU1, FedProx): FedBand closely matches OrgU1 and FedProx, achieving comparable or higher minimum and average accuracies despite significant bandwidth reduction. For instance, at $\beta = 0.0008$ and moderate skew ($\alpha = 0.4$), FedBand attains a minimum accuracy of 89%, notably surpassing OrgU1 (72%) and FedProx (80%) (see Fig. 5). This improvement indicates FedBand's effective bandwidth prioritization of critical updates.

Interestingly, FedProx performs better in this 1D traffic classification scenario compared to it's behavior in image classification, particularly under non-IID conditions. This enhanced performance arises from the simpler patterns and more distinguishable features present in 1D traffic data, allowing the proximal regularization term in FedProx to better manage local update divergence and client heterogeneity.

Comparison with Sparsification-based Baselines (Fixed Sparsification, Sparse FedProx): FedBand consistently outperforms Fixed sparsification across varying levels of skew. For instance, at $\beta = 0.0008$, $\alpha = 0.4$, FedBand achieves a 7% improvement in minimum accuracy compared to Fixed sparsification (89% vs. 82%). Moreover, Sparse FedProx performs similarly to FedBand in average accuracy but lags behind in minimum accuracy under more skewed conditions ($\alpha = 0.1$) (see Fig. 6), reflecting FedBand's superior adaptability to heterogeneous environments.

2) Efficiency of Communication and Training Convergence: Similar to CIFAR-10 results, FedBand shows notable convergence speed advantages by dynamically prioritizing critical updates. Under bandwidth constraints ($\beta = 0.0008$), FedBand achieves high accuracy significantly faster than Fixed sparsification, Sparse FedProx, and OrgU2, due to selective bandwidth allocation of impactful gradients. This demonstrates FedBand's effective strategy in reducing communication overhead and accelerating convergence.

3) Dynamic Compression Ratio Adaptability and Scalability: FedBand consistently exhibits dynamic compression ratio (CR) adaptability to both bandwidth constraints (β) and data heterogeneity (α), mirroring trends observed with CIFAR-10. Under moderate constraints ($\beta = 0.0008$), the CR varies from 0.0 to 0.0121, broadening as data skew increases ($\alpha = 0.1$), highlighting FedBand's robustness in prioritizing critical updates. Conversely, Fixed sparsification employs static CRs, limiting adaptability. These results confirm FedBand's superior efficiency and scalability in dynamically allocating bandwidth, particularly beneficial under highly heterogeneous and stringent bandwidth constraints.

Emulation Results: To assess FedBand under real-world constraints, we conducted Mininet-based emulations using CIFAR-10 with a network setup of 60MB bandwidth, 5ms latency, and a packet loss rate of 10^{-5}, simulating practical wireless conditions. Unlike idealized PyTorch or TensorFlow simulations that neglect network effects, Mininet models bandwidth contention, congestion, and delay variations, offering a more realistic evaluation of FL performance.

FedBand demonstrated significantly faster rounds than the unsparsified model (OrgU1) by reducing transmission overhead and optimizing packet delivery. Each round—including model transmission, local training, gradient compression, and aggregation—completed in 7.32s for FedBand, compared to 21.26s for OrgU1, achieving a 2.9× speedup. This improvement stems from FedBand's selective gradient transmission, which minimizes network congestion and prioritizes high-impact updates.

In an ideal setting without network overheads, FedBand completed a round in 6.64s, compared to 10.60s for OrgU1, indicating that while network constraints introduce delays, FedBand consistently improves communication efficiency. These results highlight its practical benefits in real-world FL deployments, mitigating communication bottlenecks and enhancing scalability under bandwidth and latency constraints.

VI. Conclusions

We addressed the challenge of federated learning (FL) under strict bandwidth constraints by proposing *FedBand*, a dynamic sparsification method that adaptively adjusts compression ratios based on client data heterogeneity and bandwidth limitations.

TABLE I: Accuracies with various models under different bandwidth β and α settings for CIFAR-10 and UTMobileNet2021 datasets.

Dataset	Condition	Models	Min Acc	Avg Acc
CIFAR-10	β 0.0018, α 20.0	OrgU1 "Original"	**0.72**	**0.80**
		FedProx $\mu = 1.0$	**0.72**	**0.80**
		FedProx $\mu = 0.1$	**0.74**	**0.80**
		Fixed top-k	0.56	0.74
		FedBand **(Ours)**	**0.74**	**0.82**
		Sparse FedProx $\mu = 1.0$	0.54	0.70
		Sparse FedProx $\mu = 0.1$	0.54	0.70
	β 0.0018, α 0.5	OrgU1 "Original"	**0.60**	**0.77**
		FedProx $\mu = 1.0$	**0.60**	**0.77**
		FedProx $\mu = 0.1$	0.55	**0.77**
		Fixed top-k	0.51	0.70
		FedBand (**Ours**)	**0.64**	**0.79**
		Sparse FedProx $\mu = 1.0$	0.50	0.65
		Sparse FedProx $\mu = 0.1$	0.50	0.65
	β 0.0009, α 20.0	OrgU1 "Original"	**0.72**	**0.80**
		FedProx $\mu = 1.0$	**0.72**	**0.80**
		FedProx $\mu = 0.1$	**0.74**	**0.80**
		Fixed top-k	0.50	0.65
		FedBand **(Ours)**	**0.74**	**0.80**
		Sparse FedProx $\mu = 1.0$	0.49	0.63
		Sparse FedProx $\mu = 0.1$	0.49	0.63
UTMobileNet	β 0.0008, α 0.4	OrgU1 "Original"	**0.72**	**0.95**
		FedProx $\mu = 1.0$	**0.80**	**0.95**
		Fixed top-k	0.82	0.95
		FedBand (**Ours**)	**0.89**	**0.95**
		Sparse FedProx $\mu = 1.0$	0.80	**0.95**
	β 0.0008, α 0.1	OrgU1 "Original"	0.38	**0.88**
		FedProx $\mu = 1.0$	0.60	**0.94**
		Fixed top-k	0.38	0.87
		FedBand (**Ours**)	0.49	**0.90**
		Sparse FedProx $\mu = 1.0$	0.48	**0.90**
	β 0.0005, α 0.4	OrgU1 "Original"	**0.72**	**0.95**
		FedProx $\mu = 1.0$	**0.80**	**0.95**
		Fixed top-k	0.72	0.93
		FedBand (**Ours**)	**0.80**	**0.95**
		Sparse FedProx $\mu = 1.0$	0.77	**0.95**

Extensive experiments demonstrated that FedBand consistently outperforms static sparsification methods and sparsified FedProx across various conditions, achieving higher accuracy, faster convergence, and enhanced fairness. Furthermore, FedBand matches the performance of unsparsified baselines (OrgU1 and FedProx with full model transmission) while significantly reducing communication overhead, highlighting its suitability for efficient and scalable FL deployments in bandwidth-constrained and heterogeneous environments.

Acknowledgements

This work was partially supported by the NSF CNS grant 210698. The research was also partially sponsored by the OUSD (R&E)/RT&L and was accomplished under Cooperative Agreement Number W911NF-20-2-0267. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the ARL, OUSD(R&E)/RT&L or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[1] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, 2020.

[2] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," *arXiv:1704.05021*, 2017.

[3] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic Federated Learning," in *Proc. ICML*, 2019.

[4] J. Wang, Y. Lu, B. Yuan, B. Chen, P. Liang, C. De Sa, C. Re, and C. Zhang, "CocktailSGD: Fine-tuning foundation models over 500Mbps networks," in *Proc. ICML*, 2023.

[5] The University of Texas at Austin, "UTMobileNet2021 dataset," available at https://utexas.app.box.com/s/okrimcsz1mn9ec4j667kbb00d9gt16ii.

[6] R. Kerkouche, "Differentially private federated learning for bandwidth and energy constrained environments," *Performance Eval. [cs.PF]*, (Ph.D. thesis) Université Grenoble Alpes, 2021.

[7] S. Wang, A. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE JSAC.*, vol. 37, no. 6, 2019.

[8] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian Data Analysis*, 3rd ed. Boca Raton, FL, USA: CRC Press, 2013.

[9] Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., & Smith, V., "Federated optimization in heterogeneous networks," in *Proc. MLSys*, 2020.

[10] B. Isik, F. Pase, D. Gunduz, T. Weissman, and M. Zorzi, "Sparse random networks for communication-efficient federated learning," in *Proc. ICLR*, 2023.

[11] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16.

[12] J. Prusa, T. M. Khoshgoftaar, D. J. Dittman, and A. Napolitano, "Using random undersampling to alleviate class imbalance on tweet sentiment data," in *Proc. IEEE Int. Conf. Inf. Reuse Integr.*, 2015.

[13] Cloudflare, "What is the control plane? — Control plane vs. data plane," *Cloudflare Learning Center*, 2024. Available: https://www.cloudflare.com/learning/network-layer/what-is-the-control-plane/.

[14] A. Krizhevsky, "The CIFAR-10 dataset," available at https://www.cs.toronto.edu/~kriz/cifar.html, 2009.

[15] X. Zhang, J. Xiong, X. Wu, Z. Zhang, and L. Zhou, "Federated learning with adaptive communication compression under dynamic bandwidth and unreliable networks," *Inf. Sci.*, vol. 540, 2020.

[16] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "Tackling the objective inconsistency problem in heterogeneous federated optimization," in *Proc. NeurIPS*, 2020.

[17] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," in *Proc. NeurIPS*, Long Beach, CA, USA, 2017.

[18] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *Proc. ICML*, 2020.

[19] A. Z. Tan, H. Yu, L. Cui, and Q. Yang, "Towards personalized federated learning," *IEEE Trans. Neural Netw. Learn. Syst.*, 2022.

[20] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE CVPR*, 2017.

[21] M. Zhang, K. Sapra, S. Fidler, S. Yeung, and J. M. Alvarez, "Personalized federated learning with first-order model optimization," *arXiv:2012.08565*, 2020.

[22] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning: A meta-learning approach," *arXiv:2002.07948*, 2020.

[23] D. Li and J. Wang, "FedMD: Heterogeneous federated learning via model distillation," *arXiv:1910.03581*, 2019.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, 2016.

[25] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. AISTATS*, 2017.

[26] F. Sattler, S. Wiedemann, and K.-R. Müller, "Sparse binary compression: Towards distributed deep learning with minimal communication," in *Proc. IJCNN*, IEEE, 2019.

[27] Mininet, "Mininet: An instant virtual network on your laptop (or other PC)," available at https://mininet.org/.