Dean&Francis
ISSN 2959-6157

Survey of Communication Modification on Federated Learning

Xi Wang

Ohio State University, Columbus, United States
wang.20299@buckeyemail.osu.edu

Abstract:

The widespread adoption of artificial intelligence technologies, particularly deep learning, has exposed significant security vulnerabilities that pose substantial challenges to cyberspace safety. Traditional cloud-centric distributed machine learning relies on centralized data collection from participants, making the system vulnerable to security breaches and privacy violations during data exchange and model updates. These risks often result in system performance degradation or sensitive data exposure. Federated Learning emerges as a privacy-preserving distributed machine learning paradigm that mitigates these issues. By facilitating encrypted model parameter exchanges between clients and a central parameter server, while retaining raw data locally, Federated Learning, unlike other training model, enables collaborative model training with significantly reduced privacy leakage risks, at the same time it maintains the performance of the whole training process. However, as deep learning models scale and Federated Learning tasks grow more complex, communication overhead becomes a major barrier to deployment. Consequently, optimizing FL communication efficiency has become a critical research focus.

Keywords: Federated learning; Model compression; Communication optimizations.

1. Introduction

In the period of big data, rapid advancements in the internet development and web applications have driven explosive growth in data generated at the network edge [1].

Traditional machine learning models rely on centralized training of large-scale datasets, typically implemented through deep neural networks (DNNs). However, real-world data faces significant constraints-including privacy regulations and commercial competition that prevent centralized data pooling.

These challenges render centralized model training increasingly impractical in practical applications [2]. This article provides a comprehensive and systematic survey of communication optimization methods and techniques for Federated Learning in recent years. We commence with a formal exposition of communication efficiency and its influencing factors, sub-

sequently classifying pertinent research based on optimization objectives. The scope of our review covers diverse aspects, ranging from underlying hardware and network topology organization to communication protocols, application layer communication strategies, and parameter compression. Furthermore, we delineate the developmental trajectory of this research domain, propose future research directions. And furnish novel insights to inform forthcoming investigations into Federated Learning communication optimization.

2. Federated Learning

Federated Learning Model is an essential improvement within the development of machine learning technology [3]. It is an advanced deep learning model that allows local clients to exchange and aggregate the model from the local model without sharing their private information [4]. This concept was first introduced by Google in 2016 and has quickly become a popular collaborative training method since it solved the security problem, which clients' sensitive data is being sent to the central server and being leaked [5].

The core objective of Federated Learning (FL) is to enable a set of clients N to collaboratively train a deep learning model through iterative rounds, minimizing a global loss function $f(w)$, where w represents the model parameters [6]. This is formalized by the optimization problem:

$$\min_{w} f(w) := \frac{1}{|N|}\sum_{n=1}^{|N|} f_n(w). \quad (1)$$

Here, $fn(w)$ denotes the local loss function for client n, and the global objective $f(w)$ aggregates knowledge from all clients to achieve a generalized model [6].

During each Federated Learning round, the client $n \in N$ executes local training using its private data: it updates its local parameters w_t^n over one or more epochs by computing gradient descent steps $\nabla l\left(w_t^n; m\right)$ on minibatches of fixed size m. This results in the updated parameters $w_{t+1}^n = w_t^n - \eta \nabla l(w_t^n; m)$, where denotes the learning rate. Following local training, only these updated model parameters w_{t+1}^n are communicated back to the server for aggregation; raw training data and intermediate gradients are never shared, preserving data privacy [6].

Federated Learning sends deep learning model training tasks to edge devices and lets the clients leverage their local data to train models locally. These locally trained models are then uploaded to the central server (PS) and aggregated into a global model. This deep learning method enables the collaborative training of a model incorporating the essential information from the diverse datasets of all participants without requiring the sharing of raw data. The communication pattern in federated learning follows a classic client-server architecture. Therefore, we will subsequently refer to the participants as clients and the parameter server as the server [7].

In realistic applications, centralized architecture is more frequently used [5]. The process is illustrated in Fig. 1 [6]. First, the central server initialized a global model to prepare for an update.

Second, the central server chooses the clients that satisfy the condition of the federated learning strategy and sends the global model to the chosen clients so they can present a local update. Third, the local clients download the global model and train it based on their local data. The updated model parameters are uploaded to the central server for aggregation. Fourth, the central server uses the chosen aggregation strategy to aggregate a new model after receiving the updated model parameters and distributes the new global model to the clients [8].

ISSN 2959-6157

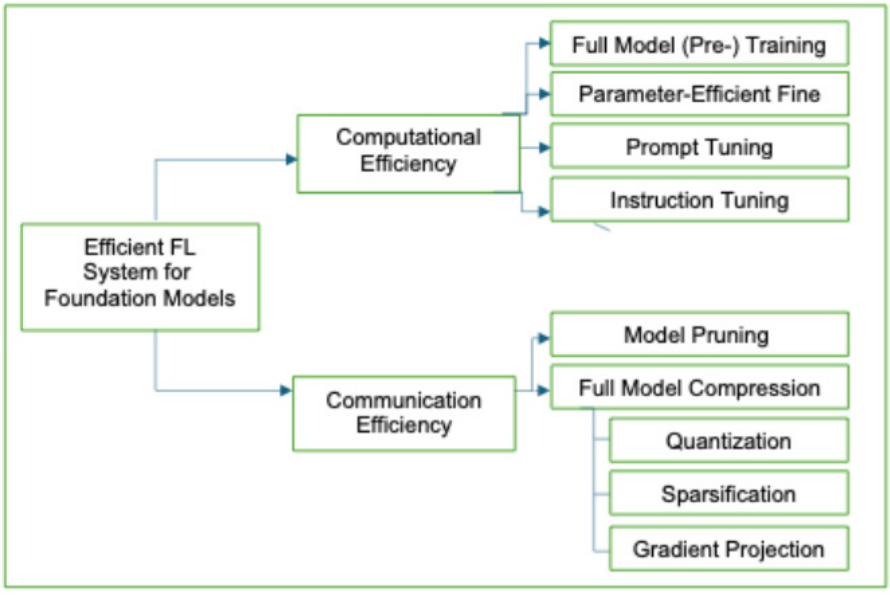

Fig. 1: Taxonomy Fundatation Model

When the Federated Learning model uses smaller models with fewer than 1 million parameters, the time spent on model parameter exchange between clients and the central server can be similar to the time spent on local model training. Usually, the computation should dominate the whole process. As the model's parameters increase, the time spent on communication is eventually larger than the computation time. This property makes the federated learning model desirable for large-scale training tasks, since it enables the training with models having hundreds of millions of parameters or more. However, the cost of transmitting model updates remains substantial. Hence, optimizing training efficiency and communication is the primary focus [6].

Nowadays, unlike before, the deep learning model can consist of billions or even trillions of learnable parameters. The federated Learning model needs to exchange the model's parameters frequently to get updates. Therefore, the communication overhead occurs during this process, which is a challenge for Federated Learning to improve. Also, with the involvement of large-scale models and the increase in participants, the clients' network speed may meet the limitation of a few hundred kbps to Mbps. The clients may be blocked by the communication burden. The heterogeneity and randomness of possibly a huge number of local devices in the learning process may also rapidly enlarge the communication cost. Since devices usually vary in hardware and network bandwidth. The network issue may also cause the local devices to randomly drop out. It is also inevitable to meet the heterogeneity problem with a large-scale Federated Learning model [9].

Therefore, how to handle these issues has become the common goal of different researchers. Generally, the total amount of data that needs to be transmitted and the bandwidth setting have a huge influence on the communication efficiency [10].

3. Communication Optimization Method for Federated Learning Model

The overall target of communication optimization is to reduce the total amount of updated models' parameters during the transmission process, increase the speed at which updated parameters of models are transmitted, and reduce the time spent on data exchange between the central server and the local server. The communication efficiency is present as the following formula:

$$E \propto \frac{Acc}{C \times f \times P \times T}. \tag{2}$$

Communication efficiency (E) in Federated Learning is defined as a metric proportional to the global model accuracy (Acc) and inversely proportional to the product of four key factors:

1. Client selection ratio (C): The fraction of clients participating per communication round.

2. Communication frequency (f): The ratio of total local training rounds needed (N) to the number of rounds performed per federated round.

3. Parameter size (P): The total bits transmitted per communication.

4. Round time (T): The sum of local training (t_{train}), transmission (t_{trans}), and server aggregation (t_{agg}) time per federated round.

Therefore E , it increases with higher accuracy but diminishes with larger client cohorts, more frequent communication, communication overhead increase, or slower round completion.

There are four categories of methods to optimize communication efficiency [7].

1. Parameter compression methods: aim to reduce the volume of model parameters transmitted per communication round. When global model accuracy remains unchanged, decreasing the transmitted parameter volume directly enhances communication efficiency.

2. Model update strategies methods: aim to enhance communication efficiency by modifying the frequency and mechanism of model updates. These approaches reduce overall communication costs by either decreasing the number of communication rounds or indirectly minimizing the total round time T.

3. System architecture-oriented optimization methods: enhance communication efficiency by holistically redesigning the federated learning infrastructure, including network topology adjustments, dynamic client scheduling (optimizing the client selection ratio C), and combinatorial system modeling. These approaches collectively reduce communication bottlenecks, such as transmission latency and synchronization overhead, through strategies like hierarchical aggregation networks, resource-aware participant selection, and joint configuration of architectural components. By simultaneously optimizing interdependent variables (notably C and round time T), this paradigm improves overall efficiency without compromising model performance.

4. Communication protocol optimization methods target the transmission mechanism and data content exchanged between clients and servers in federated learning. By redesigning communication protocols, such as adopting efficient encoding schemes, message compression, or asynchronous messaging. These techniques directly reduce transmission latency and parameter transfer time, thereby minimizing the total communication round time $T = t_{train} + t_{trans} + t_{agg}$. This optimization enhances overall efficiency without altering model architecture or training processes [7].

3.1 Parameters Compression Methods:

Since the main contents in Federated Learning are the models' parameters exchange between the central server and clients, including model weight, gradient descent, and update. Other extra information within the data transmission is basically redundancy. Therefore, we can try to compress the model parameters and reduce the redundant information during transmission, then the amount of data transmission per round can be largely reduced, and finally, reduce the total data amount of communication [7].

Cui invents a method for optimizing communication in Federated Learning. The method involves the central server first gathering key distributed training parameters, including the model's learning rate and each client's data proportion. It then dynamically determines the optimal number of compression centroids for the current iteration using these parameters and a predefined compression objective function. The server sends both the current global model and this centroid count to the clients. Each client trains the model locally using its own data to generate model updates, compresses these updates using a compression rate derived from the provided centroid count, and sends the compressed data back to the server. Finally, the server aggregates the compressed updates from all clients to produce the updated model for that iteration, significantly improving overall model precision. The modification process is illustrates as Fig. 2 below [11].

ISSN 2959-6157

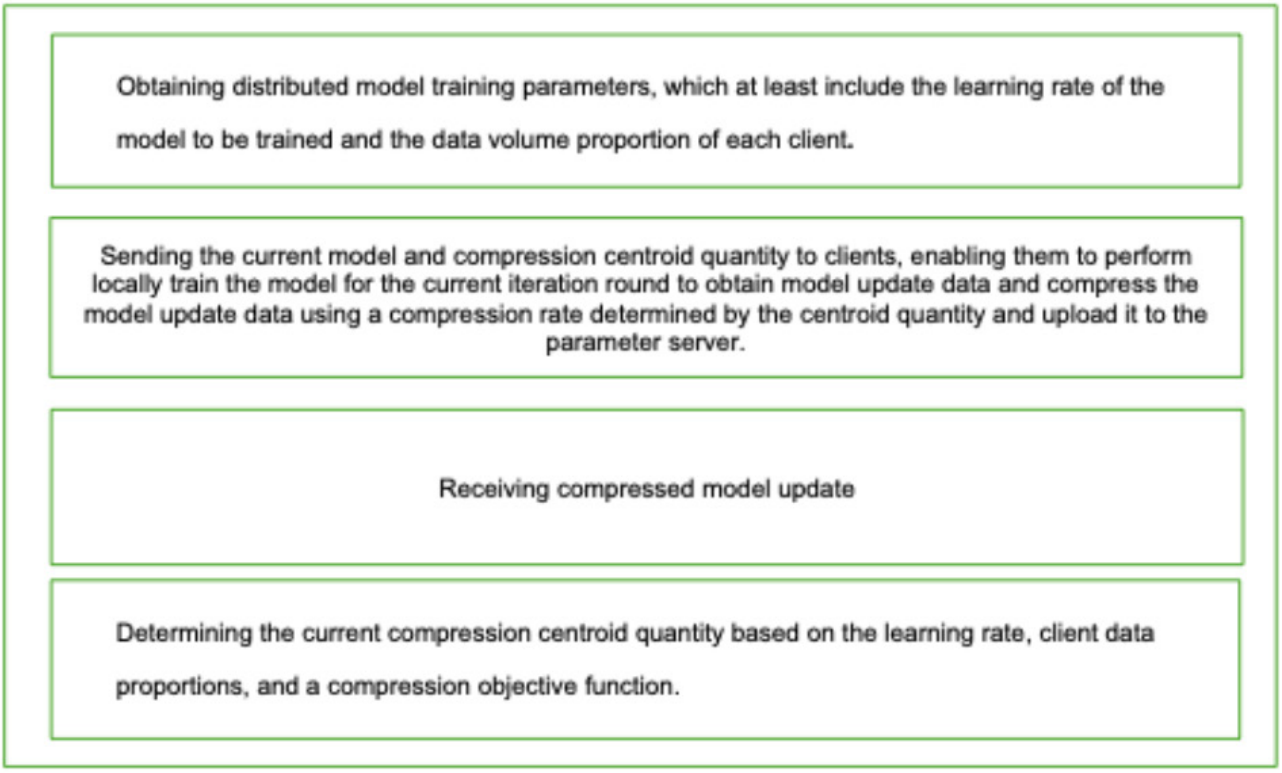

Fig. 2: Optimization Method on Communication Compression

Based on the learning rate of the model to be trained, the percentage of data amount from clients, and the predefined compression objective function, we can determine the optimal number of compression centroids by the formula below:

$$\Lambda_t = \frac{hd}{log_2 Z_t d + Z_t d}.\#(3)$$

Λ_t denote the compression rate at iteration t, determined by the uncompression bit width h, the model update dimension d, and the controid count Z_t for that iteration [11].

Nguyen proposes an approach that performs the pruning on the central server instead of on individual local devices. Unlike conventional approaches that execute all operations on diverse local devices, this method not only avoids the higher computational costs associated with on-device pruning but also resolves the issue of aggregating disparate pruning masks. These differing masks arise due to variations in local models, complicating the creation of a unified global model [12].

This approach relies on structured pruning, which is removing entire filters rather than removing individual weight elements. Unstructured pruning creates a sparse network with many zero values and requires additional storage for information like sparsity masks or compressed row formats. Consequently, pruning entire filters preserves the dense structure of the global model and ensures compatibility with the limited computational capabilities of client devices [12].

The Fig. 3 below illustrates the proposed framework for pruning CNNs within a Federated Learning system. The framework operates in two distinct stages. The first stage (part a) conducts automatic architecture search via pruning, dynamically eliminating less useful filters to identify an optimized model structure. This phase prioritizes architectural efficiency over immediate accuracy, focusing solely on reducing the global model's size. Once the final architecture is determined, the process moves to the final architecture is determined, the process moves to the second stage(part b), which involves standard FL training. This stage is dedicated to training the finalized architecture to achieve the task-specific target accuracy.

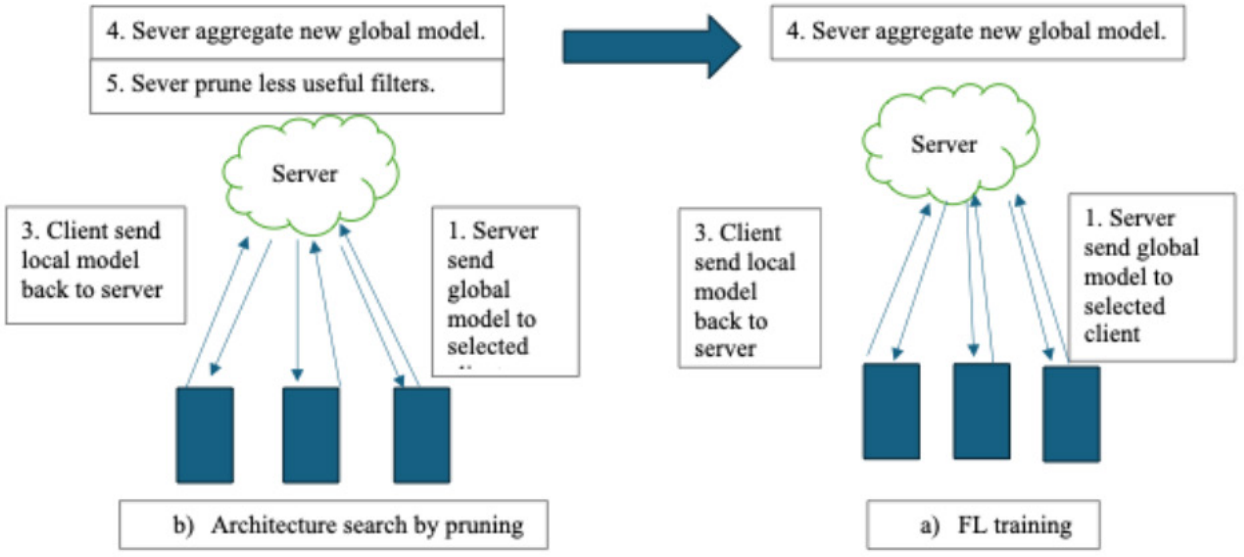

Fig. 3: An overview of our pruning scheme in Federated Learning system.

3.2 Optimization Method towards Model Update Strategy:

The model update strategy is the method that the model chooses to transmit model parameters between the central server and local devices. There are three categories of model parameter transmission methods:

1. Synchronization method, the central server has to wait for all chosen local devices to finish the upload of model parameters and then operate the aggregation. This is the most commonly used transmission method in Federated Learning.

2. Asynchronous method allows local clients to upload the model parameters immediately after the training process, then the central server returns the updated parameters, or allows local clients to continue the training process even without the parameters upload process finished or the updated parameters return process from the central server.

3. Semi-asynchronous method, this is the combination of both the synchronization method and the asynchronous method. Some existing research focuses on and optimizes the model parameters update strategy in order to reduce the time spent on the central server and local devices waiting for the response of each other, to achieve the goal of improving the communication efficiency [7].

To enhance communication efficiency in peer-to-peer networks without compromising model accuracy, Alka's team created an algorithm. This algorithm identifies the best action a device can take by evaluating key attributes of its connected peers, including their dataset size, accuracy, CPU speed, available RAM, bandwidth, number of connections, and data privacy constraints [13].

The algorithm operates in dynamic peer-to-peer Federated Learning networks where devices join unpredictably and hold non-IID data distributions. The algorithm optimizes communication efficiency by dynamically selecting the most cost-effective knowledge exchange strategy for each device. It achieves this by first exploring the network breadth-first search up to a configurable depth to identify connected peers, then ranks potential actions based on peer characteristics, such as dataset size, accuracy, CPU resources, bandwidth, and privacy constraints. Finally, it executes actions within the device's bandwidth limit, prioritizing those offering the highest expected accuracy gain per unit communication cost. This adaptive approach favors lightweight model sharing in bandwidth-scarce or high-priority scenarios and data sharing, where it accelerates convergence, reduces overhead while preserving accuracy, achieving up to 9.08% efficiency gains in time-limited knowledge transfer. The training process is illustrated in Fig. 4 below [13].

ISSN 2959-6157

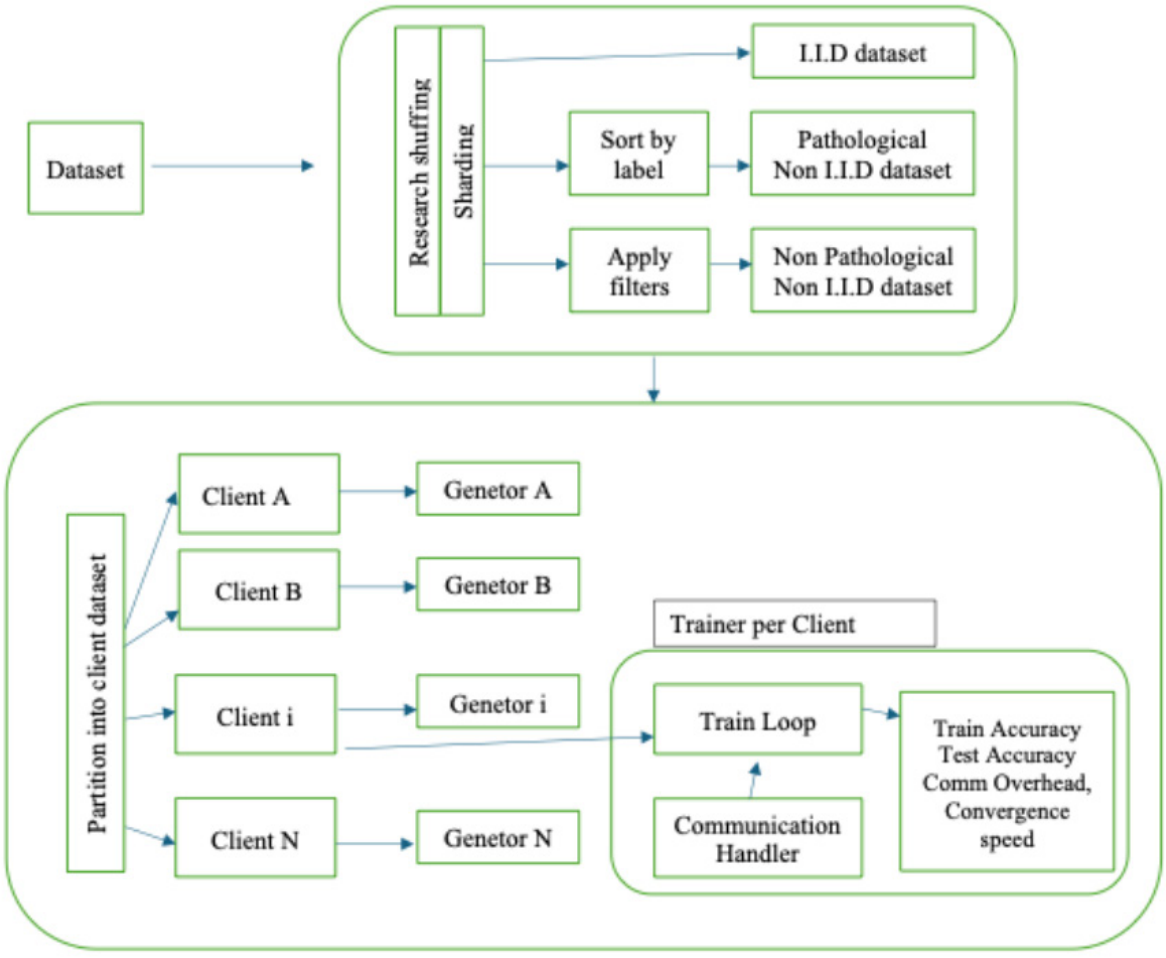

Fig. 4: Workflow depicting Dataset Handler

3.3 Optimization method toward system structure

Federated Learning system architecture optimization targets improvements in client organization and scheduling, network topology, and resource configuration. In contrast to fine grained optimizations such as modle parameter compression or update refinement, this architectural level optimization or update refinement, this architectural level optimization adopts a broader perspective. It seeks to improve federated learning efficiency by optimizing the overall system design [7].

Zhu introduces a physical layer transmission architecture for FEEL, which optimizes the communication efficiency by fundamentally redesigning the architecture. The proposed Broadband Analog Aggregation (BBA) system leverages simultaneous analog transmission over OFDM sub-channels, exploiting wireless channel superposition to compute aggregated model updates directly over the air [14]. Key structural innovations include:

1. Analog Modulation: Devices transmit high-dimensional model updates as continuous amplitude modulated symbols rather than digital bits, enabling the server to harness waveform superposition for instantaneous averaging.
2. OFDM Integration: The broadband channel is partitioned into orthogonal sub-channels, each dedicated to transmitting one model parameter. This handles frequency-selective fading while enabling parallel parameter aggregation.
3. Truncated Channel Inversion: Devices invert only strong sub-channels (above the cutoff threshold gth) to align received signal amplitudes, silencing weak channels to conserve power. This creates an SNR-truncation tradeoff (higher SNR vs. more dropped parameters).
4. Spatial Scheduling: Cell interior devices (near the server) are prioritized to limit path loss-induced SNR degradation. In low mobility networks, interior edge alternating scheduling balances data inclusion from distant devices while maintaining SNR.

3.4 Optimization Method toward Communication Protocols

Communication protocols serve as the fundamental infrastructure for model parameter transmission in Federated Learning. Implementing high-speed protocols or optimizing existing ones effectively mitigates communication latency and increases transmission efficiency.

Current research relies on optimizing underlying communication protocols in Federated Learning. Introducing high-bandwidth, low-latency protocols or adapting existing ones represents a promising research direction. High-speed protocols enable rapid, efficient bulk data transfer between networked devices through high bandwidth, low latency, and reliable transmission. Specifically, RDMA leverages zero-copy kernel bypass and CPU offload techniques to permit direct network interface access to application memory, circumventing the OS network stack. This achieves higher bandwidth, lower latency, and reduced CPY overhead [15].

Xie and his team address stragglers and high-latency bottlenecks in Federated Learning caused by system heterogeneity. The authors propose an asynchronous Federated Learning framework that eliminates synchronization barriers through dynamic client-server coordination. In this paradigm, they design a protocol where the server immediately aggregates updates from any available client, bypassing waiting periods for slower devices. To mitigate instability from outdated updates, the authors introduce adaptive aggregation weighting, prioritizing contributions from clients with fresher updates. Their strategy minimizes idle time and prevents network congestion. The authors establish formal convergence guarantees for non-IID data distributions and partial client participation. Collectively, these innovations ensure resilient performance in volatile networks, achieving empirically observed latency reduction of 3-5x compared to synchronous baselines like FedAvg [16].

4 Conclusion

This paper gives a comprehensive overview of the communication optimization method within Federated Learning. The overall range of targets covered by the communication optimization is broad, including not only the basic reduction of overall communication data amount and communication delay, but also the reduction of communication rounds and the total time spent. The optimization methods are divided into four categories, which are the Parameter Compression Methods, Optimization Method towards Model Update Strategy, Optimization Method towards System Structure, and the Optimization Method towards Communication Protocols. This paper briefly explains and analyzes the current state of research on diverse optimization methods. While communication optimization can significantly enhance the efficiency of FL, most methods incur additional computational or resource costs. For example, some methods reduce the communication rounds by increasing the communication delay. Basically, all of the parameter compression methods bring additional calculation cost, which potentially reduces the training speed. Resource settings can also be a problem that researchers have to handle, such as energy consumption and network broadband allocation. These extra prices may be enlarged on the edge. Therefore, it is crucial to consider the uniqueness of the edge environment.

To sum up, the optimization of Federated Learning is a complicated research problem; it requires researchers to consider the application and the balance of each optimization method in the real-world environment.

References

[1] WANG S, TUOR T, SALONIDIS T, etc. Adaptive Federated Learning in Resource Constrained Edge Computing Systems. IEEE Journal on Selected Areas in Communications, 2019, 37(6): 1205-1221.

[2] Li Ke, Wang Xiaofeng, Wang Hu. Personalized federated multi-task learning optimization method for heterogeneous data. Application Research of Computers, 2024, 41 (9): 2641-2648.

[3] YURDEM B, KUZLU M, GULLU M K, et al. Federated learning: Overview, strategies, applications, tools and future directions. Heliyon, 2024, 10: e38137.

[4] MAMMEN P M. Federated Learning: Opportunities and Challenges. (2021).

[5] Amjady N. Short-term hourly load forecasting using time series modeling with peak load estimation capability. IEEE Transactions on Power Systems, 2001, 16(4): 798-805.

[6] WOISETSCHLAGER H, ERBEN A, WANG S, et al. A Survey on Efficient Federated Learning Methods for Foundation Model Training. 2024.

[7] KAIROUZ P, MCMAHAN H B, AVENT B, etc. Advances and Open Problems in Federated Learning[J]. Foundations and Trends® in Machine Learning, 2021, 14(1–2): 1–210.

[8] ALEDHARI M, RAZZAK R, PARIZI R M, et al. Federated Learning: A Survey on Enabling Technologies, Protocols, and Applications. IEEE Access, 2020, 8: 140699-140725

[9] LI Q, DUAN Z, LIU Y, etc. Federated Learning on Non-IID Data: A Survey. IEEE Transactions on Neural Networks and Learning Systems, 2022, 33(12): 7149–7180.

[10] LIU X, LI G, WANG Z, etc. A Communication-Efficient Federated Learning Approach via Dynamic Mutual Distillation for Image Recognition. IEEE Transactions on Neural Networks

ISSN 2959-6157

and Learning Systems, 2023, 34(11): 8741–8753.
[11] ShenZhen University. Optimization Method, System and Device for Federated Learning Communication compression: CN 114861790 B[P]. 2023-03-17.
[12] NGUYEN T V, LE L B, AVILA A. Automatic Structured Pruning for Efficient Architecture in Federated Learning. 2024.
[13] LUGMAN A, BRANDON Y W L, CHATTOPADHYAY A. Federated Learning Optimization: A Comparative Study of Data and Model Exchange Strategies in Dynamic Networks[EB/OL]. (2024)[2024-07-27].
[14] ZHU G X, WANG Y, HUANG K B. Broadband Analog Aggregation for Low-Latency Federated Edge Learning. IEEE Transactions on Wireless Communications, 2019, 18(12): 5905-5919.
[15] DONMIC M K. Optimization of Federated Learning Protocols for Privacy-Preserving Distributed Model Training on Heterogeneous Medical Datasets with Variable Data Quality. International Journal of Artificial Intelligence Research and Development, 2023, 1(1): 99-105.
[16] XIE C, KOYEJO O, GUPTA I. Asynchronous Federated Optimization. Proceedings of the NeurIPS 2019 Workshop on Federated Learning for Data Privacy and Confidentiality. 2019.