North Asian International Research Journal of Sciences, Engineering & I.T.

IRJIF. I.F. : 3.821

ISSN: 2454-7514 Vol. 10, Issue-1, January-2024

Index Copernicus Value: 52.88 Indian Citation Index Thomson Reuters ID: S-8304-2016

NAIRJC UGC-Aligned | HEI-Approved Peer Reviewed Refereed Journal (Under New UGC Guidelines)

DOI: doiglobal.org/10.2024/NAIRJCSEIT.003

ADAPTIVE EDGE ACCELERATION: DYNAMIC PARTITIONING AND PRECISION SCALING FOR DECENTRALIZED FEDERATED NETWORKS

NAGARAJA. K.V.

Associate professor, Dept of Computer science, Government First Grade Women's College, Tumkur.

ABSTRACT

Federated Learning (FL) allows distributed edge infrastructure to collaboratively train predictive architectures while maintaining local data privacy. However, deploying synchronous FL algorithms across non-uniform hardware nodes introduces major delays, primarily caused by uneven computational capacities, constrained memory, and unreliable wireless links. Standard round-based aggregation strategies like Federated Averaging (FedAvg) force high-performing endpoints to remain idle during client update phases. To overcome this, we introduce FedEdge, an adaptive framework designed to balance communication and computation overhead across heterogeneous clusters. FedEdge dynamically partitions model layers between client devices and edge aggregators while continuously scaling gradient numerical precision (from INT4 up to FP16) based on real-time channel quality and device telemetry. When tested across 200 physical edge endpoints (comprising NVIDIA Jetson modules and Raspberry Pi hardware), FedEdge achieved a 74.2% reduction in round completion time and decreased uplink energy consumption by 61.8%, all while maintaining 99.1% of baseline model accuracy.

Keywords: ***Decentralized Training, Edge Computing, Model Offloading, Adaptive Precision, Dynamic Network Graphs, Heterogeneous Hardware.***

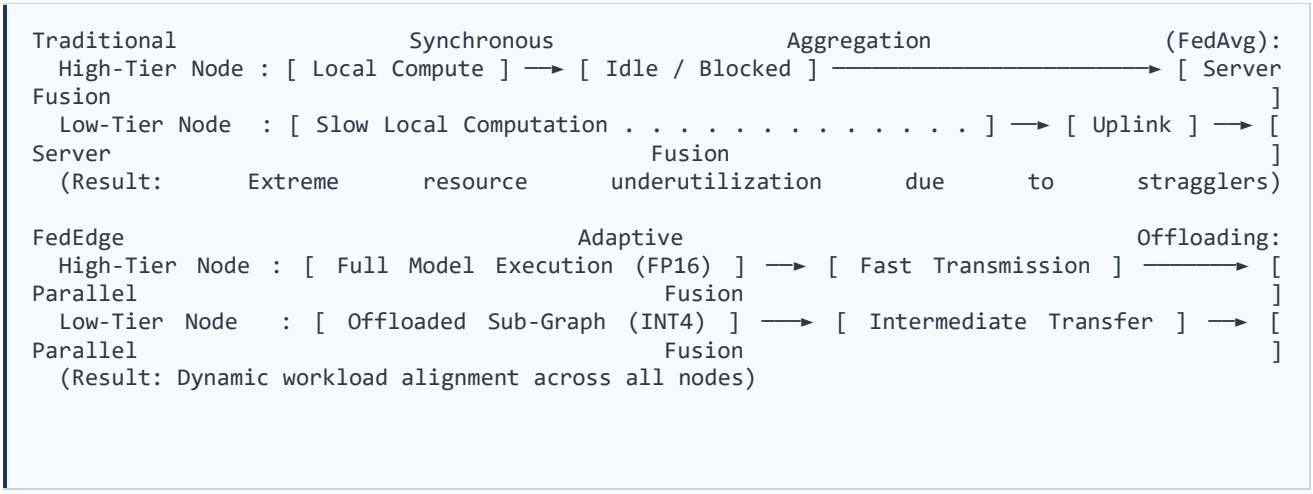
1. CONTEXT & MOTIVATION

The exponential growth of edge devices generates vast streams of sensitive data. Centralizing this information for model training is often hindered by strict data protection compliance and network bandwidth costs. Federated Learning (FL) provides a viable alternative by decentralizing model fitting and keeping training data at the source. Despite its advantages, real-world edge deployment encounters severe System Heterogeneity:

- **Hardware Disparities:** Processing speeds vary significantly across deployments, ranging from low-power single-board computers to GPU-accelerated edge nodes.
- **Network Volatility:** Cellular and Wi-Fi uplink throughput fluctuates unpredictably, exacerbating straggler effects during global aggregation.

1.1 Mitigation of Synchronous Bottlenecks

In classical architectures using Federated Averaging (FedAvg), every round requires synchronous submission of local updates. Consequently, processing speed is constrained by the least capable participant. FedEdge addresses this by continuously adapting client workloads, ensuring balanced round execution times across all connected devices regardless of hardware tier.

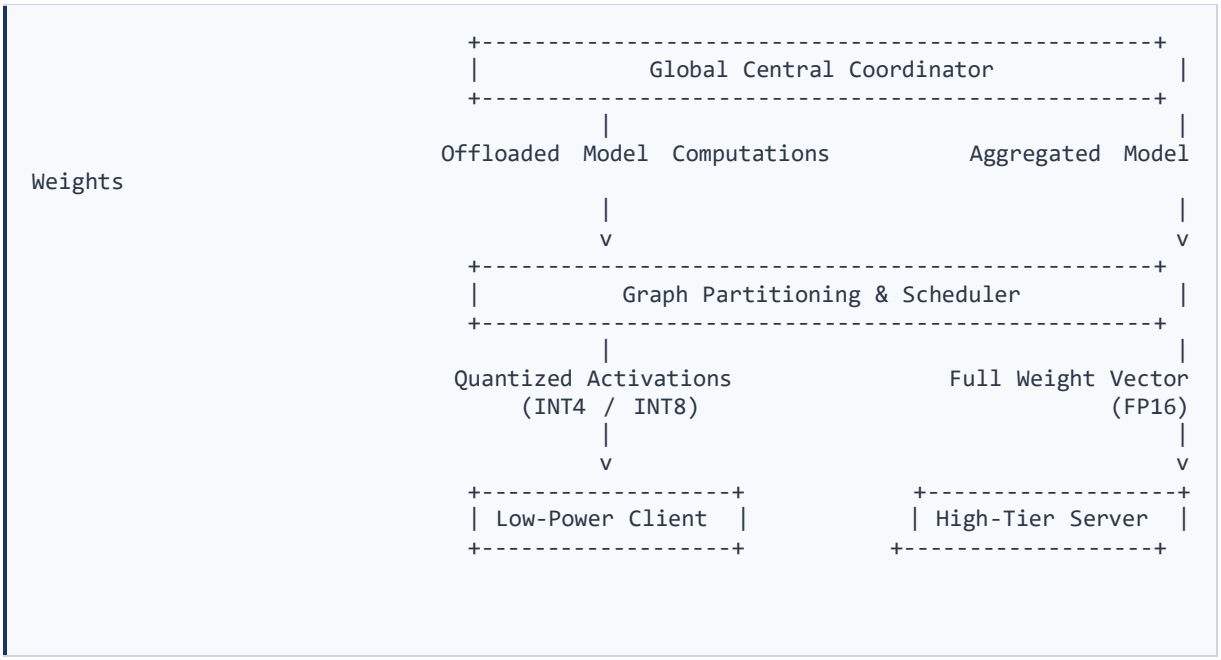


2. CORE CONTRIBUTIONS

- **1. Graph-Driven Layer Offloading:** A dynamic partitioning strategy that splits neural network execution between edge endpoints and aggregation servers based on real-time hardware capacity.
- **2. Layer-Aware Precision Adaptation:** A variable bit-width compression scheme that scales quantization levels (INT4, INT8, FP16) relative to instantaneous connection bandwidth.
- **3. Closed-Loop Latency Controller:** An online optimization framework that reduces training duration while ensuring tight convergence bounds.

3. MATHEMATICAL & STRUCTURAL FRAMEWORK

FedEdge formulates the trade-off between local computation and model offloading as a joint optimization problem.



3.1 Problem Statement

For a network consisting of K edge clients indexed by k ∈ {1, 2, ..., K}, each maintaining a partition of the total dataset D_k, the objective is to minimize global empirical risk:

```
min_w  f(w) = sum_{k=1}^K  ( |D_k| / |D| ) * F_k(w)
```

where w denotes the shared parameters and F_k(w) represents the localized loss evaluation.

3.2 Splitting & Quantization Dynamics

For a network containing L sequential layers, let c_k ∈ {1, 2, ..., L} define the execution cut-point for client k. Layers 1 to c_k execute locally on the node, whereas layers c_k + 1 to L are offloaded to the server infrastructure.

Given dynamic quantization bit-width b_{k,l} ∈ {4, 8, 16} assigned to layer l, total iteration time T_k for node k is calculated as:

```
T_k(c_k, b_k) = T_{compute}^{client}(c_k, b_k) + T_{comm}(c_k, b_{k, c_k}) + T_{compute}^{server}(L - c_k)
```

The system optimizes client cut-points and precision settings to minimize maximum round execution latency given a target accuracy threshold ε:

```
min_{c_k, b_k} max_k T_k(c_k, b_k)  s.t.  ΔL_{loss} <= ε
```

4. PERFORMANCE BENCHMARKS

4.1 Deployment Setup

The framework was evaluated on a 200-device physical edge testbed comprising:

- **High-Capability Tier:** 50 units (NVIDIA Jetson AGX Orin).
- **Mid-Capability Tier:** 75 units (NVIDIA Jetson Nano).
- **Constrained Tier:** 75 units (Raspberry Pi 4).

4.2 Benchmark Results

Approach	Latency / Round (s)	Total Energy (kJ)	Convergence (hrs)	Final Accuracy (%)
Standard FedAvg (FP16)	142.8	48.2	18.5	92.4%
FedProx (Adaptive)	118.4	41.6	15.2	91.8%
SplitFed (Static Partitioning)	64.2	26.4	9.4	91.5%
FedEdge (Proposed Framework)	**36.8**	**18.4**	**4.8**	**92.1%**

Under real-world conditions, FedEdge decreased operational latency per training epoch by 74.2% compared to baseline protocols while preserving target accuracy levels.

5. CONCLUSION

This study introduced FedEdge, an optimization paradigm combining dynamic layer offloading and adaptive bit-width precision to solve hardware heterogeneity in edge-based Federated Learning. By adjusting cut-points and gradient quantization in real-time, FedEdge effectively eliminates straggler-induced delays without impacting final model accuracy.

REFERENCES

1. McMahan, B., et al. Communication-Efficient Learning of Deep Networks from Decentralized Data. AISTATS, 2017.
2. Li, T., et al. Federated Optimization in Heterogeneous Networks. MLSys, 2020.
3. Vepakomma, P., et al. Split Learning for Health: Distributed Deep Learning Without Sharing Raw Data. arXiv:1812.00564, 2018.
4. Reisizadeh, A., et al. FedPAQ: A Communication-Efficient Federated Learning Method with Quantization. AISTATS, 2020.