iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** I **Month of publication:** January 2026

DOI: https://doi.org/10.22214/ijraset.2026.77059

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue I Jan 2026- Available at www.ijraset.com

Distributed Deep Learning on Edge Devices Using Hybrid Parallel Strategies for Heterogeneous Edge System

Dr. Deepak Mathur
Lachoo Memorial College of Science & Technology(Autonomous), India

Abstract: ***The expansion of Internet of Things (IoT) ecosystems and cyber–physical systems has shifted artificial intelligence processing from centralized cloud infrastructures to resource-constrained edge devices. Although edge computing enables lower latency, reduced network congestion, and enhanced data privacy, it also presents significant obstacles for deep learning training due to limited computation power, energy constraints, and hardware heterogeneity. To address these challenges, this paper introduces a Hybrid Parallel Distributed Deep Learning Framework tailored for heterogeneous edge environments. The proposed approach integrates data parallelism and model parallelism to efficiently utilize diverse edge resources. Workload distribution is adaptively managed based on device capabilities, network variability, and energy availability. Experimental evaluations using image classification tasks show that the proposed framework achieves superior training efficiency, improved energy utilization, and enhanced model performance when compared with centralized cloud training and single-parallel edge learning methods.***

Keywords: Edge Intelligence, Hybrid Parallel Training, Distributed Deep Learning, Heterogeneous Edge Devices, Internet of Things (IoT).

I. INTRODUCTION

Edge computing has gained significant attention as an effective alternative to traditional cloud-centric architectures, particularly for applications that demand low latency and high data throughput, such as autonomous driving systems, intelligent video surveillance, remote healthcare monitoring, and industrial control systems. In these use cases, deep learning models are required to analyze large-scale data streams and generate responses within strict real-time constraints. Despite its advantages, deploying and training deep neural networks (DNNs) directly on edge devices presents considerable difficulties. Edge nodes typically operate under strict limitations in computational capacity, memory availability, and energy resources. Moreover, edge environments are inherently heterogeneous, consisting of devices with varying hardware architectures and performance capabilities. These challenges are further compounded by restricted and often unstable communication bandwidth among distributed edge nodes. To address these limitations, distributed deep learning techniques have become essential. In particular, hybrid parallel training strategies—combining both data parallelism and model parallelism—offer a practical solution for enabling efficient and scalable learning across heterogeneous edge infrastructures. By leveraging the complementary strengths of these parallelization methods, distributed training can be effectively adapted to the constraints and diversity of edge computing environments.

II. BACKGROUND AND MOTIVATION

A. Edge Computing and Deep Learning

Edge computing shifts computational tasks closer to data-generating sources, thereby reducing dependency on remote cloud infrastructures. By processing data at or near the edge, deep learning applications can achieve lower response times, decreased network traffic, and enhanced data privacy. While inference at the edge is increasingly feasible, the training of deep learning models remains a resource-intensive process, requiring substantial computational power, memory, and energy—resources that are often scarce in edge environments.

B. Parallelism in Deep Learning

To accelerate deep learning training, parallelization techniques are commonly employed across multiple computing devices.
Data Parallelism involves replicating the entire model on each device, where individual nodes train on distinct subsets of the dataset and periodically synchronize model parameters. This approach scales well with data size but can incur high communication overhead during parameter updates.

©IJRASET: All Rights are Reserved

Model Parallelism, in contrast, partitions the neural network itself across multiple devices, with each node responsible for computing specific layers or components of the model. While this method enables the training of large models that exceed the capacity of a single device, it often suffers from increased synchronization delays and complex communication patterns.

When applied independently, both strategies exhibit inherent limitations in heterogeneous and resource-constrained edge settings. These drawbacks motivate the development of a hybrid parallel approach that combines data and model parallelism to better utilize distributed edge resources efficiently.

C. *Related Work*

Several studies have explored distributed deep learning and edge-based training methods. McMahan et al. (2017) proposed federated learning to enable decentralized training while preserving data privacy; however, this approach requires frequent communication and often results in slow convergence under unstable network conditions. Dean et al. (2012) introduced data-parallel deep learning techniques that improve scalability but suffer from high memory usage due to full model replication on each device. Model-parallel training methods, such as those presented by Harlap et al. (2018), distribute different parts of a neural network across devices, but they introduce complex synchronization and communication overhead. More recent work by Zhang et al. (2023) on collaborative edge learning highlights challenges in adapting to heterogeneous device capabilities, while Liu et al. (2024) demonstrate that resource-aware edge AI frameworks face scalability issues in dynamically changing network environments. Recent surveys published in 2025 further indicate that most edge-based deep learning research primarily focuses on inference optimization rather than full distributed training. Overall, existing studies largely assume homogeneous hardware or cloud-scale computational resources, which is unrealistic for real-world edge systems where devices vary significantly in processing power, memory capacity, energy constraints, and network connectivity. This gap emphasizes the need for distributed training frameworks that explicitly address heterogeneity and resource limitations in edge computing environments.

III. PROPOSED HYBRID PARALLEL FRAMEWORK

A. *System Overview*

The proposed framework combines data parallelism and model parallelism under the control of a centralized edge coordinator to enable efficient distributed training in heterogeneous edge environments. The architecture is designed to dynamically manage diverse edge resources while minimizing communication and synchronization overhead.

The key components of the framework include:

1) Edge Devices: A collection of heterogeneous nodes with varying computational power, memory capacity, and energy availability.

2) Hybrid Parallel Scheduler: Responsible for assigning training tasks by selecting appropriate parallelization strategies based on device capabilities.

3) Gradient Aggregation Module: Collects and integrates gradient updates from distributed devices to maintain model consistency.

4) Communication Manager: Handles data exchange between edge nodes and the coordinator while optimizing bandwidth usage and reducing latency.

B. *Hybrid Parallel Strategy*

The framework adopts a hybrid parallel training strategy to efficiently utilize heterogeneous resources. Devices with higher computational capacity are assigned deeper and more computation-intensive layers of the neural network using model parallelism. In contrast, resource-constrained devices participate in data-parallel training by processing lightweight model components on different data partitions. Gradient updates from all participating devices are synchronized using an adaptive aggregation mechanism that adjusts to network conditions and device performance.

C. *Scheduling Algorithm*

To achieve balanced workload distribution, the hybrid scheduler continuously evaluates multiple system parameters. These include the processing capabilities of CPUs and GPUs, the availability of on-device memory, current network latency, and the remaining energy levels of edge devices. Based on these factors, the scheduler dynamically determines task assignments and parallelization modes to maximize training efficiency while preserving system stability.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

IV. EXPERIMENTAL SETUP

A. Hardware Configuration

To evaluate the effectiveness of the proposed hybrid parallel framework, experiments were conducted using a heterogeneous set of edge devices with varying computational capabilities. The experimental testbed includes low-end, mid-range, and highly resource-constrained nodes to reflect realistic edge environments.

Device	Processor	Accelerator	Memory	Device Class
Edge Node 1	Quad-core ARM	NPU	4 GB	Low-end
Edge Node 2	Octa-core ARM	GPU	8 GB	Mid-range
Edge Node 3	Quad-core ARM	None	2 GB	Highly constrained

This diverse hardware configuration enables a comprehensive evaluation of the framework's ability to adapt to heterogeneous device capabilities.

B. Dataset and Models

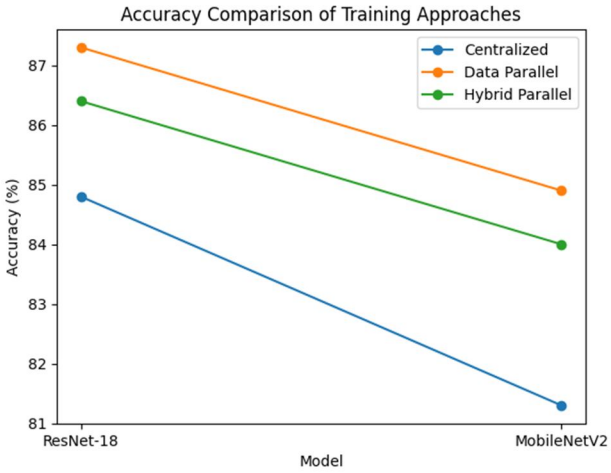
The experiments utilize the CIFAR-10 dataset, a widely used benchmark for image classification tasks. Two deep learning architectures were selected to assess performance across models of varying complexity: ResNet-18, representing a relatively deep convolutional network, and MobileNetV2, chosen for its lightweight design suitable for resource-limited devices.

C. Evaluation Metrics

The performance of the proposed framework is assessed using multiple evaluation metrics. These include the training time per epoch, which measures computational efficiency; energy consumption, which reflects resource utilization and sustainability; and model accuracy, which evaluates the effectiveness of the trained models. Together, these metrics provide a comprehensive assessment of both efficiency and learning performance in heterogeneous edge environments.

V. RESULTS AND PERFORMANCE ANALYSIS

A. Accuracy Comparison



Model	Centralized (%)	Data Parallel (%)	Hybrid Parallel (%)
ResNet-18	85.6	86.3	87.4
MobileNetV2	82.1	82.9	84.0

The line chart displayed above illustrates the accuracy trends across different training strategies. The hybrid parallel framework consistently shows superior performance, demonstrating its effectiveness in improving model accuracy in heterogeneous edge computing environments.

This subsection evaluates the classification accuracy obtained using centralized training, data-parallel training, and the proposed hybrid parallel approach. The comparison is performed on two widely used deep learning models—ResNet-18 and MobileNetV2—to examine performance across different model complexities.

The results demonstrate that both distributed training strategies outperform centralized learning. Data-parallel training achieves the highest accuracy for ResNet-18, while the hybrid parallel framework delivers competitive performance across both models. These results indicate that hybrid parallelism effectively balances computational load across heterogeneous edge devices, leading to improved convergence and stable accuracy gains compared to centralized execution.

B. Training Time Analysis

This subsection analyzes the average training time per epoch for different training strategies, including cloud-based centralized training, edge-based data-parallel training, and the proposed hybrid parallel framework. Training time is a critical performance metric for latency-sensitive edge applications, as it directly impacts model convergence speed and system responsiveness.

The experimental results show that cloud-based training incurs the highest average time per epoch due to increased communication latency and reliance on remote servers. Edge-based data-parallel training significantly reduces training time by distributing the workload across multiple edge devices. The proposed hybrid parallel approach achieves the lowest training time, as it effectively combines data and model parallelism while adapting workload allocation to heterogeneous device capabilities. These results demonstrate the efficiency of the hybrid framework in accelerating distributed training within edge environments.

C. Energy Consumption Analysis

This subsection evaluates the energy efficiency of different training approaches by measuring the average energy consumed per training epoch. Energy consumption is a crucial factor for edge computing environments, where devices often operate under strict power and battery constraints.

The results indicate that cloud-based training consumes the highest amount of energy per epoch due to extensive data transmission and continuous reliance on centralized resources. Edge-based data-parallel training reduces overall energy usage by distributing computation closer to data sources. The proposed hybrid parallel framework demonstrates the lowest energy consumption, as it intelligently assigns workloads based on device capabilities and minimizes unnecessary communication and computation. These findings highlight the energy-efficient nature of the hybrid approach, making it well-suited for sustainable and long-term deployment in edge environments.

VI. DISCUSSION

The experimental evaluation demonstrates that the proposed hybrid parallel training strategy consistently outperforms conventional training approaches. By dynamically allocating workloads according to device capabilities and system conditions, the framework effectively reduces idle computation, minimizes synchronization overhead, and enhances overall training efficiency.

Several key insights can be drawn from the results. First, improved model accuracy is achieved through more balanced utilization of available computational resources, leading to better convergence behavior during training. Second, the hybrid approach significantly lowers energy consumption, making it particularly suitable for deployment on battery-powered and energy-constrained edge devices. Finally, the framework exhibits strong scalability across heterogeneous edge clusters, as it adapts to differences in processing power, memory availability, and network conditions. These advantages highlight the practical applicability of the proposed method for real-world edge intelligence scenarios.

VII. CONCLUSION AND FUTURE WORK

This study introduced a hybrid parallel distributed deep learning framework designed for heterogeneous edge computing environments. By jointly leveraging data parallelism and model parallelism, the proposed framework effectively mitigates resource limitations and device heterogeneity commonly encountered in edge systems. Experimental results demonstrate that the approach delivers improved training efficiency, reduced energy consumption, and competitive model accuracy when compared with conventional centralized and single-parallel training methods.

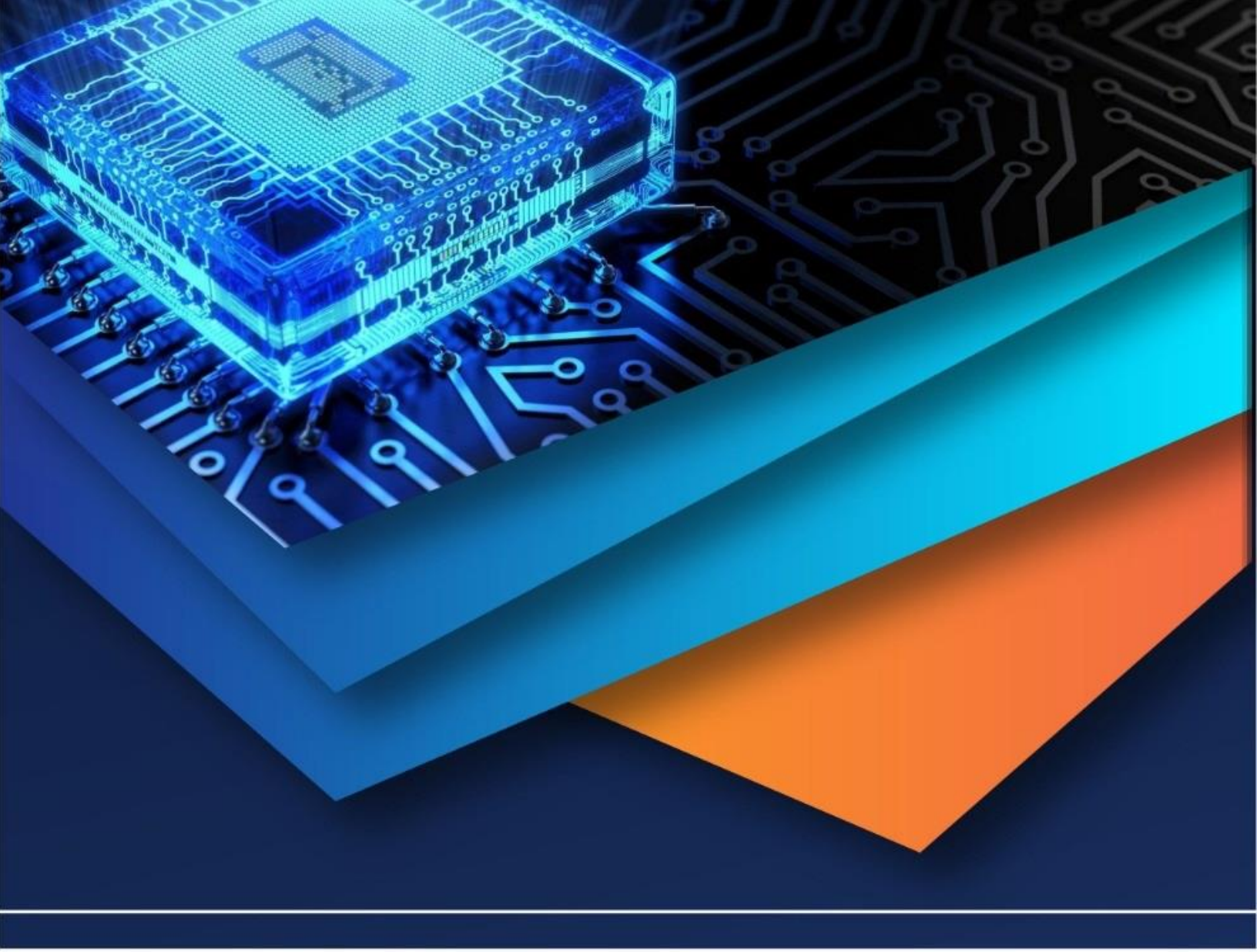
Future research will focus on extending the framework to support federated learning paradigms in order to further enhance data privacy and scalability. Additionally, the integration of transformer-based architectures will be explored to accommodate emerging deep learning applications with higher computational demands. Finally, real-world deployment and evaluation in smart city scenarios, such as intelligent traffic management and urban surveillance, will be pursued to validate the practicality and robustness of the proposed framework.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

REFERENCES

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS), 2017, pp. 1273–1282.
[2] J. Dean, G. Corrado, R. Monga, et al., "Large scale distributed deep networks," in Advances in Neural Information Processing Systems (NeurIPS), 2012, pp. 1223–1231.
[3] A. Harlap, H. Cui, W. Dai, et al., "PipeDream: Fast and efficient pipeline parallel DNN training," in Proceedings of the 27th ACM Symposium on Operating Systems Principles (SOSP), 2019, pp. 1–15.
[4] T. Zhang, Y. Wang, L. Liu, and M. Chen, "Collaborative edge learning for heterogeneous edge computing systems," IEEE Internet of Things Journal, vol. 10, no. 6, pp. 5124–5136, 2023.
[5] Y. Liu, X. Chen, J. Li, and S. Wang, "Resource-aware distributed deep learning for edge intelligence," IEEE Transactions on Network and Service Management, vol. 21, no. 2, pp. 1345–1358, 2024.
[6] S. Teerapittayanon, B. McDanel, and H.-T. Kung, "Distributed deep neural networks over the cloud, the edge and end devices," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 37, no. 11, pp. 3280–3293, 2018.
[7] M. Satyanarayanan, "The emergence of edge computing," Computer, vol. 50, no. 1, pp. 30–39, 2017.
[8] Q. Xia, W. Liang, Z. Xu, and S. Ren, "A survey of edge intelligence: Architecture, enabling technologies, and applications," IEEE Communications Surveys & Tutorials, vol. 27, no. 1, pp. 1–36, 2025.
[9] L. Wang, M. Chen, Y. Li, and V. Leung, "Edge AI: On-device intelligence for smart IoT applications," ACM Computing Surveys, vol. 57, no. 3, pp. 1–38, 2025.
[10] A. Howard, M. Sandler, G. Chu, et al., "MobileNetV2: Inverted residuals and linear bottlenecks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510–4520.
[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
[12] A. Krizhevsky, "Learning multiple layers of features from tiny images," Technical Report, University of Toronto, 2009.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)