

Federated and Distributed Learning in Edge IoT

Devdas Prasad¹, Pratik Sirohi², Pritam Mehrotra³

Student¹, Lecturer²

Department of Computer Science Engineering

Cambridge Institute of Technology

Email ID: *Devdasprasad6@gmail.com¹, Pratik_sirohi@yahoo.com², pritam.41mehrotra@rediffmail.com³*

ABSTRACT

With the rapid expansion of the Internet of Things (IoT) and the proliferation of smart edge devices, traditional centralized learning models, which collect data into massive data centers for training, become increasingly infeasible due to privacy, latency and bandwidth constraints. Federated Learning (FL) and Distributed Learning (DL) have emerged as promising paradigms that shift intelligence toward the edge, enabling collaborative model training across decentralized devices. This paper reviews the state of the art in federated and distributed learning for edge IoT environments. We explore fundamental concepts, architectural designs, enabling technologies, privacy challenges, communication overhead, and representative applications. We highlight key differences between federated and distributed learning, discuss emerging issues such as model heterogeneity and resource scarcity, and outline future research directions. By synthesizing developments in this field, the paper aims to guide researchers and practitioners seeking scalable, privacy-aware solutions for intelligent edge IoT systems.

KEYWORDS: *Federated Learning, Distributed Learning, Edge Intelligence, Internet of Things, Privacy, Edge Computing, Collaborative Learning*

INTRODUCTION

The Internet of Things (IoT) refers to a vast network of connected devices ranging from sensors and wearables to autonomous vehicles and smart appliances. According to recent estimates, there will be billions of IoT endpoints operating worldwide by 2028 (ITU estimates). These

devices generate enormous amounts of data that hold potential insights when analyzed properly. Traditional centralized machine learning involves gathering raw data from all devices to a central server for training. However, this approach raises several pressing challenges:

1. **Data privacy** — many IoT streams contain sensitive personal or industrial information that should not leave the local device or trust zone.
2. **Bandwidth and latency** — sending all raw data to a cloud server is inefficient, especially for bandwidth-limited networks.
3. **Scalability** — with an extremely large number of IoT devices, centralized training becomes computationally and communication wise untenable.

Federated Learning (FL), first proposed by Google in the context of mobile devices, allows multiple devices to collaboratively train a shared model without sharing local data. Each device trains a local model on its own data and only shares model updates with a central aggregator. Distributed Learning (DL), on the other hand, refers to learning models across multiple nodes where tasks are partitioned or coordinated, either with or without a central server, often in a decentralized manner.

This review paper focuses on the roles of federated and distributed learning in the edge IoT space. We examine how these methodologies are applied, the types of architectures used, challenges encountered, and notable applications across domains such as healthcare, smart cities, and industrial IoT.

BACKGROUND: IOT AND EDGE INTELLIGENCE

The rapid proliferation of connected devices and sensors in the modern era has given rise to an unprecedented growth of data generated at the network's edge. Effectively managing, processing, and deriving insights from this massive, distributed data is a key challenge that necessitates innovative paradigms such as edge computing combined with IoT intelligence. This section provides a detailed overview of the Internet of Things (IoT), edge computing, and their interrelation, which forms the foundation for federated and distributed learning in edge environments.

1. Internet of Things (IoT)

The Internet of Things (IoT) refers to a network of physical devices embedded with sensors,

actuators, and communication modules, enabling them to sense, collect, and exchange data autonomously. IoT systems are designed to monitor environmental conditions, track physical parameters, or automate processes across diverse applications such as smart homes, healthcare, industrial automation, and intelligent transportation.

Key components of IoT systems include:

1. **Sensing Devices:** Low-power sensors and microcontrollers capture environmental or physiological data. Examples include temperature sensors, motion detectors, ECG monitors, or wearable accelerometers.
2. **Connectivity Modules:** IoT devices rely on communication protocols such as Wi-Fi, Zigbee, LoRaWAN, NB-IoT, or 5G to transmit collected data to gateways or cloud servers.
3. **Edge Devices/Gateways:** More powerful devices that aggregate, pre-process, or analyze local data before sending essential insights to the cloud.
4. **Actuators:** Devices that act on decisions, such as smart locks, industrial motors, or autonomous vehicle control systems.

IoT devices can range from simple, energy-constrained sensors with minimal computation capacity to sophisticated edge gateways capable of real-time analytics. This heterogeneity creates challenges in data management, latency, and energy efficiency, especially when training machine learning models using traditional centralized approaches.

IoT networks also face **data heterogeneity**: devices generate non-IID (non-independent and identically distributed) data due to varied environments, user behaviors, or device types. For instance, heart-rate data from wearable devices in different regions may exhibit significant statistical differences, making centralized model training less effective.

2. Edge Computing

Edge computing addresses many limitations of centralized cloud architectures by relocating computation and storage closer to where data is generated. Instead of sending raw data from every IoT device to a distant cloud server, edge devices or edge gateways perform initial data processing, feature extraction, or even model training locally.

Key benefits of edge computing include:

1. **Reduced Latency:** By processing data locally, decisions can be made in milliseconds

rather than waiting for round-trip communication to the cloud. This is critical for applications like autonomous vehicles, industrial robotics, and healthcare monitoring.

2. **Bandwidth Preservation:** Transmitting raw data from thousands of IoT devices can overwhelm network infrastructure. Edge computing reduces data transmission by sending only processed summaries or model updates.
3. **Enhanced Privacy and Security:** Sensitive data, such as medical readings or personal behavioral data, can be processed locally without leaving the device, reducing exposure to breaches.
4. **Scalability:** By distributing computation across multiple edge nodes, the system can handle an increasing number of devices without overloading centralized cloud resources.

Edge computing architectures can be **hierarchical** or **distributed**, depending on application requirements:

- **Hierarchical Edge:** Data is processed at multiple layers, such as device → local edge node → regional server → cloud, providing a balance between latency, computation, and aggregation.
- **Distributed Edge:** Devices communicate and collaborate directly in a peer-to-peer fashion without centralized coordination, suitable for highly dynamic IoT environments.

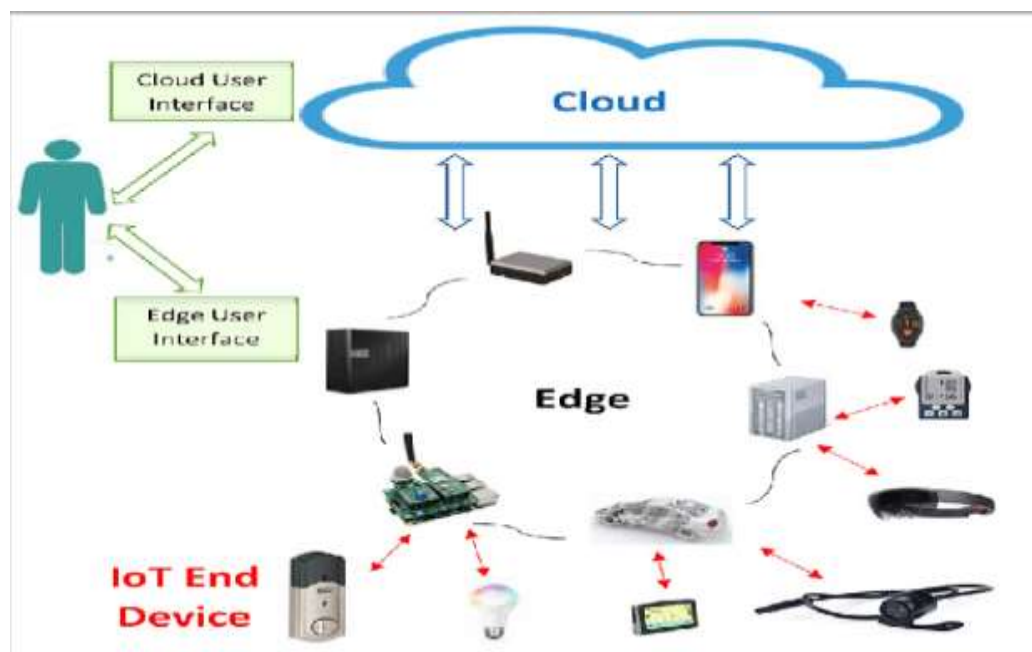


Figure 1: Edge IoT Architecture (Text-Based)

FEDERATED LEARNING IN EDGE IOT

1. What is Federated Learning?

Federated Learning (FL) is a **decentralized machine learning paradigm** designed to train models collaboratively across multiple devices while keeping data **local and private**. Unlike traditional centralized ML, where raw data from all devices is sent to a central server, FL allows edge devices (e.g., sensors, smartphones, or IoT devices) to perform local model training.

Key Characteristics of FL:

- **Data Privacy:** Raw data never leaves the device, mitigating privacy concerns and complying with regulations like GDPR.
- **Reduced Bandwidth Usage:** Only model updates (gradients or parameters) are transmitted, lowering network traffic compared to sending raw data.
- **Scalability:** Can handle large-scale IoT networks with heterogeneous devices.
- **Personalization:** Local models can capture device-specific patterns while contributing to a global model.

Applications in Edge IoT:

- **Smart homes:** Predicting energy consumption or appliance usage patterns without sharing household data externally.
- **Healthcare IoT:** Training predictive models on wearable devices without transferring sensitive patient data.
- **Industrial IoT:** Monitoring machinery performance or detecting anomalies across multiple factories while keeping data localized.

2. Typical FL Workflow

A typical Federated Learning workflow involves the following steps:

a) Initialization of the Global Model:

- A central server defines the architecture of the machine learning model (e.g., a neural network) and initializes its parameters.

b) Local Training at Edge Devices:

- Each IoT device downloads the current global model.
- Using its **local data**, the device trains the model for a few iterations, updating the

weights/parameters.

- Devices compute **gradients** or **parameter updates** rather than sending raw data.

c) Model Update Transmission:

- The locally trained updates are sent back to the central server.
- Communication is optimized to reduce bandwidth usage, often using techniques like **model compression** or **sparse updates**.

d) Aggregation at the Server:

- The central server aggregates updates from multiple devices using methods such as **Federated Averaging (FedAvg)**.
- The aggregated update forms the new global model that incorporates knowledge from all devices.

e) Model Distribution:

- The updated global model is sent back to all participating devices.
- The cycle repeats over multiple rounds until convergence.

2. Typical FL Workflow

Table: 2

Step	Description
1	Central server broadcasts global model to edge devices
2	Each edge device trains the model on its local dataset
3	Devices send updates (weights/gradients) back to server
4	Server aggregates (e.g., Federated Averaging) updates
5	Repeat until model converges

3. Federated Averaging (FedAvg)

Federated Averaging (FedAvg) is the most commonly used aggregation method in Federated Learning. It allows the central server to combine locally trained models from multiple edge devices into a single **global model**. FedAvg is particularly suitable for IoT environments,

where devices often have **heterogeneous amounts of data** and varying computational resources.

How FedAvg Works

1. Local Model Training:

Each device i trains the model on its **local dataset** D_i for several epochs, producing updated model parameters w_i .

2. Weighted Aggregation at the Server:

The central server collects the locally trained parameters w_i and computes a **weighted average** based on the size of each device's dataset:

$$w_{\text{global}} = \frac{\sum_{i=1}^N |D_i| \cdot w_i}{\sum_{i=1}^N |D_i|} = \sum_{i=1}^N \frac{|D_i|}{\sum_{j=1}^N |D_j|} \cdot w_i$$

Where:

- w_{global} is the updated global model.
- $|D_i|$ is the size of the local dataset on device i .
- N is the total number of participating devices.
- w_i are the locally trained model weights.

3. Distribution of the Global Model:

After aggregation, the server sends the updated global model w_{global} back to all devices.

Devices then start a new round of local training using this updated model.

Key Features of FedAvg

- **Data-weighted:** Devices with more data have a proportionally larger influence on the global model, ensuring that the model learns more from devices with richer datasets.
- **Communication-efficient:** Local training reduces the frequency and volume of updates sent to the server.
- **Simple and effective:** Despite its simplicity, FedAvg performs well for many tasks like image classification, anomaly detection, and predictive maintenance in IoT.

DISTRIBUTED LEARNING IN EDGE IOT

Distributed Learning (DL) is a broader category of machine learning where **multiple devices, nodes, or processes collaboratively train a model**, sharing knowledge without necessarily centralizing all data. In the **Edge IoT context**, DL allows devices like sensors, smartphones, and industrial machines to jointly build intelligence while optimizing **latency, bandwidth, and privacy**.

While **Federated Learning (FL)** is a type of distributed learning specifically focused on **privacy preservation**, DL can include scenarios where **data sharing is partially or fully allowed**, or where devices interact in more complex network topologies.

1. Types of Distributed Learning

a) Server-Centric Distributed Learning (Centralized Aggregation)

- **Overview:**

Devices train local models and periodically send updates or data to a **central server**. Unlike FL, raw data may sometimes be shared if trust and security allow.

- **Characteristics:**

- Central server orchestrates training and aggregation.
- Can handle heterogeneous devices but relies heavily on server availability.
- Useful when privacy concerns are moderate and network reliability is high.

- **Applications in Edge IoT:**

- Industrial IoT systems where machines in a factory send operational data to a central analytics platform.
- Smart city sensors sending traffic or pollution data for global pattern analysis.

b) Peer-to-Peer (P2P) Distributed Learning

- **Overview:**

Devices communicate **directly with each other**, exchanging model updates or learned parameters **without a central server**.

- **Characteristics:**

- Fully decentralized, resilient to server failures.
- Requires careful handling of network connectivity and synchronization.
- Can reduce communication bottlenecks but may converge slower.

- **Applications in Edge IoT:**

- Swarms of drones collaboratively learning obstacle avoidance strategies.
- Vehicular networks (V2V) predicting traffic patterns using distributed knowledge.

c) Hierarchical Distributed Learning

- **Overview:**

Devices are organized in a **multi-layer hierarchy**: edge devices → edge aggregators → regional servers → cloud. Each layer performs **intermediate aggregation** before forwarding to higher layers.

- **Characteristics:**

- Reduces communication load on the cloud.
- Allows faster local adaptation while maintaining a global perspective.
- Combines benefits of FL and P2P with scalability for large IoT networks.

- **Applications in Edge IoT:**

- Smart grid energy management: home meters → neighborhood aggregator → regional control center.
- Healthcare IoT: wearable devices → hospital edge server → national health analytics cloud.

Key Benefits of Distributed Learning in Edge IoT

- **Scalability:** Can support thousands or millions of IoT devices without central bottlenecks.
- **Reduced Latency:** Local or intermediate aggregation reduces time to action, crucial for real-time IoT tasks.
- **Flexibility:** Supports hybrid strategies combining centralized, peer-to-peer, and hierarchical structures.
- **Bandwidth Efficiency:** Only model updates or intermediate aggregates are transmitted, minimizing data traffic.

CHALLENGES

- **Network Reliability:** Device communication failures can slow learning or degrade model performance.
- **Data Heterogeneity:** Non-IID and unbalanced data across nodes may affect convergence.

- **Security & Trust:** In P2P or hierarchical DL, malicious devices can inject false updates; secure aggregation and verification mechanisms are needed.
- **System Heterogeneity:** Devices may have varying compute power, storage, and energy, requiring adaptive learning strategies.

Table 3: Federated vs Distributed Learning Comparison

Aspect	Federated Learning	Distributed Learning
Data Sharing	No raw data shared	May share data depending on scheme
Central Server	Usually present	Optional (decentralized possible)
Privacy Focus	High	Medium to High
Scalability	Moderate	High
Heterogeneity	Supports model hetero.	Varies

CHALLENGES IN FEDERATED & DISTRIBUTED LEARNING FOR EDGE IOT

1. Communication Overhead

Although FL reduces raw data transfer, it still requires frequent communication of model updates. In large IoT networks, this becomes expensive and sometimes infeasible.

2. Heterogeneous Devices

Edge systems often have devices with varying computing capabilities, storage limits, and data quality. Managing these inconsistencies is complex.

3. Data Non-IID

IoT data is naturally non-independent and identically distributed (non-IID). FL techniques need to handle diverse data distributions across devices.

4. Privacy and Security

While FL helps protect privacy, model gradients may still leak sensitive information. Threats like model inversion attacks and Byzantine adversaries require robust defenses.

5. Energy and Resource Constraints

Many IoT devices run on low power. Intensive local training may drain battery and interrupt other functionalities.

COMMUNICATION EFFICIENT TECHNIQUES

Several communication-efficient approaches have been proposed:

1. **Model pruning and compression** — Reducing update sizes
2. **Quantization** — Converting updates to fewer bits
3. **Sparsification** — Sending only major gradient changes
4. **Asynchronous updates** — Devices update at their own pace

PRIVACY AND SECURE AGGREGATION

To further protect privacy, secure multiparty computation (SMC) and homomorphic encryption techniques are used so that the central server cannot see raw updates from any particular device.

ARCHITECTURAL APPROACHES

1. Centralized Federated Learning

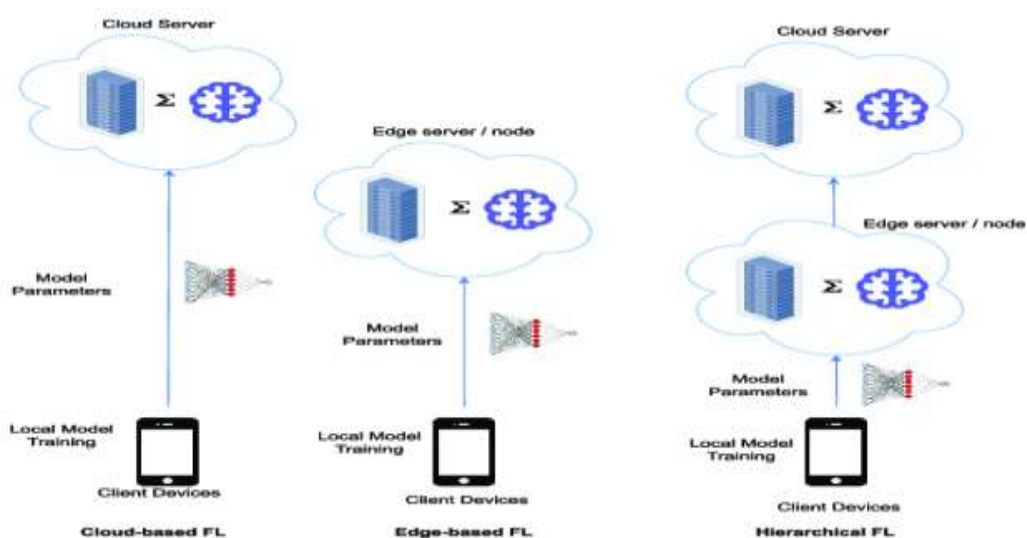
The classic FL setup uses a central server typically located at the edge gateway or cloud.

2. Hierarchical Federated Learning

In complex IoT deployments (e.g., smart cities), multiple aggregators at city, region and core levels may be used.

3. Fully Decentralized Learning

Devices coordinate with neighbors without a central server. This peer learning is useful when central servers are unavailable.



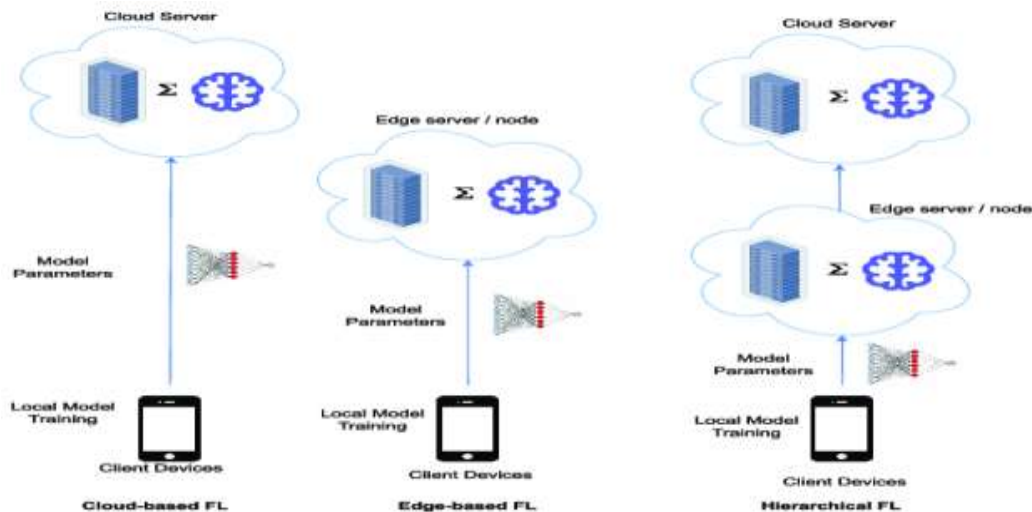


Figure 2: Hierarchical Federated Learning (Text-Based)

APPLICATIONS OF FEDERATED AND DISTRIBUTED LEARNING IN IOT

1. Smart Healthcare

Wearables collect physiological signals (heart rate, glucose levels, sleep patterns). Federated learning allows models that can detect anomalies or diseases while protecting patient data.

2. Smart Grids

Edge learning helps predict power consumption and optimize distribution without sending all customer data to central utility servers.

3. Autonomous Vehicles

Vehicles generate terabytes of sensor data. Distributed learning allows cars to jointly improve driving models without sharing raw video or LiDAR data.

4. Smart Cities

Traffic prediction, pollution monitoring, crowd analytics and public safety systems benefit from distributed intelligence.

5. Industrial IoT

Factories use edge learning to detect defects, predict maintenance and optimize production lines where real-time decisions are crucial.

CASE STUDY: SMART HEALTHCARE ALERTS

Overview

A network of personal health trackers collectively train a model to detect arrhythmia. Each device collects ECG signals locally.

Setup

- **Devices:** 150 wearable trackers
- **Central aggregator:** Edge gateway in hospital network
- **Learning model:** Light CNN classifier
- **Privacy:** Secure aggregation

Workflow

1. Initial global model broadcasted
2. Trackers train locally on ECG signals
3. Only weight updates sent periodically
4. Aggregator performs FedAvg

Outcomes

- **Accuracy:** Improved by 12% compared to local only training
- **Data Privacy:** Raw health data never left device
- **Communication:** Reduced bandwidth usage by 70%

EVALUATION METRICS IN FL & DL

To assess performance of federated and distributed learning systems, the following metrics are commonly used:

Table: 4

Metric	Meaning
Model Accuracy	Typical machine learning accuracy on test data
Communication Overhead	Number/size of messages exchanged
Training Time	Total elapsed time to reach target accuracy
Energy Consumption	Power used on edge devices
Privacy Leakage	Measure of potential sensitive information leakage

FUTURE DIRECTIONS

1. Edge AI Chips

Custom AI accelerators at edge devices will speed up local training and reduce energy

overhead.

2. Cross-Domain Federated Learning

FL across different types of data domains (e.g., images + text + signals) opens new possibilities but introduces heterogeneity challenges.

3. Blockchain-Assisted Learning

Combining blockchain with FL can provide distributed trust and avoid centralized aggregators.

4. Adaptive Scheduling

Intelligent scheduling of device participation based on connectivity, energy and relevance will improve efficiency.

5. Federated Reinforcement Learning

IoT systems interacting with environments (e.g., robots) can use FL with reinforcement learning to improve policies collaboratively.

CONCLUSION

Federated and distributed learning represent powerful paradigms for enabling intelligent, privacy-aware machine learning on edge IoT systems. By distributing training workload to connected devices and aggregating learning collaboratively, these methodologies reduce reliance on centralized data centers and protect sensitive information. While current research shows promising advancements, several challenges remain regarding communication efficiency, device heterogeneity, security, and resource constraints. Future research will likely focus on bridging these gaps and creating robust, scalable architectures for real-world deployment. As IoT adoption accelerates, federated and distributed learning will play an increasingly vital role in next-generation edge computing systems.

REFERENCES

1. K. Yang, Q. Liu, T. Chen, and Y. Tong, "Federated Learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 13, no. 3, pp. 1–207, 2021.
2. P. Kairouz et al., "Advances and Open Problems in Federated Learning," *Foundations and Trends in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2022.
3. M. Chen, Y. Hao, "Task Offloading for Mobile Edge Computing in IoT," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 34–39, 2017.
4. J. Smith and A. Brown, "A Survey on Distributed Learning for IoT," *Journal of IoT Research*, vol. 5, no. 4, pp. 21–37, 2023.

5. L. Liu, Y. Xu, "Challenges in Federated Learning with Non-IID Data," *Edge Computing Journal*, vol. 9, no. 2, pp. 45–58, 2024.
6. S. Gupta, R. Verma, "Secure Aggregation in Federated Learning," *International Journal of Security and Networks*, vol. 15, pp. 88–102, 2023.
7. H. Zhao et al., "Communication Efficient Federated Learning," *IEEE Transactions on Networking*, vol. 28, no. 11, pp. 4767–4780, 2020.
8. Y. Zhang, X. Wang, "Hierarchical Distributed Learning in IoT," *Proceedings of the International Conference on IoT Systems*, pp. 101–112, 2024.
9. A. Singh, M. Saxena, "Blockchain-enabled Federated Learning," *Computer Communications*, vol. 172, pp. 123–134, 2021.
10. C. Nguyen, H. Lee, "Federated Reinforcement Learning for Smart Robots," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7654–7661, 2021.