IIT(ISM) Dhanbad

Federated Learning in IoT–Fog–Cloud Hierarchies Using Progressively Deepening Split Neural Network

11 November 2025

Department : Computer Science & Engineering

Guide : Prof. Prasanta K. Jana

Session : 2025–2026

Presented by :

Pranav Gupta 22JE0710

Saransh Shivhare 22JE0867

Ayush Shaw 22JE0221

Part A: B.Tech Project

Introduction : Traditional Federated Learning in IoT–Fog–Cloud Architecture

1. Overview

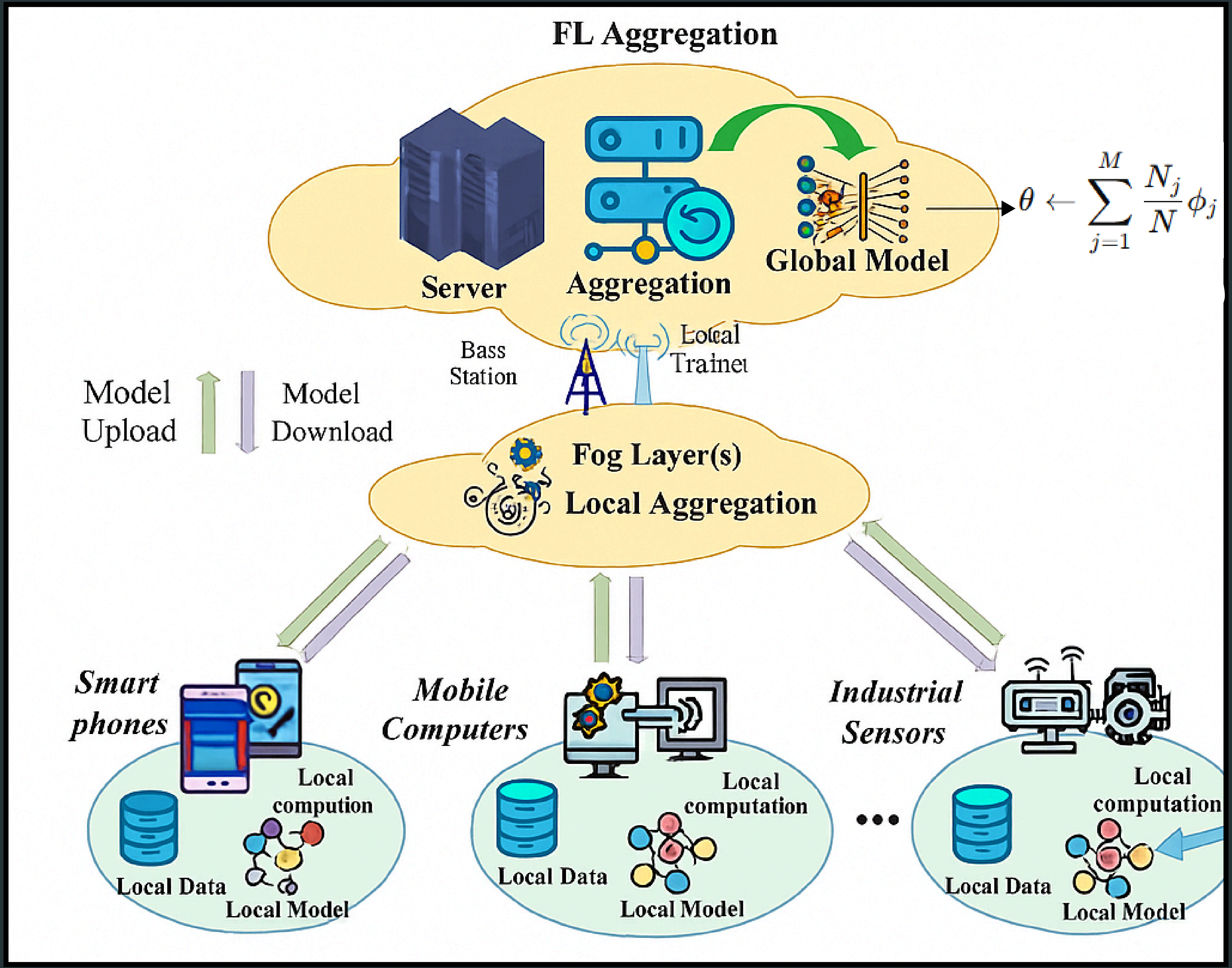
Federated Learning (FL) is a distributed machine learning paradigm that allows multiple IoT devices (clients) to collaboratively train a global model without sharing their raw data.

In the IoT–Fog–Cloud hierarchy, the FL process is extended to multiple tiers:

- IoT Layer (Clients): Perform local training on private datasets.
- Fog Layer: Aggregates updates from multiple IoT devices in its region.
- Cloud Layer: Performs global aggregation and optimization of the model.

This hierarchical setup helps distribute computation and reduce bandwidth usage compared to a fully cloud-centric system.

Introduction : Traditional Federated Learning in IoT–Fog–Cloud Architecture



Introduction : Traditional Federated Learning in IoT–Fog–Cloud Architecture

4. Limitations in Traditional FL–Fog–IoT Systems

Aspect	Limitation	Impact
Resource Constraints	IoT devices are weak (limited CPU, RAM, power)	Training heavy models locally is infeasible
Non-IID Data	Clients have dissimilar, skewed data	Poor global convergence and unstable accuracy
Synchronous Updates	All clients must finish training per round	Delays due to slow or offline devices
High Communication Overhead	Frequent cloud–fog–IoT synchronization	Increased latency and bandwidth usage
Flat Model Design	Fixed model architecture across tiers	Doesn’t match heterogeneous hardware or context
Context Loss	Aggregation averages out regional diversity	Global model fails to adapt locally

Motivation & Problem Statement

- Centralized cloud training causes **high latency** and **heavy bandwidth usage**.
- **Data privacy** concerns arise when sensitive device data is uploaded to cloud servers.
- IoT devices are **resource-constrained** — limited memory, energy, and computation power.
- **Heterogeneous devices** lead to non-uniform data distributions and inconsistent model performance.

Design a hierarchical learning framework where computation and aggregation occur closer to the edge (using fog nodes) leading to improved performance and privacy

Objectives & Scope

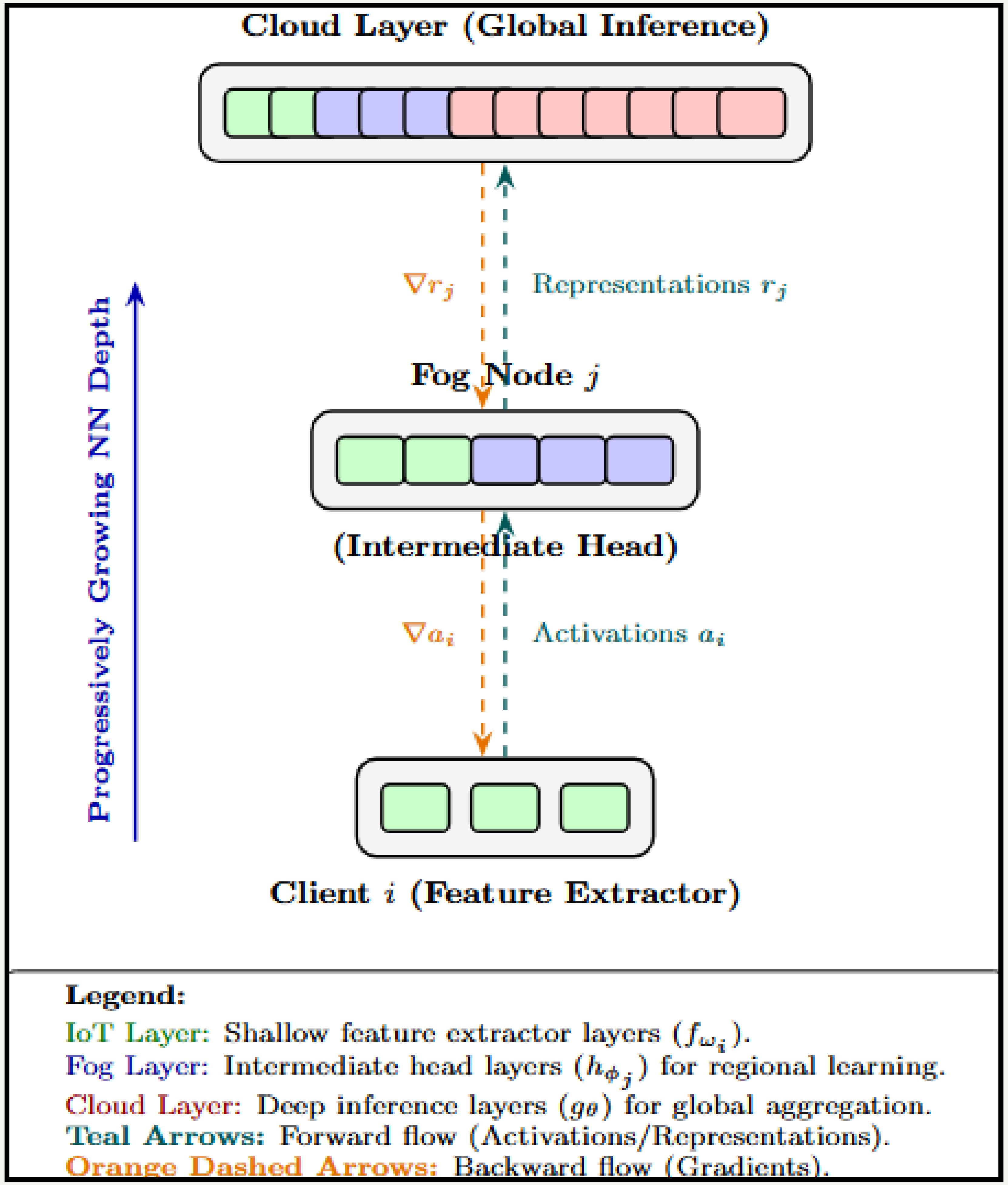
01 **Design hierarchical FL architecture**

02 **Preserve privacy**

03 **Reduce latency & bandwidth use**

04 **Progressive model construction**

System Architecture



System Architecture

The **global model** is constructed progressively as a composition of tier-specific subnetworks:

$$\mathcal{M} = g_\theta \circ h_\phi \circ f_{\omega_i}$$

where:

- f_{ω_i}: client-level base model (parameters ω_i)
- h_ϕ: fog-level regional model (parameters ϕ)
- g_θ: cloud-level global model (parameters θ)

Thus, a complete model for client c_i at fog node f_j is:

$$\mathcal{M}_{i,j}(x) = g_\theta(h_\phi(f_{\omega_i}(x)))$$

Each tier's submodel captures progressively broader context — local (f_{ω_i}), regional (h_ϕ),

Each tier has its own classification loss:

$$\mathcal{L}_{total} = \alpha\,\mathcal{L}_{client} + \beta\,\mathcal{L}_{fog} + \gamma\,\mathcal{L}_{cloud}$$

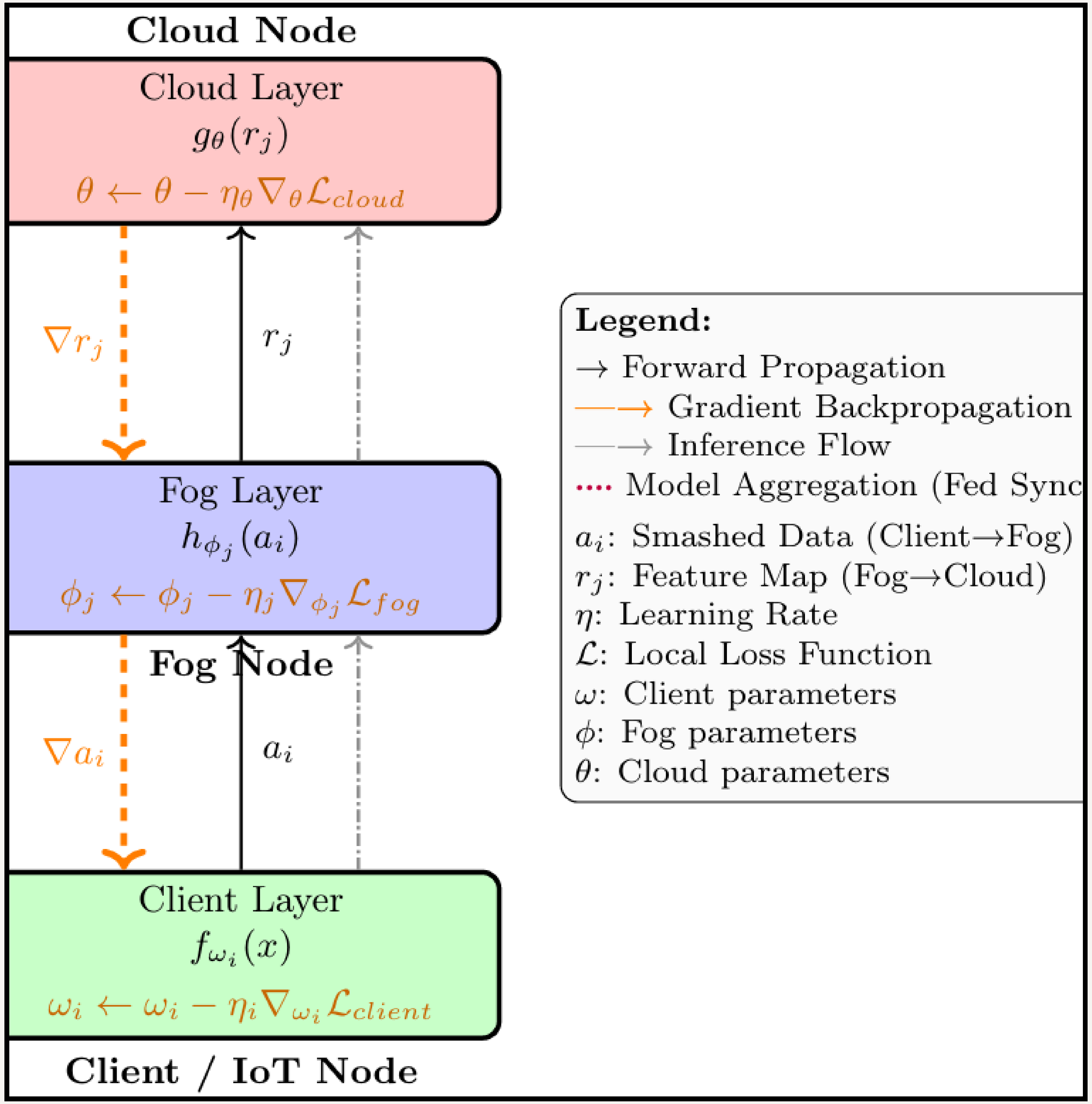
where:

- $\mathcal{L}_{client}, \mathcal{L}_{fog}, \mathcal{L}_{cloud}$ are cross-entropy losses from local, regional, and global heads respectively.
- Weights α, β, γ gradually shift toward cloud dominance as the network matures.

Implementation Details

Aspect	Suggestion
1.Frameworks	**PyTorch** + **Flower/FedML** for federated orchestration; **PySyft** for SplitNN simulation; **MQTT/gRPC** for client–fog communication.
2. Datasets	**CIFAR-10**, **Tiny ImageNet** (visual tasks), **Edge-IIoTset** (industrial IoT), **UCI-HAR** (sensor data).
3. Model Partitioning	**IoT:** initial convolution/feature layers → **Fog:** mid-level heads → **Cloud:** final inference layers.
4. Training Setup	**IoT:** forward pass only; **Fog:** forward + backward; **Cloud:** global aggregation every 10–15 epochs. **Optimizer:** Adam (LR = 1e−3).
5. Metrics	Accuracy, Communication Cost, Latency, Convergence Speed, Energy Consumption.
6. Baselines	FedAvg, SplitNN, Hierarchical FL.

Workflow



Cloud Node
Cloud Layer
$g_\theta(r_j)$
$\theta \leftarrow \theta - \eta_\theta \nabla_\theta \mathcal{L}_{cloud}$
∇r_j
r_j
Fog Layer
$h_{\phi_j}(a_i)$
$\phi_j \leftarrow \phi_j - \eta_j \nabla_{\phi_j} \mathcal{L}_{fog}$
Fog Node
∇a_i
a_i
Client Layer
$f_{\omega_i}(x)$
$\omega_i \leftarrow \omega_i - \eta_i \nabla_{\omega_i} \mathcal{L}_{client}$
Client / IoT Node
Legend:
Forward Propagation
Gradient Backpropagation
Inference Flow
Model Aggregation (Fed Sync
a_i: Smashed Data (Client→Fog)
r_j: Feature Map (Fog→Cloud)
η: Learning Rate
$\mathcal{L}$: Local Loss Function
ω: Client parameters
ϕ: Fog parameters
θ: Cloud parameters

Novelty and Contributions

- **Progressive Neural Network Growth:** The model expands hierarchically from IoT → Fog → Cloud, unlike traditional flat or parallel federated learning (FL) structures.
- **Hybrid Split-Federated Training:** Integrates the strengths of both SplitNN and Federated Learning (FL) through dynamic layer partitioning for optimized training.
- **Zero-Compute Client Inclusion:** Enables participation of devices with no local compute capability by utilizing smashed data, enhancing inclusivity in heterogeneous IoT environments.
- **Hierarchical Aggregation Mechanism:** Implements a multi-tier aggregation process (Client → Fog → Cloud) to minimize communication overhead while ensuring global synchronization.
- **Adaptive Resource Utilization:** Dynamically reallocates network layers and model updates based on available computational resources across IoT, Fog, and Cloud tiers.

Advantages

1. Compute Efficiency	Clients offload heavy computation to Fog, enabling ultra-light IoT participation.
2. Communication Reduction	Only activations and gradients (not full model weights) are exchanged, saving bandwidth.
3. Low Latency	Fog nodes act as intermediate computation units, minimizing end-to-end delay.
4. Scalability	The hierarchical setup naturally supports new IoT or fog nodes without retraining the entire system.
5. Privacy Preservation	Raw data never leaves the IoT devices.
6. Progressive Adaptability	The model depth adapts dynamically to available computational tiers.
7. Fault Tolerance	If a fog or client node fails, learning continues at other levels.

Conclusion

01 Summary

Our multi-tier federated learning approach leverages IoT, fog, and cloud layers to improve scalability and reduce latency. By progressively building the neural network (adding layers at fog/cloud), we allow IoT devices to participate with lightweight models while still achieving rich global models. Privacy is preserved since raw data never leaves devices. The fog-layer aggregation relieves the cloud of every single update, cutting communication overhead.

02 Contributions

We propose an adaptive, hierarchical FL framework tailored for IoT–Fog–Cloud environments, addressing key challenges of device heterogeneity and network constraints.

Part B: Internship Work

Internship Overview

amazon

Name	Company	Team	Project(s)	Tech Stack
Saransh Shivhare	Amazon	CMT Systems (Map Tech)	1. Enhanced Bulk ASIN Lookup system to support cross-marketplace queries and improve catalog accuracy. 2. Developed data validation and integrity framework ensuring reliable domain-competitor mappings across platforms. 3. Integrated Amazon Lens Visual-Search Sidebar into Manual Mapping Tool	Java, TypeScript, Spring, Coral Service Framework, React, AWS (S3, DynamoDB, SQS, SNS, Athena)

Internship Overview

zepto

Name	Company	Team	Project(s)	Tech Stack
Pranav Gupta	Zepto	Data Science (Growth and Pricing)	1. Competitor Price Analysis of E-commerce Market using Causal Inference and Bayesian Models 2. Inventory prediction optimisation for dynamic pricing using Survival Modelling and Statistical Analysis	Python Deep Learning and Bayesian Libraries (numpy, pandas, pgmpy, EconML), Apache Spark, Databricks

Internship Overview

accenture

Name	Company	Team	Project(s)	Tech Stack
Ayush Shaw	Accenture	Data & AI (Project iDEA – IOCL)	1. AIML-based Analytics Asset Development Focused on building analytical models and dashboards to derive actionable insights on market and inventory dynamics for IOCL. 2. Dry-Out Analysis and Root-Cause Identification Performed data-driven investigation of dry-out patterns across Indian regions, identifying high-loss trends and optimization opportunities.	Python (pandas, numpy, matplotlib), Power BI, Figma, Excel, Accenture proprietary tools

IIT(ISM) Dhanbad

Thank You

11 November 2025