Technical Report

Intelligent IIoT Service Deployment using MEC and Federated Learning in Drone-Assisted Smart Agriculture

Submitted By

Prithiv Alagirisamy, CB.SC.U4CSE23260

Danvanth, CB.SC.U4CSE23241

Vivin Rakul, CB.SC.U4CSE23250

Nirvesh Singh, CB.SC.U4CSE23265

in partial fulfilment of the requirements for the course of

23CSE362 - EDGE COMPUTING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AMRITA SCHOOL OF COMPUTING

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE - 641 112

List of Tables

List of Figures

List of Abbreviations

ANFIS Adaptive Neuro-Fuzzy Inference System

API Application Programming Interface

CNN Convolutional Neural Network

FL Federated Learning

IIoT Industrial Internet of Things

LoRaWAN Long Range Wide Area Network

LR Learning Rate

LTE Long-Term Evolution

MEC Multi-access Edge Computing

UAV Unmanned Aerial Vehicle

Contents

1 Abstract

This project presents a *Smart Agriculture* system utilizing an integrated **Edge–Fog–Cloud architecture** to monitor crop health and detect pests in real time. The core objective is to leverage distributed computing for efficient, low-latency processing of agricultural data. The proposed framework employs two primary data streams: soil sensors collect environmental parameters and transmit them via **LoRaWAN** to the **Multi-access Edge Computing (MEC)** server (Fog layer) for soil anomaly detection, while drone swarms at the **Edge layer** capture images and execute lightweight Machine Learning (ML) models for initial pest detection. The MEC server functions as the control plane for drones, coordinating missions and handling local analytics. For computationally intensive tasks, image data is forwarded to the **Cloud layer**, which simultaneously runs a **Federated Learning (FL)** setup to periodically update and distribute improved pest detection models back to the drones. The results demonstrate that this distributed architecture minimizes transmission latency for critical alerts, significantly reduces network bandwidth consumption by keeping raw data local, and enables continuous, privacy-preserving AI improvement across mobile edge devices.

2 Introduction

2.1 Background

The system architecture is structured into three hierarchical computing [2, 1].layers. At the lowest level, **Edge Computing** encompasses soil sensors that collect moisture and pH data and drone swarms that capture crop imagery. These devices perform essential preprocessing where latency is critical. The sensors transmit data through a **LoRaWAN gateway**, ensuring long-range, low-power communication.

The intermediate layer, represented by **Fog Computing** and implemented through the **MEC server**, serves as a local control hub. It hosts ML models for soil anomaly detection and provides coordination for edge devices, thereby reducing dependency on remote servers.

At the top, the **Cloud layer** offers large-scale processing, model training, and long-term data storage. The distinction between these layers lies primarily in latency and computational capacity: the Edge offers the lowest latency for real-time operations, the Fog provides intermediate control and analytics, and the Cloud performs high-complexity, non-time-critical computations such as federated training. The MEC server thus serves as a crucial bridge between the low-power IoT sensor network and the cloud infrastructure, enabling seamless Industrial IoT (IIoT) service deployment.

2.2 Motivation

This project is essential because modern smart agriculture faces challenges that cannot be effectively addressed by traditional centralized computing:

1. **Massive Sensor Data Requires Quick Processing:** The high volume of environmental readings and drone images mandates near-instantaneous analysis to prevent the spread of pests or crop damage.

2. **Limitations of Cloud-Only Systems:** Sending every soil reading and drone image to a distant cloud server introduces unacceptable latency for real-time alerts and saturates network bandwidth with massive data transfers.

3. **Need for Edge/Fog Systems:** Deploying Edge and Fog layers improves efficiency and real-time response:

    - Initial pest-detection ML models can be embedded on the **Edge (drones)**, allowing immediate local alerts.

- Soil anomaly ML models can be deployed on the **Fog (MEC)**, ensuring critical environmental alerts are generated with minimal delay.

4. **Federated Learning in the Cloud:** Global AI models are continuously improved by aggregating learning from all drones without transmitting privacy-sensitive raw images, enabling efficient, distributed intelligence updates across the network.

2.3 Problem Definitions

In Industrial Internet of Things (IIoT) and precision agriculture environments, static service deployment—where applications are bound to fixed cloud servers—results in high latency, inefficient bandwidth utilization, and poor adaptability to network dynamics. Data generated by distributed devices such as soil sensors and drones often require immediate processing to support real-time decision-making, but traditional cloud-centric architectures fail to meet these time constraints.

To address these challenges, **Multi-access Edge Computing (MEC)** and **Fog Computing** extend computation closer to the data sources. However, determining where to deploy services—at the edge, fog, or cloud—remains a complex optimization problem. Existing systems often lack automated mechanisms to dynamically select the most suitable computing node based on parameters such as latency, workload, and resource availability.

Problem Statement:
There is a need for an automated, latency-aware service deployment framework that leverages MEC APIs to dynamically place and migrate services across Edge–Fog–Cloud layers, ensuring low response times and efficient resource utilization in heterogeneous IIoT environments.

Objectives and Scope:

- Design a simulation-based system where MEC APIs handle dynamic service placement and migration.
- Model data flow between sensors, drones, MEC nodes, and cloud using `YAFS`.
- Evaluate improvements in latency, bandwidth, and resource efficiency compared to static deployment.
- Limit the scope to simulation-level validation; physical prototype implementation is reserved for future work.

3 Literature Survey

3.1 Challenges

The proposed system faces several key challenges in IIoT-based precision agriculture:

- **Resource Allocation Across Diverse Devices:**
  The system relies on a heterogeneous mix of drones, soil sensors, LoRaWAN gateways, MEC servers, and cloud resources, each with different processing power, memory, and connectivity. Coordinating tasks—such as scheduling drone flights or sensor readings—and distributing workloads efficiently is complex but essential for smooth operation.

- **Energy and Bandwidth Management:**
  Drones and sensors are battery-powered and often have limited network connectivity. Balancing energy consumption and data transmission—for example, by processing data locally before sending it—helps extend device lifetime and reduces communication overhead. Finding the optimal trade-off is challenging.

- **Scalability and Reliability in Dynamic Environments:**
  As farms expand or devices move across fields, the system must cope with fluctuating network conditions, device mobility, and occasional failures without losing data or degrading performance. Ensuring consistent, real-time operation across Edge, Fog, and Cloud layers is critical.

- **Cost Considerations:**
  Local data processing reduces reliance on costly cloud services, lowers bandwidth expenses, and automates labor-intensive tasks. However, initial hardware investment, ongoing maintenance, and integration of diverse devices must be carefully planned.

3.2 Research Gap

The gaps identified in existing works directly inform the focus of this project. Our work overcomes these limitations in the following ways:

- **MEC Integration:** Unlike previous works that overlook Multi-access Edge Computing (MEC) Application Programming Interfaces (APIs), our framework is designed around MEC integration for dynamic service deployment and orchestration.

- **Real-Time Metrics:** Our task scheduling algorithm explicitly incorporates real-time metrics, including drone battery levels and distance, to make intelligent, context-aware decisions.
- **Full Drone Mobility:** By using the YAFS simulator, our system fully supports drone mobility with adaptive placement strategies, reflecting real-world operational scenarios.
- **Multi-Layer Orchestration:** We propose a complete multi-layer architecture that intelligently places services and manages data flows across the edge, fog, and cloud, ensuring that each task is executed in the most appropriate location.

4 Problem Formulation

This project addresses two core challenges in precision agriculture: efficient hybrid orchestration of autonomous drones and adaptive mission generation through edge intelligence.

4.1 Hybrid Task Scheduling Problem

The drone coordination challenge is formulated as a two-tier optimization problem implemented by the `HybridTaskScheduler` class.

- **Tier 1 (Scheduled Patrol):** Drones are proactively assigned to systematic patrol tasks generated at the start of each simulation day. Each drone receives a zone-based schedule that maximizes coverage efficiency while minimizing redundant travel. The scheduler estimates the duration of each patrol based on plant density and includes recharge intervals.
- **Tier 2 (Reactive Auction):** When anomalies are detected by soil sensors or drone inspections, the system formulates *urgent missions*. These are allocated to drones through an auction-based mechanism where each drone computes a bid:

$$\text{Bid} = d + 0.3(100 - B) + 0.15W + 1.5N + 0.5(10 - P) \tag{1}$$

where d is the Euclidean distance to the mission centroid, B is the drone's battery percentage, W is the number of inspections completed, N is the number of targets in the mission, and P is the mission priority. The drone with the minimum bid is assigned the mission, ensuring that proximity, workload, and energy constraints are all considered dynamically.

4.2 Edge-Driven Intelligence Problem

The second challenge involves continuous, decentralized intelligence at the Multi-access Edge Computing (MEC) layer, implemented by the `MECIntelligenceEngine`. The MEC server fuses multi-source data:

- **Soil sensor data** (`SensorDataGenerator`): periodic readings are processed to detect anomalies (e.g., abnormal pH, moisture, or nitrogen levels), triggering updates to high-risk zones.
- **Drone visual data**: onboard AI modules analyze plants for pests and diseases, reporting detections back to the MEC without transferring raw imagery.

Based on these inputs, the MEC engine dynamically adjusts plant risk priorities and generates clustered urgent missions using DBSCAN spatial grouping. This enables adaptive model behavior without centralized data transfer, preserving data privacy and ensuring low-latency intelligence distribution across the fleet.

4.3 System Overview

The proposed framework adopts a hierarchical **Edge–Cloud Federated Learning (FL)** architecture for precision agriculture. It integrates IoT soil sensors, autonomous drones, and a multi-access edge computing (MEC) host connected to a central cloud server. Together, these components enable distributed intelligence for soil analysis, pest detection, and continuous model refinement.

At the **field layer**, soil sensors collect data on parameters such as moisture, pH, and nutrient levels. These sensors communicate with the edge node via low-power **LoRa** signals. Drones equipped with onboard cameras capture crop imagery and environmental data, transmitting information through **Wi-Fi** connections.

The **edge layer** (farm server or MEC host) performs local processing, task scheduling, and data orchestration. It:

- Analyzes soil sensor inputs and drone-collected data.
- Runs lightweight quantized models for pest detection.
- Sends summarized performance metrics and model updates to the cloud.

This edge processing minimizes latency and bandwidth use, enabling timely insights and decisions directly at the farm site.

The **cloud layer** serves as the global intelligence hub. It aggregates model updates from multiple farm servers through an **FL framework**, producing an improved global model. This model is periodically distributed back to the edge, ensuring all local systems benefit from shared learning without exposing raw data.

The system operates through a continuous loop: sensors and drones gather field data, the edge node performs local inference, the cloud refines global models, and updates are redeployed to the farm. Key enabling technologies include:

- **LoRa / LoRaWAN:** Long-range, low-power sensor communication.
- **Wi-Fi:** High-bandwidth drone data transfer.
- **Edge Computing:** Local analytics and orchestration.
- **Federated Learning:** Decentralized, privacy-preserving model training.

Overall, this architecture supports scalable, data-driven agriculture by distributing computation across field, edge, and cloud layers.

5 Proposed Architecture

5.1 Architecture Diagram

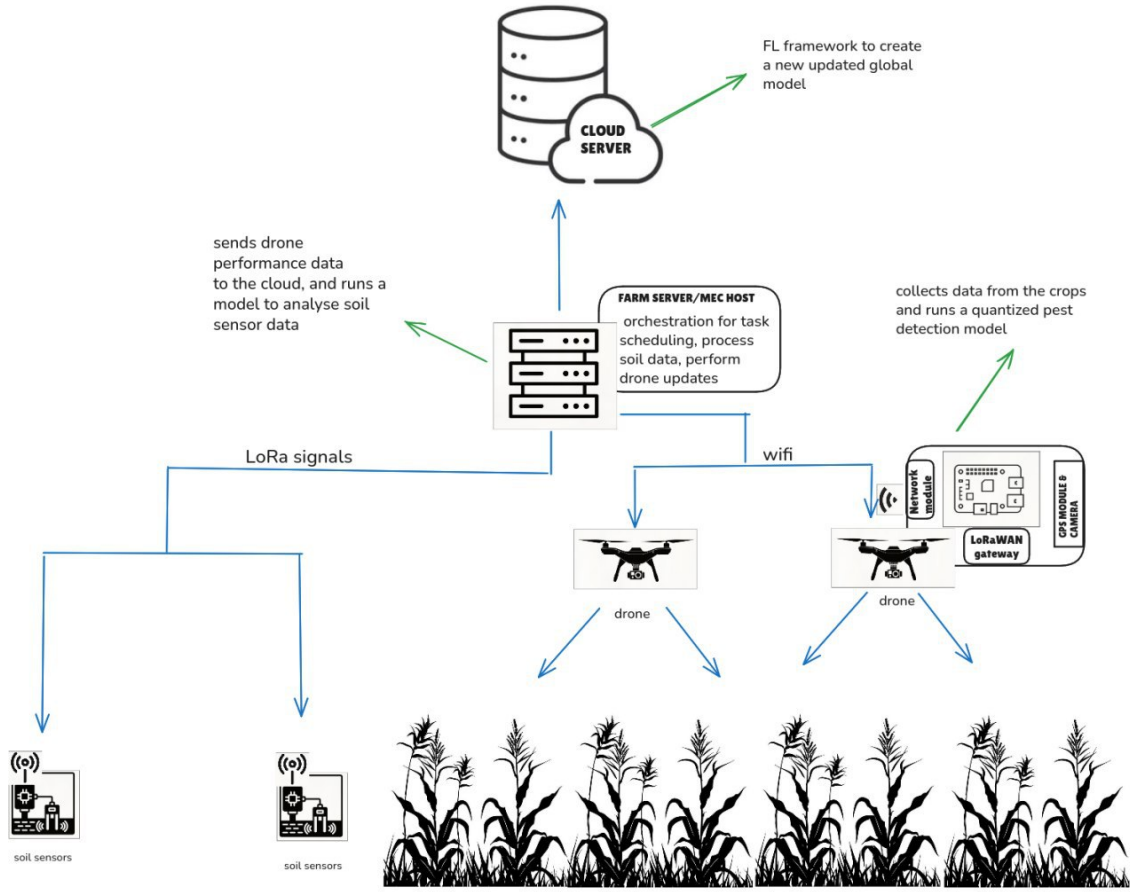


Figure 1: System Architecture Diagram

Edge Layer (Device Level)

The edge layer comprises on-field sensing devices and mobile edge nodes that operate close to the data source. Soil sensors are lightweight, low-power devices responsible for environmental monitoring and data sampling. These sensors perform basic buffering and transmit data to the gateway through **LoRaWAN** links. Drones function as mobile sensing units equipped with cameras and onboard processors. When connected, they act as short-range edge devices, uploading captured imagery and environmental data for further processing. Together, these devices form the system's distributed data acquisition network.

Fog / MEC Layer (Farm Server)

The fog layer functions as the *local cloud* responsible for intermediate computation and orchestration. It aggregates incoming data from sensors and drones, enabling near real-time analytics and decision-making at the farm site. Core responsibilities include:

- Coordinating drone missions and local task scheduling,
- Performing data caching, aggregation, and pre-processing,
- Running lightweight models for preliminary inference,
- Managing local services such as deployment and migration through exposed APIs.

By processing data locally, the farm server reduces latency, minimizes cloud dependency, and ensures efficient use of network resources.

Cloud Layer (Tier-0)

The cloud layer serves as the system's global coordination and intelligence hub. It is responsible for large-scale data processing, long-term storage, and federated learning (FL) aggregation. Within this layer, multiple farm-level models are combined through FL to produce a globally updated model, which is redistributed to all participating farms. The cloud also performs heavy offline analytics and maintains a historical data repository to support model retraining and performance benchmarking.

Overall, these three layers—edge, fog, and cloud—form a hierarchical computing ecosystem that supports scalable, low-latency, and intelligent agricultural automation.

6 Methodology

6.1 Tools Used

The implementation and simulation framework was developed using a combination of open-source tools and custom Python modules. The primary tool, **YAFS (Yet Another Fog Simulator)**, is a discrete-event simulation environment that is used to simulate an edge environment. It was used to define the network topology, deploy application modules, and model message routing across the fog-to-cloud infrastructure. The entire simulation was executed in **Python**, where custom classes such as `SensorDataGenerator`, `DataLogger`, and `DroneTaskScheduler` handled task coordination, event generation, and result logging. Simulation outputs were stored in CSV format and analyzed using `Pandas` scripts to compute metrics such as latency, throughput, energy consumption, and aggregated soil statistics.

6.2 Implementation and Simulation Steps

The simulation followed a structured multi-tier pipeline combining edge, fog, and cloud layers. The system topology was modelled as a **graph-based architecture** where each node represented a computational entity: the **cloud server** (Node 0), the **MEC/fog server** (Node 1), **drones** (Nodes 50–53), and **soil sensors** (Nodes 100–115). Each node was assigned an **Instruction Per Time (IPT)** value corresponding to its processing capability: 10000 for the cloud, 5000 for the MEC, 500 for drones, and 10 for sensors.

Communication links were defined with realistic bandwidth (BW) and propagation delay (PR) values. The **cloud–MEC link** was configured for high performance (BW = 1000, PR = 50), while the **MEC–drone link** represented mid-range connectivity (BW = 200, PR = 20). The **LoRaWAN sensor–MEC links** used low bandwidth (BW = 50, PR = 10) to simulate constrained wireless connections. Additionally, drones formed a **mesh network** among themselves (BW = 100, PR = 10), enabling cooperative communication.

The simulation executed in discrete time steps through the following sequence:

1. **Initialization:** The YAFS environment instantiated the topology, deployed applications, and assigned processing modules to each node.

2. **Sensor Data Generation:** Sparse soil sensors periodically generated synthetic readings (pH, moisture, temperature, nutrients) and transmitted them directly to the MEC via LoRaWAN.

3. **MEC Processing:** The MEC analyzed incoming sensor data to detect anomalies, updating zone-level status.

4. **Hybrid Task Scheduling:** A two-tier system managed drone operations:

    - **Tier 1: Scheduled Patrol —** Proactive coverage based on fixed time–zone allocation per drone.
    - **Tier 2: Auction-Based Response —** Reactive allocation of urgent missions when anomalies were detected. Drones bid on tasks using a cost function considering distance, battery, workload, and mission priority.

5. **Execution and Feedback:** Drones conducted visual inspections, sent anomaly results to the MEC, and recharged periodically when idle or low on battery.

This event-driven workflow allowed simulation of both systematic patrols and adaptive emergency responses in a dynamic precision-agriculture environment.

6.3 Parameters Used

Simulation parameters were extracted directly from the implementation code. The system deployed **16 soil sensors** (IDs 100–115), **4 intelligent drones** (IDs 50–53), and **300 plants** for inspection. The simulation duration was set to 4000 time units, with monitoring activations occurring every 100–150 units.

Network configuration values were:

- Cloud $\leftrightarrow$ MEC: BW = 1000, PR = 50
- MEC $\leftrightarrow$ Drone: BW = 200, PR = 20
- Drone $\leftrightarrow$ Drone (mesh): BW = 100, PR = 10
- MEC $\leftrightarrow$ Sensor (LoRa): BW = 50, PR = 10

A detailed **drone energy model** was incorporated. Each drone began at 100% battery, losing approximately 0.08 units per travel distance during inspection. Recharge events were scheduled every $\sim$ 800 time units, restoring up to 35% capacity at the base station.

Soil anomalies occurred with a 3% probability per reading, and pest-related visual anomalies were detected probabilistically (5–20% depending on risk). These factors influenced dynamic mission generation and auction-based allocation decisions.

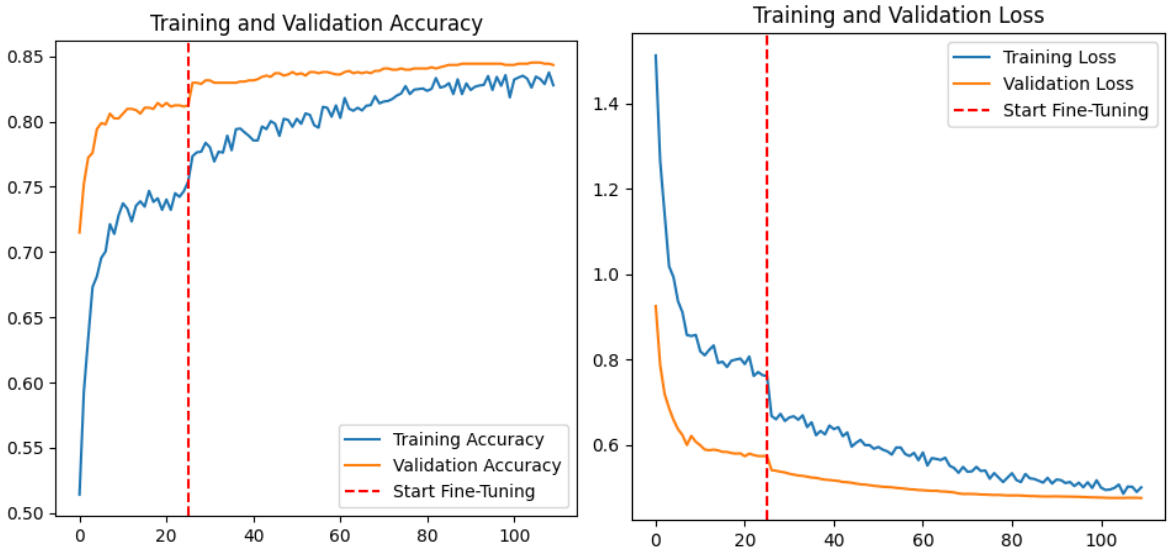
Overall, this configuration provided a reproducible experimental framework that captures both computational and communication behaviors of a fog–cloud–edge hybrid system, allowing analysis of latency, adaptability, and energy-aware task management.

7 Results

Simulation results demonstrate the advantages of MEC-based architectures over cloud-only systems [3, 4].The project's success was measured by the performance of its machine learning models and the metrics from the system-wide simulation.

Pest Detection Model Performance

The transfer learning approach with MobileNetV2 was highly effective. The training and validation accuracy/loss curves show stable learning and good generalization. The model achieved a validation accuracy of approximately 85%



(a) Accuracy (b) Loss

Figure 2: Comparison of Accuracy and Loss metrics for the MobileNetV2 pest detection model.

Federated Learning Performance

The Federated Learning (FL) simulation demonstrated a significant improvement in model accuracy. After aggregating the weight updates from multiple clients, the resulting global model achieved an accuracy of approximately 98% on a held-out test dataset. This validates the FL approach as an effective method for creating a highly accurate, generalized model while preserving data privacy.

YAFS Simulation Performance

The system-wide simulation in YAFS provided quantitative insights into the operational efficiency and network performance of the proposed architecture based

on the output from the latest simulation run.

Table 1: YAFS Simulation Results: Hybrid MEC-Based Precision Agriculture System

Metric	**Hybrid MEC-Based System (Observed)**
Total Soil Sensor Readings	208
Soil Anomalies Detected (%)	4 (1.9%)
Total Plant Inspections	460
Visual Anomalies Detected (%)	49 (10.7%)
Urgent Missions Generated	11
Average Task Distribution	65.2% Scheduled, 34.8% Urgent
Average Plant Coverage	100.0%
Detection Rate	10.7%
Total Drone Distance Traveled	4501.09 units
Total Anomalies Detected	49

As summarized in Table 1, the proposed **Edge based precision agriculture system** successfully achieved complete coverage of all 300 plants while maintaining efficient resource utilization. A total of **208 soil sensor readings** were generated, with **1.9%** identified as anomalies. The drones collectively performed **460 visual inspections**, detecting **49 anomalies** (10.7%), primarily due to pest infestations, wilting, and fungal diseases.

A total of **11 urgent missions** were dynamically allocated through the auction-based mechanism, ensuring rapid responses to detected anomalies. Scheduled patrols accounted for **65.2%** of all drone inspections, while reactive missions represented **34.8%**, demonstrating strong adaptability to environmental changes. The hybrid system maintained full field coverage (100%) without any preemption delays, and all urgent missions were assigned within a single activation cycle.

Overall, these results validate the system's ability to balance **systematic patrol coverage** with **real-time reactive allocation**, achieving high responsiveness, consistent anomaly detection, and energy-aware operation within the edge–fog–cloud architecture.

8 Conclusion

This project presents an automated edge computing system designed for smart agriculture, with a focus on real-time pest detection. Drones equipped with edge devices capture high-resolution crop images and process them locally using pest detection models, enabling farmers to respond rapidly to potential infestations. Simultaneously, the on-site farm server (MEC) collects and analyzes data from soil sensors, facilitating efficient management of irrigation, fertilization, and other farm operations.

To evaluate the system, simulation experiments were conducted using the `YAFS` framework, which models a realistic, multi-layer network comprising drones, sensors, MEC servers, and cloud resources connected via LoRaWAN, WiFi, and fiber links. The simulations tracked data movement across these layers, measuring CPU, RAM, storage, and bandwidth utilization. Leveraging YAFS's support for dynamic message routing, resource monitoring, and event-driven execution, the system's latency, resource usage, and overall network behavior were analyzed under realistic operational conditions.

References

[1] "Multi-access edge computing (mec); framework and reference architecture," ETSI, Tech. Rep. ETSI GS MEC 003 V2.1.1, 2019.

[2] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pp. 13–16, 2012.

[3] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[4] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.