

A Federated Learning Framework for Optimizing Edge Computing with Semantic Offloading

Avi Deb Raha*, Apurba Adhikary*, Mrityunjoy Gain**, and Choong Seon Hong*

*Department of Computer Science and Engineering, Kyung Hee University, Yongin-si 17104, Republic of Korea

**Department of Artificial Intelligence, Kyung Hee University, Yongin-si 17104, Republic of Korea

avi@khu.ac.kr, apurba@khu.ac.kr, gain@khu.ac.kr, cshong@khu.ac.kr

Abstract—Traditional real-time edge tasks, such as monitoring and control, often rely on centralized processing, which introduces single points of failure, high communication costs, and delays, particularly in latency-critical environments. Federated Learning (FL) offers a distributed alternative by enabling collaborative model development without the need to upload local data, thereby reducing the communication overhead and enhancing privacy. However, the limited computational resources of edge devices constrain their ability to store extensive data and run complex deep learning models. To address these challenges, we propose a novel hybrid framework that integrates FL to optimize real-time task execution and decision making. Our architecture consists of three layers: edge devices, regional edge servers (RES), and cloud servers (CS). The RESs, in collaboration with CS, train high-capacity models for themselves and lightweight models for edge deployment, enabling efficient initial data analysis on the edge. To address the sub-optimal performance of lightweight models in complex scenarios, we introduce a semantic offloading mechanism based on uncertainty estimation. When the uncertainty of the model exceeds a predefined threshold, the data are dynamically offloaded to the RES, and high capacity models enhance decision making for these complex cases. This collaborative strategy ensures that most data are processed locally with minimal latency, whereas complex or ambiguous instances benefit from the RES's superior computational capabilities. Experimental results demonstrate the effectiveness of the proposed method, achieving task execution accuracy improvements of 2.59% to 5.68% over using only lightweight models, closely matching centralized computing systems. Additionally, the proposed method realizes bandwidth savings of 58.73% to 86.86% compared to relying solely on sophisticated RES models across diverse scenarios.

Index Terms—Federated Learning, Hybrid Framework, Semantic Offloading, Edge Devices, Distributed Systems.

I. INTRODUCTION

In Industry 4.0, the demand for efficient real-time smart task execution systems—such as monitoring and decision-making—has surged across sectors like transportation, agriculture, healthcare, and infrastructure [1]. By utilizing sophisticated deep learning models [2], these systems enhance safety,

optimize performance, and enable proactive maintenance. Traditionally, they rely on centralized or cloud-based processing [3], where data from edge devices is sent to a central unit for analysis [4]. However, training these models requires transferring large volumes of data, leading to challenges in managing vast real-time data streams at a single point, ensuring data privacy, and incurring high communication costs. Additionally, centralized architectures are prone to single points of failure, which can result in privacy breaches, service unavailability, and data transmission delays that compromise system responsiveness in large-scale environments [3]–[6].

Federated Learning (FL) [7]–[11] addresses some of these challenges by enabling shared model training across decentralized edge devices without uploading raw data [12], [13]. This distributed approach eliminates single points of failure, reduces communication overhead by transmitting only model updates, and lowers latency through local processing. Additionally, FL simplifies data management and enhances privacy by keeping data local, making it ideal for real-time task execution systems handling sensitive or large-scale data.

However, despite its advantages, implementing FL in edge systems presents significant challenges. Edge devices—such as smartphones, IoT sensors, and Unmanned Aerial Vehicles (UAVs)—are typically resource-constrained [14], possessing limited computational power, storage capacity, and energy resources [1], [15], [16]. These limitations prevent them from storing extensive datasets and running the sophisticated deep learning models required for complex data analysis and real-time task execution, including monitoring, identification, decision-making, and control [1], [16]. At the same time, as previously discussed, relying solely on centralized models introduces issues such as increased latency, higher bandwidth consumption, and potential single points of failure due to the need to transmit large volumes of data to central servers [3]–[5]. Therefore, there is a pressing need for architectures that can balance the computational demands and performance requirements of real-time task execution with the inherent constraints of edge devices while also minimizing latency and bandwidth usage.

To address these challenges, we propose a novel hybrid framework that leverages both edge computing and centralized processing in a FL setup. The framework includes a semantic offloading mechanism that decides whether data should be

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. RS-2023-00207816), (No. RS-2024-00352423), and supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2024-RS-2023-00258649) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation) *Dr. CS Hong is the corresponding author.

uploaded for further analysis or processed locally. This decision is based on the semantic complexity or uncertainty of the data samples, optimizing real-time task execution by balancing computational demands and performance. This architecture is structured into three distinct layers: Edge Devices, Regional Edge Servers (RES), and Cloud Servers. In this system, edge devices collect data and perform initial inference using a lightweight deep learning model suited to their limited resources. When the model's output indicates high complexity reflected by elevated entropy, the semantic offloading mechanism enables the edge device to transmit the data to the RES for more advanced analysis using a sophisticated deep learning model. The RES collaborates with other RESs and the central cloud server within the FL system to train these models. Once training is complete, the lightweight model is deployed to the edge devices for efficient inference.

In various applications, such as healthcare [17], [18] and intelligent transportation system (ITS) [19], [20], data is provided directly to RESs like hospitals [17] or road side units (RSUs). Securing data at the RES level is essential as it ensures that sensitive raw data remains localized, thereby reducing the risk of data breaches, enhancing privacy, and ensuring compliance with data protection regulations. During inference, edge devices utilize the lightweight model for rapid processing, while the RES employs the high-complexity model for more advanced analysis when necessary. This hybrid approach effectively balances computational demands, minimizes latency and bandwidth usage, and leverages the strengths of both edge and centralized systems to enhance the overall scalability, reliability, and responsiveness of real-time task execution. The following are the key contributions:

- We propose a novel hybrid framework with a semantic offloading mechanism that optimizes real-time task execution in edge computing environments by integrating FL with a three-layer architecture consisting of edge devices, RESs, and cloud servers.
- The proposed framework utilizes FL to collaboratively train resource-efficient models tailored to the computational capabilities of each layer. Lightweight models are deployed on edge devices to handle essential tasks with minimal overhead, while more sophisticated models are deployed on RESs, which have greater computational resources.
- We introduce a semantic offloading strategy based on uncertainty estimation. This mechanism assesses the complexity of data samples by estimating the model's entropy and determines when to offload data from edge devices to regional servers, enhancing decision-making.
- Extensive experiments demonstrate the effectiveness of our hybrid FL framework. Our approach increases task execution accuracy by 2.59% to 5.68% compared to processing solely on edge devices, closely matching the performance of high-capacity models. Additionally, it achieves bandwidth savings of 58.73% to 86.86% compared to exclusively using sophisticated RES layer models across diverse scenarios.

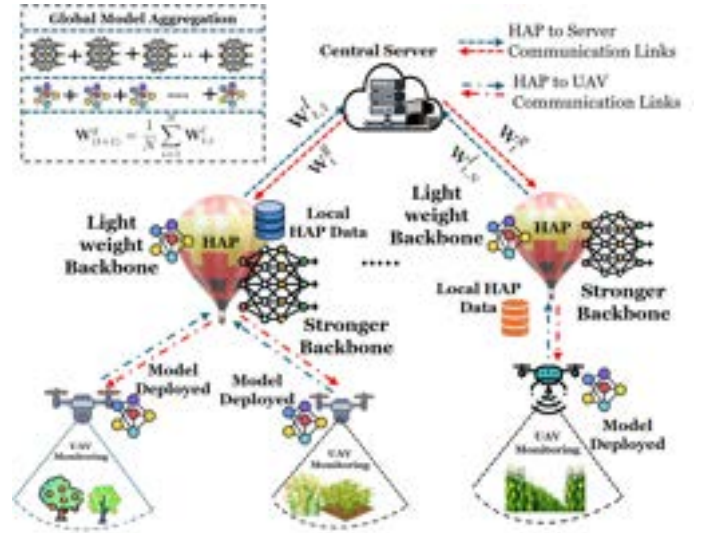


Figure 1: System Model for A Federated Learning Framework for Optimizing Edge Computing with Semantic Offloading.

II. SYSTEM MODEL AND PROBLEM DESCRIPTION

For the purposes of this study, we specifically considered High Altitude Platforms (HAPs) as the RES and Unmanned Aerial Vehicles (UAVs) as the edge devices. However, the framework is generalizable to other scenarios, such as hospitals functioning as RESs with smartphones as edge devices in healthcare, or RSUs as RESs with Connected and Autonomous Vehicles (CAVs) as edge devices in intelligent transportation systems. This adaptability makes the framework highly applicable to various distributed environments, supporting real-time monitoring and intelligent decision-making across diverse domains. In the following subsections, we outline the system model used in this study, provide an overview of FL, describe the inference phase, and explain the semantic offloading conditions.

A. System Model

Considering a central server (CS) coordinates a network of H HAPs, represented by the set $\mathcal{H} = \{h_1, h_2, \dots, h_H\}$, as shown in Fig. 1. Each HAP $h_i \in \mathcal{H}$ supervises a group of Unmanned Aerial Vehicles (UAVs), denoted by the set $\mathcal{U}_{h_i} = \{u_{i1}, u_{i2}, \dots, u_{iU_i}\}$, where U_i is the number of UAVs operating under HAP h_i . The UAVs are tasked with monitoring designated areas and executing a specific task represented as $\mathcal{T}$. Given the inherent resource limitations of UAVs, including restricted computational power, storage capacity, and energy availability, each UAV $u_{i,j} \in \mathcal{U}_{h_i}$ is equipped with a lightweight deep learning model, denoted by $\mathcal{M}_{u_{i,j}}^{\text{light}}$, to efficiently perform the task $\mathcal{T}$. The UAVs may encounter complex or ambiguous data samples $\mathbf{x}_{ij} \in \mathcal{X}$ for which $\mathcal{M}_{u_{i,j}}^{\text{light}}$ lacks sufficient confidence to make accurate predictions. In such cases, the UAV $u_{i,j}$ requires offloading the uncertain sample $\mathbf{x}_{ij}$ to its supervising HAP h_i for more robust processing. Each HAP h_i is equipped with a more

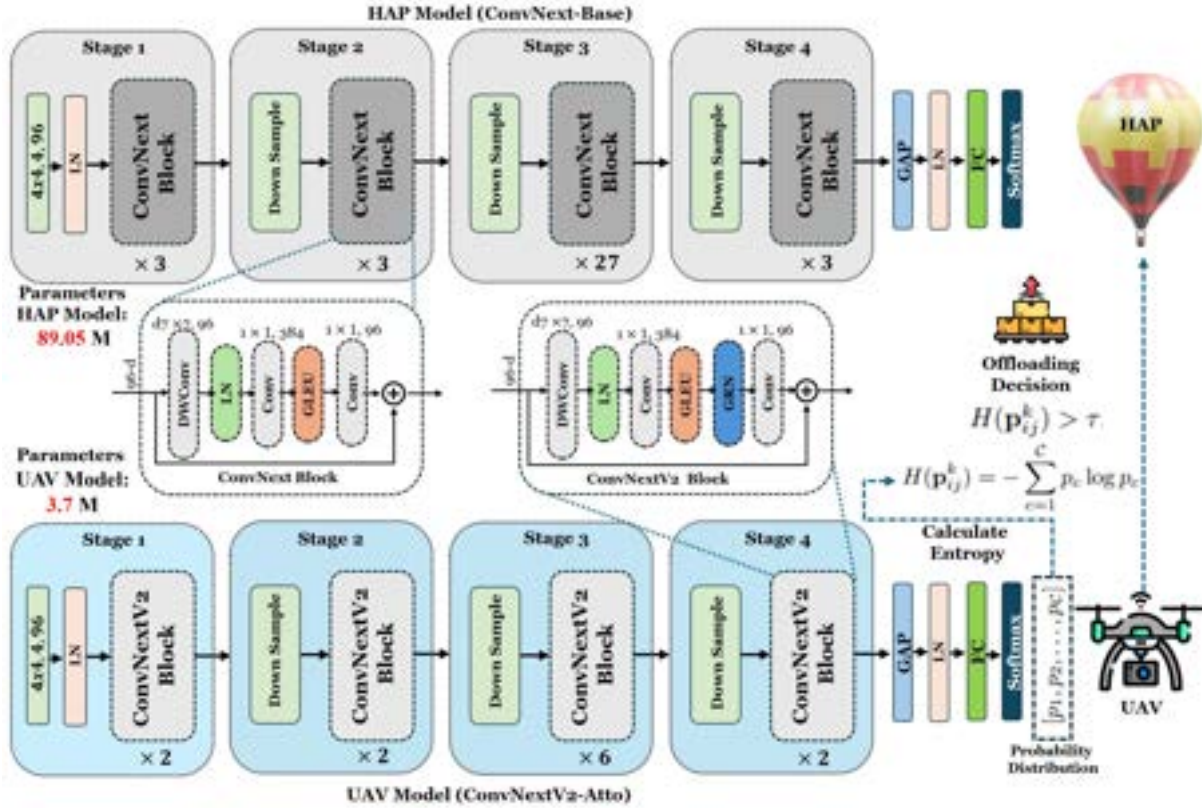


Figure 2: Model Architectures for the HAPs (ConvNext-B) and UAVs (ConvNextV2-Atto) and Offloading strategy.

powerful deep learning model, $\mathcal{M}_{h_i}^{\text{strong}}$, capable of handling these offloaded samples and providing more accurate and reliable results. To effectively train both the lightweight models on UAVs and the stronger models on HAPs, we utilized a FL framework. Within this framework, each HAP $h_i \in \mathcal{H}$ aggregates data from its associated UAVs to construct a local dataset $\mathcal{D}_{h_i} = \{(\mathbf{x}_{ij}^k, y_{ij}^k)\}_{k=1}^{|\mathcal{D}_{h_i}|}$, where $|\mathcal{D}_{h_i}|$ represents the number of data points collected by HAP h_i . Here, $\mathbf{x}_{ij}^k$ denotes the k -th data sample collected by UAV u_{ij} , and y_{ij}^k corresponds to the target output associated with $\mathbf{x}_{ij}^k$, depending on the specific task $\mathcal{T}$. For instance, for classification tasks, $y_{ij}^k \in \{1, 2, \dots, C\}$, where y_{ij}^k represents a class label. For object detection tasks, $y_{ij}^k = \{(c_n, b_n)\}_{n=1}^N$, where c_n denotes the class label and $b_n = (x, y, w, h)$ specifies the bounding box coordinates for each detected object in the sample. In this study, we considered the task $\mathcal{T}$ involves the classification of insect pests based on visual data captured by the UAVs. Therefore, $\mathbf{x}_{ij}^k \in \mathcal{X}$ denotes an input image within the set $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, where each image $\mathbf{x}_n \in \mathcal{X}$ has dimensions (C, H, W) , representing the color channels, height, and width, respectively. Consequently, $y_{ij}^k \in \{1, 2, \dots, C\}$, where each label $y_{ij}^k \in \mathcal{Y}$ indicates the specific species of insect pest present in image $\mathbf{x}_{ij}^k$, with $\mathcal{Y}$ representing the set of all possible labels. This FL-based approach facilitates collaborative model training while preserving data privacy at HAP level and minimizing communication overhead between UAVs and HAPs.

B. Federated Learning Framework

The objective is to coordinate collaboration between the HAPs and the CS to collectively train two shared global models: a lightweight model for UAVs and a stronger model for HAPs. We denote these models as $\mathcal{F}^{\text{light}}(\mathbf{W}_g^{\text{light}}; \mathbf{x})$ and $\mathcal{F}^{\text{strong}}(\mathbf{W}_g^{\text{strong}}; \mathbf{x})$, where $\mathbf{W}_g^{\text{light}}$ and $\mathbf{W}_g^{\text{strong}}$ represent the global model parameters for the lightweight and stronger models, respectively. Since both models follow the same training methodology, we will refer to their global model parameters collectively as $\mathbf{W}_g$ in this section for simplicity. The local empirical risk (loss function) at HAP h_i is defined as [21], [22]:

$$L_{h_i}(\mathbf{W}_i) = \frac{1}{|\mathcal{D}_{h_i}|} \sum_{k=1}^{|\mathcal{D}_{h_i}|} \mathcal{L}(\mathcal{F}(\mathbf{W}_i; \mathbf{x}_{ij}^k), y_{ij}^k), \quad (1)$$

where $\mathbf{W}_i$ are the local model parameters at HAP h_i , and $\mathcal{L}$ denotes the loss function. For classification tasks, we use the cross-entropy loss:

$$\mathcal{L}(\mathcal{F}(\mathbf{W}_i; \mathbf{x}_{ij}^k), y_{ij}^k) = - \sum_{z=1}^C \mathbf{1}_{(y_{ij}^k=z)} \log p_z(\mathbf{x}_{ij}^k; \mathbf{W}_i), \quad (2)$$

where $p_z(\mathbf{x}_{ij}^k; \mathbf{W}_i)$ is the predicted probability of class z for input $\mathbf{x}_{ij}^k$ computed by the model $\mathcal{F}(\mathbf{W}_i; \mathbf{x}_{ij}^k)$, and $\mathbf{1}_{(y_{ij}^k=z)}$ is the indicator function that equals 1 if $y_{ij}^k = z$ and 0 otherwise.

After local training, each HAP h_i updates its local model parameters $\mathbf{W}_i$ by minimizing the local empirical risk $L_{h_i}(\mathbf{W}_i)$. The updated local model parameters are then sent to the central server. The central server aggregates these local updates

to form the new global model parameters $\mathbf{W}_g$ using the Federated Averaging algorithm:

$$\mathbf{W}_g = \sum_{i=1}^H \frac{|\mathcal{D}_{h_i}|}{\sum_{j=1}^H |\mathcal{D}_{h_j}|} \mathbf{W}_i, \quad (3)$$

where H is the total number of HAPs. This aggregated global model $\mathbf{W}_g$ encapsulates the collective knowledge from all HAPs without requiring access to their raw data, thereby preserving data privacy at the local level. Upon completing the training of both models, the lightweight model is deployed to each UAV, while the more robust model is deployed to the HAPs.

C. Inference Phase, Semantic Offloading and Communication Model

During the inference phase, each UAV u_{ij} utilizes its lightweight model $\mathcal{M}_{u_{ij}}^{\text{light}}$ for rapid and resource-efficient processing of the tasks $\mathcal{T}$. However, when the model's decision for a sample x_{ij} is low it needs to offload the sample x_{ij} to the HAP h_i . This offloading can be represented as an binary decision as follows:

$$\delta_{ij}(x_{ij}) = \begin{cases} 1, & \text{if } \text{Conf}_{\mathcal{M}_{u_{ij}}^{\text{light}}}(x_{ij}) < \tau, \\ 0, & \text{if } \text{Conf}_{\mathcal{M}_{u_{ij}}^{\text{light}}}(x_{ij}) \geq \tau, \end{cases} \quad (4)$$

Where $\delta_{ij}(x_{ij})$ represents the offloading binary decision variable and τ represent a threshold value. $\delta_{ij}(x_{ij}) = 1$ indicates that the sample x_{ij} is offloaded to the HAP h_i for processing, while $\delta_{ij}(x_{ij}) = 0$ means the sample is processed locally by the UAV u_{ij} . For transmitting the samples from UAV u_i to HAP h_j , we considered the following communication model based on Line-of-Sight (LoS) transmission. The Free-Space Path Loss (FSPL) in a LoS environment is given by:

$$\text{PL}_{\text{LoS}}(d, f) = 20 \log_{10} \left(\frac{4\pi df}{c} \right) \quad (5)$$

where d is the distance between UAV u_i and HAP h_j (in meters), f is the carrier frequency of the communication signal (in Hertz), c is the speed of light in vacuum. The distance d between the UAV and HAP can be calculated using the Euclidean distance in three-dimensional space:

$$d = \sqrt{(x_{u_i} - x_{h_j})^2 + (y_{u_i} - y_{h_j})^2 + (h_{u_i} - h_{h_j})^2} \quad (6)$$

where $(x_{u_i}, y_{u_i}, h_{u_i})$ are the coordinates of UAV u_i , $(x_{h_j}, y_{h_j}, h_{h_j})$ are the coordinates of HAP h_j . The Signal-to-Noise Ratio (SNR) for the communication link is defined as:

$$\text{SNR}_{\text{LoS}} = \frac{P_{\text{tx}}}{N_0 \cdot B} \cdot 10^{-\frac{\text{PL}_{\text{LoS}}(d, f)}{10}} \quad (7)$$

where P_{tx} is the transmission power of UAV u_i , N_0 is the noise power spectral density, B is the bandwidth of the communication channel. The HAP h_i processes the offloaded sample x_{ij} using its stronger model $\mathcal{M}_{h_i}^{\text{strong}}$, providing enhanced accuracy for these uncertain cases. The overarching goal of the hybrid architecture is to achieve a balance between maintaining high performance and maximizing operational efficiency.

III. SOLUTION APPROACH

To achieve the overall objective of the study, we aim to ensure that both the lightweight models deployed on UAVs ($\mathcal{M}_{u_{ij}}^{\text{light}}$) and the stronger models on HAPs ($\mathcal{M}_{h_i}^{\text{strong}}$) are robust and effective for the task $\mathcal{T}$.

A. Model Architectures

1) *Stronger Model on HAPs*: For the HAPs, we employ the ConvNeXt-B model [23], a convolutional neural network architecture pretrained on the ImageNet-22K dataset. ConvNeXt-B is suitable for handling complex and high-resolution images due to its deep architecture and large capacity. For the HAPs, we employ the ConvNeXt-B model [23] pretrained on the ImageNet-22K dataset. ConvNeXt-B is well-suited for handling complex and high-resolution images due to its deep architecture and high capacity. Recent studies have shown that ConvNeXt models perform effectively in FL settings [9]. The ConvNeXt-B block is defined as:

$$\mathbf{Y} = \mathbf{X} + \mathcal{F}(\mathbf{X}) \quad (8)$$

$$\mathcal{F}(\mathbf{X}) = \text{Cv}_{1 \times 1}(\text{GELU}(\text{Cv}_{1 \times 1}(\text{LN}(\text{DC}_{7 \times 7}(\mathbf{X})))) \quad (9)$$

Here, $\text{DC}_{7 \times 7}$ denotes a depthwise convolution with a 7×7 kernel, LN is layer normalization, $\text{Cv}_{1 \times 1}$ is a pointwise convolution, and GELU is the Gaussian Error Linear Unit activation function. $\mathbf{Y}$ is the output feature map with a residual connection to $\mathbf{X}$.

2) *Lightweight Model on UAVs*: For the UAVs, we utilize the ConvNeXtV2-Atto model [24], a lightweight variant designed for resource-constrained environments. The ConvNeXtV2-Atto block is defined as:

$$\mathbf{Y} = \mathbf{X} + \mathcal{F}'(\mathbf{X}) \quad (10)$$

$$\mathcal{F}'(\mathbf{X}) = \text{Cv}_{1 \times 1}(\text{GELU}(\text{Cv}_{1 \times 1}(\text{LN}(\text{GRN}(\text{DC}_{k \times k}(\mathbf{X})))))) \quad (11)$$

Here, k is the kernel size optimized for efficiency, $\text{DC}_{k \times k}$ denotes a depthwise convolution with a $k \times k$ kernel, and GRN represents the Gated Residual Network module. Both the architectures are shown in Fig. 2.

B. Semantic Offloading Mechanism Based on Uncertainty

To determine whether a data sample $\mathbf{x}_{ij}^k$ should be processed locally on the UAV or offloaded to the HAP, we implement an offloading mechanism that evaluates the semantic complexity of the sample using uncertainty estimation with the model $\mathcal{M}_{u_{ij}}^{\text{light}}$. Specifically, the entropy of the model's output probability distribution is used as a measure of uncertainty because it effectively quantifies the unpredictability in the model's predictions. A higher entropy indicates greater uncertainty, suggesting that the sample has higher semantic complexity and may benefit from being offloaded to the HAP for more accurate processing. Conversely, lower entropy reflects higher confidence, allowing the UAV to handle the processing locally efficiently.

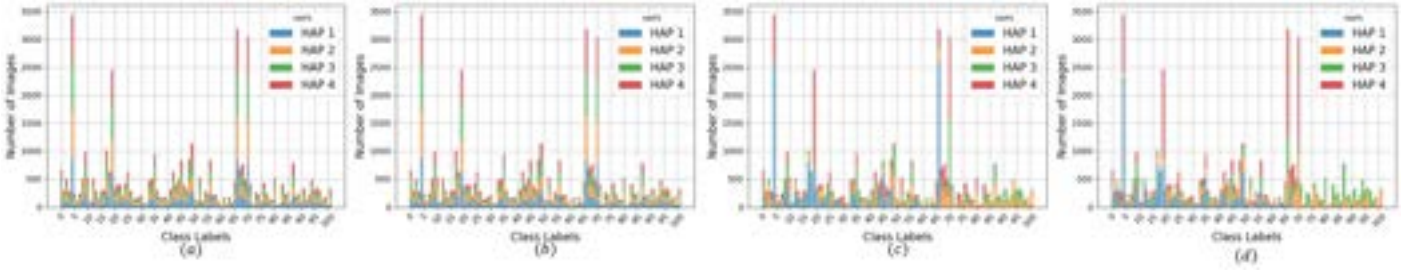


Figure 3: Training Dataset Distribution.

1) *Entropy Calculation*: For a given input $\mathbf{x}_{ij}^k$, the lightweight model produces a probability distribution over the C classes:

$$\mathbf{p}_{ij}^k = \mathcal{F}^{\text{light}}(\mathbf{W}_g^{\text{light}}, \mathbf{x}_{ij}^k) = [p_1, p_2, \dots, p_C], \quad (12)$$

where p_c is the predicted probability for class c . The entropy $H(\mathbf{p}_{ij}^k)$ of this distribution is calculated as:

$$H(\mathbf{p}_{ij}^k) = - \sum_{c=1}^C p_c \log p_c. \quad (13)$$

2) *Offloading Decision*: The UAV u_{ij} decides to offload the sample $\mathbf{x}_{ij}^k$ to its supervising HAP h_i if the entropy exceeds a predefined threshold τ . The offloading decision $\delta_{ij}(\mathbf{x}_{ij}^k)$ is defined as:

$$\delta_{ij}(\mathbf{x}_{ij}^k) = \begin{cases} 1, & \text{if } H(\mathbf{p}_{ij}^k) > \tau, \\ 0, & \text{otherwise,} \end{cases} \quad (14)$$

where $\delta_{ij}(\mathbf{x}_{ij}^k) = 1$ indicates that the sample is offloaded to the HAP, and $\delta_{ij}(\mathbf{x}_{ij}^k) = 0$ means the sample is processed locally by the UAV.

IV. EXPERIMENTAL SETTINGS

A. Dataset

For this study, we utilize the IP-102 dataset [25], a comprehensive pest detection dataset specifically designed for insect pest classification. The dataset is composed of 102 distinct classes and a total of 75,222 images. It is divided into training, validation, and testing sets with a split ratio of 6:1:3, respectively. This extensive and well-balanced dataset provides a comprehensive foundation for training and evaluating our FL models in diverse scenarios. Fig. 3 shows the training data distribution for different.

B. Federated Learning Settings

We conduct our FL experiments using a total of four clients, each representing a distinct HAP in the network. To simulate realistic data distributions, we consider both independent and identically distributed (IID) and Non-IID settings.

- **IID Setting**: In the IID scenario, the data is evenly and randomly distributed among the four clients, ensuring that each client has a similar representation of all pest classes.
- **Non-IID Setting**: To emulate more challenging and realistic conditions, we implement Non-IID data distributions using the Dirichlet distribution with concentration parameters α set to 0.3, 0.5, and 0.7. The Dirichlet distribution

controls the degree of data heterogeneity across clients, where lower values of α result in more skewed and uneven class distributions among clients.

All models are trained with a learning rate of 0.001, optimized using the SGD optimizer. Training is conducted over 20 communication rounds which local model updates are aggregated at the central server using the FedAvg algorithm.

V. RESULTS AND DISCUSSION

Fig. 4 presents the testing convergence curves with respect to communication rounds for different settings. Specifically, Fig. 4(a) displays the curve for the IID setting, while Figs. 4(b), (c), and (d) correspond to Dirichlet distributions with parameters $\alpha = 0.3$, $\alpha = 0.5$, and $\alpha = 0.7$, respectively. The results indicate that the ConvNext-B model consistently outperforms the smaller ConvNextv2-Atto model across all scenarios. This consistent performance gap between the lightweight and stronger models highlights the limitations of lightweight models which will be deployed on UAVs due to their constrained computational resources. Semantic offloading can significantly mitigate this performance gap by leveraging the computational capabilities of HAPs to handle more complex tasks. Furthermore, the FL system significantly surpasses the performance of individual client models for both the lightweight and strong models. This demonstrates that FL can effectively boost model performance while maintaining data privacy and accommodating resource limitations.

Table I provides a comprehensive performance comparison between UAV and HAP models under both IID and non-IID scenarios, with varying distribution parameters (α) and offloading thresholds (τ). The semantic offloading mechanism consistently enhances the overall classification accuracy across all settings. In the IID scenario with $\tau = 0.3$, the overall accuracy improves from the UAV model's 67.29% to 72.40%, indicating a 5.11% enhancement. This improvement highlights the effectiveness of offloading semantically complex samples to the HAPs, which possess stronger computational capabilities. However, as the threshold τ increases, the overall accuracy gradually decreases due to fewer samples being offloaded, thereby reducing the benefits derived from HAP processing. For instance, at $\tau = 0.9$, the overall accuracy in the IID case drops to 69.88%, with an offloading percentage of only 11.19%. This trend underscores the trade-off between

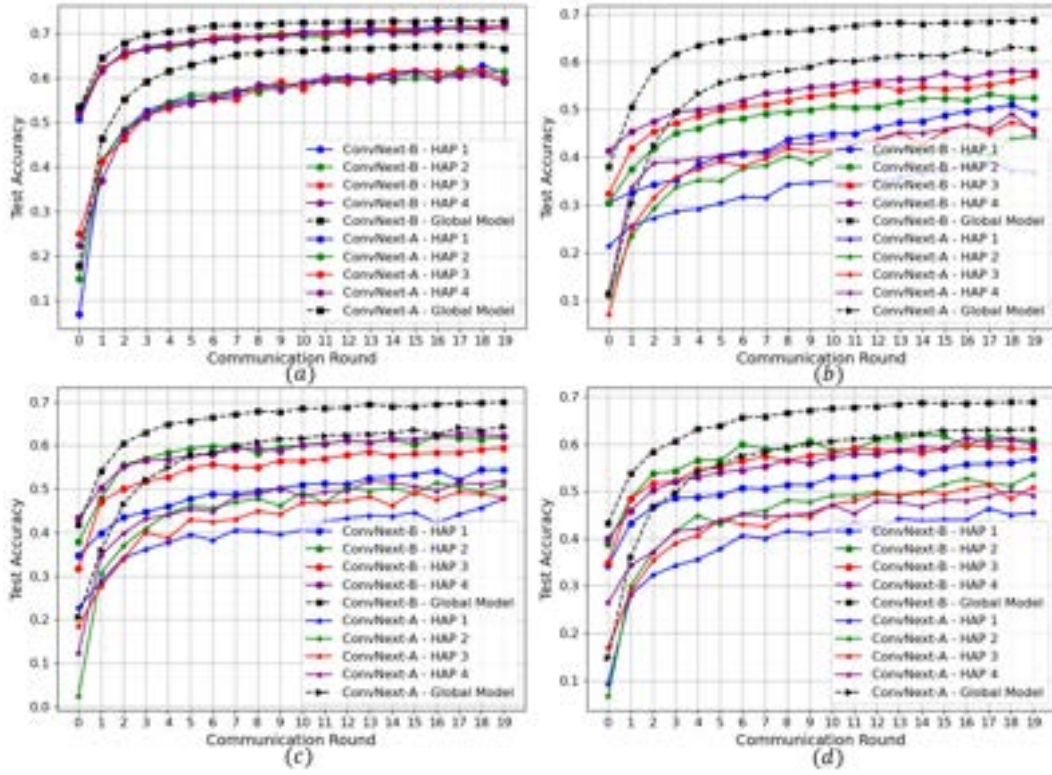


Figure 4: Testing performance convergence curve for different scenarios.

TABLE I: Comparison of UAV and HAP Models in IID and Non-IID Scenarios with Various Beta and Threshold Values

Case	$\alpha(\alpha)$	Thr. (τ) Value	Basic Accuracy		After Offloading				Off. (Per.)
			HAP Model	UAV Model	Overall Accuracy	Improvement Over UAV	Model Accuracy After Offloading		
IID	N/A	0.3	72.97%	67.29%	72.40%	+5.11%	82.48%	50.11%	31.14%
		0.5			71.73%	+4.44%	79.00%	48.95%	24.19%
		0.7			70.66%	+3.37%	75.68%	46.08%	16.96%
		0.9			69.88%	+2.59%	73.11%	44.27%	11.19%
NONIID	0.3	0.3	68.63%	63.01%	68.69%	+5.68%	82.66%	50.05%	42.85%
		0.5			68.27%	+5.26%	78.70%	48.47%	34.48%
		0.7			67.56%	+4.55%	75.06%	46.02%	25.82%
		0.9			66.56%	+3.55%	71.64%	43.53%	18.06%
	0.5	0.3	69.93%	64.25%	69.90%	+5.65%	81.88%	50.57%	38.28%
		0.5			69.42%	+5.17%	78.42%	48.74%	30.33%
		0.7			68.31%	+4.06%	74.61%	45.75%	21.82%
		0.9			67.22%	+2.97%	71.33%	42.93%	14.47%
	0.7	0.3	68.84%	63.17%	68.67%	+5.50%	81.28%	49.28%	39.39%
		0.5			68.16%	+4.99%	77.66%	47.33%	31.31%
		0.7			67.04%	+3.87%	73.64%	44.42%	22.59%
		0.9			66.05%	+2.88%	70.26%	42.54%	15.20%

maintaining high accuracy and minimizing communication overhead, emphasizing the need for an optimal threshold selection.

In non-IID scenarios, varying the α values of the Dirichlet distribution simulates different degrees of data heterogeneity among clients. The results demonstrate that the proposed framework maintains robustness against data heterogeneity, as the overall accuracy remains relatively stable across different α values. For example, with $\alpha = 0.3$ and $\tau = 0.3$, the overall accuracy reaches 68.69%, showing a 5.68% improvement over the UAV model. Similar improvements are observed for

$\alpha = 0.5$ and $\alpha = 0.7$, indicating that the semantic offloading mechanism effectively compensates for data heterogeneity by allowing HAPs to handle more complex and diverse samples.

Additionally, the UAV models exhibit higher accuracy on the non-offloaded samples, suggesting confident local predictions, while the HAP models effectively manage the more uncertain and complex samples despite their inherent complexity. This division of labor ensures that UAVs operate efficiently without overburdening their limited computational resources.

Table I also highlights the offloading percentages across different scenarios. As the threshold τ increases, the percent-

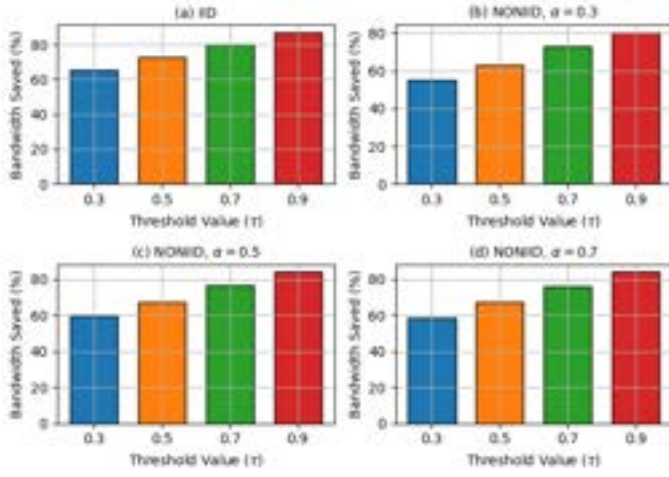


Figure 5: Bandwidth Saving Percentage Compared to Centralized Computing Systems for different Scenarios.

age of samples offloaded to HAPs decreases, which aligns with the observed reduction in accuracy improvements. This relationship emphasizes the critical balance between offloading enough samples to enhance accuracy and minimizing the communication overhead to conserve bandwidth.

Fig. 5 depicts the percentage of bandwidth saved (%) compared to centralized systems for both IID and NONIID data distributions with varying α parameters. From the fig, we can see that as τ increases, more samples are processed locally, resulting in higher bandwidth savings. Specifically, in the IID scenario, bandwidth savings grow from 65.49% at $\tau = 0.3$ to 86.86% at $\tau = 0.9$. For NONIID distributions, the savings also increase with higher τ values: from 55.04% to 80.22% when $\alpha = 0.3$, from 59.49% to 83.84% when $\alpha = 0.5$, and from 58.73% to 84.06% when $\alpha = 0.7$. These results demonstrate that higher entropy thresholds consistently lead to greater bandwidth savings across all data distributions. Additionally, as the α value increases, indicating a reduction in the degree of non-IIDness, the bandwidth savings in NONIID scenarios approach those of the IID case. This highlights that less severe non-IID data distributions benefit more from the semantic offloading mechanism, enhancing overall bandwidth efficiency.

Overall, the experimental results confirm that the semantic offloading mechanism enhances classification performance by intelligently allocating uncertain samples to more capable HAPs. Adjusting the threshold τ allows for a balance between accuracy and resource utilization. Additionally, the federated learning framework demonstrates robustness against data heterogeneity, maintaining consistent performance across different α values in Non-IID settings.

VI. CONCLUSION

In this paper, we proposed a novel hybrid federated learning framework to optimize real-time task execution and decision-making processes in edge computing environments. Our architecture consists of three layers: Edge Devices, Regional Edge

Servers (RES), and Cloud Servers. By training both a high-capacity model and a lightweight model within the federated learning system, we effectively addressed the limitations of edge devices' computational resources. The lightweight models deployed on the edge devices enable rapid and resource-efficient initial data analysis. When the model's confidence is low, data is dynamically offloaded to the RES, where the high-capacity models provide enhanced decision-making for uncertain cases. Our dual-model approach ensures that most data is processed locally with minimal latency, while complex or ambiguous instances benefit from the RES's superior computational capabilities. This strategy effectively balances response speed, communication costs, and data management. The experimental results demonstrated significant improvements in classification accuracy and system performance under both IID and Non-IID data distributions. By leveraging the strengths of both lightweight and high-capacity models, our framework addresses the critical challenges in real-time edge computing applications, paving the way for more efficient and scalable systems. Looking ahead, we plan to integrate knowledge distillation and semantic communication to further enhance the framework's efficiency and scalability. Additionally, extending its application to diverse scenarios, such as smart healthcare and autonomous transportation, will validate its adaptability and robustness in real-world environments.

REFERENCES

- [1] A. D. Raha, M. S. Munir, A. Adhikary, Y. Qiao, and C. S. Hong, "Generative AI-driven Semantic Communication Framework for NextG Wireless Network," *arXiv preprint arXiv:2310.09021*, 2023.
- [2] M. Gain, A. D. Raha, A. Adhikary, K. Kim, and C. S. Hong, "Open ran embracing continual learning: Towards nextg adaptive traffic analysis," in *NOMS 2024-2024 IEEE Network Operations and Management Symposium*, 2024, pp. 1–7.
- [3] F. Deng, W. Mao, Z. Zeng, H. Zeng, and B. Wei, "Multiple diseases and pests detection based on federated learning and improved faster r-cnn," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–11, 2022.
- [4] F. Deng, Z. Zeng, W. Mao, B. Wei, and Z. Li, "A novel transmission line defect detection method based on adaptive federated learning," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–12, 2023.
- [5] A. N. Patel *et al.*, "A trustable federated learning framework for rapid fire smoke detection at the edge in smart home environments," *IEEE Internet of Things Journal*, pp. 1–1, 2024.
- [6] M. Abughazalah, W. Alsaggaf, S. Saifuddin, and S. Sarhan, "Centralized vs. decentralized cloud computing in healthcare," *Applied Sciences*, vol. 14, no. 17, p. 7765, 2024.
- [7] B. McMahan *et al.*, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 54. Fort Lauderdale, Florida, USA: PMLR, 20–22 Apr 2017, pp. 1273–1282.
- [8] Y. Qiao, M. S. Munir, A. Adhikary, H. Q. Le, A. D. Raha, C. Zhang, and C. S. Hong, "MP-FedCL: Multiprototype Federated Contrastive Learning for Edge Intelligence," *IEEE Internet of Things Journal*, vol. 11, no. 5, pp. 8604–8623, 2024.
- [9] A. D. Raha, A. Adhikary, M. Gain, Y. Qiao, and C. S. Hong, "Boosting federated domain generalization: The role of advanced pre-trained architectures," *arXiv preprint arXiv:2409.13527*, 2024.
- [10] Y. Qiao, H. Q. Le, M. Zhang, A. Adhikary, C. Zhang, and C. S. Hong, "Fedccl: Federated dual-clustered feature contrast under domain heterogeneity," *Information Fusion*, vol. 113, p. 102645, 2025.

- [11] Y. Qiao, C. Zhang, H. Q. Le, A. D. Raha, A. Adhikary, and C. S. Hong, "Knowledge distillation in federated learning: Where and how to distill?" in *2023 24th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2023, pp. 18–23.
- [12] Z. Mo, Z. Zhang, and K.-L. Tsui, "Distance Aware Risk Minimization for Domain Generalization in Machine Fault Diagnosis," *IEEE Internet of Things Journal*, 2024, doi:10.1109/JIOT.2024.3441253.
- [13] X. Liu, W. Yu, F. Liang, D. Griffith, and N. Golmie, "Toward Deep Transfer Learning in Industrial Internet of Things," *IEEE Internet of Things Journal*, vol. 8, no. 15, pp. 12 163–12 175, 2021, doi:10.1109/JIOT.2021.3062482.
- [14] M. Gain, A. D. Raha, B. Biswas, A. K. Bairagi, A. Adhikary, and R. Debnath, "Leo satellite oriented wildfire detection model using deep neural networks: A transfer learning based approach," in *2024 6th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT)*. IEEE, 2024, pp. 214–219.
- [15] B. Lyu, H. Yuan, L. Lu, and Y. Zhang, "Resource-constrained neural architecture search on edge devices," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 1, pp. 134–142, 2022.
- [16] A. Gutierrez-Torre *et al.*, "Automatic distributed deep learning using resource-constrained edge devices," *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 15 018–15 029, 2022.
- [17] A. D. Raha, M. Gain, R. Debnath, A. Adhikary, Y. Qiao, M. M. Hassan, A. K. Bairagi, and S. M. S. Islam, "Attention to monkeypox: An interpretable monkeypox detection technique using attention mechanism," *IEEE Access*, vol. 12, pp. 51 942–51 965, 2024.
- [18] R. S. Antunes, C. André da Costa, A. Küderle, I. A. Yari, and B. Eskofier, "Federated learning for healthcare: Systematic review and architecture proposal," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 13, no. 4, pp. 1–23, 2022.
- [19] A. D. Raha, K. Kim, A. Adhikary, M. Gain, and C. S. Hong, "Advancing ultra-reliable 6g: Transformer and semantic localization empowered robust beamforming in millimeter-wave communications," *arXiv preprint arXiv:2406.02000*, 2024.
- [20] M. S. Munir, K. T. Kim, S. F. Abedin, M. G. R. Alam, W. Saad, and C. S. Hong, "Cognitive behavior-in-the-loop: Towards an attentive driving in intelligent transportation systems," *IEEE Transactions on Industrial Informatics*, pp. 1–10, 2024.
- [21] M. Gain, A. D. Raha, and R. Debnath, "Ccc++: Optimized color classified colorization with segment anything model (sam) empowered object selective color harmonization," *arXiv preprint arXiv:2403.11494*, 2024.
- [22] M. Gain *et al.*, "Ccc: Color classified colorization," *arXiv preprint arXiv:2403.01476*, 2024.
- [23] Z. Liu *et al.*, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11 976–11 986.
- [24] S. Woo, S. Debnath, R. Hu, X. Chen, Z. Liu, I. S. Kweon, and S. Xie, "Convnext v2: Co-designing and scaling convnets with masked autoencoders," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 16 133–16 142.
- [25] X. Wu *et al.*, "Ip102: A large-scale benchmark dataset for insect pest recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8787–8796.