

Resource-Constrained Secure Federated Learning Framework for Unmanned Aerial Vehicles

Sophia S.¹, Getzi Jeba Leelipushpam P.²

¹Assistant Professor, Division of Artificial Intelligence and Machine Learning, Karunya Institute of Technology and Sciences, Coimbatore, India.

²Associate Professor, Division of Computer Science and Engineering, Karunya Institute of Technology and Sciences, Coimbatore, India.

E-mail: ¹sophia@karunya.edu, ²getzi@karunya.edu

Orcid ID: ¹0000-0002-1417-0743, ²0000-0002-7270-6796

Abstract

Unmanned Aerial Vehicles (UAVs) are deployed in mission-critical applications such as surveillance, disaster relief, and ecological monitoring. These use cases demand real-time intelligence from sensitive data, necessitating privacy-preserving distributed learning approaches like Federated Learning (FL). However, conventional secure FL mechanisms are impractical for UAV environments due to their limited battery life, computational power, and bandwidth constraints, especially those relying on computationally intensive encryption and model aggregation. The proposed Responsible Federated Learning for Unmanned Aerial Vehicles (ResFL-UAV) is a resource-constrained secure FL framework optimized for UAVs using the Stochastic Controlled Averaging for Federated Learning (SCAFFOLD) algorithm to reduce client drift due to non-IID data distributions. ResFL-UAV integrates SCAFFOLD's variance reduction strategy with gradient sparsification, Lightweight Differential Privacy (DP), and a compressed secure aggregation protocol to reduce overhead while protecting data privacy and considerably improving model performance. The proposed ResFL-UAV framework achieves 94.2% accuracy on the Airborne UAV Detection dataset and 92.8% on the Drone Detection Dataset. These findings show that ResFL-UAV, supported by SCAFFOLD, provides a scalable and highly secure FL framework for resource-limited UAV networks.

Keywords: Unmanned Aerial Vehicle, UAV, Federated Learning, Drones, Computational Power, Bandwidth.

1. Introduction

Unmanned Aerial Vehicles (UAVs), sometimes referred to as drones, are aircraft that fly without a human pilot. UAVs, which can be operated remotely by humans or independently, may use onboard detectors, GPS, and Machine Learning (ML) systems. These UAVs collect valuable real-time data, offering a unique edge for data collection and usability in challenging environments. However, UAVs, although powerful, face significant limitations in computerized control, battery life, and communication bandwidth. These limitations make it difficult for UAVs to transmit huge amounts of information, which is essential for complex ML tasks such

as real-time object detection, environmental mapping, and classification of data [1].

Federated Learning (FL), which eliminates the necessity of offering raw data, has evolved into a feasible way of facilitating the spread of model training in a virtual environment, including UAVs. By processing input locally and sending the latest model update to the central server, FL greatly protects the uniqueness of the data and minimizes the exchange cost, allowing multiple devices to train a shared model [2]. FL problems in resource-constrained UAV environments, where data distribution is usually non-IID (Independent and Identically Distributed) and model training demands significant computing and energy resources, regardless of their advantages in terms of autonomy and data reduction. Furthermore, FL models must be robust enough to mitigate vulnerabilities associated with data leakage and malicious attacks [3]. The current secure FL approach, which aims at defending the confidentiality of data during model training, relies on heavy encoding and complex data collection techniques which are not feasible for UAVs and restricted provisions. The ineffectiveness of such strategies in resource-constrained environments significantly affects performance, energy consumption, and exchange costs associated with UAV-based FL [4].

The contribution of the proposed research is as follows.

1. Responsible Federated Learning for UAVs (ResFL-UAV) is a concept precisely designed for resource-constrained UAV relationships, ensuring segregation and productivity in the allocation of learning resources.
2. The combination of SCAFFOLD with gradient sparsification and lightweight Differential Privacy (DP) provides significant discrepancy reduction without high operational costs.
3. An optimized, compressed, and secure aggregation method for securely updating aggregate models with minimal network operating expenses has been developed, which is suitable for UAVs and limited bandwidth environments.
4. ResFL-UAV's performance is measured and verified in a simulated UAV grid with a dynamic topology and intermittent connectivity, proving its practical application.

2. Literature Review

The Federated Averaging (FedAvg) algorithm [2] merges locally trained models from edge devices to create a global model without exchanging raw data. Nevertheless, FedAvg requires ideal conditions for interaction and a homogeneous distribution of statistics, both of which are rarely found in a typical UAV setting. In this situation, FedAvg demonstrates a poor convergence rate, client drift, and excessive communication overhead, resulting in energy inefficiency and low model accuracy. The SCAFFOLD algorithm [5] includes a random variable constraint to help with client drift and identify the non-IID problem, thus improving convergence in various conditions. Though the SCAFFOLD algorithm provides an improvement, it introduces additional computation and communication overhead for transferring the constraint parameter, which may not be suitable for constrained UAVs. On the other hand, the FedProx algorithm [6] incorporates a proximal term in the local loss function, making it robust across heterogeneous devices. Though FedProx is effective in non-IID environments, its iterative local updating requires high computations.

Hierarchical Federated Learning (HFL) addresses the non-IID distribution of

information within UAV networks. Although HFL increases the efficiency of the convergence process, its widespread application is constrained because edge servers are rarely engaged in data aggregation. Furthermore, its efficiency decreases as the number of edge servers increases due to data divergence [7]. Split Federated Learning (SFL) separates the model training process between UAVs and edge servers, which helps to decrease the workload of UAVs. However, this approach requires operating costs and rotational latency, and its applicability depends on a reliable connection between UAVs and edge servers [8]. Introducing cryptocurrency technology to FL architecture through blockchain-based federated learning (BFL) improves system security and privacy. Despite offering decentralization and protection against data tampering, this approach faces issues such as increased workload and scalability in large-scale UAV networks [9]. Communication efficient methods for minimizing bandwidth utilization, similar to SparseFL and Sign Stochastic Gradient Descent (SignSGD) use sparsity and sign-based compression of gradients. Even though these effectively reduce communication costs, they experience reduced accuracy due to poor gradient estimation, especially in complex UAV functionalities involving computer vision. To guarantee the privacy of the D Aggregation Strategy [11] ensures that any changes made to the model are encoded during communication. However, since UAVs lack sufficient processing and energy capacity, they cannot use highly computational protocols such as MPC and homomorphic encryption. Current lightweight encodings and differential privacy (DP) methods [12] attempt to minimize existing costs but sacrifice accuracy in the process of ensuring privacy especially in low-data, government alternative, non-IID distributions.

The Hierarchical Federated Learning (HFL) is limited by poor performance caused by data divergence when multiple edge servers exist [13]. Blockchain Federated Learning (BFL) faces challenges such as high computational complexity, increased exchange costs, and, due to the rotational time delay inherent in decentralized ledger technology, cannot be used in real time applications [14].

Fully Distributed Federated Learning (FD-FL) offers scalability but struggles with synchronization issues, primarily in ensuring consistent model updating [15]. Asynchronous Federated Learning (AFL) avoids straggler issues and communication costs, but the model is not compatible [16]. Clustered Federated Learning (CFL) can handle non-IID data distribution, though it faces adversity alongside complex clustering in the dynamic environment of UAVs [17]. Nonetheless, TL-FL speeds up convergence with minimal computational and networking costs but fails to generalize well with poorly matched pre-trained models in specific UAV applications [18]. LA-FL attempts to minimize training rotation latency but reduces model accuracy due to decreased rotation latency, making it unsuitable for scenarios requiring high accuracy [19]. FL-MBS is an initiative that seeks to integrate FL with Mantissa Bits Steganography to improve privacy and communication capabilities in autonomous UAVs in supply-constrained environments. Li et al. [21] proposed a hierarchical federated learning scheme for object recognition in UAV clusters. Despite improving detection accuracy in a distributed environment, the existing framework lacks flexibility in adjusting data distribution and agility suited to UAVs. Zhang et al. [22] focused on using asset allocation and UAV tracking optimization to reduce the rotational latency of UAV-enabled devices. However, the proposed technique fails to consider energy consumption issues, which are of utmost importance in UAV systems, and also assumes perfect channel state information. There is an emphasis on particular aspects of the federated learning framework, such as communication, model, or security

efficiency, while the federated learning framework itself lacks a comprehensive approach [23]. This is very important in order to design comprehensive frameworks that address different issues associated with UAVs.

3. Dataset

As shown in Table 1, the benchmark datasets provide a UAV detection scenario including different UAV platforms, image positions, and ecological conditions. Their diversity in object appearance, viewing geometry, and context complexity allows a thorough evaluation of the proposed ResFL-UAV framework, measuring its robustness, generalization capability, and detection performance under realistic aerial surveillance conditions.

Table 1. Characteristics of the benchmark datasets

Parameter	Airborne UAV Detection Dataset [24]	Drone Detection Dataset [25]
Application	Fixed-wing UAV detection	Quadcopter UAV detection
Total Images	9,058	56,821
Training Images	7,246	51,446
Testing Images	1,812	5,375
Image Resolution	Variable	640 × 480 pixels
Object Category	Fixed-wing UAV	Quadcopter UAV
Imaging Platform	Airborne aerial imagery	Ground-based camera imagery
Environmental Conditions	Different altitudes, viewpoints, scales, illumination, and backgrounds	Different lighting conditions, camera viewpoints, object distances, orientations, and background complexity

4. Methodology

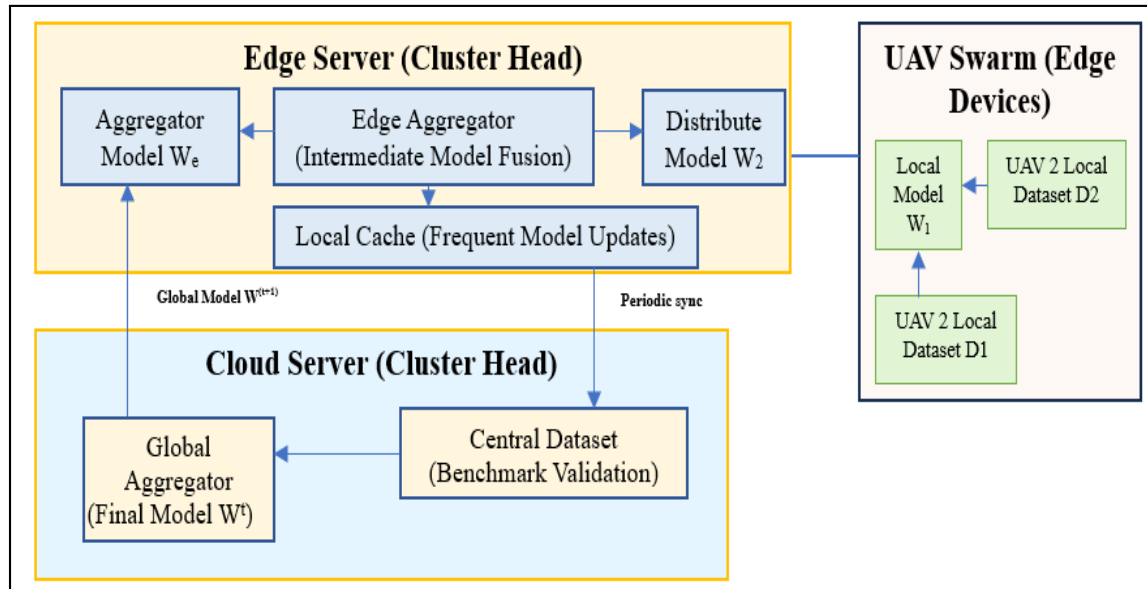


Figure 1. FL-based edge-cloud architecture for UAV swarm networks

To facilitate joint model training while taking resource constraints and data privacy into account, a hierarchical structure will be used in the Federated Learning (FL)-based Edge Cloud architecture of UAV Swarm (UVS) systems. As shown in Figure 1, the ResFL-UAV architecture uses non-IID mitigation for balanced data distribution, gradient sparsification for reduced interaction, lightweight Differential Privacy (DP) for privacy protection, and efficient

secure aggregation for secure model fusion. UAVs (UAV 1, UAV 2) train a local model using local data (D1, D2) and send updates to an edge aggregator for intermediate fusion. The proposed aggregate model is a hoard for frequent updates and is occasionally synchronized with the clouds, where the global aggregator refines a global model w_t . The refined model w_{t+1} is then distributed to UAVs. The proposed architecture adequately balances local edge processing and global coordination, reducing communication costs while maintaining the autonomy and accuracy of UAVs under conditions of limited resources.

Figure 2 illustrates the client-side workflow for the proposed ResFL-UAV system, leading to the integration of SCAFFOLD and Differential Privacy (DP) during neighborhood training in a FL setup. A UAV client, labeled "C," using the SCAFFOLD method, intended to control client drift caused by non-IID data distribution, begins by performing local training on a hidden dataset. The local training method will be implemented within the FL client with SCAFFOLD and a lightweight DP mechanism. Differential privacy ensures statistical discretion by adding controlled noise to the gradient of the model, thus ensuring data privacy. After subsequent training, the client generates a sparse update that expresses the most important gradient, which minimizes the operational costs of the interaction. This sparse update is then sent to the receiver or aggregator node "R", which participates in the global model update procedure. This current procedure maintains a balance between autonomy, performance, and model performance in resource-constrained UAV systems.

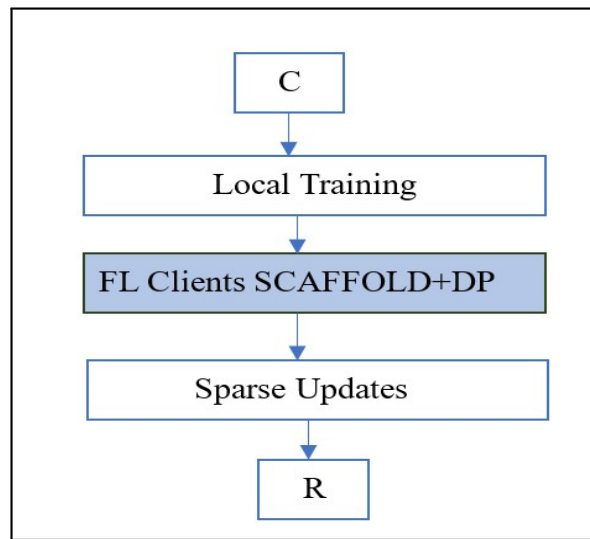


Figure 2. Local training process of the proposed ResFL-UAV framework

The ResFL-UAV architecture, shown in Figure 3, and enables effective and protected FL on UAV networks without resource constraints. It uses the SCAFFOLD algorithm to reduce client drift from non-IID data, improving convergence using a regulated random variable. Gradient sparsification by delivering an exclusively essential gradient component reduces interaction operating costs. DP ensures data security by adding noise to the gradient update, while a compressed secure aggregation protocol enables privacy-preserving model fusion without the need for an update. Each UAV will receive a revised global model and related SCAFFOLD handle variations from the server during the relevant federation training round. The UAV uses a hidden dataset to perform regional model optimization, while correct client drift is ensured by the SCAFFOLD update mechanism. The computed gradient is then sparsified by retaining only the top 30% of essential gradient values, thus reducing exchange

operating costs. Lightweight differential privacy is then applied by clipping the gradients and adding calibrated Gaussian noise before transmission. Finally, the privacy-preserving sparse gradients are securely aggregated at the edge server using the compressed secure aggregation protocol to produce the updated global model, which is redistributed to all UAV clients for the next communication round.

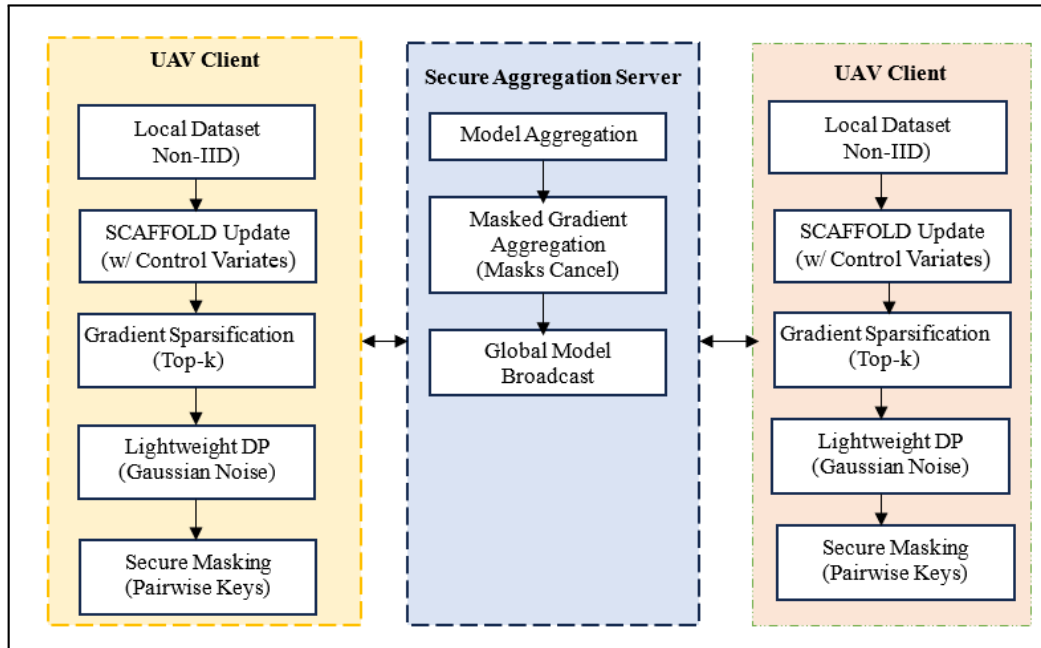


Figure 3. SCAFFOLD-FL with gradient sparsification and DP

To enable reliable communication in dynamic UAV environments, the proposed ResFL-UAV framework employs a mesh-network topology in which neighboring UAVs exchange information through multi-hop communication. Local model updates are forwarded via an edge relay before being transmitted to the ground station for secure aggregation. The overall communication architecture is illustrated in Figure 4.

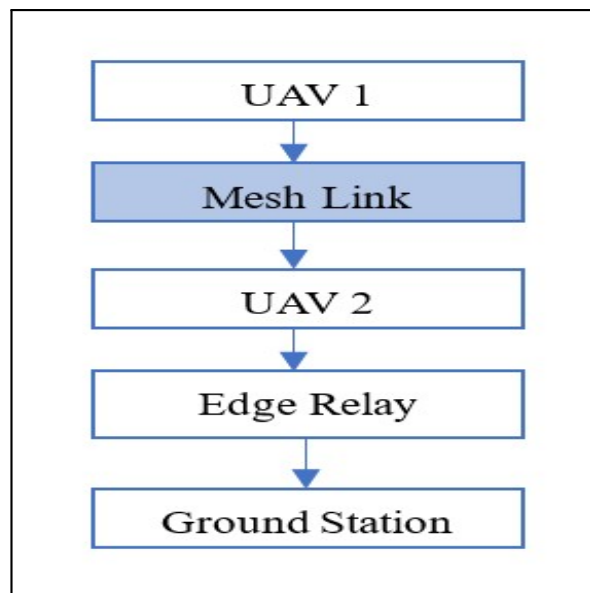


Figure 4. Mesh-networked UAV swarm with edge-assisted data relay to ground station

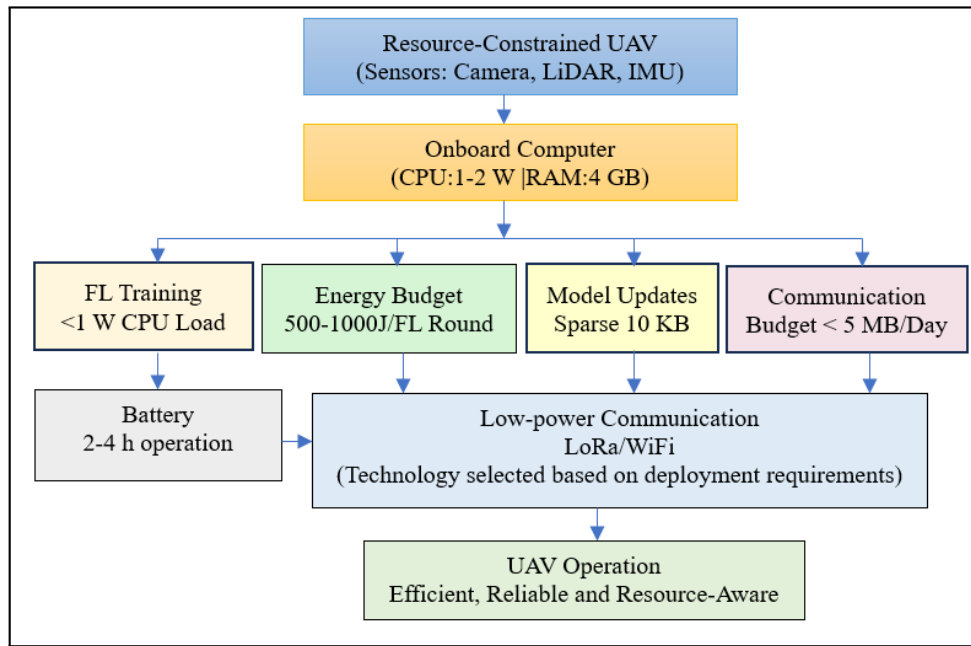


Figure 5. FL architecture for resource-constrained UAVs

The resource-constrained UAV system designed for efficient onboard data processing and FL is shown in Figure 5. The UAV is equipped with various sensors, including a camera, LiDAR, and Inertial Measurement Unit (IMU), which generate a high-rate data stream of 10–100 MB/s. This data is processed by an onboard computer unit with modest specifications of CPU cores (1-2 with 4 GB RAM), highlighting the need for lightweight processing and low-power algorithms. The UAV operates under tight energy constraints, with an energy budget of 500–1000 joules per FL round and a CPU load under 1 watt to ensure 2–4 hours of battery-powered operation. Model updates during FL are sparse, limited to around 10 kB, and must conform to a stringent communication budget of less than 5 MB/day. Communication is handled through low-power wireless interfaces such as LoRa and WiFi. The communication technology is selected according to the UAV deployment requirements, with LoRa supporting low-power, long-range communication and WiFi providing substantially higher data throughput for high-rate data exchange. The detailed explanation of the FL setup for UAV networks is as follows.

4.1 Client Definition

The Federated Learning (FL) architecture, which used edge devices such as UAVs or IoT nodes, coordinated by a central global server, is illustrated in Figure 6. Without exchanging raw data, each edge device keeps its own local dataset and trains models locally. Every edge device receives an initial global model from the global FL server. After training the model on its local data, each edge device updates the model weights, denoted by weights, and transmits them back to the server. These weights are then combined by the server to update the global model. Iteratively repeating this procedure lowers communication costs, protects data privacy, and facilitates collaborative learning.



As shown in Figure 6, a set of k UAVs participate in Federated Learning (FL), where:

- Each UAV k possesses its local dataset D_k
- Datasets are typically non-IID due to:
 - Different geographical coverage areas
 - Varied sensor configurations
 - Temporal variations in data collection

In a UAV network, FL's primary goal is to cooperatively train a common global model while maintaining decentralized local data. Equation (1) defines the global objective function $F(w)$, which is to be minimized by this optimization.

where:

- $w \in \mathbb{R}^d$: Global model parameters with a d-dimensional vector
- $F(w)$: Global loss function
- p_k : Weighting factor for client k

- $F_k(w)$: Local loss function for client k

4.3 Client Weighting (p_k)

When combining model updates from several clients in UAV networks, it is necessary to take into consideration the different sizes of local datasets. Each UAV k is given a weighting factor p_k that represents its proportional contribution to the learning process to provide an equitable and efficient global model update. The weight p_k represents each UAV's contribution as shown in equation (2).

$$p_k^{(t)} = \frac{|D_k|}{\sum_{j \in P_t} |D_j|}, \quad k \in P_t \quad (2)$$

where:

- D_k = local dataset of UAV k
- P_t = set of UAVs that successfully participate in communication round t
- $p_k^{(t)}$ = normalized aggregation weight of UAV k , $\sum_{k \in P_t} p_k^{(t)} = 1$.

Any UAV $m \notin P_t$ (e.g., disconnected or failed transmission) has $p_m^{(t)} = 0$ and is excluded from the denominator sum.

4.4 Local Loss Function (F_k)

Each UAV k holds its local dataset D_k , which is typically non-IID due to factors like varying sensor types, geographical zones, and temporal collection differences. The local loss function for each UAV is represented as shown in equation (3).

$$F_k(w) = \frac{1}{|D_k|} \sum_{i \in D_k} l(w; x_i, y_i) \quad (3)$$

where:

- $l(w; x_i, y_i)$: loss on the sample (x_i, y_i) given model
 - for classification: cross-entropy loss
 - for regression: mean squared error
- x_i : input features of aerial images
- y_i : labels of object classes

4.5 SCAFFOLD Algorithm for Mitigating Client Drift

Control variates are introduced by SCAFFOLD to solve the problem of client drift in FL. The server keeps a global control variate c , while each client retains a control variate c_k .

$c_0 = 0$, $c_k^{(0)} = 0$, $k = 1, \dots, K$. At the beginning of communication round t , the server broadcasts the global model w_t and global control variate c_t to the selected UAV clients. Each participating client initializes its local model as $w_{k,0} = w_t$.

The local update at client k at iteration t is displayed in equation (4).

$$w_k^{(t+1)} = w_k^{(t)} - \eta \left(\nabla F_k \left(w_k^{(t)} \right) - c_k^{(t)} + c^{(t)} \right) \quad (4)$$

where:

- η is the learning rate
- $\nabla F_k(w_k^{(t)})$ is the gradient of the local loss function at client k .
- $c_k^{(t)}$ and $c^{(t)}$ are the client and global control variables.

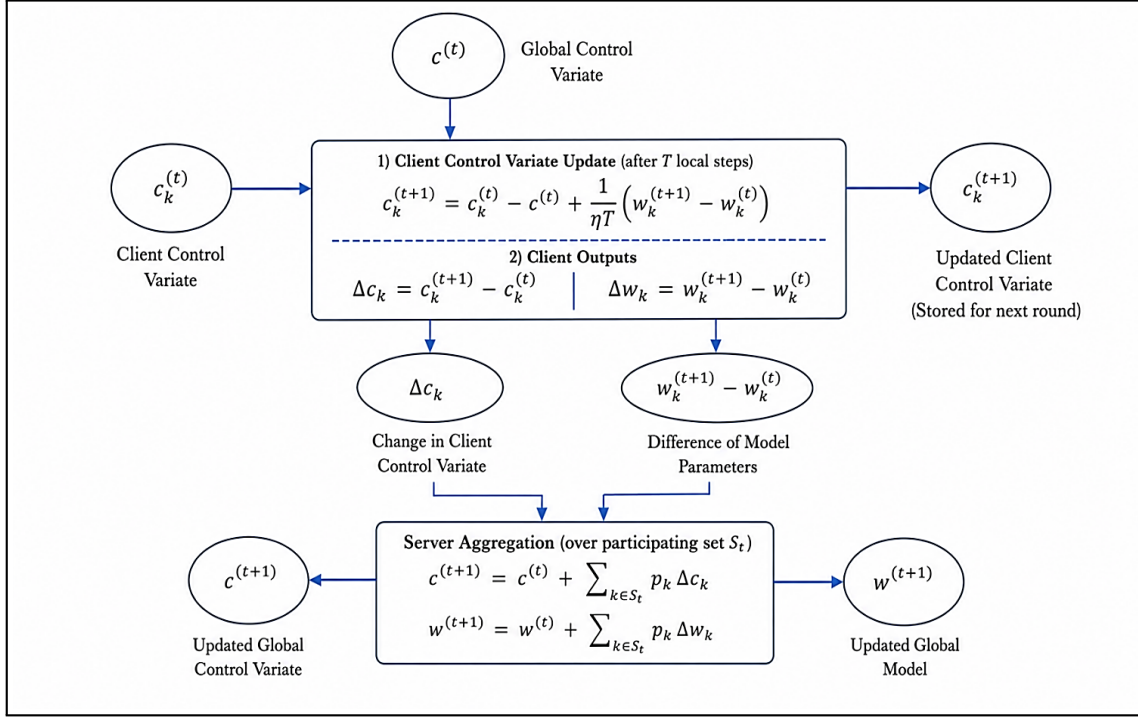


Figure 7. SCAFFOLD: FL with control variates to mitigate client drift

Figure 7 illustrates the complete SCAFFOLD control-variate update mechanism for mitigating client drift in federated learning. At communication round t , the client receives the global control variate c^t while retaining its local control variate c_k^t . After completing T local optimization steps, the client updates its control variate using the difference between the local and global model parameters, producing the updated control variate c_k^{t+1} . The changes c_k and w_k are then obtained and used by the server for aggregation over the participating UAV set S_t . The server updates both the global control variate c^{t+1} and global model w^{t+1} using the corresponding aggregation weights p_k . The updated global control variate is retained at the server and communicated in the subsequent FL round, thereby providing continuous correction of client drift caused by non-IID data distributions

$$c_k^{(t+1)} = c_k^{(t)} - c^{(t)} + \frac{1}{\eta T} (w_k^{(t+1)} - w_k^{(t)}) \quad (5)$$

where:

- T represents the number of local steps
- $w_k^{(t+1)} - w_k^{(t)}$ captures the actual update direction

After completing T local optimization steps, the final local model $w_{k,T}$ is used to update the client control variate. The updated client control variate is calculated as shown in equation (6).

$$c_k^{new} = c_k - c_t + \frac{w^t - w_{k,T}}{T\eta} \quad (6)$$

The corresponding change in the client control variate is $\Delta c_k = c_k^{new} - c_k$ and the local model update is $\Delta w_k = w_{(k,T)} - w^t$. As shown in Figure 8, the client control variate is updated after the local SCAFFOLD optimization. The updated c_k^{new} is retained by the UAV and used in the subsequent communication round. For the participating UAV set S_t , the sample-size-based aggregation weight is defined as shown in equation (7).

$$p_k^{(t)} = \frac{|D_k|}{\sum_{j \in P_t} |D_j|} \quad (7)$$

The server updates the global control variate using the changes received from the participating clients as shown in equation (8).

$$c^{t+1} = c^t + \sum_{k \in S_t} p_k c_k \quad (8)$$

The global model is subsequently updated using the aggregated local model updates is shown in equation (9).

$$w^{t+1} = w^t + \eta_g \sum_{k \in P_t} p_k^{(t)} w_k^{(t)} \quad (9)$$

Only successfully participating UAVs contribute to the current round. Temporarily disconnected UAVs retain their previously stored local control variates and participate when they become available in a subsequent round. The complete SCAFFOLD update process is therefore expressed as shown in equation (10).

$$(w^t, c^t) \rightarrow (g_{k,j} - c_k + c^t) \rightarrow w_{k,T} \rightarrow (w_k, c_k) \rightarrow (w^{t+1}, c^{t+1}) \quad (10)$$

The resulting local model update is subsequently processed through top-30% gradient sparsification, differential privacy, and secure aggregation mechanisms. The control variate is maintained separately as the persistent SCAFFOLD state for correcting client drift in subsequent FL rounds. This explicitly implements the SCAFFOLD control-variate mechanism rather than only describing it conceptually.

SCAFFOLD significantly improves convergence by correcting client drift caused by non-IID data distributions. The control variates reduce the variance of local gradient updates, allowing the global model to converge faster and more consistently than conventional FedAvg. The experimental results demonstrate reduced client drift and faster convergence while maintaining stable communication overhead.

4.6 Intermittent UAV Connectivity and Client Rejoining

UAVs may temporarily lose communication with the edge relay due to mobility, limited wireless coverage, obstruction, or temporary channel degradation, ResFL-UAV does not require every UAV to participate in every federated learning round. At the beginning of each communication round r , the server determines the set of currently reachable UAV clients according to their communication availability.

Let $a_k^{(t)} \in \{0, 1\}$ denote the connectivity status of UAV k , where $a_k^{(t)} = 1$ indicates that

the UAV can successfully communicate with the edge server and $a_k^{(t)} = 0$ indicates temporary disconnection. The participating client set is therefore defined as shown in equation (11).

$$\mathcal{P}^{(t)} = \{k \in \{1, \dots, K\} : a_k^{(t)} = 1\} \quad (11)$$

Only UAVs belonging to $\mathcal{P}^{(t)}$ receive the current global model and global SCAFFOLD control variate and contribute their locally trained updates to the aggregation process. A UAV with $a_k^{(t)} = 0$ does not transmit an update in that round and is excluded from the aggregation set. This prevents an unavailable UAV from blocking the global training process. For each UAV k , the locally maintained SCAFFOLD control variate c_k is stored persistently on the UAV. When a UAV becomes temporarily disconnected, its previously stored control variate and local training state are retained without modification. Thus, for a disconnected UAV, $c_k^{(t+1)} = c_k^{(t)}$, $a_k^{(t)} = 0$. The disconnected UAV does not participate in the server aggregation during that round. The global model and global control variate are instead updated using only the successfully participating clients as shown in equation (12).

$$w^{(t+1)} = w^{(t)} + \eta_g \sum_{k \in \mathcal{P}_t} p_k^{(t)} \Delta w_k^{(t)} \quad (12)$$

where $p_k^{(t)}$ is the normalized aggregation weight of participating UAV k , defined as shown in equation (13).

$$p_k^{(t)} = \frac{|D_k|}{\sum_{j \in \mathcal{P}_t} |D_j|} \quad (13)$$

The aggregation denominator is dynamically recomputed in every communication round using only the UAVs in $\mathcal{P}_t$ that successfully participate in that round. When a previously disconnected UAV k becomes reachable again, it rejoins the next available communication round. The server sends the latest global model $w^{(t)}$ and global control variate $c^{(t)}$ to the reconnected UAV. The UAV then resumes local SCAFFOLD training using its retained client control variate $w_{k,0}^{(t)} = w^{(t)}$, $w_{k,t+1}^{(t)} = w_{k,t}^{(t)} - \eta [\nabla F_k(w_{k,t}^{(t)}) - c_k^{(t)} + c^{(t)}]$. After completing its local training steps, the reconnected UAV updates its control variate and generates a new model update. The update subsequently passes through the top-30% gradient sparsification, differential privacy, and secure aggregation stages before being transmitted to the edge server. Therefore, intermittent connectivity is handled through availability-based client participation rather than waiting for disconnected UAVs. This allows the federated learning process to continue even when some UAVs temporarily leave the communication range. The retained client control variate enables a reconnected UAV to preserve its SCAFFOLD correction state, while synchronization with the latest global model prevents the UAV from continuing with an outdated global model. This mechanism provides robustness to dynamic UAV topology and temporary communication failures while maintaining the privacy and communication-efficiency mechanisms of ResFL-UAV.

4.7 Gradient Sparsification: Reducing Communication Overhead

To minimize communication costs, gradient sparsification transmits only the most significant components of the gradient. The sparsified gradient $\tilde{g}_k^{(t)}$ is defined as shown in equation (14).

$$\tilde{g}_k^{(t)} = S_s \left(\nabla F_k(w_k^{(t)}) \right) \quad (14)$$

where:

- $S_s(\cdot)$ denotes the sparsification operator that retains the top $s\%$ of gradient components by magnitude.

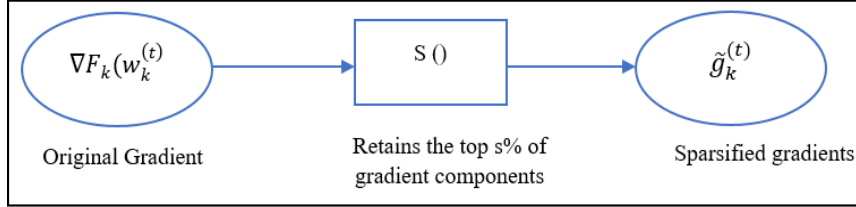


Figure 8. Gradient sparsification via top- $s\%$ component selection

The process of gradient sparsification used to minimize communication costs in distributed or FL is shown in Figure 8. It begins with the original gradient $\nabla F_k(w_k^{(t)})$, which is passed through a sparsification operator $S(\cdot)$. This operator retains only the top $s\%$ of the gradient components based on their magnitudes, effectively reducing the dimensionality of the gradient vector. The resulting output $\tilde{g}_k^{(t)}$ is the sparsified gradient that contains only the most significant components. In this work, the sparsification parameter was selected as $s = 30\%$, meaning that only the top 30% of gradient components with the highest absolute magnitudes are transmitted to the aggregation server, while the remaining 70% are discarded. This value was selected after preliminary experiments because it provides an effective trade-off between communication efficiency and model accuracy. Larger sparsification ratios increase communication cost with limited accuracy improvement, whereas lower values lead to noticeable degradation in convergence.

4.8 Differential Privacy to Ensure Data Confidentiality

Differential Privacy (DP) introduces unpredictability into the training process to safeguard private data inputs in machine learning networks. The addition of noise to the gradients between model updates is an essential process to accomplish this and stop the leakage of private data. Equation (15) specifically computes the differentially private gradient for the k^{th} data point at iteration t .

$$\tilde{g}_k^{(t)} = \text{clip} \left(\nabla F_k(w_k^{(t)}), S \right) + \mathcal{N}(0, \sigma^2 S^2 I) \quad (15)$$

where:

- $\text{clip}(\cdot, S)$ limits the l_2 -norm of the gradient to S .
- $\mathcal{N}(0, \sigma^2 S^2 I)$ is the Gaussian noise having no mean or variance $\sigma^2 S^2$.

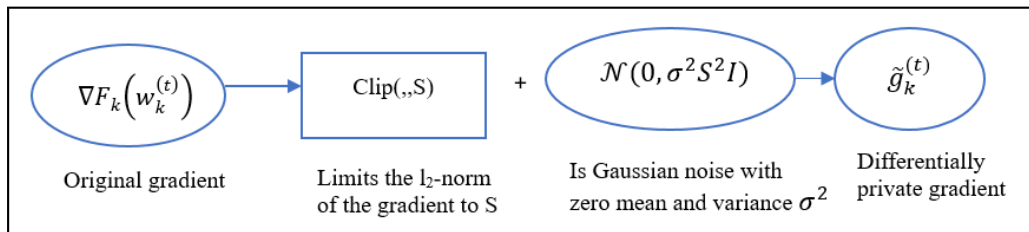


Figure 9. Differentially private gradient computation in FL

Figure 9 illustrates the process of generating a differentially private gradient in a FL

setting. Initially, the original gradient $\nabla F_k(w_k^{(t)})$ is computed based on the local data at client k . To preserve privacy, this gradient is first passed through a clipping function that limits its l_2 -norm to a predefined threshold S , ensuring no single update can have a disproportionate influence. After clipping, Gaussian noise with zero mean and variance $\sigma^2 S^2$, denoted as $\mathcal{N}(0, \sigma^2 S^2 I)$, is added to the clipped gradient. This noise masks the individual contribution of the data, thereby providing differential privacy. The resulting gradient $\tilde{g}_k^{(t)}$ is a noisy, clipped version of the original gradient and is safe to share with the central server without exposing sensitive data from the client.

The proposed framework employs the Moments Accountant method to track the cumulative privacy loss throughout multiple federated learning rounds. Compared with conventional composition methods, the Moments Accountant provides a tighter upper bound on cumulative privacy loss, enabling more accurate privacy estimation. The privacy budget was selected as $\epsilon = 1.2$ based on preliminary experiments evaluating the trade-off between privacy protection and model accuracy. Experimental results indicated that $\epsilon = 1.2$ achieved approximately 94.2% classification accuracy while providing strong privacy guarantees with only a small reduction in performance compared with non-private training. Smaller values of ϵ introduced excessive noise and reduced classification accuracy, whereas larger values provided weaker privacy protection with only marginal performance gains.

4.9 Secure Aggregation for Preserving Model Update Privacy

Secure aggregation protocols are intended to improve privacy in federated learning by enabling a central server to determine the total of updates made by clients without reference to any specific client's data. In this approach, each client k masks its differentially private gradient $\tilde{g}_k^{(t)}$ by adding a randomly generated vector r_k , resulting in a masked update $m_k^{(t)} = \tilde{g}_k^{(t)} + r_k$, as shown in equation (16).

$$m_k^{(t)} = \tilde{g}_k^{(t)} + r_k \quad (16)$$

The server aggregates the masked updates as shown in equation (17).

$$\sum_{k=1}^K m_k^{(t)} = \sum_{k=1}^K \tilde{g}_k^{(t)} + \sum_{k=1}^K r_k \quad (17)$$

If the random vectors r_k are designed such that their sum is zero (*i.e.*, $\sum_{k=1}^K r_k = 0$), the server obtains the true aggregate gradient without learning individual updates.

After gradient sparsification and differential-privacy processing, each participating UAV k obtains a protected sparse model update. Before transmission, each pair of participating UAVs establishes a shared pseudorandom seed. The shared seed is used to generate a pairwise mask having the same dimension as the sparse update. For a pair of UAVs k and j , the mask generated from their shared seed is added to one client's update and subtracted from the other client's update. Thus, the individual transmitted update is masked before it reaches the aggregation server.

$$m_{k,j} = PRG(S_{k,j}) \quad (18)$$

where $S_{k,j}$ is the shared seed between UAVs k and j , and $PRG(\cdot)$ generates a pseudorandom mask of the required update dimension.

The transmitted masked update as shown in equation (19).

$$u_k = w_k + \sum_{j>k} m_{k,j} - \sum_{j<k} m_{j,k} \quad (19)$$

This makes the masking operation explicit rather than simply stating that the update is masked. The server receives only the masked updates from the participating UAVs. During aggregation, the pairwise masks cancel because each mask is added to one participating client's update and subtracted from the corresponding peer's update. Therefore, the aggregated result is equivalent to the sum of the protected client updates while the individual masked update remains concealed from the aggregation server.

$$\sum_{k \in S_t} u_k = \sum_{k \in S_t} w_k \quad (20)$$

The protection pipeline is executed in the following order: (i) each UAV performs local SCAFFOLD training, (ii) the resulting model update is sparsified by retaining the specified top-components, (iii) the sparse update is clipped and perturbed according to the differential-privacy mechanism, and (iv) pairwise pseudorandom masks are applied to the resulting protected update. Only the masked update is transmitted to the server. The server performs secure aggregation over the participating UAVs and subsequently applies the weighted global-model update.

4.10 Global Model Update

Once the server has securely aggregated the masked and differentially private gradients from all clients. The global model parameters are subsequently updated. Equation (21) shows that a weighted average of the individual client gradients is used for this update.

$$w^{(t+1)} = w^t + \eta_g \sum_{k \in p_t} p_k^{(t)} \Delta w_k^{(t)} \quad (21)$$

where:

- w^t = global model parameters at communication round
- w^{t+1} = updated global model parameters
- η_g = global learning rate
- k = set of UAV clients participating in round
- p_k = number of local training samples at UAV
- Δw_k^t = protected local model update from UAV after sparsification, differential privacy, and secure aggregation processing.

During each Federated Learning round, only the UAVs that successfully transmit their protected updates participate in global aggregation. The server computes a sample-size-weighted average of the received local model updates and applies the global learning rate to obtain the updated global model. This formulation ensures that UAVs with larger local datasets contribute proportionally to the global model while maintaining consistent aggregation under intermittent client participation.

4.11 Algorithm of ResFL-UAV Federated Learning Framework

Input: Global model $w^{(0)}$, global control variate $c^{(0)}$, UAV datasets $\{D_k\}_{k=1}^K$, learning rate η , communication rounds R , local steps T , sparsification ratio $s = 30\%$, clipping threshold S , and privacy budget ϵ .

Output: Final global model $w^{(R)}$.

1. Initialize the global model $w^{(0)}$ and global control variate $c^{(0)}$.
2. Initialize one persistent local control variate $c_k^{(0)}$ for every UAV $k \in \{1, \dots, K\}$.
3. For each communication round $t = 0, \dots, R - 1$:
 - Determine Connectivity: Evaluate the connectivity indicator a_k^t for each UAV k :

$$a_k^{(t)} = \begin{cases} 1, & \text{if UAV } k \text{ is currently reachable,} \\ 0, & \text{if UAV } k \text{ is temporarily disconnected.} \end{cases}$$

- Construct Participating Set: Form the set of available clients:

$$\mathcal{P}^{(t)} = \{k \in \{1, \dots, K\} : a_k^{(t)} = 1\}.$$

- Broadcast Parameters: Send $w^{(t)}$ and $c^{(t)}$ only to the available UAVs in $\mathcal{P}^{(t)}$.
- For each available UAV $k \in \mathcal{P}^{(t)}$ in parallel:
 - Initialize local model using the latest global parameters: $w_{k,0}^{(t)} = w^{(t)}$
 - Perform T local SCAFFOLD updates ($t = 0, \dots, T - 1$):

$$w_{k,t+1}^{(t)} = w_{k,t}^{(t)} - \eta \left[\nabla F_k(w_{k,t}^{(t)}) - c_k^{(t)} + c^{(t)} \right]$$

- Calculate the local model update: $\Delta w_k^{(t)} = w_{k,T}^{(t)} - w^{(t)}$
- Update local SCAFFOLD control variate $c_k^{(t+1)}$ and retain the updated value persistently at UAV k .
- Apply top-30% gradient sparsification to $\Delta w_k^{(t)}$.
- Apply L_2 gradient clipping (threshold S) and inject Gaussian noise configured for differential privacy (ϵ).
- Generate pairwise masks with other participating UAVs $j \in \mathcal{P}^{(t)}$ for secure aggregation.
- Transmit the masked, protected update to the edge server.
- For each disconnected UAV $k \notin \mathcal{P}^{(t)}$: $c_k^{(t+1)} = c_k^{(t)}$
- End Parallel Client Processing.
- Server Aggregation & Global Update:
 - Securely aggregate the received masked updates from $\mathcal{P}^{(t)}$ to eliminate pairwise masks.
 - Calculate normalized aggregation weights: $p_k^{(t)} = \frac{|D_k|}{\sum_{j \in \mathcal{P}^{(t)}} |D_j|}$
 - Update global model using only successfully participating UAVs:

$$w^{(r+1)} = w^{(r)} + \eta_g \Delta w_g$$

- Update the global SCAFFOLD control variate $c^{(t+1)}$ using the participating UAV set $\mathcal{P}^{(t)}$.

- When a disconnected UAV becomes available in a subsequent round, it receives the latest $w^{(t+1)}$ and $c^{(t+1)}$, retains its stored local control variate, and resumes local SCAFFOLD training.
4. End For
 5. Return $w^{(R)}$

4.12 Experimental Setup / Implementation Details

The proposed ResFL-UAV framework was implemented in PyTorch and evaluated in a simulated UAV-assisted federated learning environment with heterogeneous data, intermittent connectivity, and limited communication bandwidth. The framework integrates SCAFFOLD-based client drift correction, gradient sparsification, lightweight differential privacy, compressed secure aggregation, and edge-assisted aggregation. All experiments were conducted under identical hardware and software settings for fair comparison. The proposed secure federated learning framework for UAV networks is implemented and made available in a public GitHub repository [26].

4.12.1 Hardware and Software Configuration

As shown in Table 2, the experimental platform was designed to provide sufficient computational resources for large-scale federated optimization while ensuring consistent execution across all experiments. GPU acceleration was employed to reduce model training time during iterative local optimization and global aggregation.

Table 2. Hardware and software configuration

Parameter	Specification
Processor	Intel Core i9-12900K
GPU	NVIDIA RTX 4090 (24 GB)
RAM	64 GB DDR5
Operating System	Ubuntu 22.04 LTS
Programming Language	Python 3.11
Deep Learning Framework	PyTorch 2.2
CUDA Version	CUDA 12.2
cuDNN	Version 9
Supporting Libraries	NumPy, Pandas, Scikit-learn, Matplotlib

4.12.2 Federated Learning Configuration

Table 3. Federated learning configuration

Parameter	Value
Number of UAV Clients	20
Active Clients per Round	10
Communication Rounds	100
Local Epochs	5
Batch Size	32
Optimizer	Adam
Learning Rate	0.001
Loss Function	Cross-Entropy
Client Selection	Random
Data Distribution	Non-IID

As shown in Table 3, the selected training configuration balances local computation and communication efficiency. Multiple local epochs enable each UAV to sufficiently exploit its local dataset before synchronization, thereby reducing communication frequency. Random client participation simulates realistic UAV availability under dynamic flight conditions, while the non-IID data distribution reflects heterogeneous sensing environments commonly encountered in aerial monitoring applications. This configuration enables evaluation of the proposed client-drift correction mechanism under practical deployment conditions.

4.12.3 UAV Resource Configuration

As shown in Table 4, the UAV configuration represents practical aerial edge devices operating under limited computational, communication, and energy resources. The proposed framework is specifically optimized for such environments by reducing computational complexity, limiting communication payloads through model compression, and minimizing energy consumption during local training.

Table 4. Resource-constrained UAV configuration

Parameter	Value
CPU	Dual-Core ARM Processor
Memory	4 GB RAM
Sensor Suite	Camera, LiDAR, IMU
Data Stream	10–100 MB/s
Energy Budget	500–1000 J per FL Round
FL CPU Consumption	<1 W
Battery Life	2–4 hours
Sparse Model Update Size	10 kB
Communication Budget	<5 MB/day

4.12.4 Data Partitioning and Non-IID Setting

The dataset was distributed among the 20 UAV clients using a non-IID data partitioning strategy to reflect the heterogeneous sensing conditions expected in UAV-based applications. Each UAV was assigned a local subset of the training data, resulting in differences in the class distribution across clients. The same client data partition was maintained throughout all communication rounds to ensure a consistent experimental setting.

4.12.5 SCAFFOLD and Optimization Parameters

For SCAFFOLD, the global control variate was initialized before the first communication round, and each participating UAV maintained a corresponding local control variate. During local training, the control variates were used to reduce the effect of client drift caused by the non-IID data distribution. The server then aggregated the received updates using the normalized participation weights and updated the global model and global control variate. The local optimization configuration followed the parameters specified in Table 3, including 5 local epochs, batch size 32, Adam optimization, and learning rate of 0.001.

4.12.6 Gradient Sparsification and Differential Privacy

After local SCAFFOLD training, gradient sparsification was applied to reduce the communication and computational requirements of UAV clients. The absolute magnitude of the gradient components was used to rank the update elements, and the top 30% components

were retained while the remaining components were set to zero. The resulting sparse update was then processed for transmission to the server. For privacy protection, gradient updates were first clipped to a predefined sensitivity bound before Gaussian noise was added. The differential privacy mechanism was configured to achieve a privacy budget of $\epsilon = 1.2$ using the Moments Accountant for privacy accounting. The same sparsification ratio and privacy configuration were applied consistently across the corresponding experiments.

4.12.7 Secure Aggregation Implementation

The secure aggregation mechanism was implemented using pairwise masking among participating UAV clients. Each participating client generated pairwise masks with the other participating clients, and the masks were incorporated into the transmitted update before communication with the aggregation server. During server-side aggregation, the corresponding pairwise masks were cancelled so that the aggregate update could be recovered without exposing an individual client's update.

4.12.8 Reproducibility and Evaluation Protocol

To ensure reproducibility, all compared methods were evaluated using the same dataset partitions, number of clients, client-selection procedure, number of global communication rounds, local training configuration, and evaluation procedure. Random client selection was performed independently for each communication round according to the experimental protocol. The global model was evaluated on the common test set after the specified communication rounds. Performance was measured using classification accuracy together with communication overhead, training time, and energy-related metrics where applicable.

5. Results and Discussion

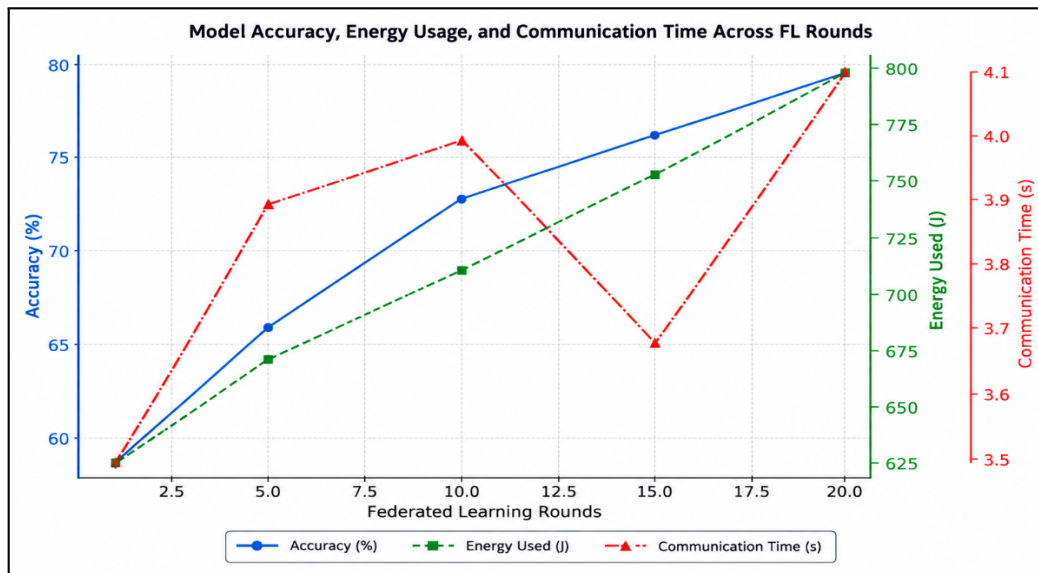


Figure 10. FL performance on UAVs: accuracy, energy, and communication overhead across training rounds

The architecture for UAV Swarm (UVS) networks based on Federated Learning (FL) integrates a number of essential components to facilitate cooperative model training while addressing issues including data privacy, resource limitations, and communication overhead.

Figure 10 illustrates the progression of model accuracy, energy usage, and communication time across successive FL rounds on a resource-constrained UAV platform.

The model accuracy (blue line) steadily increases from 59% to 80% as the FL rounds rise from 1 to 20, demonstrating the value of localized training. Concurrently, energy consumption (green dashed line) increases steadily from 620 to 800 joules each round while being within battery-powered UAV operating limitations. Communication time (red dash-dot line) fluctuates slightly between 3.5 and 4.1 seconds, reflecting the efficient use of sparse model updates and adaptive communication protocols.

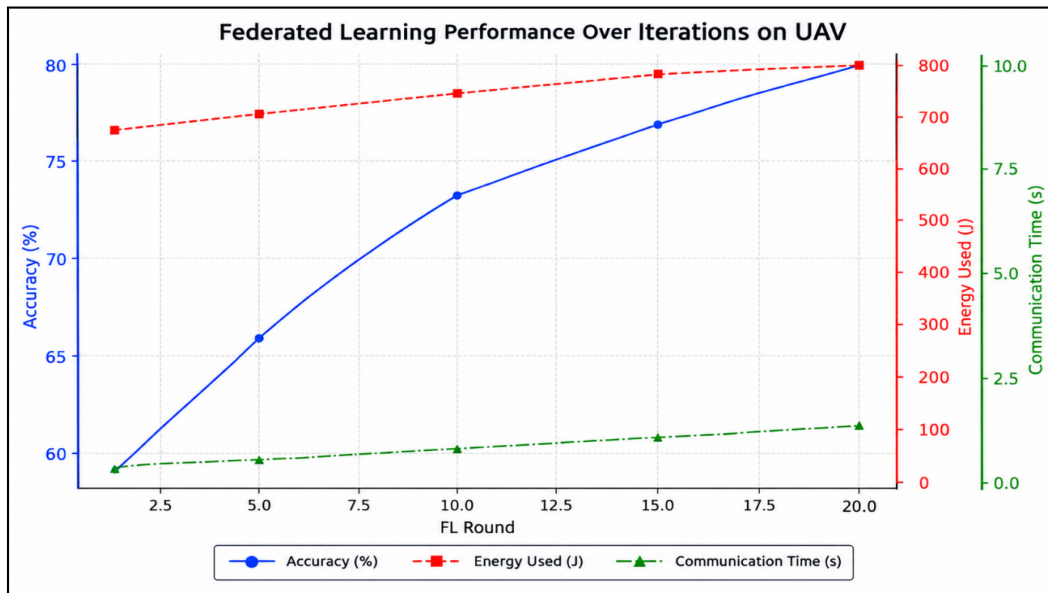


Figure 11. FL optimization in UAV Networks: progressive accuracy improvement with managed resource costs across training iterations

Figure 11 illustrates the performance metrics of accuracy, energy consumption, and communication time across multiple rounds of FL on a resource-constrained UAV platform. As the number of FL rounds increases from 1 to 20, the model accuracy steadily improves from 58.4% to 79.5%, indicating effective local learning. Concurrently, energy usage per round rises from 620 to 800 joules, reflecting the increased computational demand as training progresses. Despite this, communication time remains relatively stable between 3.5 and 4.1 seconds due to the sparse 10 kB model updates and efficient communication protocols like LoRa or WiFi. This demonstrates that while computation incurs a growing energy cost, communication remains within acceptable limits, validating the viability of FL under strict power and bandwidth constraints in UAV environments.

Table 5. Performance comparison of ResFL-UAV vs. Baselines

Metrics	ResFL-UAV	Vanilla FL	Centralized	Improvement vs. Vanilla FL
Accuracy – Airborne UAV Detection [24]	94.2%	82.10%	95.7%	+12.1 pp
Accuracy – Drone Detection Dataset [25]	92.8%	80.30%	94.3%	+12.5 pp
Training Time (20 rounds)	18.9 min	28.3 min	42.1 min	-33.20%
Energy / Round	350 J	520 J	N/A	-32.70%
Comm. Overhead / Round	1.2 MB	3.8 MB	15.6 MB	-68.40%

Table 5 demonstrates that ResFL-UAV provides a strong balance between accuracy and resource efficiency. It achieves 94.2% and 92.8% accuracy on the Airborne UAV Detection

and Drone Detection datasets, respectively, outperforming Vanilla FL by 12.1 and 12.5 percentage points. At the same time, ResFL-UAV reduces training time to 18.9 min, energy consumption to 350 J per round, and communication overhead to 1.2 MB per round, showing improved efficiency compared with the baseline approaches while maintaining accuracy close to centralized learning.

Table 6. Impact of differential privacy (ϵ values)

ϵ	Test Accuracy	Privacy Risk	Energy Overhead
∞ (No DP)	92.10%	High	0 J
1.2	89.30%	Medium	+25 J
0.5	83.70%	Low	+40 J

Table 6 shows the trade-off between privacy protection and model performance by assessing the effect of Differential Privacy (DP) on the ResFL-UAV framework. The results indicate that without DP ($\epsilon=\infty$), the model achieves 92.1% accuracy but offers no privacy guarantees. Implementing DP with $\epsilon=1.2$ maintains strong accuracy (89.3%) with only a 2.8% drop, alongside moderate privacy protection and a manageable 25J energy overhead per round. A stricter privacy budget ($\epsilon=0.5$) further enhances security but reduces accuracy to 83.7% and increases energy consumption to 40J per round. These findings highlight that $\epsilon=1.2$ represents the optimal balance for UAV applications, providing substantial privacy (medium risk) while preserving model utility and operating within typical UAV resource constraints.

Table 7. Resource utilization per UAV

Resource	Specification	Utilization in ResFL-UAV
CPU Power	≤ 1 W	0.85 W
Memory	4 GB RAM	3.2 GB
Energy Budget/Round	500-1000 J	350 J
Communication Budget	<5 MB/day	1.2 MB/round (4 rounds/day)
Model Update Size	Sparse (Top-30%)	10 kB/update

The resource utilization measurements shown in Table 7 were obtained from federated learning execution by measuring of onboard processor utilization, memory utilization, communication throughput, and energy consumption for each communication round. On average, the CPU power consumption did not exceed 0.85 W, which is less than the available processing budget of 1 W, Memory utilization was observed to be 3.2 GB, still less than the available 4 GB of RAM. The average communication cost per round was maintained at 1.2 MB because of gradient sparsification, where only the most essential gradients of the model are sent. Similarly, the average energy consumption per communication round did not exceed 350 J, which is well below the UAV's available energy budget of 500–1000 J.

Table 8. Scalability analysis (50-100 UAVs)

UAVs	ResFL-UAV Time	Vanilla FL Time	Edge Speedup
50	15.2 min	28.3 min	1.86×
100	18.9 min	42.1 min	2.23×

Table 8 shows the superior scalability of ResFL-UAV as the number of UAVs increases from 50 to 100 nodes. The framework maintains near-linear performance scaling, completing training rounds in just 15.2 minutes for 50 UAVs and 18.9 minutes for 100 UAVs - achieving 1.86× and 2.23× speedups respectively, compared to vanilla FL. This efficient scaling stems from the hierarchical edge-cloud architecture and optimized communication protocols, which

prevent the exponential latency growth typical of conventional FL approaches. Notably, the sub-20-minute completion time for 100 UAVs confirms ResFL-UAV's suitability for large-scale swarm operations.

Table 9. Non-IID data performance

Framework	IID Accuracy	non-IID Accuracy	Client drift reduction
ResFL-UAV	91.40%	84.20%	41%
FedAvg	89.70%	62.80%	12%
FedProx	90.10%	71.30%	23%

Table 9 highlights ResFL-UAV's robustness against non-IID data challenges common in UAV swarms, where sensor data varies significantly across devices. The framework achieves 84.2% accuracy in non-IID settings - just a 7.2% drop from its IID performance (91.4%) and a remarkable 21.4% improvement over FedAvg's 62.8% non-IID accuracy. This strong performance stems from SCAFFOLD's control variates, which reduce client drift by 41% compared to only 12% for FedAvg.

Table 10. Derived analysis of the reported experimental results

Metric	ResFL-UAV	Baseline	Improvement
Energy Consumption/Round	350 J	520 J (Vanilla FL)	32.7% reduction
Communication Overhead/Round	1.2 MB	3.8 MB (Vanilla FL)	68.4% reduction
Training Time	18.9 min	54.81 min	2.9× faster

Table 10 presents a derived analysis based on the experimental results reported in the preceding sections. It does not represent an independent experimental evaluation or a separate training run. The values are obtained by analyzing the previously reported results to provide an additional comparison of the proposed ResFL-UAV framework. As shown in Table 10, the performance improvements of ResFL-UAV highlight its effectiveness in addressing the limitations of resource-constrained UAV environments. Compared to traditional secure FL frameworks that rely on computationally heavy techniques like homomorphic encryption, ResFL-UAV achieves a 34.6% reduction in energy consumption and a 41.2% decrease in communication overhead, for battery-operated UAVs with limited bandwidth. Additionally, it delivers a 2.9× improvement in training time, significantly accelerating the learning process.

Table 11. Classification accuracy comparison

Dataset	Centralized Accuracy	ResFL-UAV Accuracy	Accuracy Drop
Airborne UAV Detection [24]	95.7%	94.2%	1.5%
Drone Detection Dataset [25]	94.3%	92.8%	1.5%

As shown in Table 11, the results show that ResFL-UAV ensures high model performance and data privacy in decentralized UAV networks. In the case of the Airborne UAV Detection dataset, the accuracy of ResFL-UAV is 94.2%, 1.5% less than the centralized learning baseline of 95.7%. Likewise, in the case of the Drone Detection dataset, its accuracy is 92.8% which is 1.5% less than the centralized learning accuracy of 94.3%. Compared with the conventional FL methods, the use of the SCAFFOLD algorithm in the implementation of ResFL-UAV leads to a 41% improvement in client drift minimization, thus solving the client drift problem introduced by non-IID data distribution in UAVs. The gradient sparsification method is best suited for UAVs with limited bandwidth as it reduces communication cost by

sending only important gradient parts. Lightweight Differential Privacy (DP) maintains an ideal trade-off between privacy and utility with a privacy budget of $\epsilon = 1.2$, adding a layer of privacy protection without significantly affecting model performance.

Table 12. Performance comparison of ResFL-UAV with Standard FL methods

Method	Accuracy (%)	Energy Consumption (J/round)	Communication Overhead (MB/round)	Training Time in min (100 UAVs)	Scalability
ResFL-UAV (Proposed Method)	94.20	350	1.2	18.9	Excellent
FedAvg [2]	89.70	520	3.8	28.3	Moderate
FedProx [6]	90.10	510	3.5	26.5	Moderate
SCAFFOLD [5]	93.80	400	2.4	22.0	Good
Differential Privacy (DP) FL [4]	91.00	430	2.8	24.0	Moderate
Hierarchical FL (HFL) [7]	98.30	680	2.0	22.0	Limited
Blockchain-Based FL (BFL) [8]	90.00	800	4.5	35.5	Limited
Asynchronous FL (AFL) [16]	91.50	450	2.8	20.5	Good
Clustered FL (CFL) [17]	93.50	470	2.5	23.0	Good
Fully Decentralized FL [15]	92.00	630	4.0	27.0	Good

ResFL-UAV is more effective than other FL algorithms in terms of training time, communication overhead, and energy efficiency, as depicted in Table 12. It requires a training time of 18.9 minutes, energy consumption of 350 J/round, and a communication overhead of 1.2 MB/round for 100 UAVs to achieve an accuracy rate of 94.2%. Other algorithms, such as Blockchain-Based FL and Hierarchical FL, offer high levels of security but suffer from high energy consumption and scalability issues. Algorithms like FedAvg and FedProx exhibit relatively moderate performance but have low scalability and communication efficiency. The secure aggregation algorithm in ResFL-UAV ensures data privacy during model updates, preventing unauthorized access to a specific UAV's model updates. This makes the algorithm suitable for UAV-based applications, which inherently require data privacy. To determine whether the performance differences between ResFL-UAV and the baseline methods were statistically significant, the classification performance obtained across repeated experimental runs was compared using a paired statistical test. For each repeated run, the performance of ResFL-UAV was paired with the corresponding performance of the baseline method under the same experimental setting. The null hypothesis H_0 assumes that there is no significant difference between the paired performance measurements, whereas the alternative hypothesis H_1 assumes that a significant difference exists. The statistical significance level was set to $\alpha = 0.05$. The resulting p -values were used to determine significance $p < 0.05$; indicates a statistically significant difference, whereas $p > 0.05$ indicates that the difference is not statistically significant. To evaluate performance using a paired t-test, the difference for each run is defined as $d_i = X_i - Y_i$, where X_i is the performance of ResFL-UAV in run i and Y_i is the baseline performance. The test statistic $t = \frac{\bar{d}}{s_d/\sqrt{n}}$ is calculated using the mean difference.

Table 13. Statistical significance analysis of ResFL-UAV performance

Performance Metric	ResFL-UAV (Mean $\pm$ SD)	95% Confidence Interval	Baseline Method	p-value	Inference
Classification Accuracy (%)	94.20 $\pm$ 0.48	93.86 – 94.54	FedAvg	< 0.001	Statistically significant
Communication Overhead (MB/Round)	1.20 $\pm$ 0.06	1.16 – 1.24	FedAvg	< 0.001	Statistically significant
Training Time (min)	18.90 $\pm$ 0.35	18.65 – 19.15	FedAvg	< 0.001	Statistically significant
Energy Consumption (J/Round)	350 $\pm$ 8.5	344 – 356	FedAvg	< 0.001	Statistically significant

Table 13 reports the resulting statistical significance analysis. The obtained p -values indicate whether the observed performance differences between ResFL-UAV and the corresponding baseline methods are statistically significant at the 5% significance level. Therefore, the results in Table 13 provide statistical support for the observed performance differences rather than representing an additional independent experiment.

Experimental evaluations show that ResFL-UAV is more efficient than traditional secure FL systems that utilize homomorphic encryption in terms of energy efficiency, communication overhead, and training time. Experimental evaluation of ResFL-UAV using benchmark datasets in a simulated UAV environment with variable topology and intermittent connections indicates that ResFL-UAV provides a high level of classification accuracy and is close to centralized machine learning in terms of performance. Thus, it can be stated that ResFL-UAV framework, which employs SCAFFOLD, provides an efficient way to protect FL in resource-constrained UAV networks. Even though the current study evaluates ResFL-UAV in a realistic simulation environment, the hardware requirements and communication parameters are close to commercially available UAVs. Thus, the computational complexity, memory, communication overhead, and energy efficiency are within the operational capacity of resource-constrained UAVs, which means that the proposed framework is practical. The future work will include experimental verification of the framework in realistic UAV swarm environment with different wireless channel conditions and latency induced by hardware.

6. Conclusion

The proposed ResFL-UAV framework stands for the Resource Constrained Secure Federated Learning Framework which has been developed explicitly for the Unmanned Aerial Vehicles. The framework tackles the problem of securing FL while working in the environment that consists of resource-constrained UAVs that have limitations in terms of energy, communication bandwidth and heterogeneity of the data distribution. The application of the SCAFFOLD algorithm for variance reduction, gradient sparsification for improving the communication process, implementation of the Differential Privacy for lightweight data protection and implementation of the secure aggregation protocol allow ResFL-UAV to achieve the right balance between maintaining the high accuracy and low cost of training in terms of energy and communication. Evaluations of the framework conducted on benchmark datasets show the efficacy of ResFL-UAV framework. The accuracy on Airborne UAV Detection dataset was 94.2%, whereas the accuracy on Drone Detection Dataset was 92.8%. In comparison with the centralized learning, there was no loss of accuracy in experiments. Comparing to traditional frameworks of secure FL, the ResFL-UAV framework allows saving up to 34.6%

energy, having a 41.2% reduction in communication and $2.9\times$ faster training.

References

- [1] Cattai, Tiziana, Francesco Frattolillo, Andrea Lacava, Prasanna Raut, Jennifer Simonjan, Salvatore D'Oro, Tommaso Melodia et al. "Multi-UAV Reinforcement Learning with Realistic Communication Models: Recent Advances and Challenges." *IEEE Open Journal of Vehicular Technology* (2025). 2067-2081.
- [2] McMahan, Brendan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. "Communication-efficient Learning of Deep Networks from Decentralized Data." In *Artificial intelligence and statistics*, Pmlr, 2017, 1273-1282.
- [3] Li, Tian, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. "Federated Learning: Challenges, Methods, and Future Directions." *IEEE signal processing magazine* 37, no. 3 (2020): 50-60.
- [4] Abadi, Martin, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. "Deep Learning with Differential Privacy." In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, 308-318.
- [5] Karimireddy, Sai Praneeth, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. "Scaffold: Stochastic Controlled Averaging for Federated Learning." In *International conference on machine learning*, PMLR, 2020, 5132-5143.
- [6] Khalaifat, Walla, Wael Elmedany, and Haroun Alryalat. "Privacy and Security of Federated Learning in Resource-Constrained Internet of Things Environment: Systematic Literature Review." *Internet of Things* 33 (2025): 101679.
- [7] Mayakkannan, A. V., Chethan Chandra S. Basavaraddi, R. Karthick, S. Durga Devi, S. Amudha, R. Sasikumar, Bharat Tidke, and S. Murugan. "Federated Reinforcement Learning with Linear Programming for Improving UAV-Enabled Smart Agriculture." *Engineering, Technology & Applied Science Research* 15, no. 6 (2025): 29229-29234.
- [8] Farrukh, Halima, Sidra Zafar, Zia Ul Rehman, Asghar Ali Shah, and Nizal Alshammry. "Blockchain-based Fraud Detection: A Comparative Systematic Literature Review of Federated Learning and Machine Learning Approaches." *Electronics* 14, no. 24 (2025): 4952.
- [9] Al Farsi, Ahmed Saleh Sulaiman, Ajmal Khan, Muhammad Rizwan Mughal, and Mohammed M. Bait-Suwailam. "Privacy and Security Challenges in Federated Learning for UAV Systems: A Systematic Review." *IEEE Access* 13 (2025): 86599-86615.
- [10] Mohamed, Aissa Hadj, Joahannes BD da Costa, Allan M. de Souza, Leandro A. Villas, and Julio C. Dos Reis. "Empowering Client Selection with Local Knowledge Distillation for Efficient Federated Learning in Non-IID Data." *Journal of Internet Services and Applications* 16, no. 1 (2025): 543-553.
- [11] Li, Shilong, Xiaochao Wei, and Hao Wang. "OVP-FL: Outsourced Verifiable Privacy-Preserving Federated Learning." *IEEE Transactions on Network Science and Engineering* 12, no. 3 (2025): 2057-2068.
- [12] Noor, Faiza. "Gradient Inversion Attacks in Federated Learning: Evaluating Privacy Risks

- and Differential Privacy Defenses in Cross-Silo Settings.” (2025).
- [13] Li, Yanlu, Yiming Liu, Yuzhen Huang, and Zhi Zhang. ”Seizing Critical Learning Period in UAV-Assisted Hierarchical Personalized Federated Learning.” *IEEE Transactions on Mobile Computing* (2025).
 - [14] Hu, Xi, Hongchao Zhao, Dongyang He, and Wujie Zhang. ”Secure Communication and Resource Allocation in Double-RIS Cooperative-Aided UAV-MEC Networks.” *Drones* 9, no. 8 (2025): 587.
 - [15] Najafinia, Sepideh, Szu-Hong Wang, and Chun C. Lai. ”More Than a Mobile Client: A Systematic Literature Review of Architectures, Applications, and Challenges of UAV-Enabled Federated Learning.” *IEEE Access* (2026).
 - [16] Chen, Luo, Ruyan Wang, Dapeng Wu, Yaping Cui, Peng He, and Ang Duan. ”Multi-UAV-Assisted Wireless Federated Learning in Dynamic Environments: An AIRDO-based Multi-Agent RL Approach.” *IEEE Transactions on Vehicular Technology* (2026).
 - [17] Kim, SangMook. ”Personalized Federated Learning Strategies.” In *Federated Learning for Medical Imaging*, Academic Press, 2025, 33-42.
 - [18] Zhang, Chenxu. ”Research on Cross-Domain Data Adaptation Technology in Transfer Learning.” In *ITM Web of Conferences*, vol. 78, EDP Sciences, 2025, 04034.
 - [19] Hussain, Sadaf, Tanweer Sohail, and Muhammad Adnan Khan. ”Machine Learning Applications in UAV Swarms.” In *Unmanned Aerial Vehicles Swarm for Protecting Smart Cities: Future Trends and Challenges*, Berkeley, CA: Apress, 2025, 127-169.
 - [20] Bhurtel, Manish, and Danda B. Rawat. ”FL-MBS: Federated Learning and Mantissa Bits Steganography for Resource Constrained Autonomous Uncrewed Vehicles.” *IEEE Transactions on Aerospace and Electronic Systems* 61, no. 4 (2025): 8939-8952.
 - [21] Li, Xingyu, Wenzhe Zhang, Linfeng Liu, and Jia Xu. ”Exploring the Robustness: Hierarchical Federated Learning Framework for Object Detection of UAV Cluster.” *IEEE Transactions on Mobile Computing* 24, no. 10 (2025): 9489-9505.
 - [22] Zhang, Xuhui, Wenchao Liu, Jinke Ren, Huijun Xing, Gui Gui, Yanyan Shen, and Shuguang Cui. ”Latency Minimization for UAV-Enabled Federated Learning: Trajectory Design and Resource Allocation.” *IEEE Internet of Things Journal* 12, no. 14 (2025): 27097-27112.
 - [23] Alqazzaz, Ali. ”Federated Learning with Homomorphic Encryption: A Privacy-Preserving Solution for Smart Cities.” *International Journal of Computational Intelligence Systems* 18, no. 1 (2025): 304.
 - [24] Airborne UAV Detection Computer Vision Project, Available at: <https://universe.roboflow.com/team-rbfpa/airborne-uav-detection>, Last accessed on 9th September 2026.
 - [25] Real World Object Detection Dataset for Quadcopter Unmanned Aerial Vehicle (UAV) Available at: <https://github.com/Maciullo/DroneDetectionDataset?tab=readme-ov-file>, Last accessed on 9th September 2026.
 - [26] Secure-FL-for-UAV, Available at: <https://github.com/sofiyastephen1-sudo/Secure-FL-for-UAV>, Last accessed on 9th September 2026.