

# Joint Self-Organizing Maps and Knowledge-Distillation-Based Communication-Efficient Federated Learning for Resource-Constrained UAV-IoT Systems

Gad Gad<sup>ID</sup>, *Graduate Student Member, IEEE*, Aya Farrag, *Graduate Student Member, IEEE*,

Ahmed Aboulfotouh, *Member, IEEE*, Khaled Bedda, *Member, IEEE*,

Zubair Md. Fadlullah<sup>ID</sup>, *Senior Member, IEEE*, and Mostafa M. Fouda<sup>ID</sup>, *Senior Member, IEEE*

**Abstract**—The adoption of Internet of Things (IoT) and monitoring devices in 5G and beyond networks has been widespread. Unmanned aerial vehicles (UAVs) have shown success in connecting rural and remote areas due to the high cost of deploying infrastructures like cellular network base stations and optical fiber connections in vast landscapes with sparse populations. The constrained energy of UAVs results in limited coverage area and flight time, which in turn reduces the potential of UAVs to provide task-oriented wireless communication links. In this article, we explore path optimization and transmission organization algorithms to minimize flight time and extend the range of UAVs performing collaborative federated learning (FL) among geographically dispersed nodes communicating through wireless connections offered by UAVs coupled with device-to-device (D2D) networks. The UAV orchestrates FL between spatially scattered homes via long-range radio wireless communication. We formulate the drone path optimization as a traveling salesman problem (TSP) and employ self-organizing maps (SOM) for path planning. Additionally, knowledge distillation (KD)-based FL is used to reduce communication overhead for the resource-constrained UAV-IoT system. Experimental results demonstrate SOM’s ability to represent the topological structure of nodes and produce a cost-efficient Hamiltonian cycle, from which the drone path is derived. Our results demonstrate the communication efficiency and utility of KD-based FL compared to model-based FL methods. The proposed hybrid solution enables energy-constrained UAVs to perform FL over large areas leveraging a shared data set for KD and a SOM-based path optimization algorithm.

Manuscript received 12 September 2023; revised 27 November 2023; accepted 21 December 2023. Date of publication 2 January 2024; date of current version 25 April 2024. This work was supported in part by the New Frontiers in Research Fund (NFRF) Explore under Grant NFRFE-2019-00313, and in part by the Natural Sciences and Engineering Research Council of Canada under Discovery Grant RGPIN-2020-06260. (*Corresponding author: Mostafa M. Fouda*)

Gad Gad and Zubair Md. Fadlullah are with the Department of Computer Science, Western University, London, ON N6G 2V4, Canada (e-mail: ggad@uwo.ca; zfadlullah@ieee.org).

Aya Farrag, Ahmed Aboulfotouh, and Khaled Bedda are with the Department of Computer Science, Lakehead University, Thunder Bay, ON P7B 5E1, Canada (e-mail: afarrag@lakeheadu.ca; aboulof@lakeheadu.ca; kbedda@lakeheadu.ca).

Mostafa M. Fouda is with the Department of Electrical and Computer Engineering, College of Science and Engineering, Idaho State University, Pocatello, ID 83209 USA, and also with the Department of Electrical Engineering, Faculty of Engineering at Shoubra, Benha University, Cairo 11629, Egypt (e-mail: mfouda@ieee.org).

Digital Object Identifier 10.1109/JIOT.2023.3349295

**Index Terms**—Federated learning (FL), human activity recognition (HAR), knowledge distillation (KD), self-organizing map (SOM), unmanned aerial vehicle (UAV) path optimization.

## I. INTRODUCTION

WITH the emergence of digital technologies, such as wireless communication networks, the Internet of Things (IoT) [1], and low-power wide-area networks (LPWANs), there has been significant progress in the realm of edge device connectivity with a myriad of applications [2]. For instance, with the rapid proliferation of off-the-shelf monitors and wearable devices, IoT-enabled home monitoring for collecting human activity for tracking wellness appeared as a popular application among urban users. However, in rural and remote areas, IoT-based data offloading remains difficult due to inadequate communication infrastructures since mobile operators and network service providers often find it challenging, as well as expensive, to invest in building robust network infrastructures in such areas. Satellite-based continuous data offloading also is impacted by cost and harsh weather conditions in many rural areas. As a viable solution, recent researchers considered the incorporation of aerial networks with unmanned aerial vehicles (UAVs) [3], [4], [5] to offload IoT data from users at locations with limited Internet infrastructures, and thus avoid placing more burden on the already fragile terrestrial network systems in those areas. This article investigates flight time minimization to enhance UAVs ability in IoT applications in which UAVs orchestrate communication and schedule transmission among geographically dispersed nodes as part of a distributed learning task, especially in the development of advanced health monitoring systems in areas with limited Internet infrastructure.

Long-range radio (LoRa), a communication technology under the LPWAN umbrella, facilitates the transfer of data over long distances with minimal power consumption among edge devices. It is a cost effective and efficient option for multiagent IoT applications, despite having a low-bandwidth rate [6]. The use of UAV-assisted LoRa systems is on the rise in various applications, such as health monitoring in rural

areas [7], package delivery [8], precision agriculture [9], localization [10], and forestry monitoring [11]. While UAVs present many possibilities for IoT applications, they are not without shortcomings, e.g., providing a stable aerial communication link between sensor nodes and drones, network planning, limited battery-life, and trajectory optimization [12], [13], [14].

In this article, to address the challenge of providing healthcare monitoring in rural areas with limited Internet connectivity, we propose a drone-aided LoRa (DORA) system, which studies path optimization and node transmission scheduling of a Drone performing federated learning (FL) via a LoRa wireless communication link. In the proposed system, the drone acts as a server orchestrating FL across nodes, where the task is to guide the drone on a near-optimal path to complete a FL task [15], [16], [17], [18] in which dispersed nodes collaborate to train local machine learning models on distribute data without raw data or model weight sharing, leveraging knowledge distillation (KD).

We first analyze the flight time dividing it into waiting and moving time. We then write the moving time in terms of the distance covered by the drone formulating the optimization problem as a modified traveling salesman problem (TSP), and employing self-organizing map (SOM) for efficient path planning. This ensures that the UAV follows a path that covers all network nodes while minimizing the covered distance and flight time. On the other hand, to minimize the waiting time, in which the drone waits for uploading/downloading of FL packages, our conceptualized FL involves the development of a lightweight collaborative algorithm that operates on the edge computing devices in the network. In this vein, we propose a KD-based FL algorithm to leverage soft labels (SLs) calculated on a shared data set employed for knowledge transfer across nodes with significantly less communication overhead compared to that achieved by model-based FL methods. The traffic model of the KD-based FL is presented and compared with the traffic model of the standard FL algorithm (e.g., FedAvg).

To ensure the reliability of communication between the user-homes and UAVs acting as aerial access points, we develop a window-based algorithm similar to TCP/IP, called the reliable data transfer (RDT) layer, to ensure that data are transmitted from homes to UAVs without packet losses.

The main contributions of our work are as follows. We propose the DORA network system composed of two sequential parts: 1) path planning and 2) communication-efficient FL. First, DORA guides the drone through the participant nodes following a distance-efficient path derived from the Hamiltonian cycle calculated using SOM. Second, DORA leverages a shared proxy data set to distill local knowledge learned by the local model across nodes to perform KD-based FL. The objective of the proposed system is to enhance the capability of drones to function as servers in FL and edge analytics applications in rural areas lacking adequate connectivity.

The remainder of this article is organized as follows. Section II describes the relevant research work on the areas of UAV-IoT networks, path planning and optimization, and communication-efficient FL algorithms. Section III presents the considered UAV-IoT system model. Section IV formulates the drone path optimization problem as a variant of TSP and details our approach to find the optimal path using SOM. Next,

Section VI discusses implementation details, the LoRa propagation model, and our proposed RDT protocol. In Section V, an overview of model-based FL algorithms followed by our proposed KD-based FL algorithm is presented. Section VII discusses the experiments and performance evaluation, including path optimization using SOM, data preprocessing splitting details, and the baseline FL algorithms considered. The experiments conducted to evaluate the proposed approach are presented and discussed in Section VII-D. Section VIII highlights the limitations of the proposed approach and suggests future work. Finally, Section IX concludes this article by summarizing the contributions of the proposed DORA network for a communication-efficient FL algorithm.

## II. RELATED WORK

In this section, we present the related work on UAV-IoT systems with a focus on LoRa-based communication and FL, respectively, to understand the state-of-the-art and the existing research gap as the motivation behind the research in this article.

### A. UAV-IoT Systems Employing LoRa Communication

Recently, the idea of UAV-enabled LoRa network systems was successfully applied for many applications, such as agriculture [9], healthcare [19], [20], and tracking air quality [21], [22], [23]. For instance, researchers in [24] designed a UAV-based onboard relay and ground intelligent terminal for environmental monitoring in remote areas. However, in their study, the LoRa module was adopted only to wake up the high-power 5-GHz module which was the main communication technology employed for data collection. The work in [25] employed a similar infrastructure for livestock monitoring in rural farms, while other researchers applied it to remote sensing, i.e., forestry monitoring and precision agriculture [11], [26], [27].

For healthcare applications, UAVs provide a feasible solution for healthcare delivery and other health-related services to individuals in difficult-to-reach locations [28], [29]. Using UAVs in health care reduces emergency response time [30] and improves health access to remote regions. While the low-operation cost for UAVs is tempting, their deployment still faces challenges, such as the limited battery life and payload capacity; stability in intense weather conditions; and accident risks mandating adequate training for the UAV operating and maintaining personnel.

In this work, we leverage UAV-LoRa-based device-to-device (D2D) networks to perform edge health analysis in which we aim to train distributed machine learning models on human activity recognition (HAR) using a UAV as a server.

In the domain of delivery and transportation systems, the work in [10] adopted drones as mobile gateways in LoRa networks for products localization and tracking, whereas [8] deployed LoRaWAN as a secondary communication mode for drone delivery systems. Researchers in [31] exploited the LPWAN system, while including UAVs as on-demand nodes, to consume lower energy and provide wide-range surveillance in intelligent transportation systems (ITSs). Other applications leveraging similar UAV-IoT systems include live

TABLE I  
MATHEMATICAL TERMS AND NOTATIONS

Symbol	Description	Symbol	Description
$TFT$	Total Federated Learning Time	$FL$	Federated Learning
$SOM$	Self Organizing Maps	$KD$	Knowledge Distillation
$R_d$	data rate bits/s	$V^d$	speed of drone m/s
$d_{i,j}$	distance between nodes i and j	$N_v$	Number of vertices in the graph
$\mathbf{P}^d$	trajectory of the drone	$T_L$	time of learning phase of FL (in seconds)
$\delta$	duration of a time slot (in seconds)	$N_t$	Number of time slots
$e_{i,j}$	edge between nodes i and j	$p_n^d$	Drone coordinates at time slot n
$T_M$	Moving time of the drone	$T_W$	Waiting time of the drone
$N_k$	number of user homes	$N_t$	Number of time slots
$T_{USL}$	time of upload phase	$T_{DSL}$	time of download phase
$R$	number of FL rounds	$p_k$	coordinates of the kth household
$H^d$	the altitude of the drone	$H$	Hamiltonian cycle
$G^d$	near-optimum drone path derived from $H$	$D_k$	private dataset of user k
$D_p$	public dataset used for KD	$i_{win}$	index of the winner node
$W_{win}$	Weights of the winner node	$KN(I, x)$	distance between $i^{th}$ row in the SOM network and x
$AP_k$	Home access point for user $k$	$AP^d$	Drone access point (server)
$x_{i,j}$	binary decision variable	$\theta^r$	Weights of server-controlled model at round $r$
$\theta_k^r$	Trained weights for client $k$ at round $r$	$SL$	Soft Labels
$D_p$	Proxy dataset	$f_k$	Local model architecture for client $k$
$\mathcal{L}_{CCE}$	Categorical Cross Entropy function	$N_k$	Number of clients
$CS_k^r$	Compressed soft labels for client $k$ at round $r$	$CS^r$	Aggregated compressed soft labels at round $r$

broadcasting [32] and disaster management [33]. However, to the best of our knowledge, our work in this article is the first research effort to systematically design a UAV-assisted LoRa-based IoT network system with an aim to fulfill the communication requirements of healthcare service provisioning to users in rural/remote areas.

Although the utilization of DORA networks has been demonstrated to be useful in various use-cases, there were some issues identified by researchers while designing these networks. For example, to minimize the energy consumption of end devices for packet transmission, the research work in [34] devised a method to jointly optimize the 3-D UAV trajectory, scheduling strategies, and transmission parameters of edge user devices. In order to reduce the UAV flight distance and time, researchers of [35] introduced optimization methods for cluster heads selection and path planning. Their work demonstrated the viability of teaching–learning-based genetic algorithm (GA) for path planning that accounted for a significant reduction of the UAV flight distance (by 30%) compared to the baseline GA. They also introduced an adaptive data rate strategy to reduce the data transmission time. Another question that arose with implementing this type of network was how close the LoRa-enabled drone needs to maintain flight over the LoRa-enabled sensing devices to collect data with a given quality of data collection. To address this point, attempts were made in [26] and [36] to estimate the optimal UAV deployment height. Based on these findings, there are two key challenges in such UAV-IoT network systems, namely, 1) the UAV path optimization and 2) the reliability of the data transfer between LoRa-enabled devices.

### B. Federated Learning in Human Activity Recognition IoT Systems

In the remainder of this section, we investigate the current progress of distributed learning in IoT systems. Among various

distributed learning paradigms, FL, commonly referred to as FL, is a machine learning framework for distributed model training where each client keeps her data private and only shares model updates [37], [38], [39], [40]. In standard FL setups, these updates, which could be weights or gradients, are aggregated by the server into global weights prior to their broadcast to clients to replace the local model weights/gradients and train for a few more epochs [41], [42], [43].

Wearable HAR systems produce a significant amount of data from different sensors that need to be locally analyzed via collaborative learning. In this vein, FL has been applied to wearable HAR systems to train these systems on distributed data as they are generated on edge devices without the need to outsource them to a remote server/cloud to perform central training. Moreover, existing FL algorithms consider fine-tuning and transfer learning strategies to improve the accuracy of clients on their local data. For instance, Zeng et al. [44] proposed an approach for automatically extracting discriminative features for HAR using a convolutional neural network (CNN). The work in [45] presented FLAME, a user-centered FL algorithm for the multidevice environment (MDE), in which a user may own multiple devices producing data and participating in FL. Applying FL to MDE is more challenging since data are usually not independently and identically distributed (non-IID). In real-world HAR system deployment, it is hard to access sufficient labeled data to train high-quality models; therefore, semi-supervised learning has been employed to address the issue of labeled data scarcity. Presotto et al. [46] presented FedHAR, an FL method, which integrates semi-supervised learning to annotate the stream of unlabeled sensor data in a semi-automatic fashion. FedHAR also employs transfer learning to personalize the global model for each client. Adopting FL as a distributed learning paradigm faces several challenges, such as susceptibility to data and system heterogeneity; and large communication overhead. In

this work, we employ the KD mechanism in the context of FL to distill knowledge across FL clients in the form of SLs. This model-agnostic approach has many advantages over model-based FL algorithms in which model parameters are shared, including communication efficiency and an independently designed local model. On the other hand, in model-based FL algorithms, a global model architecture is imposed on all participant clients.

Perhaps the work by Donevski et al. [17] is the most relevant to our research since it studies the problem of path optimization of a drone-aided network to complete a FL round within a predefined deadline in order to achieve fairness in terms of minimizing staleness. The work compared two algorithms to optimize the drone trajectory: 1) reinforcement learning (RL) and 2) successive convex programming (SCP). The primary optimization criterion used by [17] to minimize the learning discrepancy between nodes by controlling the drone trajectory is given by an average-anchored staleness (AAS)

$$\max_{\mathbf{P}^d} \frac{1}{K} \sum_{k=1}^K T_k(\mathbf{P}^d) - s(\mathbf{P}^d) \quad (1)$$

where  $T_k(\mathbf{P}^d)$  denotes the number of epochs that client  $k$  learns each round,  $\mathbf{P}^d$  indicates the drone's path which is being optimized, and  $s(\mathbf{P}^d)$  refers to the maximum learning discrepancy between any two clients given by

$$s(\mathbf{P}^d) = \max(|T_k(\mathbf{P}^d) - T_l(\mathbf{P}^d)|) \quad \forall k \neq l; k, l \in K. \quad (2)$$

### III. CONSIDERED SYSTEM MODEL

In this section, we present our considered UAV-IoT system model designed to offload HAR data from wearable user devices to be locally analyzed by lightweight collaborative learning.

We consider a target area (TA) represented by an  $M \times M$  map, containing  $N_k$  households, each denoted by  $k \in \mathcal{K} = \{1, 2, \dots, N_k\}$ , with home  $k$  having spatial coordinates  $\mathbf{p}_k = (x_k, y_k)$ . The map is divided into  $M^2$  blocks, each with a side length of  $s$  meters, resulting in a total side length  $S = M \cdot s$ . Each household has an access point  $AP_k$ , which consists of a microcontroller (e.g., Raspberry Pi board) and an IoT interface (e.g., LoRa module). Each  $AP_k$  connects to health-monitoring wearable devices via Bluetooth low energy (BLE) to extract sensory data and train a deep learning model in a distributed manner using a lightweight, communication FL algorithm. Table I lists all the symbols used in this article with their corresponding descriptions.

We define a circular coverage area (CA)  $CA_k$  with radius  $r = (s/2)$  centered at the coordinates of each household  $\mathbf{p}_k \forall k \in \mathcal{K}$ .

The drone access point  $AP_d$  can communicate with the access point of the  $k$ th user  $AP_K$  only if the drone is within the CA of that access point, i.e., the distance between  $AP_d$  and  $AP_K$  is below  $r$ . Therefore, the data rate of the LoRa wireless link between the drone and the  $k$ th node,  $Dr(k, \mathbf{p}_n^d)$  can either be zero, if the drone is outside the CA of the node, or a constant

TABLE II  
USED LoRA, DISTANCE, AND RDT PARAMETERS IN OUR EXPERIMENTS

Parameter	Value
Spread Factor	12
Bandwidth	125 KHz
Carrier frequency	915 MHz
Coding rate	4/5
Programmed Preamble	7
LoRa engine	SX1276
CA radius	50 meters
VD	8 meters
Delay	3 seconds
Window size	3 packets
Timeout	3 seconds

denoted as  $R_b$ , if the drone is inside the CA of the node it is communicating with.  $Dr(k, \mathbf{p}_n^d)$  is given by

$$Dr(k, \mathbf{p}_{d,n}^d) = \begin{cases} R_b, & \text{if } d(k, \mathbf{p}_n^d) \leq r \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

Here,  $Dr(k, \mathbf{p}_n^d)$  represents the data rate of the drone at point  $\mathbf{p}_n^d$  of its trajectory  $\mathbf{P}^d$ , and  $R_b$  is the nominal data rate according to the considered LoRa propagation model presented in Section VI-B. The data rate is a function of  $\mathbf{p}_n^d$  because as the drone moves horizontally across the map, the data rates change.

The UAV, which acts as the FL server, travels, equipped with an access point  $AP_d$ , across the TA commencing from point  $A$  at  $\mathbf{p}_A = (x_A, y_A)$ , and passing through each household, waiting to upload and download data, before returning to the starting point. The UAV flight duration can be therefore divided into rounds corresponding to FL rounds. Each round consists of two parts, namely, the upload SLs (USLs) phase and the download SLs (DSLs) phase, as depicted in Fig. 1. We set the radius of CA  $r = 50$  m based on our drone-LoRa distance test shown in Fig. 2 using the parameters specified in Table II.

The Round Time  $T$  is the time the drone takes to perform one round of FL and it is given by

$$T = T_{USL} + T_{DSL}. \quad (4)$$

We divide  $T$  into  $N_t$  equal time slots  $n \in \{0, 1, 2, \dots, N_t - 1\}$ , each lasting  $\delta$  seconds, such that  $T = N_t \cdot \delta$ , and  $T_{USL} = T_{DSL} = (N_t/2) \cdot \delta$ . The drone follows a trajectory  $\mathbf{P}^d = \{\mathbf{p}_n^d\} \in \mathbb{R}^{N_t \times 2} \forall n \in \{1, 2, \dots, N_t\}$ , a sequence of coordinates where each element  $\mathbf{p}_n^d$  represents the drone's coordinates at time slot  $n$  while it is performing FL across nodes. The first and last points have the same coordinates:  $\mathbf{p}_A = \mathbf{p}_0^d = \mathbf{p}_{N_t-1}^d = (x_A, y_A)$ . The horizontal distance between the drone and node  $k$  is given by

$$d(k, \mathbf{p}_n^d) = \sqrt{(x_n^d - x_k)^2 + (y_n^d - y_k)^2}. \quad (5)$$

We denote the horizontal speed of UAV as the constant  $V^d$ . We also assume the altitude of the drone to be a constant given by  $H^d$  since in this article we only focus on path planning in the 2-D plane containing the map of the TA.

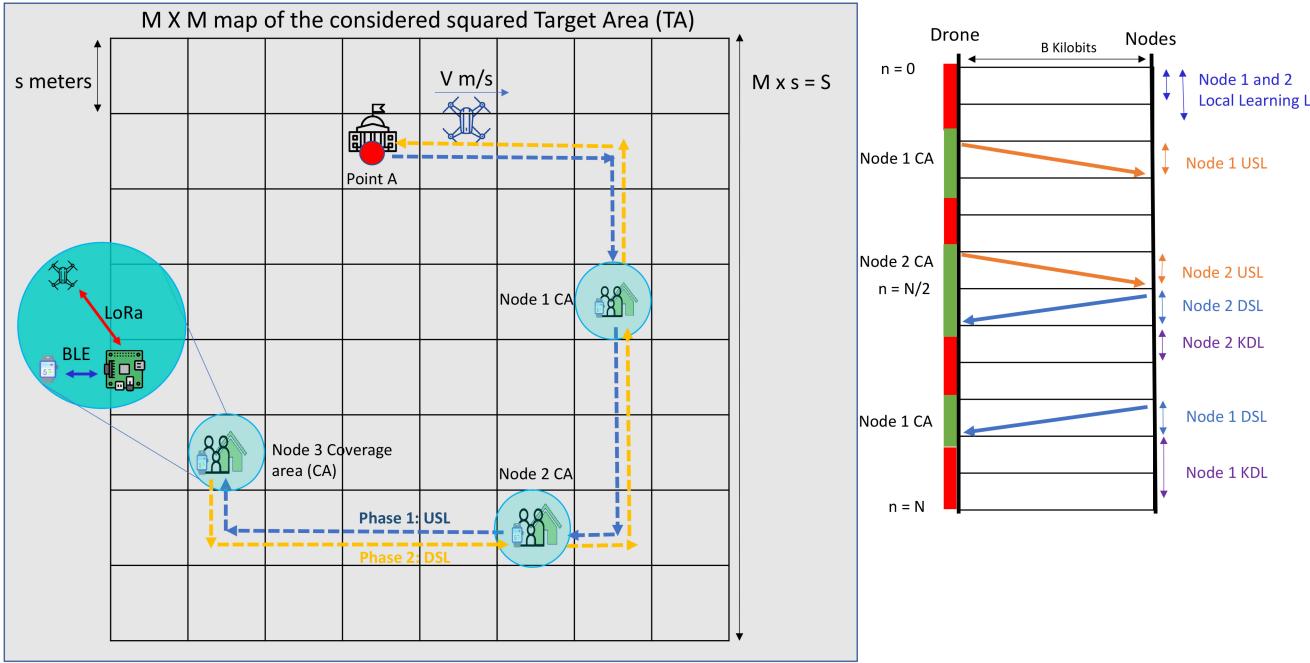


Fig. 1. Left: The drone traverses an optimized path while performing KD-based FL over two passes across the considered TA. The drone communicates lightweight messages via LoRa link only if it is inside the CA. Right: Timeline of the FL communication phases for a map of one drone and two nodes.

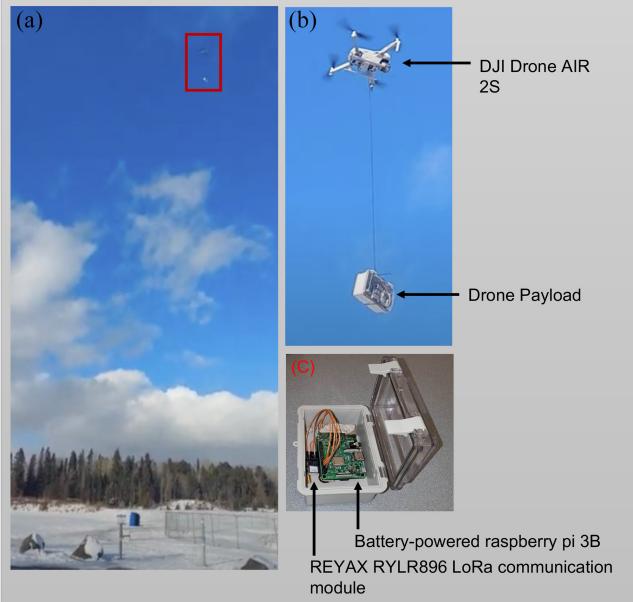


Fig. 2. Drone access point  $AP^d$  LoRa communication with house access point  $AP_i$  inside the CA.

Considering the limited battery life of the drone, we aim to minimize the drone flight time while it is performing FL denoted as total FL time (TFT) given by

$$\text{TFT} = R \cdot T = R \cdot (T_{\text{USL}} + T_{\text{DSL}}) \quad (6)$$

where  $R$  is the number of FL rounds.  $T_{\text{USL}}$  and  $T_{\text{DSL}}$  are the upload and download phase time, respectively.

In each round, in the USL phase, the UAV visits each node  $p_k \forall k \in \mathcal{K}$  to receive FL updates. These updates can be model weights  $\theta$  or SLs depending on the FL algorithm. We

denote the size of FL updates from user  $k$  as  $B_k$  via a LoRa wireless link. Moreover, we assume the size of the update packages uploaded and downloaded to all users are the same, i.e.,  $|\theta_k| \forall k \in \mathcal{K}$ . Once the UAV has collected the update of the last node  $P_{N_k-1}^d$  in its trajectory  $P^d$ , they are aggregated into global updates. The DSL phase commences from the last node back to the node from which the UAV started the round. In this phase, the UAV returns along the USL path, stopping at each node to download the aggregated updates.

We can further breakdown  $T_{\text{USL}}$  as follows:

$$T_{\text{USL}} = T_M + T_W, \quad (7)$$

where  $T_M$  and  $T_W$  denote the drone moving time and waiting time, respectively. During  $T_M$ , the drone moves from one node/household to the next. Before the drone leaves a node, it waits for  $T_W$  seconds to upload or download the FL update package. To realize our goal of minimizing the drone FL time, we can further breakdown  $T_M$  and  $T_W$  as

$$T_M = \sum_{i=1}^{N_k} \sum_{j=1, j \neq i}^{N_k} \frac{(d_{ij}x_{ij})}{Vd} \quad (8)$$

$$T_W = \sum_{i=1}^{N_k} \frac{B_k}{R_b} \quad (9)$$

where  $d_{i,j}$  is the distance between the nodes  $i$  and  $j$ ,  $x_{i,j}$  is a binary decision variable to indicate whether the path between nodes  $i$  and  $j$  is used in the drone trajectory. We can rewrite (6) as

$$\min_{x_{i,j}, B} \text{TFT} = \min_{x_{i,j}, B} T_M + T_W \quad (10)$$

$$= \min_{x_{ij}, B} 2R \left( \sum_{i=1}^{N_k} \sum_{j=1, j \neq i}^{N_k} \frac{(d_{ij}x_{ij})}{V} + \sum_{i=1}^{N_k} \frac{B_k}{R_b} \right). \quad (11)$$

We model  $T_M$  and  $T_W$  separately since we assume there is no overlapping between both durations. That is, the drone traverses the participant nodes to upload/download FL packages. When the drone wants to send FL package to node  $k$ , it flies to node  $k$  following the optimized path as part of  $T_M$ . When the drone enters the CA of the  $k$ th user,  $T_W$  starts where the drone downloads the FL package.

To this end, we propose a hybrid solution to tackle the complex optimization problem in (11) by using SOM to optimize the drone path to reduce  $T_M$ , and KD-based FL to reduce the  $T_W$  part of the optimization problem by using Soft Labels SL instead of model weights  $\theta$ .

For the first problem, we formulate the path optimization problem as follows, the drone traverses a graph  $G = (V, E)$ , where  $V = \{v_1, v_2, \dots, v_{N_v}\}$  is the set of  $N_v = K + 1$  vertices (home access points  $AP_i \forall i \in I$  and the starting point A which has location  $\{x_A, y_A\}$ );  $E = \{e_{ij}\} \forall i, j \in V, i \neq j$  is the set of edges; and  $w_{ij}$  is the weight (distance) of edge  $e_{ij}$  between vertex  $v_i$  and vertex  $v_j$ .

To this end, we tackle the drone path optimization problem in two steps: 1) formalizing the drone path optimization as a TSP problem and finding a Hamiltonian cycle  $H$  and 2) we apply post-processing steps to derive the near-optimum drone path  $G^d$  from  $H$  where  $G^d$  is the path the drone should follow to perform the two FL phases USL and DSL with minimum  $T_M$ .

The Hamiltonian cycle  $H$  is a permutation of vertices forming a closed loop that visits each path once. To ensure that this permutation minimizes the distance  $D$  covered by the drone, TSP minimizes the sum of distances masked by modeling the edges included in  $H$  using a binary decision variable  $x_{ij}$ , corresponding to each edge  $e_{ij} \in E$ , where

$$x_{ij} = \begin{cases} 1, & \text{if } e_{ij} \text{ is included in } H \\ 0, & \text{otherwise.} \end{cases} \quad (12)$$

The TSP is formally defined as

$$\min_H \sum_{e(ij) \in H} w_{ij} = \min_{x_{ij}} \sum_{i=1}^{N_v} \sum_{j=1, j \neq i}^{N_v} w_{ij}x_{ij} \quad (13)$$

under the constraints.

1) Each node is visited exactly once

$$\sum_{j=1, j \neq i}^{N_v} x_{ij} = 1 \quad \forall i \in V. \quad (14)$$

2) No mini-cycles (subtours) are allowed, meaning any cycle in the solution must include all nodes

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \quad \forall S \subset V, 2 \leq |S| \leq N_v - 1. \quad (15)$$

Notice that the first constraint states each node is visited once, however, in our case each node is visited twice (one time in the USL phase and another in the DSL phase). To address this, We manually remove the last edge connecting the starting

vertex to the last vertex ( $v_{N_v}$ ) of  $H$ , we then append the vertices from the second-to-last vertex ( $v_{N_v-1}$ ) to the starting vertex following the same path in reverse order. This will produce the drone path  $G^d$  which can be used to produce the drone trajectory  $\mathbf{P}^d$ .

#### IV. DRONE PATH OPTIMIZATION USING SELF-ORGANIZING MAPS

In this section, we propose a SOM-based path optimization approach to solve the modified TSP problem formulated earlier.

SOM is a type of unsupervised competitive network used for clustering [47]. SOM is related to two families of competitive learning algorithms called winner-takes-all (WTA) and winner-takes-most (WTM) [48]. The goal is to learn a weighted network that represents the patterns in a graph of nodes as defined by the distance function.

In WTA, a list of vectors, referred to the network  $W$ , is constructed and is trained as follows. An input node  $x$  is given, and we calculate the distance to each row  $W_i$  corresponding to a neuron. The neuron whose weights are most correlated to the current node emerges as the winner node  $W_{win}$  with an index  $i_{win}$ . Only the winning neuron modifies its weights as

$$W_{win} = W_{win} + \eta(x - W_{win}) \quad (16)$$

where  $\eta$  is the learning rate and  $x$  stands for the current input node. WTA only allows one neuron (the winner neuron) to be updated in each iteration which slows performance. A natural extension to WTA is the WTM strategy. In WTM, the concept of winner neighborhood is introduced to refer to neurons close to the winner neuron. The farther the neighboring neuron from the winner, the smaller the modification that is applied to its weights. This process can be described as

$$W_i = W_i + \eta KN(i, i_{win})(x - W_i) \quad (17)$$

for all neurons  $i$  that belong to the winner's neighborhood as defined by  $KN(i, i_{win})$

$$KN(i, x) = \begin{cases} 1 & \text{for } d(i, i_{win}) \leq \lambda \\ 0 & \text{for others} \end{cases} \quad (18)$$

where  $d(i, i_{win})$  is the distance between the winning neuron and  $i$ th neuron.  $\lambda$  denotes the neighborhood radius.

Note that we use the term neurons to point to learnable parameters, and nodes to refer to the coordinates of graph vertices which in our case represent user-home coordinates and the drone starting point coordinates on the TA map. After many iterations of (17), the neurons in  $W$  become closer to the coordinates of the input nodes  $x$ , and they reflect the topological structure of the nodes.

Improved neighborhood function has been proposed [48] to select neighborhood neurons by employing a Gaussian kernel as the neighborhood function

$$g(i, i_{win}, \sigma) = e^{-\frac{d(i, i_{win})^2}{2\sigma^2}} \quad (19)$$

where  $\sigma$  defines the spread of the distribution which controls the strength of the applied modification. Using  $g(i, i_{win})$ , we

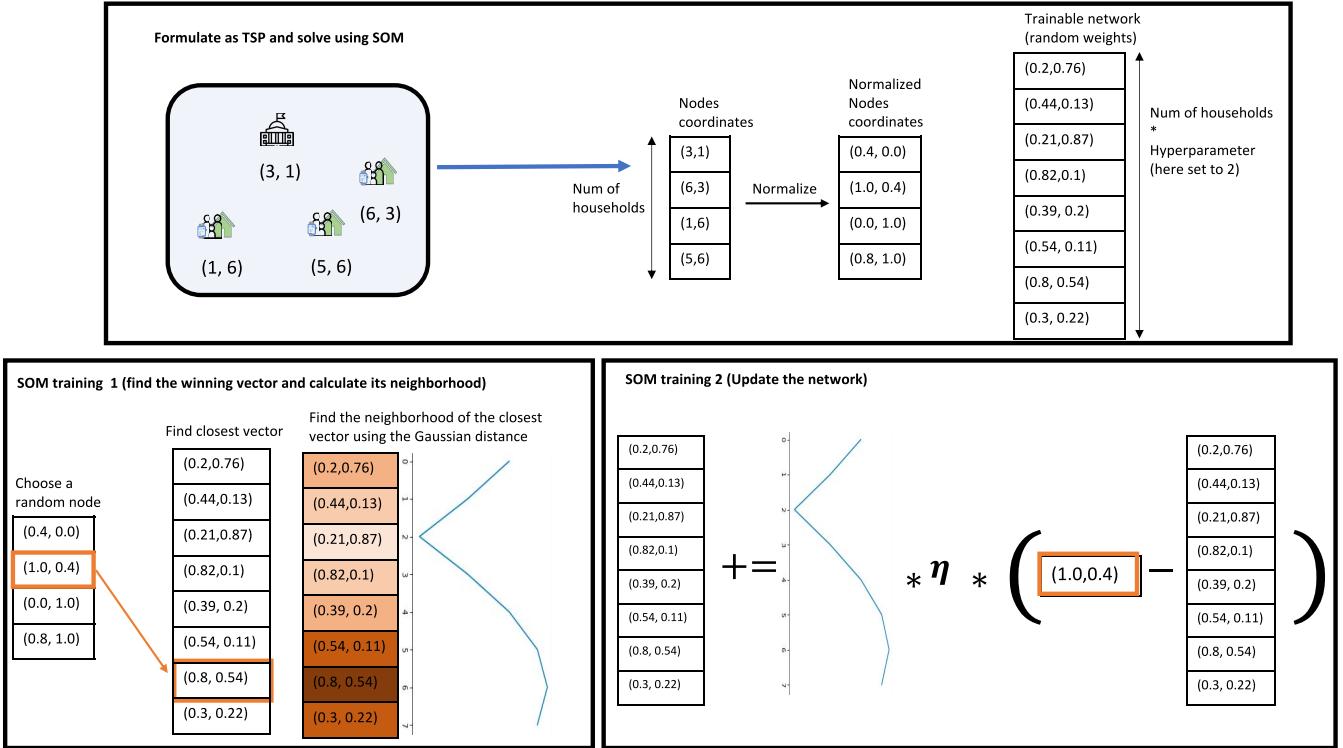


Fig. 3. To obtain a distance-efficient drone path, we formulate the problem as a TSP and construct a trainable network to train a SOM to obtain a near-optimal Hamiltonian path.

can rewrite (17). Thus, we arrive at the following equation in our implementation of SOM:

$$W_i = W_i + \eta g(i, i_{\text{win}}), \sigma)(x - W_i). \quad (20)$$

To help convergence as training progresses, decaying coefficients are employed for the learning rate and the neighborhood spread

$$\eta_{e+1} = \gamma_\eta \eta_e, \quad \sigma_{e+1} = \gamma_\sigma \sigma_e \quad (21)$$

where  $\gamma_\eta$  and  $\gamma_\sigma$  are the decaying coefficients for the learning rate  $\eta$  and the standard deviation  $\sigma$ , respectively. The subscript  $e$  refers to the current SOM iteration.

To train SOM for TSP, we apply (20) on the graph nodes as training examples for many iterations. The stopping criteria can be reaching a certain number of iterations or when  $\eta$  or  $\sigma$  decays to a certain value.

For inference, after training the SOM, we utilize the learned network of neurons to construct an approximate TSP tour. This is performed by going through each node and calculating the index  $i_{\text{win}}$  of the closest neuron (the winner neuron  $W_{\text{win}}$ ) for each node. We choose the size of  $W$ , denoted as  $N_w$ , to be eight times the size of the graph  $N_v$ . Since the number of neurons is usually larger than the number of nodes and not each neuron  $W_i$  will map to a node, we cannot directly use indices of winner neurons to construct the TSP tour. Instead, the obtained list of neuron indices is sorted and the sorting index is used to reorder the graph nodes into a near-optimal TSP tour  $C$  given by

$$C = \langle v_{c_1}, v_{c_2}, \dots, v_{c_{N_c}} \rangle \quad (22)$$

where  $N_c = N_v + 1$ . The Hamiltonian cycle is given by

$$H = \langle e_{h_1}, e_{h_1}, \dots, e_{h_{N_h}} \rangle \quad (23)$$

where  $N_h = N_c - 1$ . Finally, to calculate  $G^d$ , which is the drone's tour from  $H$ , we start by removing the last edge in  $H$   $e_{N_h-1}$  which connects the second-to-last vertex of  $C: v_{c_{N_c-1}}$  to the last vertex  $v_{c_{N_c}}$ . We also remove the last vertex in  $C$ , which is the same as the starting vertex, to obtain an open tour of vertices  $C' = \langle v_{c_1}, v_{c_2}, \dots, v_{c_{N_c-1}} \rangle$ .  $C'$  represents the path of the USL phase. To consider the drone's DSL path, we reverse the order of vertices in the open tour  $C'$  to obtain  $\tilde{C} = \langle v_{c_{N_c-1}}, v_{c_{N_c-2}}, \dots, v_{c_1} \rangle$ .

The path of the drone that minimizes the distance  $D$  can finally be defined as the concatenation of  $C'$  and  $\tilde{C}$  as

$$G^d = \langle v_{c_1}, v_{c_2}, \dots, v_{c_{N_c-1}}, v_{c_{N_c-2}}, \dots, v_{c_1} \rangle. \quad (24)$$

The number of nodes in the drone optimal path,  $G^d$ , is given by  $N_G = 2 \times (N_c - 1)$ . The distance  $D$  covered by the drone while following  $G^d$  is minimized since it is derived from the near-optimal Hamiltonian cycle  $H$  calculated by SOM.

#### A. Drone Path Optimization Temporal and Memory Complexity Analysis

To analyze the time and space complexity of the modified SOM algorithm for drone path optimization, we need to consider the major operations involved. These operations include distance calculations, weight updates, and the construction of the near-optimal TSP tour.

During training, we need to calculate the distance between each input node  $x$  and each neuron  $W_i$  for each SOM iteration

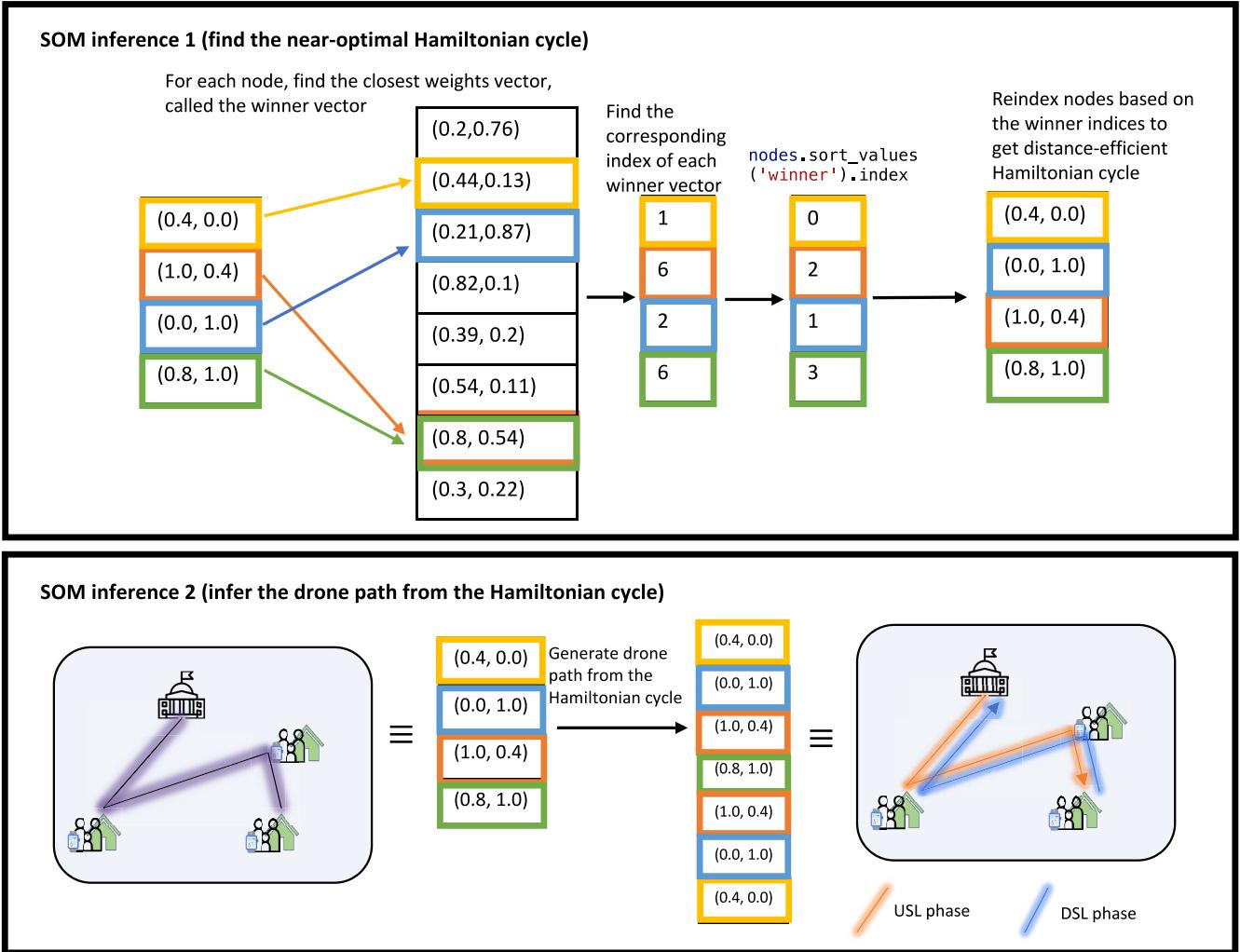


Fig. 4. After training the SOM network on the topography of the graph, we use the trained network to infer the near-optimal Hamiltonian path  $H$ . After that, the drone path can be derived from  $H$ . Note in the toy example, both the second and the fourth node point to the same weight vector, so the order of these two nodes (households) in the Hamiltonian cycle is arbitrary.

to find the winning neuron  $W_{\text{win}}$ . In the considered problem with  $N_v$  nodes and a network  $W$  of size  $N_w$ , the distance calculations result in a complexity of  $O(N_v \cdot N_w)$ . In the weights update step, the used Gaussian neighborhood function  $g(i, i_{\text{win}})$  generates a Gaussian distribution centered at  $W_{\text{win}}$  with a standard deviation  $\sigma$  to weight the modification applied to each neuron. Thus, the weight update complexity is  $O(N_w)$ . Fig. 3 shows how the given household graph is formulated as a TSP problem and how SOM is trained. As the algorithm consists of  $E$  iterations, the overall time complexity for training is  $O(E(N_v \cdot N_w + N_w))$ .

For inference, we need to calculate the winning neuron for each node, requiring a time complexity of  $O(N_v \cdot N_w)$ . Then, constructing the near-optimal TSP tour using the winning neuron indices involves sorting the indices with a complexity of  $O(N_v \log N_v)$ . The steps of SOM inference to obtain the drone path are shown in Fig. 4. Consequently, the total time complexity for the inference phase is  $O(N_v \cdot N_w + N_v \log N_v)$ .

The space complexity is determined by the storage requirements for the input nodes and the weighted network. Storing

the output drone path nodes requires  $O(N_v)$ , and storing the weighted network requires  $O(N_w)$ . Therefore, the overall space complexity for the algorithm is  $O(N_v + N_w)$ .

## V. KNOWLEDGE DISTILLATION-BASED FEDERATED LEARNING

To minimize the waiting time  $T_W$ , the size of the FL update package received from or sent to user  $k$ ,  $B_k$ , has to be reduced. We study KD-based FL as an alternative to model-based FL for our considered resource-constrained IoT system, that leverages SLs for sending updates with significantly smaller sizes compared to model weights used in standard FL. FL is a distributed learning framework for training deep learning models on data scattered across many clients [16], [18]. In this work, the drone access point  $AP^d$  functions as the server and the users' home access points  $AP_k \forall k \in K$  represent clients with each having its own private local data set  $D_k$  where  $D = \{D_k\}_{k=0}^K$  is collected from the wearable devices used in each household.

We discuss two FL strategies. The first is the standard FL where the server controls the model architecture  $f$  for all clients. In this FL setting, for each global round  $r$ , client  $k$  downloads the weights  $\theta^r$  of the server-controlled model  $f$ , trains it on the local data  $D_k$ , and finally uploads the trained weights  $\theta_k^r$  to be aggregated at the server as  $\theta^{r+1}$  of model  $f^{r+1}$ . The aggregated weights  $\theta^{r+1}$  of round  $r$  are the initial weights broadcasted to clients in the next round  $r+1$ .

The second FL paradigm considered in this work is KD-based FL. In this FL method, updates are in the form of Soft Labels SL instead of model weights, which significantly reduces communication costs. KD-based FL assumes that a public data set  $D_p$  is shared among all clients to calculate their local soft labels  $SL_k \forall k \in \mathcal{K}$ . Since the KD-based FL method shares SLs, it can be regarded as a model-agnostic method. Therefore, there is no server-controlled model architecture  $f$  in KD-based FL. Instead, each client independently designs its local model architecture  $f_k$  according to its unique resource needs/availability.

In the remainder of the section, we present traffic models for both the model-based and communication-efficient KD-based FL, followed by an overview of the augmented KD-based FL framework. Then, we propose employing mixup augmentation and compression techniques to enhance performance and further reduce communication overhead.

#### A. Model-Based FL Traffic Model

Each global round in model-based FL [17] consists of three ordered phases: 1) the download phase (DL); 2) the learning phase (L); and 3) the upload phase (UL).

In the DL phase, each node downloads the weights of the server-controlled global model  $f$  denoted as  $\theta^r$ , where  $r$  is the current global round. In the L phase, each client updates his local copy of weights using

$$\theta_k^{r'} \leftarrow \theta_k^r - \eta \cdot \frac{1}{|D_i|} \sum_{(x,y) \in D_i} \nabla \mathcal{L}(\theta_{t-1}, f(\theta_k^r, x), y) \quad (25)$$

where  $\eta$  denotes the learning rate,  $\theta_k^{r'}$  denotes the locally trained weights of client  $k$  at round  $r$ , and  $\mathcal{L}$  is the loss function. In the case of a classification problem, a Categorical Cross Entropy function is usually used  $\mathcal{L} = \mathcal{L}_{CCE}$ .

Finally, in the UL phase, clients upload the locally trained weights to be aggregated at the server as

$$\theta^{r+1} \leftarrow \frac{1}{N_k} \sum_{i=1}^{N_k} \theta_k^{r'}. \quad (26)$$

This process is repeated until convergence or until a predetermined number of rounds have been completed.

Uploading and downloading model weights requires high-bandwidth communication. In our drone-based system, the drone flight time is limited, therefore, we use a different FL approach that is based on the concept of KD.

#### B. Knowledge Distillation-Based Federated Learning Traffic Model

In centralized settings, KD is a technique used to train a student model using a trained teacher model, i.e., given an

unlabeled data set and a trained teacher model, the goal is to use the SLs produced by the teacher model to train a student model. To apply KD to the FL context, we use a proxy data set  $D_p$  shared with all clients to calculate their SLs on it. Then these local SLs are sent to the server to be aggregated into global SLs and sent back to clients to train on the  $(D_p, SL')$  labeled data set.

In FL with KD, each node has its private data set,  $D_k$ , and another public data set,  $D_p$ , that is shared with all other nodes and is used to transfer knowledge. A major advantage of KD-based FL over model-based FL, which was presented in the previous section, is that the former gives nodes the freedom to design their own model architecture  $f_k$ . On the other hand, model-based FL [17], [49] assumes a server-controlled model architecture.

The global cycle in KD-based FL consists of four ordered phases: 1) local learning (LL) phase; 2) USLs phase; 3) DSLs; and 4) KD learning (KDL) phase.

Both the USL and DSL phases are described earlier. In these phases, SLs are uploaded from nodes/clients to the server (the drone), then the server aggregates and downloads the global SLs to nodes, respectively.

In the LL phase, Cross Entropy Loss (CCE) is used, as follows, to train the locally designed model  $f_k$  on the locally labeled data set

$$\theta_k^{r'} \leftarrow \theta_k^r - \eta \cdot \frac{1}{|D_k|} \sum_{(x,y) \in D_k} \nabla \mathcal{L}_{CCE}(\theta_k^r, f_k(x), y). \quad (27)$$

After the local clients have trained their models on the local data, each client uploads its local SLs  $SL_k^r$  to the drone in the USL phase. In the DSL phase, the drone traverses back its USL path and downloads the aggregated SLs  $SL'$  to be used for training in the KDL phase as follows:

$$SL' \leftarrow \sum_{i=1}^{N_k} \frac{P_i^r \cdot SL_k^r}{\sum_{k=0}^{N_k} P_k^r}. \quad (28)$$

In the KDL phase, distance functions like the mean squared error (MSE) or Kullback–Leibler divergence (KD) loss are used to train  $f_i$  on the public unlabeled data set leveraging the downloaded SLs. This process can be described by the following equation:

$$\theta_k^{r+1} \leftarrow \theta_k^{r'} - \eta \cdot \frac{1}{|D_p|} \sum_{(x,y) \in (D_p, SL')} \nabla \mathcal{L}_{KD}(\theta_k^{r'}, f_k^*(x), y) \quad (29)$$

where  $\theta_k^{r'}$ ,  $\theta_k^{r+1}$ , and  $f_k^*(x)$  represent, the model weights after local data set training, the model weights after KD training, and similar architecture to  $f_k$  constructed by removing the last layer [50] or increase the SoftMax temperature [51] to smooth the distribution of the output vector, which is the SLs  $SL_k = f_k^*(D_p)$ .

1) *Augmented Knowledge Distillation-Based Federated Learning*: To improve the performance of KD-based FL methods, Gad and Fadlullah [52] and Gad et al. [53] proposed a KD-based FL algorithm that uses mixup augmentation [54] to synthesize a data set  $D_{Aug}^r$  each global round  $r$ , as a result of applying Mixup augmentation to  $D_p$  and  $D_d^r$  as given by

$$D_{Aug}^r = \lambda^r \cdot D_d^r + (1 - \lambda^r) \cdot D_p \quad (30)$$

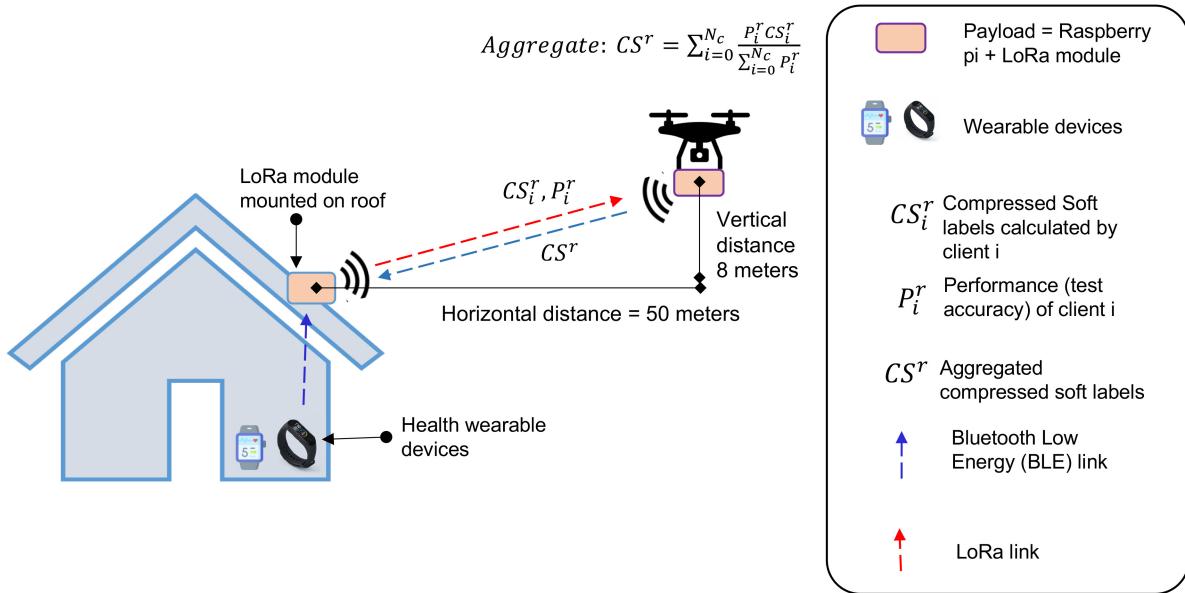


Fig. 5. Overview of the proposed KD-based FL algorithm on a DORA network. The drone exchanges messages with the nodes/houses within a CA around each house defined as a circle with a radius of 50 m.

where  $D_d^r$  is a permuted version of  $D_p$  with a permutation seeded with  $\rho^r$ , and  $\lambda^r \sim \text{Beta}(\alpha^r, \alpha^r)$ . The  $\text{Beta}(a, b)$  is the beta distribution.

Both  $\rho^r$  and  $\alpha^r$  are broadcasted each round  $r$  to ensure KD training is performed on a new synthesized data set  $D_{\text{Aug}}^r$  while ensuring it is the same data set on all clients. The use of mixup augmentation [54] and the permutation of the public data set also helps to prevent overfitting and improve generalization.

2) *Proposed Compressed Federated Learning With Augmented Knowledge Distillation:* Compression SLs can reduce their size by normalizing  $SL_k^r$  using Min-Max normalization, multiplying by 255, and then casting type from floating point (46 bits) to unsigned integer (8 bytes) [55]. The compressed local SLs,  $CS_k^r$ , are then sent to the server to be aggregated as

$$CS^r \leftarrow \sum_{i=1}^{N_k} \frac{P_i^r \cdot CS_k^r}{\sum_{k=0}^{N_k} P_k^r} \quad (31)$$

where  $CS^r$  is broadcasted by the server to all clients to train on it. Before that,  $CS^r$  is normalized and cast back to float.

Fig. 5 illustrates how FL can be implemented under the proposed Drone-LoRa system (DORA) to enable smart health analytics on edge devices in rural areas.

At each house, the access point module is fixed in a high position from which it can extract data from wearable devices held by residents of the house through Bluetooth. The access point module also listens for the drone access point carried by the drone, which is scheduled to establish a LoRa connection.

When the drone enters the CA of any of the house access points  $\{AP_k\} \forall k \in K$ , the drone access point attempts to establish a connection with the home access point. If the home access point is detected, through the acknowledgment packets, the drone access point  $AP_d$ , asks for updates from the home access point  $AP_k$ . In the context of FL with KD, these local updates are SLs, which have a small size and can be quickly

TABLE III  
COMMUNICATION OVERHEAD BETWEEN MODEL-BASED AND MODEL-AGNOSTIC FL METHODS

Method	Weighting	MS
FedAKD	UW	$ SL_k^r  +  SL^r $
	PW	$ SL_k^r  +  P_k^r  +  SL^r $
Comp FedAKD	UW	$ CS_k^r  +  CS^r $
	PW	$ CS_k^r  +  P_k^r  +  CS^r $
FedAvg	UW	$ \theta_k^r  +  \theta^r $
	PW	$ \theta_k^r  +  P_k^r  +  \theta^r $

sent without much waiting time from the drone. In the case of standard FL (FedAvg), the local updates are model weights. In other scenarios, the home access point may choose to send the encrypted health records collected from wearable devices to be analyzed at the community server.

The time needed to exchange updates with each home access point is estimated as 9.8 min, including 4.8 min for sending the compressed SLs, according to Table VII. Our proposed CFedAKD approach requires less waiting time compared to FedMD/FedAKD and FedAvg, which require 41.2 min and over 30 h, respectively, due to the relatively large size of the model file and the low bandwidth of LoRa + RDT.

## VI. RELIABLE COMMUNICATION PROTOCOL FOR LORA-BASED IOT DEPLOYMENT SETTING

In this section, we describe the LoRA-based IoT setting and present our designed reliable communication protocol for LoRA. We consider a flexible yet powerful microcontroller board, i.e., Raspberry Pi 3 (RP3), operated with the Raspbian Unix-like operating system (OS) in all our access points as it comes equipped with a Bluetooth module and two sets of universal asynchronous receiver-transmitter (UART) modules for serial communication. As for the LoRa module, we employ

Reytax's RYLR896 868/915-MHz transceiver module [56], which can provide long-range spread spectrum communication up to 15 km. The LoRa module is controlled by the Raspberry Pi using the serial UART pins (pin 8 and pin 10), by connecting the receiver pin (RX) of the LoRa module to the transmitter pin (TX) of the Raspberry Pi and vice versa. We also attach Pisugar 3 chargeable batteries to all our access points to ensure portability.

For the serial communication between the LoRa module and the Raspberry Pi, we set the Baud Rate to 115 200 bits per second (bps) to ensure a maximum communication speed over the serial channel as recommended by the manufacturer. Although the mini-UART is the default module used for serial communication in Raspberry Pi 3, we select the PL011 UART module to be our serial port rather than the default mini-UART module. This is because the PL011 UART, which is originally utilized by the Bluetooth module in RP3, is more reliable than the mini-UART module since the latter exhibits a reduced feature set and has its baud rate reliant on the CPU clock speed [57]. To enable the usage of PL011 UART as the serial port, we create a service that disables the Bluetooth module and allocate this port for the LoRa serial communication every time the LoRa module is about to transmit or receive data. This approach allows us to ensure a successful transfer and decoding of serial commands with a fixed baud rate between the LoRa module and the microcontroller. The commands for setting the LoRa parameters, setting addresses, sending and receiving are implemented according to the AT commands guide provided by Reytax [56].

To facilitate the communication process between our APs, each AP is assigned a unique identification number (ID) stored as an environment variable in the Raspbian OS and utilized by our software to set the address of the LoRa module attached to each AP. The IDs of all  $AP_h$ s are kept anonymous to maintain the data privacy of the users and they are only used by the  $AP_d$  to establish a separate communication channel with each home-based access point. Since the LoRa communication is known to be unreliable (i.e., lossy communication), we illustrate our proposed RDT layer that uses a TCP/IP-like method to track the sent packets and resend those if no acknowledgment is received from the receiver. Also, we demonstrate the LoRa parameters associated with LoRa point-to-point communication.

#### A. Proposed Reliable Data Transfer Protocol

Due to the lossy nature of LoRa communication, there are lost and out-of-order packets. Therefore, we implement a reliable transport protocol, which transforms the unreliable LoRa D2D link into a reliable one, such that the content can be transferred between LoRa-equipped devices reliably and efficiently. We choose to implement the Go-Back-N protocol [58], which is one of the window-based protocols for RDT. The protocol implementation provides a high-level API for the LoRa-equipped devices to be used such that files can be seamlessly transferred.

The are two types of packets in the Go-Back-N protocol: 1) the data packet which includes a chunk of data transmitted

(a)	Packet ID	LoRa address	Data
(b)	Packet ID	LoRa address	Acknowledgment number

Fig. 6. Structure of data and acknowledgment packets. (a) Data packet structure. (b) Acknowledgment packet structure.

from the sender to the receiver and 2) the acknowledgment packet transmitted from the receiver to the sender. The data packet includes three fields, i.e., the packet ID, LoRa address, and actual data or payload. The packet ID is a unique ID for each packet while the LoRa address refers to the address of the sending LoRa-equipped device. The payload includes a chunk of data intended for the receiver. On the other hand, while the acknowledgment packet shares the packet ID and LoRa address fields with the data packet, the data field is replaced by an acknowledgment number field, which includes the ID of the last received packet. The structure of the packets is shown in Fig. 6(a) and (b).

In the Go-Back-N protocol, the sender is allowed to send multiple packets without waiting for an acknowledgment from the receiver. However, the number of unacknowledged packets is limited to  $N$ , which denotes the window size. The protocol uses cumulative acknowledgment, hence, the interpretation of the acknowledgment packet by the sender is that all the packets up to the acknowledgment number have been received correctly.

When a LoRa-equipped device needs to transmit content (e.g., sensed data from HAR devices) to another LoRa device, it calls the sending function with the receiver address, file information, and protocol parameters. The protocol information consists of the window size and timeout interval. Then, the following steps are considered.

- 1) Divide the content into data packets and each packet is given a unique ID in chronological order  $\{1, \dots, \text{number of packets}\}$ .
- 2) Add a packet with ID 0 that includes control information, such as *filename* and *number of packets to be sent*.
- 3) Transmit the first  $N$  packets to the receiver and wait for an acknowledgment receipt.

The sender then responds to the following events.

- 1) *Receipt of an Acknowledgment*: If the received acknowledgment number is more than the window base, the sender slides the window forward and sends the packets inside the window.
- 2) *Timeout Event*: If the sender does not receive an acknowledgment for a period of time equal to the *timeout interval*, it resends all the packets within the window and restarts the timer.

The sender terminates once the last packet is acknowledged. The receiver function only deals with one event which is the *receipt of a data packet*. If the sender address and packet ID match the expected ones, the receiver formats an acknowledgment packet with the proper acknowledgment

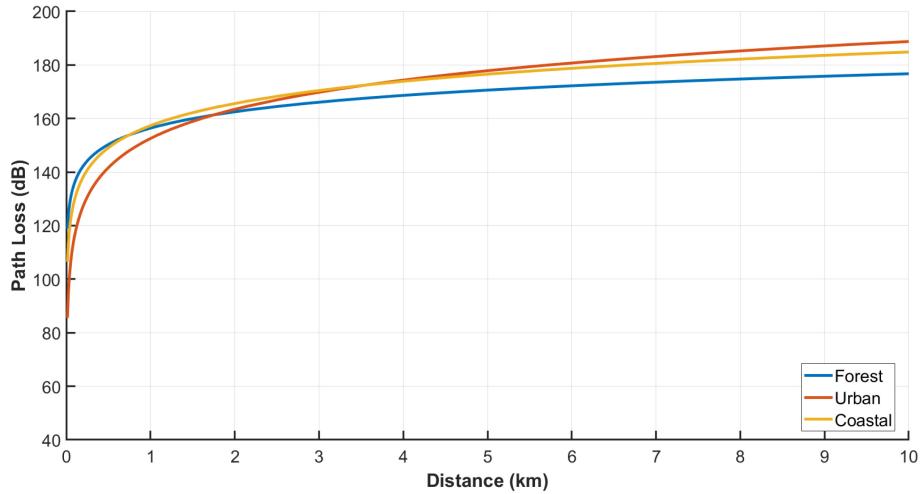


Fig. 7. Path loss versus distance.

TABLE IV  
PARAMETERS OF PATH LOSS MODEL IN DIFFERENT ENVIRONMENTS [59]

Environment	$PL(d_0)$	$\eta$	$\sigma$
Urban	74.85	2.75	11.25
Coastal	43.96	3.62	27.51
Forested	95.52	2.03	6.87

number and sends it to the sender. Otherwise, the packet is discarded. The receiver terminates when all the packets are received.

### B. LoRa Propagation Model

We adopt a single slope log-distance model for the path loss associated with the LoRa point-to-point link. Accordingly, the path loss  $PL(d)$  measured in dB is expressed as follows:

$$PL(d) = PL(d_0) + 10\eta \log_{10}\left(\frac{d}{d_0}\right) + \mathcal{X}_\sigma \quad (32)$$

where  $PL(d_0)$  denotes the path loss measured at the reference distance  $d_0$ ,  $\eta$  indicates the path loss exponent,  $d$  refers to the distance between the transmitter and the receiver, and  $\mathcal{X}_\sigma$  represents random variable which accounts for shadow fading. Based on [59], we consider the empirical characterization of the LoRa point-to-point link in three environments: 1) urban; 2) coastal; and 3) forested environments. Log-normal shadowing is assumed  $\mathcal{X}_\sigma \sim \mathcal{N}(0, \sigma)$  and three different single slope models for each environment are fitted to estimate the values of  $PL(d_0)$ ,  $\eta$  and  $\sigma$ . The parameters are summarized in Table IV. The three models are illustrated in Fig. 7 assuming that  $\mathcal{X}_\sigma = 0$  which can be considered as the expected value of the path loss at distance  $d$ .

There are several transmission parameters for LoRa modulation, such as spreading factor (SF), bandwidth (BW), and coding rate (CR). The SF directly influences the sensitivity of the receiver which ranges from  $-111$  dBm to  $-148$  dBm, while the bandwidth and CR influence the nominal bitrate. We choose to operate with the LoRa parameters in Table II with our proposed protocol designed earlier.

The corresponding receiver sensitivity for this combination of parameters is  $-136$  dBm. Using the models in Fig. 7 and assuming that the transmitter radiates with its maximum power ( $20$  dBm), the maximum communication distance for the urban, coastal and forest settings is  $0.89$  km,  $1.24$  km, and  $0.95$  km; respectively. Furthermore, the nominal communication rate can be calculated as follows based on [60]:

$$R_b = SF \times \frac{\left(\frac{4}{4+CR}\right)}{\left(\frac{2SF}{BW}\right)}. \quad (33)$$

For the chosen parameter setting, using our proposed protocol, The nominal rate is  $293$  bps for the considered IoT setting.

## VII. EXPERIMENTS AND PERFORMANCE EVALUATION

### A. Evaluating Drone Path Optimization Using SOM

Since finding the optimum drone path depends on the Hamiltonian cycle  $H$  obtained as a solution for TSP, we evaluate the performance of our algorithm based on the quality of the obtained  $H$ . For that, we use map instances provided by the National TSP maintained by the University of Waterloo. For each map, the distances of both the optimal route and the SOM-generated route are provided. We report the execution time and the quality of the obtained  $H$ . The quality of the route is calculated based on how far the obtained route distance is from the optimal value using  $(D_{SOM}/D_{optimal})$ , where  $D_{SOM}$  and  $D_{optimal}$  represent the distance SOM-generated route and the distance of the optimal route, respectively. We used three city maps of Qatar, Egypt, and Canada with the number of cities (nodes on the map) equal 194, 7146, and 4663, respectively.

### B. Federated Learning Baselines

We compare the following FL algorithms in terms of performance and communication overhead.

TABLE V  
TRAIN/TEST SPLITTING AND LOCAL SETS PARTITIONING OF THE HAR DATA SETS USED IN THIS WORK

Dataset	HARS	Depth	Harbox	IMU
<b>Total train set size</b>	10,800 samples	3,544 samples	22,657	683 samples
<b>Total test set size</b>	2,947 samples	-	-	-
<b>Local train size (per client):</b>	200 x 6 = 1,200 samples	418-696 samples	80-737 samples	133-146 samples
<b>Local test size (per client):</b>	2,947 samples	895-902 samples	55-317 samples	58-63 samples
<b>Public/Shared set size:</b>	736 samples	449 samples	185 samples	136 samples

TABLE VI  
CHARACTERISTICS OF THE FOUR HAR DATA SETS USED TO EVALUATE THE FL ALGORITHMS CONSIDERED IN THIS WORK

Dataset	HARS	Depth	Harbox	IMU
<b>Number of activities</b>	6	5	5	3
<b>Activities</b>	Walk, Walk up-stairs, Walk down-stairs, Sit, Stand, and Lay	Good, Ok, Victory, Stop, and Fist	Walking, Hopping, Phone calls, Waving, and Typing	Walking in corridor, Walking upstairs, and Walking downstairs
<b>Sensors</b>	Smartphone Sensors	Intertial	Depth camera	9-axis IMU
<b>Data modality</b>	Tabular	Image	Tabular	Tabular
<b>Data dimension</b>	(561)	(30,30,1)	(900)	(900)
<b>Number of subjects (data collectors)</b>	30	-	121	7
<b>Number of clients (In the FL experiment)</b>	9	8	115	6

- 1) *FedAvg*: Standard FL algorithm, aggregates client model weights each global round. Not suitable for our app due to limited LoRa bandwidth.
- 2) *FedMD*: Model-agnostic FL algorithm, clients communicate SLs calculated on shared public data set. Suitable for IoT apps with limited communication resources.
- 3) *FedAKD*: Model-agnostic FL algorithm. Unlike FedMD, FedAKD augments the shared public set each global round using Mixup augmentation to improve accuracy. Clients communicate their SLs calculated on an augmented version of the shared public data set. The augmentation variables are controlled by the server to generate the same augmented version across clients.
- 4) *Compressed FedAKD*: Same as FedAKD, but with SLs compressed from float-point 64-bit values to 8-bit integer values.

The communication overhead analysis for model-based FL and KD-based FL algorithms is outlined in Table III. The table provides a comparative analysis, elucidating the distinct communication dynamics of each algorithm, including the methodologies employed and the various weighting schemes.

### C. Data Set Preparation

This section discusses four different data sets used to evaluate model-based and model-agnostic FL (FL) algorithms for HAR. The data sets are as follows.

- 1) *IMU*: An IMU-based data set that includes walking activities performed by seven participants in two different buildings.
- 2) *Depth*: A depth-camera data set with five types of gestures performed by two subjects in three different environments.

- 3) *HARS*: A smartphone-sensors data set that includes recordings of 30 individuals performing daily activities while wearing a waist-mounted smartphone with embedded inertial sensors.
- 4) *HARBox*: The HARBox data set, which collected data using an Android app from 121 users performing five different activities.

These data sets vary in terms of size, dimensionality, and heterogeneity, providing a diverse set of challenges for evaluating FL algorithms for HAR.

Table V provides information on train/test and local sets partitioning for four HAR data sets. HARS has the largest train set with 10 800 samples followed by Harbox, Depth, and IMU. Only HARS has a total test set size of 2947 samples. The local train and test sizes are provided per client, ranging from 133-146 samples for IMU to 895-902 samples for Depth. The table also shows the public/shared set size for each data set, with HARS having the largest size of 736 samples.

Table VI provides information on the characteristics of the four HAR data sets, including the number of activities, specific activities, sensors used, data modality and dimension, and the number of subjects and clients. HARS has six activities, while the other data sets have five or three. The table shows that HARS and Harbox use tabular data, Depth uses image data with a dimension of (30,30,1), and IMU uses tabular data with a dimension of 900.

### D. Performance Evaluation

In this section, we first evaluate the performance of our proposed RDT protocol for the LoRa-equipped IoT environment. Then, we evaluate the performance of our modified

**TABLE VII**  
SENDING DEEP MODEL CONTENT VERSUS SLs USING LoRa + RDT PROTOCOL-BASED COMMUNICATION DETAILS. THE PROPOSED PROTOCOL AVOIDS LOSING PACKETS AS THE LoRA COMMUNICATION PROTOCOL IS ORIGINALLY UNRELIABLE (LOSSY COMMUNICATION). THE FILE SIZE OF FEDMD'S SLs (Z) AND CFEDAKD'S COMPRESSED SLs (CS) IS SIGNIFICANTLY SMALLER THAN THE SIZE OF MODEL FILES THEREFORE, THEY ARE MORE SUITABLE FOR FL IN BANDWIDTH-LIMITED ENVIRONMENTS

Federated learning algorithm	FedAvg			CFedAKD	FedMD
Basis of communication	Model 0	Model 1	Model 2	CS	Z
File size (KiloBytes: KB)	256 KB	638 KB	466 KB	3 KB	12 KB
Num packets	12,005	30,066	21,907	49	210
Packets preparation time (second:s)	0.4 s	0.77 s	0.6 s	<1 ms	<1 ms
Sending time (minute:m)	1178 m	2947 m	2147 m	4.8 m	20.6 m
Data rate (bits/second: b/s)	29 b/s	29 b/s	29 b/s	83 b/s	77 b/s
Packet rate (packet/s)	0.17	0.17	0.17	0.17	0.17

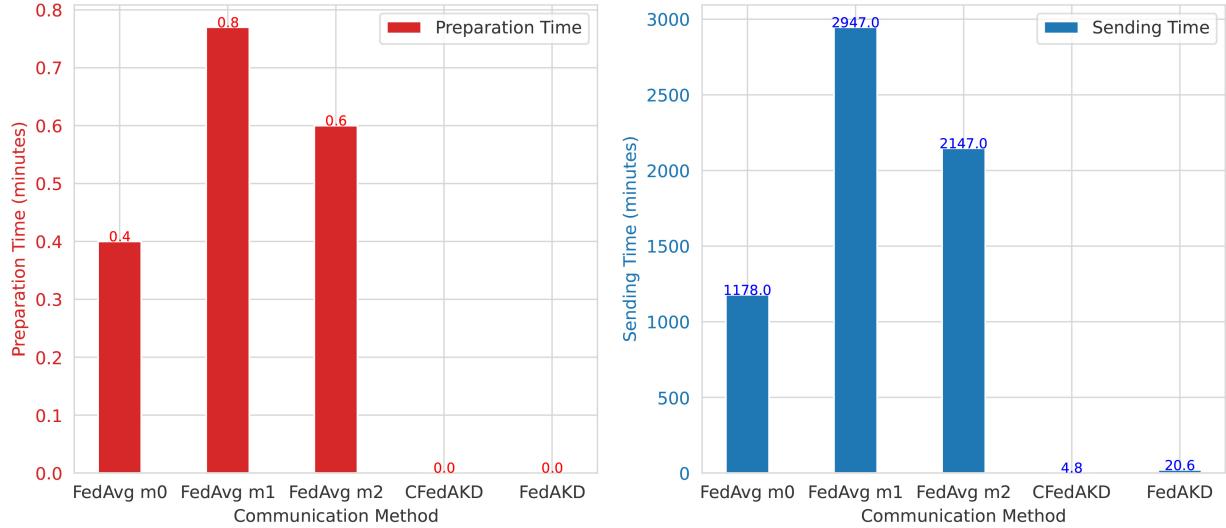


Fig. 8. Preparation and sending time of model-based and KD-based FL methods.

**TABLE VIII**  
PERFORMANCE OF SOM FOR SOLVING THE TSP ON QATAR, EGYPT, AND CANADA CITY MAP INSTANCES WITH VARYING ITERATIONS

City map	Num Iterations	Generation Time (Sec)	Path Distance	Drone Distance	Quality (APE)
Canada	100	2.06	9.64e+06	2.25e+07	647.27
	500	2.69	9.76e+06	1.86e+07	656.68
	1000	3.26	9.37e+06	1.97e+07	626.15
	5000	7.87	6.22e+06	1.30e+07	382.31
	10000	13.74	3.19e+06	7.10e+06	147.51
Egypt	100	4.35	2.32e+06	5.33e+06	1245.14
	500	5.44	2.15e+06	4.85e+06	1145.52
	1000	6.15	1.80e+06	4.68e+06	944.69
	5000	12.79	1.10e+06	1.80e+06	540.41
	10000	21.30	4.90e+05	1.00e+06	183.98
Qatar	100	0.11	2.04e+04	3.74e+04	117.85
	500	0.21	2.23e+04	3.35e+04	138.89
	1000	0.31	1.81e+04	3.33e+04	93.38
	5000	1.24	1.20e+04	2.35e+04	28.67
	10000	2.30	9.85e+03	1.96e+04	5.32

SOM-based drone path optimization technique. Furthermore, we verify the effectiveness of our proposed FL approach.

1) *Efficient Communication of KD-Based FL for LoRa-Based Communications:* In FL, clients share local updates with the server and receive global updates. Different FL algorithms use different types of updates. FedAvg [49] has been considered as the standard FL algorithm and it uses model weights, FedMD [50], and FedAKD. In our conducted

experiment, FL was performed over a LoRa communication link using our designed RDT protocol for LoRa-based communications. Table VII and Fig. 8 show the communication overhead of three FL algorithms: 1) FedAvg [49]; 2) FedMD [50]; and 3) the proposed CFedAKD method. As can be seen in Fig. 8, the packet preparation time (i.e., the time of splitting the client/server update into LoRa packets of size 250 bytes each) and the sending time of SLs in

FedAKD/FedMD and CFedAKD are x100 and x400 less than the time to prepare and send model weights in FedAvg.

*2) Performance of Drone Path Optimization Technique Using Our Modified SOM:* Table VIII shows the results of solving the TSP using SOM. Three map instances are obtained from the National TSP to evaluate the Hamiltonian cycle  $H$  learned by SOM: Qatar, Egypt, and Canada. These maps represent city coordinates. For each map, three runs with iterations 100, 1000, and 10000 were performed. The table shows the execution time for each run and the length of the obtained cycle. Additionally, using the precalculated optimal route which provided for each map, we evaluate the quality of the path obtained by SOM, denoted  $Q$ , with respect to the optimal route using absolute percentage error (APE) given by:  $Q = \text{APE} = |(D_{\text{optimal}} - D_{\text{SOM}})/D_{\text{optimal}}| \times 100\%$ , where  $D_{\text{optimal}}$  and  $D_{\text{SOM}}$  represent the length of the optimal path given for each map and the path obtained by SOM. Note that  $D_{\text{SOM}}$  is the length of the Hamiltonian cycle and not the drone path, this is because the given optimal path is a solution for TSP so it represents a Hamiltonian cycle.

$Q$  is shown in Table VIII for each solution with different parameters, and it reflects how close the found solution is to the optimal solution.

We can notice that for maps with a small number of nodes, SOM finds a route with a length close to the optimum path in a reasonable time frame. As the number of nodes increases, the quality of the found route becomes farther from the optimal solution, even with a large number of iterations.

To evaluate the quality of the SOM-based path with respect to the randomly generated path, Table IX shows the quality (measured in terms of the APE w.r.t. the given optimal path) of the randomly generated path for the considered city maps. Additionally, Fig. 9 shows a comparison between the quality of the random and the SOM-based paths. We can see that the quality SOM-based paths is less than the quality of the randomly generated path, which means that the SOM-based path is closer to the optimal path. Moreover, the value of the SOM-based path quality decreases as the number of iterations increase, this indicates that the total distance of the obtained path decreases as the SOM-based path planning algorithm runs for more iterations. This is also intuitive as SOM is a learning algorithm that starts with a network with random parameters and iteratively updates the network parameters.

*3) Comparative Results on Proposed and Conventional Federated Learning Approaches:* Table X shows the test accuracy of the following FL methods (FedMD, FedAKD, FedAvg, Compressed FedMD, and Compressed FedAKD) on the four HAR data sets (HARS, HARBOX, IMU, and Depth) under different conditions. The conditions include the fraction of clients participating in FL, whether compression using quantization is applied (which differentiates FedMD and Compressed FedMD), whether Mixup augmentation is applied (which differentiates FedMD from FedAKD), and the weighting scheme used to weight clients' contributions (SLs).

The results presented in Table X demonstrate that the FedAvg algorithm achieves the highest accuracy on the majority of data sets. This can be attributed to the utilization of

clients' weights for sending updates to the server. However, it is important to note that this approach comes with a significantly higher communication cost as compared to KD-based FL algorithms, which use SLs as the basis for communication.

In addition, it has been observed that reducing the fraction of users participating in each round results in lower average accuracy, as the algorithm requires more rounds to converge. Moreover, when comparing the performance of FedAKD and FedMD, it can be concluded that they achieve comparable results. While FedMD outperforms FedAKD in terms of accuracy on the HARS and IMU data sets, FedAKD performs better on the HARBOX and Depth data sets.

It is worth noting that both FedAKD and FedMD utilize SLs as the basis for communication update messages between clients and the server. However, the SLs are calculated on a mixup augmented version of the shared proxy data set in the case of FedAKD, instead of on the data set directly as is the case with FedMD. Despite this difference in approach, both methods have the same communication cost.

The spider plot on the left-hand side of Figure 10 demonstrates that the FedAKD algorithm outperforms the FedMD algorithm in serving the custom needs of clients' heterogeneous models. This can be observed from the larger red area that represents the per-client accuracy results of the FedAKD algorithm and the smaller blue area that represents the per-client performance under the FedMD algorithm. The larger red area in the spider plot on the left-hand side of the figure indicates that the FedAKD algorithm is better suited to meet the individual requirements of clients' models, resulting in higher accuracy scores.

On the other hand, the spider plot on the right-hand side of Figure 10 showcases the importance of weighting clients' contributions according to their accuracy on local test sets to improve the overall performance of FL algorithms. This is evidenced by the overlapping red and blue areas that represent the per-client accuracy results of FedAvg with performance-based weighting and the same algorithm using uniform weighting. The larger red area in this plot denotes that the performance-based weighting approach leads to higher per-client accuracy scores than the uniform weighting approach, emphasizing the need to consider clients' local test set accuracies for optimal performance.

## VIII. LIMITATIONS AND FUTURE WORK

*KD:* To effectively distill knowledge from clients, it is important that the distribution of the public data set  $D_p$  is similar to that of each client's local data set  $D_k$ . This requires feature engineering to get the two data sets to have close distributions. For instance, if the combined private data set  $D$  is CIFAR10, the public data set can be chosen as CIFAR100. In our study, we set both the public and private data sets to come from the same distribution. However, investigating the relationship between the effectiveness of KD and the similarity of the private and public data sets distribution is an exciting research direction.

*Path Planning:* Due to the small number of households that form the nodes in our graph, we used an unsupervised deep

**TABLE IX**  
PERFORMANCE OF THE RANDOM SOLUTION OF TSP (CORRESPONDING TO 0 ITERATIONS OF THE SOM ALGORITHM)

City map	Path Generation Time (Sec)	Path Distance	Drone Distance	Quality (APE)
Canada	5.30	4.70e+07	9.03e+07	3542.26
Egypt	8.68	5.60e+06	1.18e+07	3149.74
Qatar	0.24	6.65e+04	1.31e+05	611.08

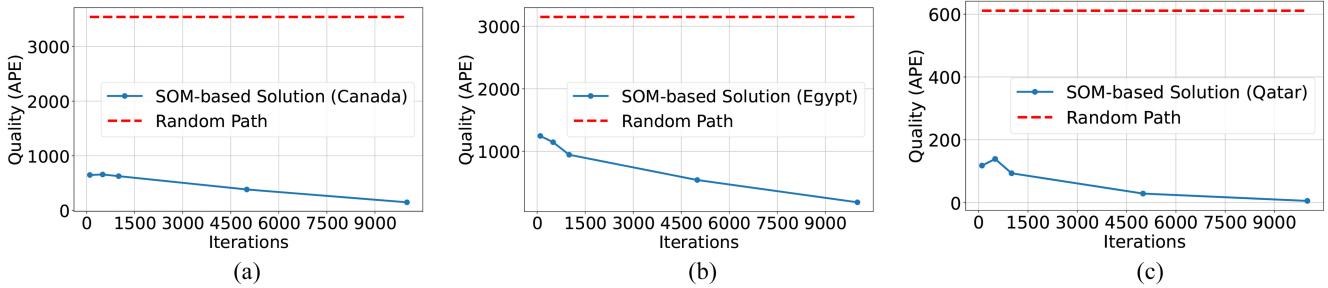


Fig. 9. Quality of drone path generated by our SOM-based algorithm versus the randomly generated drone path for three city maps. Quality is evaluated using APE. (a) APE for random path versus SOM path for Canada city map. (b) APE for random path versus SOM path for Egypt city map. (c) APE for random path versus SOM path for Qatar city map.

**TABLE X**  
PERFORMANCE OF THE THREE CONSIDERED FL METHODS ON FOUR HAR DATA SETS. *K*, *Comp*, *Aug*, AND *Weighting* REFER TO THE FRACTION OF CLIENTS PARTICIPATING IN FL AT ANY ROUND, COMPRESSION USING QUANTIZATION, WHETHER MIXUP AUGMENTATION IS APPLIED, AND THE WEIGHTING SCHEME USED TO WEIGHT USERS' SLs, RESPECTIVELY

Method	K	Comp	Aug	Weighting	HARS	HARBOX	IMU	Depth
<b>FedMD</b>	0.5	FALSE	FALSE	uniform	0.75	0.54	0.77	0.61
	0.5	FALSE	FALSE	test accuracy	0.78	0.55	0.79	0.71
	1	FALSE	FALSE	uniform	<b>0.9</b>	0.57	0.98	0.66
	1	FALSE	FALSE	test accuracy	<b>0.9</b>	0.52	0.96	0.66
	0.5	TRUE	FALSE	uniform	0.7	0.6	0.71	0.74
	0.5	TRUE	FALSE	test accuracy	0.78	0.62	0.78	0.71
<b>FedAKD</b>	1	FALSE	TRUE	uniform	0.87	0.83	0.84	0.69
	1	FALSE	TRUE	test accuracy	0.86	<b>0.86</b>	0.89	0.69
	1	TRUE	TRUE	uniform	0.89	0.85	0.88	0.69
	1	TRUE	TRUE	test accuracy	0.88	0.84	<b>0.97</b>	<b>0.75</b>
<b>FedAvg</b>	0.5	N/A	N/A	uniform	0.84	0.78	0.95	0.7
	0.5	N/A	N/A	test accuracy	0.83	0.82	0.97	0.73
	1	N/A	N/A	uniform	0.89	0.84	0.93	0.7
	1	N/A	N/A	test accuracy	0.89	0.83	<b>0.99</b>	0.73

learning-based optimization solution (SOM) for its simplicity and robustness. Comparing our implementation with other optimization frameworks like exact and heuristics algorithms is considered future work.

## IX. CONCLUSION

In this article, we proposed a DORA that provides communication-efficient FL for the purpose of health monitoring in rural areas lacking Internet connectivity. The DORA system consists of two main components: 1) drone path planning and 2) lightweight FL. We formulate the drone path as a TSP and employ SOM to find an efficient cycle that traverses all nodes. From the obtained solution, we theoretically and experimentally derive the path the drone should follow to perform FL. For the second part, we describe

the model-based FL traffic model and present the KD-based FL traffic model which is divided into two phases leveraging KD to transfer knowledge across nodes in the form of SLs instead of model weights. We introduce a communication-efficient KD-based FL algorithm: CFedAKD and test the communication overhead of CFedAKD compared to other KD-based FL methods that rely on SLs and standard FL (FedAvg). Our results show that the compressed SLs used in CFedAKD as a basis for communication are only 3 KB, compared to 12 KB for other KD-based FL methods and over 250 KB for standard FL (FedAvg). Our results show the computational efficiency of SOM in finding a high-quality TSP route, as a distance to precalculated optimal routes on multiple multiscale maps. The communication cost of KD-based FL enables it to run on low-bandwidth networks. Finally, the performance of baseline FL algorithms was evaluated on

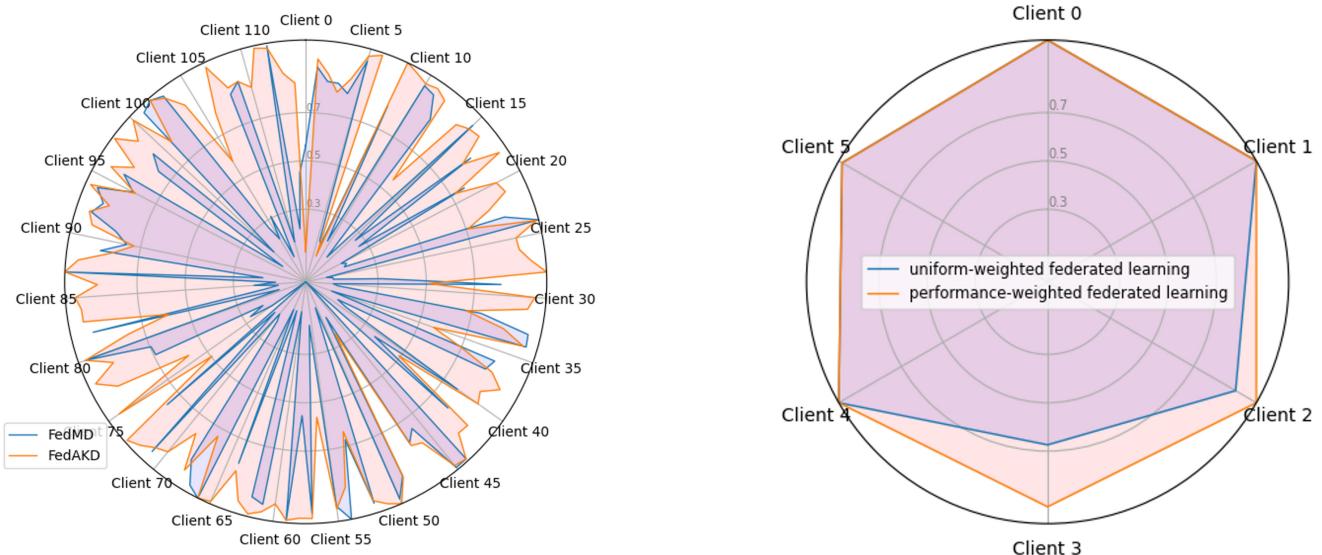


Fig. 10. Left: Per-client accuracy under two KD-based FL algorithms on the Harbox data set: FedMD (blue) and FedAKD (red). Right: Per-client accuracy under FedAvg using different weighting schemes on the HARS data set: uniform weighting versus performance-based weighting. The covered area by each algorithm indicates the achieved accuracy. We can notice that FedAKD and performance-based averaging boost the performance of a larger number of clients (the red area is larger than the blue area).

HAR data sets. Results show that KD-based FL achieves competitive performance relative to model-based FL methods with significantly less communication overhead leveraging a shared data set to transfer knowledge across the network.

## REFERENCES

- [1] N. S. Labib, M. R. Brust, G. Danoy, and P. Bouvry, "The rise of drones in Internet of Things: A survey on the evolution, prospects and challenges of unmanned aerial vehicles," *IEEE Access*, vol. 9, pp. 115466–115487, 2021, doi: [10.1109/ACCESS.2021.3104963](https://doi.org/10.1109/ACCESS.2021.3104963).
- [2] S. Verma, Y. Kawamoto, Z. M. Fadlullah, H. Nishiyama, and N. Kato, "A survey on network methodologies for real-time analytics of massive IoT data and open research issues," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1457–1477, 3rd Quart., 2017, doi: [10.1109/COMST.2017.2694469](https://doi.org/10.1109/COMST.2017.2694469).
- [3] Z. Wei et al., "UAV-assisted data collection for Internet of Things: A survey," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 15460–15483, Sep. 2022, doi: [10.1109/JIOT.2022.3176903](https://doi.org/10.1109/JIOT.2022.3176903).
- [4] N. Cheng et al., "AI for UAV-assisted IoT applications: A comprehensive review," *IEEE Internet Things J.*, vol. 10, no. 16, pp. 14438–14461, Aug. 2023, doi: [10.1109/IOT.2023.3268316](https://doi.org/10.1109/IOT.2023.3268316).
- [5] J. Tan, X. Dai, F. Tang, M. Zhao, and N. Kato, "Intelligent configuration method based on UAV-driven frequency selective surface for communication band shielding," *IEEE Internet Things J.*, vol. 10, no. 13, pp. 11507–11517, Jul. 2023, doi: [10.1109/IJOT.2023.3243387](https://doi.org/10.1109/IJOT.2023.3243387).
- [6] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," *ICT Express*, vol. 5, no. 1, pp. 1–7, 2019, doi: [10.1016/j.icte.2017.12.005](https://doi.org/10.1016/j.icte.2017.12.005).
- [7] S. Sankarasrinivasan, E. Balasubramanian, K. Karthik, U. Chandrasekar, and R. Gupta, "Health monitoring of civil structures with integrated UAV and image processing system," *Proced. Comput. Sci.*, vol. 54, pp. 508–515, Jan. 2015, doi: [10.1016/j.procs.2015.06.058](https://doi.org/10.1016/j.procs.2015.06.058).
- [8] A. Rahmadhani, R. Isswandhana, A. Giovani, and R. A. Syah, "LoRaWAN as secondary telemetry communication system for drone delivery," in *Proc. IEEE Int. Conf. Internet Things Intell. Syst. (IOTAIS)*, 2018, pp. 116–122, doi: [10.1109/IOTAIS.2018.8600892](https://doi.org/10.1109/IOTAIS.2018.8600892).
- [9] A. Pagano, D. Croce, I. Tinnirello, and G. Vitale, "A survey on LoRa for smart agriculture: Current trends and future perspectives," *IEEE Internet Things J.*, vol. 10, no. 4, pp. 3664–3679, Feb. 2023, doi: [10.1109/IJOT.2022.3230505](https://doi.org/10.1109/IJOT.2022.3230505).
- [10] V. Delafontaine, F. Schiano, G. Cocco, A. Rusu, and D. Floreano, "Drone-aided localization in LoRa IoT networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2020, pp. 286–292, doi: [10.1109/ICRA40945.2020.9196869](https://doi.org/10.1109/ICRA40945.2020.9196869).
- [11] S. Park, S. Yun, H. Kim, R. Kwon, J. Ganser, and S. Anthony, "Forestry monitoring system using LoRa and drone," in *Proc. 8th Int. Conf. Web Intell. Min. Semant. (WIMS)*, 2018, pp. 1–8, doi: [10.1145/3227609.3227677](https://doi.org/10.1145/3227609.3227677).
- [12] A. O. Hashesh, S. Hashima, R. M. Zaki, M. M. Fouda, K. Hatano, and A. S. T. Eldien, "AI-enabled UAV communications: Challenges and future directions," *IEEE Access*, vol. 10, pp. 92048–92066, 2022, doi: [10.1109/ACCESS.2022.3202956](https://doi.org/10.1109/ACCESS.2022.3202956).
- [13] T. Tazrin, M. M. Fouda, Z. M. Fadlullah, and N. Nasser, "UV-CDS: An energy-efficient scheduling of UAVs for premises sterilization," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 3, pp. 1191–1201, Sep. 2021, doi: [10.1109/TGCN.2021.3074536](https://doi.org/10.1109/TGCN.2021.3074536).
- [14] T. Tazrin, Z. Ayaz, M. M. Fouda, and Z. M. Fadlullah, "On energy-efficient UAV route scheduling to offload health data from under-served rural communities," in *Proc. IEEE Int. Conf. Internet Things Intell. Syst. (IoTaIS)*, 2021, pp. 86–91, doi: [10.1109/IoTaIS53735.2021.9628751](https://doi.org/10.1109/IoTaIS53735.2021.9628751).
- [15] C. T. Dinh, N. H. Tran, and T. D. Nguyen, "Personalized federated learning with Moreau envelopes," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 1–12.
- [16] Z. Md. Fadlullah and N. Kato, "HCP: Heterogeneous computing platform for federated learning based collaborative content caching towards 6G networks," *IEEE Trans. Emerg. Topics Comput.*, vol. 10, no. 1, pp. 112–123, Jan.–Mar. 2022, doi: [10.1109/TETC.2020.2986238](https://doi.org/10.1109/TETC.2020.2986238).
- [17] I. Donevski, N. Babu, J. J. Nielsen, P. Popovski, and W. Saad, "Federated learning with a drone orchestrator: Path planning for minimized staleness," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 1000–1014, 2021, doi: [10.1109/OJCOMS.2021.3072003](https://doi.org/10.1109/OJCOMS.2021.3072003).
- [18] Z. M. Fadlullah and N. Kato, "On smart IoT remote sensing over integrated terrestrial-aerial-space networks: An asynchronous federated learning approach," *IEEE Netw.*, vol. 35, no. 5, pp. 129–135, Sep./Oct. 2021, doi: [10.1109/MNET.101.2100125](https://doi.org/10.1109/MNET.101.2100125).
- [19] K. B. Laksham, "Unmanned aerial vehicle (drones) in public health: A SWOT analysis," *J. Family Med. Prim. Care*, vol. 8, no. 2, pp. 342–346, 2019, doi: [10.4103/jfmpc.jfmpc\\_413\\_18](https://doi.org/10.4103/jfmpc.jfmpc_413_18).
- [20] B. Hiebert, E. Nouvet, V. Jeyabalan, and L. Donelle, "The application of drones in healthcare and health-related services in north america: A scoping review," *Drones*, vol. 4, no. 3, p. 30, 2020, doi: [10.3390/drones4030030](https://doi.org/10.3390/drones4030030).
- [21] J.-M. Martinez-Caro and M.-D. Cano, "IoT system integrating unmanned aerial vehicles and LoRa technology: A performance evaluation study," *Wireless Commun. Mobile Comput.*, vol. 2019, Nov. 2019, Art. no. e4307925, doi: [10.1155/2019/4307925](https://doi.org/10.1155/2019/4307925).
- [22] L. Angrisani et al., "An innovative air quality monitoring system based on drone and IoT enabling technologies," in *Proc. IEEE Int. Workshop Metrol. Agric. Forest. (MetroAgriFor)*, 2019, pp. 207–211, doi: [10.1109/MetroAgriFor.2019.8909245](https://doi.org/10.1109/MetroAgriFor.2019.8909245).

- [23] L.-Y. Chen, H.-S. Huang, C.-J. Wu, Y.-T. Tsai, and Y.-S. Chang, "A LoRa-based air quality monitor on unmanned aerial vehicle for smart city," in *Proc. Int. Conf. Syst. Sci. Eng. (ICSSE)*, 2018, pp. 1–5, doi: [10.1109/ICSSE.2018.8519967](https://doi.org/10.1109/ICSSE.2018.8519967).
- [24] M. Zhang and X. Li, "Drone-enabled Internet of Things relay for environmental monitoring in remote areas without public networks," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7648–7662, Aug. 2020, doi: [10.1109/JIOT.2020.2988249](https://doi.org/10.1109/JIOT.2020.2988249).
- [25] M. Behjati, A. B. Mohd Noh, H. A. H. AlObaidy, M. A. Zulkifley, R. Nordin, and N. F. Abdullah, "LoRa communications as an enabler for Internet of drones towards large-scale livestock monitoring in rural farms," *Sensors*, vol. 21, no. 15, p. 5044, 2021, doi: [10.3390/s21155044](https://doi.org/10.3390/s21155044).
- [26] A. Caruso, S. Chessa, S. Escobar, J. Barba, and J. C. López, "Collection of data with drones in precision agriculture: Analytical model and LoRa case study," *IEEE Internet Things J.*, vol. 8, no. 22, pp. 16692–16704, Nov. 2021, doi: [10.1109/JIOT.2021.3075561](https://doi.org/10.1109/JIOT.2021.3075561).
- [27] M. A. Ahmed et al., "LoRa based IoT platform for remote monitoring of large-scale agriculture farms in Chile," *Sensors*, vol. 22, no. 8, p. 2824, 2022, doi: [10.3390/s22082824](https://doi.org/10.3390/s22082824).
- [28] T. Jain, A. Sibley, H. Stryhn, and I. Hubloue, "Comparison of unmanned aerial vehicle technology-assisted triage versus standard practice in triaging casualties by paramedic students in a mass-casualty incident scenario," *Prehosp. Disaster Med.*, vol. 33, no. 4, pp. 375–380, 2018, doi: [10.1017/S1049023X18000559](https://doi.org/10.1017/S1049023X18000559).
- [29] M. Al Zayer, S. Tregillus, J. Bhandari, D. Feil-Seifer, and E. Folmer, "Exploring the use of a drone to guide blind runners," in *Proc. 18th Int. ACM SIGACCESS Conf. Comput. Access.*, 2016, pp. 263–264, doi: [10.1145/2982142.2982204](https://doi.org/10.1145/2982142.2982204).
- [30] M. Balasingam, "Drones in medicinethe rise of the machines," *Int. J. Clin. Pract.*, vol. 71, no. 9, 2017, Art. no. e12989, doi: [10.1111/ijcp.12989](https://doi.org/10.1111/ijcp.12989).
- [31] V. Sharma, I. You, G. Pau, M. Collotta, J. D. Lim, and J. N. Kim, "LoRaWAN-based energy-efficient surveillance by drones for intelligent transportation systems," *Energies*, vol. 11, no. 3, p. 573, 2018, doi: [10.3390/en11030573](https://doi.org/10.3390/en11030573).
- [32] J. Mfitumukiza, V. Mariappan, M. Lee, J. Cho, and J. Cha, "A study on F8L10D-N LoRa RF module for drone based live broadcasting system," *Int. J. Adv. Cult. Technol.*, vol. 4, no. 4, pp. 1–5, 2016, doi: [10.17703/IJACT.2016.4.4.1](https://doi.org/10.17703/IJACT.2016.4.4.1).
- [33] O. A. Sarareh, A. Alsaraira, I. Khan, and P. Uthansakul, "Performance evaluation of UAV-enabled LoRa networks for disaster management applications," *Sensors*, vol. 20, no. 8, p. 2396, 2020, doi: [10.3390/s20082396](https://doi.org/10.3390/s20082396).
- [34] R. Xiong, C. Liang, H. Zhang, X. Xu, and J. Luo, "FlyingLoRa: Towards energy efficient data collection in UAV-assisted LoRa networks," *Comput. Netw.*, vol. 220, Jan. 2023, Art. no. 109511, doi: [10.1016/j.comnet.2022.109511](https://doi.org/10.1016/j.comnet.2022.109511).
- [35] Z. Zhang, C. Zhou, L. Sheng, and S. Cao, "Optimization schemes for UAV data collection with LoRa 2.4 GHz technology in remote areas without infrastructure," *Drones*, vol. 6, no. 7, p. 173, 2022, doi: [10.3390/drones6070173](https://doi.org/10.3390/drones6070173).
- [36] V. A. Dambal, S. Mohadikar, A. Kumbhar, and I. Guvenc, "Improving LoRa signal coverage in urban and sub-urban environments with UAVs," in *Proc. Int. Workshop Antenna Technol. (iWAT)*, 2019, pp. 210–213, doi: [10.1109/iWAT.2019.8730598](https://doi.org/10.1109/iWAT.2019.8730598).
- [37] T. K. Rodrigues and N. Kato, "Hybrid centralized and distributed learning for MEC-equipped satellite 6G networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 4, pp. 1201–1211, Apr. 2023, doi: [10.1109/JSAC.2023.3242700](https://doi.org/10.1109/JSAC.2023.3242700).
- [38] M. M. Badr et al., "Privacy-preserving federated-learning-based net-energy forecasting," in *Proc. SoutheastCon*, 2022, pp. 133–139, doi: [10.1109/SoutheastCon48659.2022.9764093](https://doi.org/10.1109/SoutheastCon48659.2022.9764093).
- [39] N. Nasser, Z. M. Fadlullah, M. M. Fouda, A. Ali, and M. Imran, "A lightweight federated learning based privacy preserving 5G pandemic response network using unmanned aerial vehicles: A proof-of-concept," *Comput. Netw.*, vol. 205, Mar. 2022, Art. no. 108672, doi: [10.1016/j.comnet.2021.108672](https://doi.org/10.1016/j.comnet.2021.108672).
- [40] K. Bedda, Z. M. Fadlullah, and M. M. Fouda, "Efficient wireless network slicing in 5G networks: An asynchronous federated learning approach," in *Proc. IEEE Int. Conf. Internet Things Intell. Syst. (IoTaIS)*, 2022, pp. 285–289, doi: [10.1109/IoTaIS56727.2022.9976007](https://doi.org/10.1109/IoTaIS56727.2022.9976007).
- [41] M. I. Ibrahim, M. Mahmoud, M. M. Fouda, B. M. ElHalawany, and W. Alasmari, "Privacy-preserving and efficient decentralized federated learning-based energy theft detector," in *Proc. IEEE Glob. Commun. Conf. GLOBECOM*, 2022, pp. 287–292, doi: [10.1109/GLOBECOM48099.2022.10000881](https://doi.org/10.1109/GLOBECOM48099.2022.10000881).
- [42] Y. Gupta, Z. M. Fadlullah, and M. M. Fouda, "Toward asynchronously weight updating federated learning for AI-on-edge IoT systems," in *Proc. IEEE Int. Conf. Internet Things Intell. Syst. (IoTaIS)*, 2022, pp. 358–336, doi: [10.1109/IoTaIS56727.2022.9975908](https://doi.org/10.1109/IoTaIS56727.2022.9975908).
- [43] S. Sakib, M. M. Fouda, Z. Md Fadlullah, and N. Nasser, "On COVID-19 prediction using asynchronous federated learning-based agile radiograph screening booths," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2021, pp. 1–6, doi: [10.1109/ICC42927.2021.9500351](https://doi.org/10.1109/ICC42927.2021.9500351).
- [44] M. Zeng et al., "Convolutional neural networks for human activity recognition using mobile sensors," in *Proc. 6th Int. Conf. Mobile Comput., Appl. Services*, 2014, pp. 197–205, doi: [10.4108/icst.mobicase.2014.257786](https://doi.org/10.4108/icst.mobicase.2014.257786).
- [45] H. Cho, A. Mathur, and F. Kawsar, "FLAME: Federated learning across multi-device environments," *ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 6, no. 3, pp. 1–29, 2022, doi: [10.1145/3550289](https://doi.org/10.1145/3550289).
- [46] R. Presotto, G. Civitarese, and C. Bettini, "Semi-supervised and personalized federated activity recognition based on active learning and label propagation," *Pers. Ubiquitous Comput.*, vol. 26, no. 5, pp. 1281–1298, 2022, doi: [10.1007/s00779-022-01688-8](https://doi.org/10.1007/s00779-022-01688-8).
- [47] Ł. Brocki and D. Koržinek, "Kohonen self-organizing map for the traveling salesperson problem," in *Proc. Recent Adv. Mechatron.*, 2007, pp. 116–119, doi: [10.1007/978-3-540-73956-2\\_24](https://doi.org/10.1007/978-3-540-73956-2_24).
- [48] J. Zhu, H. Ye, L. Yao, and Y. Cai, "Algorithm for solving traveling salesman problem based on self-organizing mapping network," *J. Shanghai Jiaotong Univ. (Sci.)*, early access, Nov. 12, 2022, doi: [10.1007/s12204-022-2517-3](https://doi.org/10.1007/s12204-022-2517-3).
- [49] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," 2023, *arXiv:1602.05629*.
- [50] D. Li and J. Wang, "FedMD: Heterogenous federated learning via model distillation," 2019, *arXiv:1910.03581*.
- [51] T. Kim, J. Oh, N. Kim, S. Cho, and S.-Y. Yun, "Comparing Kullback–Leibler divergence and mean squared error loss in knowledge distillation," 2021, *arXiv:2105.08919*.
- [52] G. Gad and Z. Fadlullah, "Federated learning via augmented knowledge distillation for heterogenous deep human activity recognition systems," *Sensors*, vol. 23, no. 1, p. 6, 2022, doi: [10.3390/s23010006](https://doi.org/10.3390/s23010006).
- [53] G. Gad, "Light-weight federated learning with augmented knowledge distillation for human activity recognition," M.S. thesis, Artif. Intell., Lakehead Univ., Thunder Bay, ON, Canada, 2023.
- [54] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," 2017, *arXiv:1710.09412*.
- [55] G. Gad, Z. M. Fadlullah, K. Rabie, and M. M. Fouda, "Communication-efficient privacy-preserving federated learning via knowledge distillation for human activity recognition systems," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2023, pp. 1572–1578, doi: [10.1109/ICC45041.2023.10278987](https://doi.org/10.1109/ICC45041.2023.10278987).
- [56] Rayex Technol. Co., Ltd. Co., Barrington, IL, USA). *RYLR896, 868/915-MHz LoRa® Antenna Transceiver Module*. Accessed: Feb. 20, 2023. [Online]. Available: <https://rayex.com/products/RYLR896>
- [57] "Raspberry Pi documentation-configuration." RaspberryPi.com, Accessed: Feb. 20, 2023. [Online]. Available: <https://www.raspberrypi.com/documentation/computers/configuration.html>,
- [58] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach*, 6th ed., London, U.K.: Pearson, 2012.
- [59] G. Callebaut and L. Van der Perre, "Characterization of LoRa point-to-point path loss: Measurement campaigns and modeling considering censored data," *IEEE Internet Things J.*, vol. 7, no. 3, pp. 1910–1918, Mar. 2020, doi: [10.1109/JIOT.2019.2953804](https://doi.org/10.1109/JIOT.2019.2953804).
- [60] "AN1200.22 LoRa™ modulation basics," Application Note, Semtech Corp., Camarillo, CA, USA, 2015.



**Gad Gad** (Graduate Student Member, IEEE) received the bachelor's degree in computer engineering from Nile University, Giza, Egypt, in 2021, and the master's degree in computer science from Lakehead University, Thunder Bay, ON, Canada, in 2023. He is currently pursuing the Ph.D. degree in computer science with Western University, London, ON, Canada.

His main research interests are in the areas of deep learning, federated learning, and differential privacy.

Dr. Gad is a Vector Institute AI Scholarship recipient, and he received a Best Poster Award at the NRSC 2020 Conference.



**Aya Farrag** (Graduate Student Member, IEEE) received the B.Sc. degree in electrical engineering with a specialization in communications and electronics from Alexandria University, Alexandria, Egypt, in 2018. She is currently pursuing the master's degree in computer science with Lakehead University, Thunder Bay, ON, Canada.

She is also a Teaching Assistant with Lakehead University. Her research interests are in the areas of machine learning, health analytics, and natural language processing.



**Ahmed Aboulfotouh** (Member, IEEE) received the B.Sc. degree (Hons.) in communications and information engineering from the University of Science and Technology, Gardens City, Egypt, in 2020. He is currently pursuing the M.Sc. degree in artificial intelligence with Lakehead University, Thunder Bay, ON, Canada.

He is also a Teaching Assistant with Lakehead University. His current research interests include artificial intelligence for physical-layer communication, cell-free massive MIMO, beyond massive MIMO, and semantic communication.



**Khaled Bedda** (Member, IEEE) received the bachelor's and B.Sc. (Hons.) degrees in biomedical engineering and systems from Cairo University, Giza, Egypt, in 2021. He is currently pursuing the master's degree in computer science (specialization in artificial intelligence) with Lakehead University, Thunder Bay, ON, Canada.

His research interests rely on the interdisciplinary area between deep learning, computer vision, healthcare analytics, and the Internet of Things.



**Zubair Md. Fadlullah** (Senior Member, IEEE) received the bachelor's degree in computer science and information technology from the Islamic University of Technology (IUT), Gazipur City, Bangladesh, in 2003, and the M.Sc. and Ph.D. degrees in applied information sciences from Tohoku University, Sendai, Japan, in 2008 and 2011, respectively.

He is currently an Associate Professor with the Computer Science Department, Western University, London, ON, Canada. He was an Associate Professor with Lakehead University, Thunder Bay, ON, Canada, from 2019 to 2022. Prior to that, he was an Associate Professor with the Graduate School of Information Sciences, Tohoku University, Sendai, Japan, from 2017 to 2019 and an Assistant Professor from 2011 to 2017. His main research interests are in the areas of emerging communication systems.

Dr. Fadlullah was a recipient of the Prestigious Dean's and President's Awards from Tohoku University in March 2011, and the IEEE Asia-Pacific Outstanding Researcher Award in 2015 and NEC Tokin Award for Research in 2016, for his outstanding contributions. He received several best paper awards at conferences, including IEEE/ACM IWCMC, IEEE GLOBECOM, and IEEE IC-NIDC. He is currently an Editor of IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY, and IEEE ACCESS.



**Mostafa M. Fouada** (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Benha University, Benha, Egypt, in 2002 and 2007, respectively, and the Ph.D. degree in information sciences from Tohoku University, Sendai, Japan, in 2011.

He is currently an Associate Professor with the Department of Electrical and Computer Engineering, Idaho State University, Pocatello, ID, USA. He is also a Full Professor with Benha University. He was an Assistant Professor with Tohoku University and a Postdoctoral Research Associate with Tennessee Technological University, Cookeville, TN, USA. He has (co)authored more than 220 technical publications. His current research focuses on cybersecurity, communication networks, signal processing, wireless mobile communications, smart healthcare, smart grids, AI, and IoT.

Dr. Fouada has guest edited several special issues covering various emerging topics in communications, networking, and health analytics. He is currently serving on the editorial boards for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE INTERNET OF THINGS JOURNAL, and IEEE ACCESS. He has received several research grants, including NSF Japan-U.S. Network Opportunity 3.