2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) | 978-1-6654-6483-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/PIMRC56721.2023.10294036

Communication-Efficient Federated Learning in Drone-Assisted IoT Networks: Path Planning and Enhanced Knowledge Distillation Techniques

Gad Gad*1, Aya Farrag*2, Zubair Md Fadlullah‡3, and Mostafa M. Fouda¶‖4.
*Department of Computer Science, Lakehead University, Thunder Bay, Ontario, Canada.
‡Department of Computer Science, Western University, London, ON, Canada.
¶Department of Electrical and Computer Engineering, Idaho State University, Pocatello, ID, USA.
‖Center for Advanced Energy Studies (CAES), Idaho Falls, ID, USA.
Emails: 1ggad@lakeheadu.ca, 2afarrag@lakeheadu.ca, 3zfadlullah@ieee.org, 4mfouda@ieee.org

***Abstract*—As 5G and beyond networks continue to proliferate, intelligent monitoring systems are becoming increasingly prevalent. However, geographically isolated regions with sparse populations still face difficulties in accessing these technologies due to infrastructure deployment challenges. Additionally, the high cost and unreliability of satellite Internet services make them less appealing. This paper studies the challenges of drone-aided networks and presents a communication-efficient Federated Learning (FL) system on a drone-aided Internet of Things (IoT) network to facilitate health analysis in rural areas over LoRa wireless links. The proposed approach consists of two primary components. Firstly, optimizing the drone's trajectory is theoretically formulated as a modified version of the Traveling Salesman Problem (TSP), with the Self-Organizing Map (SOM) algorithm employed for effective route planning. Secondly, the Knowledge Distillation (KD)-based FL algorithm is utilized to reduce communication overhead by leveraging soft labels. The quality of drone routes generated by the SOM is evaluated on multi-scale maps with pre-determined optimal paths. The experiments reveal SOM's ability to accurately represent node topologies and yield cost-effective Hamiltonian cycles. The KD-based FL proves to be more efficient in terms of communication than FedAvg as the former exchanges soft labels while the latter exchanges model weights, thus reducing drone waiting time and battery consumption. We showcase the performance of our KD-based FL algorithm using Human Activity Recognition (HAR) datasets, illustrating a communication-efficient alternative for distributed learning, offering competitive performance leveraging a shared dataset for knowledge transfer among IoT devices.**

***Index Terms*—Deep learning, UAV networks, drone-aided LoRa networks, edge devices, Federated Learning (FL), digital health and well-being, Knowledge Distillation, Self-Organizing Maps.**

I. Introduction

Drones, as adaptable aerial small cells (AASCs), provide significant wireless communication support to geographically dispersed nodes lacking high bandwidth terrestrial links. Recently, distributed learning in drone-assisted Internet of Things (IoT) networks via Machine/Deep Learning (ML/DL) techniques emerged as a hot research topic. The appeal of ML/DL techniques for wireless and Internet of Things (IoT) networks stems from the extensive traffic data and limitations of conventional model-based solutions in managing dynamic complexity [1]. Conventional ML frameworks, however, are cloud-focused and not well-suited for drone-supported networks due to privacy concerns, along with high network bandwidth and energy consumption requirements for data transmission. Federated Learning (FL) offers a decentralized approach enabling distributed DL model training while preserving private data and reducing network overhead [2]–[4]. Nonetheless, FL communication costs remain high for low-bandwidth IoT networks. Knowledge Distillation (KD)-based FL techniques provide a communication-efficient alternative to model-based approaches, such as [5]. Adapting KD for FL involves assuming a shared dataset, aggregating local soft labels into global ones, and broadcasting them for clients training.

Considering the power, computing, and bandwidth constraints, KD-based FL is more suitable for drone-supported networks than model-based FL, preserving data privacy and minimizing network overhead and latency [6]. Our work in this paper also explores the path planning problem in drone-enabled networks and proposes a solution based on Self-Organizing Maps (SOM) to guide the drone on a cost-efficient route according to network topology. The main contributions of our work are as follows.

- We establish a drone-assisted LoRa network, referred to as DORA, to facilitate FL in rural regions with inadequate Internet infrastructures. Our proposed DORA system consists of two sub-systems, namely drone path planning and communication-efficient KD-based FL.
- We envision a Compressed Federated Learning via Augmented Knowledge Distillation (CFedAKD) algorithm, which utilizes compression techniques to compress soft labels to minimize communication overheads. The envisioned CFedAKD algorithm accommodates client model architectures designed independently, an essential feature when working with edge devices possessing diverse hardware resources.

The remainder of the paper is structured as follows. A survey of relevant research work is presented in section II. Section III introduces the proposed drone-assisted LoRa ("long range") network system model. Section IV defines the drone path optimization problem as a modified TSP and elaborates on our method for determining the optimal path using the Self-Organizing Map (SOM) algorithm. Section V presents the traffic models for both model-based FL algorithms and the proposed KD-based FL algorithm. Section VI covers the experiments and performance evaluation, including path optimization using SOM, and FL simulations. Finally, section VII concludes the paper.

II. Related Work

Federated Learning, commonly referred to as FL in the contemporary literature, is an emerging technique for decentralized, privacy-preserving model training from private data, with applications across privacy-sensitive domains [5], [7]–[9]. A major challenge in FL consists in the communication cost incurred during model updates between the server and clients, particularly for large models. On the other hand, efforts have been directed toward improving communication efficiency via gradient compression [10], though this approach may result in performance degradation with high compression ratios. Knowledge Distillation (KD) [11], [12] offers an alternative by utilizing a trained model (teacher) to inform the training of a smaller model. KD has been integrated into FL frameworks, such as FedKD [13] and FedMD [14], to address system heterogeneity and communication overheads. In FedKD, each client employs a local mentor model and a shared mentee model, where the latter is collaboratively learned to minimize communication costs. FedMD uses a shared public dataset for distilling knowledge, with clients calculating soft labels for server exchange instead of model weights. Gad et al. [15], [16] built on this approach, incorporating Mixup augmentation [17] to enhance distillation performance.

Next, researchers have demonstrated the efficacy of Unmanned Aerial Vehicle (UAV)-enabled LoRa networks in monitoring tasks, such as air quality [18]–[20], environmental monitoring in remote areas [21], livestock monitoring [22], and forestry and precision agriculture [23]–[25]. Despite the demonstrated utility of drone-aided LoRa networks, several challenges have arisen in their design. To address energy consumption, researchers in [26] proposes a method optimizing 3D UAV trajectory, scheduling, and transmission parameters. To reduce flight distance and time, the work in [27] introduced optimization schemes and exhibited a 30% reduction in UAV flight distance using their Teaching–Learning-based Genetic Algorithm (TGA). Furthermore, the optimal UAV deployment height was investigated in [24], [28]. However, in the aforementioned research efforts, a joint consideration of privacy-preservation and UAV path optimization was not taken into consideration. We identify this research gap in the following section.

III. System Model

We examine a target area (TA) depicted by an $M \times M$ grid, including N_h residential nodes $h \in H = 1, 2, \ldots, N_h$. Residence h has spatial coordinates $\mathbf{q_h} = (x_h, y_h)$. The grid is separated into M^2 squares, each with a side length of s meters, with a total side length $S = (M \cdot s)$. Each residence has an access point AP_h, comprising a microcontroller (e.g., a Raspberry Pi board) and a low-power IoT (i.e., LoRa) module.

A drone with an access point AP_d flies within the TA, starting from point A at $\mathbf{q_A} = (x_A, y_A)$ and returning to the same point. To perform FL, the drone flight consists of two stages: the Uploading Soft Labels (USL) phase and the Downloading Soft Labels (DSL) phase. In the former phase, the drone stops at each node $q_h, \forall h \in H$ to acquire FL updates via LoRa communication. Once the drone gathers updates from the final node, the DSL phase commences, where the drone retraces the USL path, pausing at each node to download the aggregated updates via LoRa communication, and finally returning to the starting point A. Let T_{USL} and T_{DSL} represent two equal time durations in which the drone finishes the USL and DSL phases, respectively, while covering linear distances of D_{USL} and D_{DSL}. Then, the total time T and total distance covered D are given by:

$$T = T_{USL} + T_{DSL}, \quad D = D_{USL} + D_{DSL}. \tag{1}$$

The drone concludes one global FL round per journey, divided between two identical paths for uploading and downloading FL messages between AP_h (the server) and $AP_h \forall h \in H$ (the clients). The total time T is divided into N_t equal time slots $n \in 1, 2, \ldots, N_t$, each lasting t seconds, such that $T = (N_t \cdot t)$ and $T_{USL} = T_{DSL} = (\frac{N}{2} \cdot t)$. The drone's velocity V and altitude VD are constant, and it follows a path $\mathbf{P_d} = \mathbf{q_{d,n}} \in \mathbb{R}^{N_t \times 2}, \forall n \in 1, 2, \ldots, N_t$, a series of coordinates throughout its route. $\mathbf{q}_n^d$ denotes the drone's coordinates at time n. The first and last points share the same coordinates: $\mathbf{q_A} = \mathbf{q_{d,0}} = \mathbf{q_{d,N_t}} = (x_A, y_A)$. The distance from the drone to node k is given by:

$$d(k, \mathbf{p_{d,n}}) = \sqrt{(x_{d,n} - x_k)^2 + (y_{d,n} - y_k)^2}. \tag{2}$$

We establish a circular Coverage Area CA_k with a radius $r = \frac{s}{2}$, centered at $\mathbf{p_k}$ for all $k \in K$. A drone access point AP_d can only interact with another access point AP_d if the distance between them is less than the threshold distance r:

$$dr(k, \mathbf{p_{d,n}}) = \begin{cases} R_b, & \text{if } d(k, \mathbf{p_{d,n}}) \leq r \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

where $dr(k, \mathbf{p_{d,n}})$ denotes the data rate for client k at time n, and R_b represents the nominal data rate of the LoRa communication module. The data rate depends on $\mathbf{p_{d,n}}$ since it alters as the drone moves horizontally across the map.

Due to LoRa module's low bandwidth and high drop rate, we determine a coverage area (CA) with radius r, where the signal quality surpasses the receiver antenna's sensitivity, making it suitable for transmitting FL messages. This guarantees that the drone access point AP_d communicates with access point AP_k only when the distance is less than the threshold distance r, as demonstrated in eq. 3. We set the CA radius $r = 50$ meters based on our drone-LoRa distance test, using the parameters specified in Table I.

TABLE I
LoRa parameters.

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

Considering the drone's limited resources, it is crucial to minimize the total trip time T. We address this by minimizing of total linear distance D covered by the drone, and the size of FL updates MS. In the case of FedAvg, MS would be the model weights sent by clients. However, in KD-based FL, MS represents soft labels.

To minimize distance D, we consider the drone path as a graph theory problem. The drone aims to traverse a graph $G = (V, E)$, where $V = v_1, v_2, \ldots, v_{N_v}$ comprises $N_v = K+1$ vertices ($AP_i, \forall i \in I$ and the starting point A with coordinates x_A, y_A); $E = e_{ij}, \forall i, j \in V, i \neq j$ is the set of edges; and w_{ij} denotes the weight of edge e_{ij} between vertices v_i and v_j as measured by the Euclidean distance.

In the USL phase of the FL algorithm, the drone receives FL updates by following a trajectory $\mathbf{P_d}$. During the DSL phase, beginning from the last node visited in the USL, the drone retraces its USL path while downloading the aggregated soft labels SL to $AP_k, \forall k \in K$ until it returns to the starting point.

We recognize that this entire trip, which takes time T, spans distance D and includes both phases, resembles a modified Traveling Salesman Problem (TSP) and can be approached in two stages. The first stage consists of formalizing the drone path optimization as a TSP problem and finding a Hamiltonian cycle H. In the second step, we implement post-processing steps, detailed later, to derive G_d from H. Here, G_d represents the path (order of vertices) the drone should follow to execute the two FL phases USL and DSL with minimum D.

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

$$x_{ij} = \begin{cases} 1, & \text{if } e_{ij} \text{ is included in } H \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

Now, the TSP can be formally defined as:

$$\min_{H} \sum_{e(ij) \in H} w_{ij} = \min_{x_{ij}} \sum_{i=1}^{N_v} \sum_{j=1, j\neq i}^{N_v} w_{ij} x_{ij}, \tag{5}$$

under the following constraints:

1) Each node is visited exactly once:

$$\sum_{j=1, j\neq i}^{N_v} x_{ij} = 1, \quad \forall i \in V. \tag{6}$$

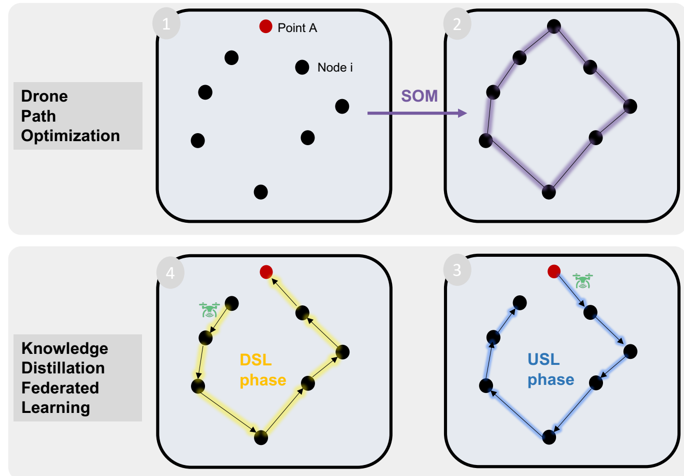

Fig. 1. The Drone optimization problem is first formulated as a variant of the Traveling Salesman Problem to obtain a near-optimal Hamiltonian cycle H. The drone path G^d is then derived from H. Finally, the drone acts as the server while performing Federated Learning which is divided into two communication phases: USL and DSL, following the G^d to minimize distance.

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

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1, \quad \forall S \subset V, 2 \leq |S| \leq N_v - 1. \tag{7}$$

An overview of our envisioned drone-assisted LoRa (DORA) network system components, including the path optimization problem and the FL phases, is depicted in Fig. 1.

IV. Drone Path Optimization

Self-Organizing Map (SOM) is a clustering algorithm [29] that belongs to a class of competitive learning algorithms known as Winner-Takes-Most (WTM) [30]. The objective of SOM entails learning a network W (list of feature vectors) that exemplifies the patterns within a graph of nodes as delineated by a distance function.

WTM involves constructing a network W, which is a list of vectors, and training it as follows: provide an input node x, calculate the distance to each row W_i that represents a neuron, and select the winning neuron with index i_{win}, whose weights exhibit the highest correlation with the current node. Neurons in proximity to the winner node are called the winner neighborhood. In each iteration, the weights of the winner neighborhood are modified as:

$$W_i = W_i + \eta KN(i, i_{win})(x - W_i), \tag{8}$$

where η denotes the learning rate while x symbolizes the present input node. i refers to all neurons belonging to the winner's neighborhood as defined by $KN(i, i_{win})$:

$$KN(i, x) = \begin{cases} 1 & \text{for } d(i, i_{win}) \leq \lambda \\ 0 & \text{for others,} \end{cases} \tag{9}$$

where $d(i, i_{win})$ signifies the distance between the winning neuron and the i-th neuron, while λ represents the neighborhood radius.

An enhanced neighborhood function was proposed in [30] to select neighborhood neurons by utilizing a Gaussian kernel as the neighborhood function:

$$g(i, i_{win}, \sigma) = e^{-\frac{d(i,i_{win})^2}{2\sigma^2}}, \tag{10}$$

where σ defines the distribution spread, controlling the modification strength. By employing $g(i, i_{win})$, eq. 8 can be rewritten as follows that we utilize in our SOM implementation:

$$W_i = W_i + \eta g(i, i_{win}), \sigma)(x - W_i). \tag{11}$$

During inference, after training the SOM, the learned network of neurons is used to construct an approximate TSP tour. This process involves iterating through each node, calculating the index i_{win} of the closest neuron (the winner neuron W_{win}) for each node. The size of W, denoted as N_w, is chosen to be eight times the size of the graph N_v. As the number of neurons typically exceeds the number of nodes and not every neuron W_i maps to a node, the indices of winner neurons cannot be used directly to construct the TSP tour. Instead, the obtained list of neuron indices is sorted, and the sorting index is employed to reorder the graph nodes into a near-optimal TSP tour C expressed by:

$$C = \langle v_{c_1}, v_{c_2}, \ldots, v_{c_{N_c}} \rangle, \tag{12}$$

where $N_c = (N_v + 1)$. The Hamiltonian cycle is denoted by:

$$H = \langle e_{h_1}, e_{h_1}, \ldots, e_{h_{N_h}} \rangle. \tag{13}$$

In this context, $N_h = (N_c - 1)$. Lastly, to calculate G_d, the drone's tour, from H, we remove the final 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 $v_{c_{N_c}}$, and we also remove the last vertex in C, identical to the starting vertex, to obtain an open tour of vertices $C' = \langle v_{c_1}, v_{c_2}, \ldots, v_{c_{N_c-1}} \rangle$. C' represents the path of the USL phase. To account for the drone's DSL path, we reverse the order of vertices in the open tour C', yielding $\widetilde{C} = \langle v_{c_{N_c-1}}, v_{c_{N_c-2}}, \ldots, v_{c_1} \rangle$.

The path of the drone that minimizes the distance D is ultimately defined as the concatenation of C' and $\widetilde{C}$, expressed as:

$$G^d = \langle v_{c_1}, v_{c_2}, \ldots, v_{c_{N_c-1}}, v_{c_{N_c-2}}, \ldots, v_{c_1} \rangle. \tag{14}$$

The total number of nodes in the drone's optimal path G_d is given by $N_G = 2 \times (N_c - 1)$. The distance D traversed by the drone while following G_d is minimized, as it is derived from the near-optimal Hamiltonian cycle H computed by the SOM.

V. Envisioned Communication-Efficient Federated Learning with Knowledge Distillation

In this section, we first present a traditional model-based FL technique, identify its limitation, and then envision a communication-efficient, KD-based FL framework. Based on this, we propose a compressed KD-based FL algorithm, referred to as CFedAKD. In both the traditional and envisioned FL frameworks, the drone access point, denoted as AP_d, serves as the server, while the access points deployed at user-homes, represented as $AP_k \forall k \in K$, act as clients. Each client possesses its own private local dataset, D_k, where $D = D_k{}_{k=1}^{K}$.

A. Model-based Federated Learning traffic model

Each global round in model-based FL [31] unfolds in three sequential phases, namely Download (DL), Learning (L), and Upload (UL).

During the DL phase, nodes retrieve the weights, θ^r, of the server-controlled global model f, with r signifying the current global round. In the L phase, clients update their local copies of weights using:

$$\theta_k^r \leftarrow \theta^r - \eta \frac{1}{|D_i|} \sum_{(x,y)\in D_i} \nabla \mathcal{L}(\theta_{t-1}, f(\theta^r, x), y), \tag{15}$$

where η represents the learning rate and $\mathcal{L}$ denotes the loss function. For classification problems, a Categorical Cross Entropy function, $\mathcal{L} = \mathcal{L}_{\mathcal{CCE}}$, is typically employed.

Lastly, in the UL phase, clients transmit the locally trained weights to the server for aggregation:

$$\theta^{r+1} \leftarrow \frac{1}{N_k} \sum_{i=1}^{N_k} \theta_k^r. \tag{16}$$

This process continues until convergence is reached or a predetermined number of rounds have elapsed.

The uploading and downloading of model weights necessitate substantial bandwidth for communication. However, in our drone-based system, drone flight time is constrained. Thus, we employ an alternative FL approach rooted in the concept of KD.

B. Knowledge Distillation-based Federated Learning Traffic Model

In order to adapt KD for the FL setting, a proxy dataset D_p is employed, which is accessible to all clients for soft label computation. Subsequently, the local soft labels are transmitted to a server for aggregation into global soft labels, which are then returned to the clients for training on the labeled dataset (D_p, SL^r).

Within FL incorporating KD, each node possesses a private dataset, D_k, in addition to a shared public dataset, D_p. The latter facilitates knowledge transfer among nodes. One notable advantage of KD-based FL in comparison to the model-based FL, is the autonomy it provides with to the nodes in terms of crafting their model architecture f_k. Conversely, model-based FL [5], [31] necessitates a server-controlled model architecture.

The global cycle for KD-based FL encompasses four sequential stages: Local Learning phase (LL), Upload Soft Labels phase (USL), Download Soft Labels (DSL), and Knowledge Distillation Learning phase (KDL). Both USL and DSL stages have been previously discussed, involving soft label uploads from nodes/clients to the server (the drone) and subsequent server aggregation and global soft label downloads to nodes, respectively.

During the LL phase, Cross Entropy Loss (CCE) is utilized to train the locally designed model f_k on the locally labeled dataset:

$$\theta^{r,'} k \leftarrow \theta_k^r - \eta \cdot \frac{1}{|D_k|} \sum_{(x,y)\in D_k} \nabla \mathcal{L}_{CCE}(\theta f_k, f_k(x), y). \quad (17)$$

Following the local training of models on the local data, each client forwards its local soft labels SL_k^r to the drone in the USL phase. In the DSL phase, the drone retraces its USL trajectory and disseminates the aggregated soft labels SL^r for KDL phase training purposes as follows,

$$SL^r \leftarrow \sum_{i=1}^{N_k} \frac{P_i^r \cdot SL_k^r}{\sum_{k=0}^{N_k} P_k^r}. \quad (18)$$

During the KDL phase, distance functions such as Mean Squared Error (MSE) or Kullback-Leibler (KL) divergence loss are employed to train f_i on the public unlabeled dataset, utilizing the downloaded soft labels. This is expressed as:

$$\theta^{r+1} k \leftarrow \theta^{r,'} k - \eta \cdot \frac{1}{|D_p|} \sum_{(x,y)\in (D_p, SL^r)} \nabla \mathcal{L}_{KD}(\theta_k^{r,'}, f_k^*(x), y), \quad (19)$$

where $\theta_k^{r,'}$, $\theta_k^{r,''}$, and $f_k^*(x)$ denote the model weights following local dataset training, model weights post knowledge distillation training, and an analogous architecture to f_k obtained via removal of the final layer [14] or increase the SoftMax temperature [12] to smooth the distribution of the output vector, which is the soft labels $SL_k = f_k^*(D_p)$.

C. Compressed KD-based FL with Compression

To enhance KD-based FL, coauthors of this work in their earlier research [15] employed mixup augmentation [17] to generate a synthetic dataset D_{Aug}^r for each global round r, by mixing D_p and a permuted version of it D_d^r as:

$$D_{Aug}^r = \lambda^r \cdot D_d^r + (1 - \lambda^r) \cdot D_p. \quad (20)$$

Mixup augmentation is an example of a Vicincal Distribution which aims to prevent overfitting and improve generalization.

We also use compression to reduce soft label sizes by applying quantization to SL_k^r, resulting in compressed local soft labels, CS_k^r, which can be sent to the server to be aggregated:

$$CS^r \leftarrow \sum_{i=1}^{N_k} \frac{P_i^r \cdot CS_k^r}{\sum_{k=0}^{N_k} P_k^r}. \quad (21)$$

Finally, the Server broadcasts CS^r to clients to use them for KD training as shown in the steps of Algorithm 1, referred to as our proposed CFedAKD approach. CFedAKD, by design, requires less waiting time compared to other standard FL methods (e.g. FedAvg), which we evaluate empirically in the following section.

VI. Performance evaluation

In this section, we evaluate the performance of our proposed DORA system. We evaluate two aspects of the system. First, the quality of the generated path using SOM is evaluated by comparing the solution generated on graphs that have pre-calculated optimal routes. Second, we evaluate the proposed Compressed

Algorithm 1 CFedAKD Algorithm for Drone-based FL Network.

Input: Public dataset: D_p, Test dataset: D_t, Local dataset of client i: D_i, Independently designed local model of client i: f_i, Number of communication rounds: R, Number of epochs for local training: E_l, Number of epochs for KD training: E_{KD}, Loss function for local training: L_l, Loss function for KD training: L_{KD}, Total number of participating clients: N_c, Fraction of clients participating at any given round: K
Output: Collaboratively trained local model f_i
Client i designs f_i and initializes θ_i
for round $r = 1$ to R **do**
  $N_k = K \cdot N_c$
  Select N_k clients randomly
  $\rho^r, \alpha^r \leftarrow$ server randomly generated
  $\lambda^r \leftarrow Beta(\alpha^r, \alpha^r)$
  Broadcast ρ^r, λ^r
  Server:
  $CS^r \leftarrow \sum_{i=1}^{N_k} \frac{P_i^r CS_i^r}{\sum_{k=0}^{N_k} P_k^r}$
  Broadcast CS^r
  Client:
  for client $i = 1$ to N_k **do**
    $D_d^r \leftarrow$ permute(D_p, ρ^r)
    $D_{Aug}^r \leftarrow$ mixup(D_p, D_d^r, λ^r)
    $S_i^r \leftarrow$ calculate soft labels on D_{Aug}^r
    $CS_i^r \leftarrow$ Quantize(S_i^r)
    $P_i^r \leftarrow$ calculate accuracy on D_t
    Client i sends CS_i^r and P_i^r to the Server
    for epoch $e = 1$ to E_{KD} **do**
      $\theta'_{f_i} \leftarrow \theta_{f_i} - \eta \frac{1}{|D_{Aug}^r|} \sum_{(x,y)\in(D_{Aug}^r, CS^r)} \nabla \mathcal{L}_{KD}(\theta_{f_i}, f_i(x), y)$
    end for
    for epoch $e = 1$ to E_L **do**
      $\theta'_{f_i} \leftarrow \theta_{f_i} - \eta \frac{1}{|D_i|} \sum_{(x,y)\in D_i} \nabla \mathcal{L}_{CCE}(\theta_{f_i}, f_i(x), y)$
    end for
  end for
end for

Federated Learning via Augmented Knowledge Distillation (CFedAKD) algorithm against conventional FedMD [14] and FedAvg [5] algorithms in terms of communication cost (size of update files).

For evaluating the quality of the SOM-generated route, since finding the optimum drone path depends on finding a solution for TSP, we evaluated the performance of our algorithm based on the quality of the obtained Hamiltonian cycle H. For that, we employed map instances provided by the National Traveling Salesman Problem website maintained by the University of Waterloo [32], Canada. For each map, the distances of both the optimal route and the SOM-generated route were provided. We reported the execution time and the quality of the obtained H. The quality of the route was calculated based on how far the obtained route distance is from the optimal value using $pathquality = \frac{D_{SOM}}{D_{optimal}}$ where D_{SOM} and $D_{optimal}$ represent the distance SOM-generated route and the distance of the optimal route, respectively. We used four city maps

TABLE II
AVERAGE TEST ACCURACY AND COMMUNICATION OVERHEAD FOR DIFFERENT CENTRAL AND KNOWLEDGE DISTILLATION-BASED FL ALGORITHMS.

Algorithm	# Soft labels	Test accuracy	Communication overhead (KB)
Local	-	30	-
Central	-	78	-
FedMD	5000	70	320
	10000	72	640
FedAKD	5000	68	320
	10000	70	640
CFedAKD	1000	65	16
	5000	65	80
	10000	64	160

of Qatar, Uruguay, Rwanda, and Zimbabwe with the number of cities (nodes on the map) equal 194, 734, 1621, and 929, respectively. We compared the following Federated Learning (FL) algorithms in terms of performance and communication overhead: FedAvg [5], FedMD [14], and FedAKD [15]. The considered FL simulations were constructed using CIFAR100 as the local dataset, split across 20 clients with each client possessing a local dataset comprising 10 samples per class. KD-based FL (e.g., FedMD/FedAKD/CFedAKD) was considered to employ a public dataset as a proxy dataset for knowledge transfer. We used CIFAR10 as a public dataset and run simulations with the following public dataset sizes: 1000, 5000, and 10,000 randomly drawn samples of the public dataset CIFAR10. Unlike FedAvg (which imposes a server-controlled local model design), one of the major advantages of KD-based FL algorithms is that they give clients control over the design of local models. Therefore, we considered a custom local model for each client with model sizes ranging from 400 KB to 5000 KB.

Since we derived the drone path from the Hamiltonian cycle H obtained by SOM, we evaluated the drone path by measuring the quality of H which was calculated as $\frac{D_{SOM}}{D_{optimal}}$ where D_{SOM} denotes the distance covered by H and $D_{optimal}$ refers to the distance covered by the optimal path. For that, we employed maps of four different cities in Qatar, Zimbabwe, Rwanda, and Uruguay, sourced from the National Traveling Salesman Problem website, maintained by the University of Waterloo. The Hamiltonian cycle H learned by SOM was assessed using these maps, which represent city coordinates. For each map, the experiment was executed with 100, 1000, and 10000 iterations. Table III shows the time taken for each run (i.e. time of all iterations), the resulting cycle length, and the quality of the solution. The quality column in Table III reveals the proximity of the identified solution to the optimal solution, with a lower value indicating a better route. It is evident that for maps with fewer nodes, SOM is able to determine a route with a length close to the optimal path within a reasonable time frame. However, as the number of nodes increases, the quality of the discovered route deviates further from the optimal solution, even when using a large number of iterations.

Next, the result in Fig. 2 compares the communication overhead of the considered FL algorithms. The bar plot demonstrates the sizes of Compressed Soft Labels (CSL), Soft Labels (SL), and Model (M) referring to model weights. M is the average of the used model sizes in FL simulations since each client in KD-based FL algorithms like FedMD [14] and FedAKD [15] independently designs its model. The size of CSL is 160 Kilo Bytes (KB) representing the soft labels calculated using 10,000 shared samples resembling the public dataset, which is significantly less than the average size of local model weights, 2500 KB, that is shared if a model-based FL algorithm (e.g., FedAvg) is used. Table II shows the performance (test accuracy) of CFedAKD and other KD-based FL algorithms. Local and central performances shown in the table refer to the average accuracy achieved by each client trained only on his local dataset, and the average accuracy achieved by training each client on the dataset gathering all local datasets, respectively.

TABLE III
PERFORMANCE OF SELF-ORGANIZING MAPS (SOM) FOR SOLVING THE TSP ON FOUR CITY MAP INSTANCES WITH VARYING ITERATIONS. THE REPORTED DURATION IS FOR ALL ITERATIONS.

City map	Optimal tour length	Iterations	# Nodes	Time (s)	Distance (m)	Quality (lower is better)
Qatar	9352	100	194	0.210	20266.01	2.17
		1000		0.490	15435.31	1.65
		10000		3.695	9967.67	1.07
Rwanda	26051	100	1621	0.648	131294.46	5.04
		1000		1.431	123616.82	4.75
		10000		8.885	38039.48	1.46
Uruguay	79114	100	734	0.282	364385.30	4.61
		1000		0.743	348757.37	4.41
		10000		5.544	104557.78	1.32
Zimbabwe	95345	100	929	0.338	441386.25	4.63
		1000		0.878	464806.02	4.87
		10000		6.448	127421.36	1.34

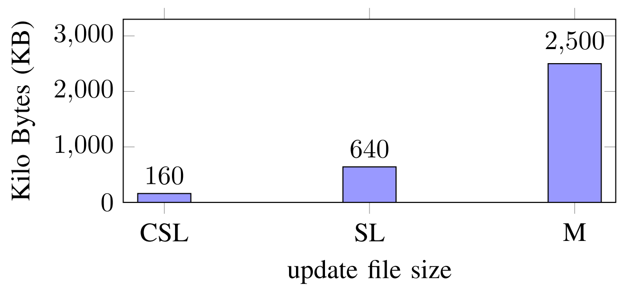

Fig. 2. Comparing communication overhead under the considered FL algorithms. Compressed Soft Labels (CSL), Soft Labels (SL), and Model weights (M) refer to the exchanged update messages in the FL algorithms: CFedAKD, FedMD, and FedAvg, respectively.

VII. CONCLUSION

In this paper, we presented a drone-assisted agile network for rural areas with LoRa payload that performs communication-efficient Federated Learning (FL) for the purpose of data offloading (e.g., health monitoring by collecting wearable data from user homes) in rural areas lacking Internet connectivity. To save energy on the drone, Self-Organizing Maps were used to optimize the drone path by formulating the path planning problem as a variant of the Traveling Salesman Problem

(TSP) and devising a drone path based on the near-optimal Hamiltonian cycle produced by the SOM. We compared two FL communication models: Knowledge Distillation-based FL and model-based FL algorithms, and introduced a communication-efficient Knowledge Distillation-based FL algorithm named Compressed Federated Learning via Augmented Knowledge Distillation (CFedAKD) algorithm. We tested the performance and communication overhead of our proposed CFedAKD compared to other KD-based FL, model-based FL, local, and central training. Experimental results demonstrated that SOM is able to find a high-quality TSP route, on multiple multi-scale city maps. Furthermore, the communication cost of KD-based FL makes it more adequate to run on low-bandwidth networks in contrast with model-based FL. KD-based FL can boost performance beyond local performance with significantly less communication cost than model-based FL methods leveraging a shared dataset to transfer knowledge across the network.

Code Availability

The implementation of SOM drone path optimization and Federated Learning simulations can be accessed at https://github.com/gadm21/Drone-LoRa-SOM-CFedAKD (last accessed on 30 June 2023).

Acknowledgement

This research was funded by the New Frontiers in Research Fund (NFRF) grant number: NFRFE-2019-00313.

References

[1] G. Gad, E. Gad, K. Cengiz, Z. Fadlullah, and B. Mokhtar, "Deep learning-based context-aware video content analysis on IoT devices," *Electronics*, vol. 11, no. 11, article no. 1785, 2022.

[2] N. Nasser, Z. M. Fadlullah, M. M. Fouda, A. Ali, and M. Imran, "A lightweight federated learning based privacy preserving B5G pandemic response network using unmanned aerial vehicles: A proof-of-concept," *Computer Networks*, vol. 205, article no. 108672, 2022.

[3] S. Sakib *et al.*, "On COVID-19 prediction using asynchronous federated learning-based agile radiograph screening booths," in *ICC 2021 - IEEE International Conference on Communications*, 2021.

[4] S. Sakib, M. M. Fouda, Z. M. Fadlullah, K. Abualsaud, E. Yaacoub, and M. Guizani, "Asynchronous federated learning-based ECG analysis for arrhythmia detection," in *2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*, 2021.

[5] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," *arXiv preprint arXiv:1602.05629*, 2016.

[6] 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 *ICC 2023 - IEEE International Conference on Communications*, 2023.

[7] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[8] Y. Gupta, Z. M. Fadlullah, and M. M. Fouda, "Toward asynchronously weight updating federated learning for AI-on-edge IoT systems," in *2022 IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS)*, 2022.

[9] K. Bedda, Z. M. Fadlullah, and M. M. Fouda, "Efficient wireless network slicing in 5G networks: An asynchronous federated learning approach," in *2022 IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS)*, 2022.

[10] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," *arXiv preprint arXiv:1712.01887*, 2017.

[11] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, 2021.

[12] T. Kim, J. Oh, N. Y. Kim, S. Cho, and S.-Y. Yun, "Comparing Kullback-Leibler divergence and mean squared error loss in knowledge distillation," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 2021, pp. 2628–2635.

[13] C. Wu, F. Wu, L. Lyu, Y. Huang, and X. Xie, "Communication-efficient federated learning via knowledge distillation," *Nature communications*, vol. 13, no. 1, article no. 2032, 2022.

[14] D. Li and J. Wang, "Fedmd: Heterogenous federated learning via model distillation," *arXiv preprint arXiv:1910.03581*, 2019.

[15] G. Gad and Z. Fadlullah, "Federated learning via augmented knowledge distillation for heterogenous deep human activity recognition systems," *Sensors*, vol. 23, no. 1, article no. 6, 2022.

[16] G. Gad, "Light-weight federated learning with augmented knowledge distillation for human activity recognition," Master's thesis, 2023.

[17] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *International Conference on Learning Representations*, 2018.

[18] J.-M. Martinez-Caro and M.-D. Cano, "IoT system integrating unmanned aerial vehicles and LoRa technology: A performance evaluation study," *Wireless Communications and Mobile Computing*, vol. 2019, article no. e4307925, Nov. 2019.

[19] L. Angrisani *et al.*, "An innovative air quality monitoring system based on drone and IoT enabling technologies," in *IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor)*, 2019.

[20] 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 *International Conference on System Science and Engineering*, 2018.

[21] M. Zhang and X. Li, "Drone-enabled internet-of-things relay for environmental monitoring in remote areas without public networks," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7648–7662, 2020.

[22] M. Behjati *et al.*, "LoRa communications as an enabler for internet of drones towards large-scale livestock monitoring in rural farms," *Sensors*, vol. 21, no. 15, article no. 5044, 2021.

[23] S. Park, S. Yun, H. Kim, R. Kwon, J. Ganser, and S. Anthony, "Forestry monitoring system using LoRa and drone," in *8th International Conference on Web Intelligence, Mining and Semantics (WIMS)*, 2018.

[24] A. Caruso *et al.*, "Collection of data with drones in precision agriculture: Analytical model and LoRa case study," *IEEE Internet of Things Journal*, vol. 8, no. 22, pp. 16 692–16 704, 2021.

[25] M. A. Ahmed *et al.*, "LoRa Based IoT Platform for Remote Monitoring of Large-Scale Agriculture Farms in Chile," *Sensors*, vol. 22, no. 8, article no. 2824, 2022.

[26] R. Xiong, C. Liang, H. Zhang, X. Xu, and J. Luo, "FlyingLoRa: Towards energy efficient data collection in UAV-assisted LoRa networks," *Computer Networks*, vol. 220, article no. 109511, 2023.

[27] 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, article no. 173, 2022.

[28] V. A. Dambal, S. Mohadikar, A. Kumbhar, and I. Guvenc, "Improving LoRa signal coverage in urban and sub-urban environments with UAVs," in *2019 International Workshop on Antenna Technology (iWAT)*, 2019.

[29] Ł. Brocki and D. Koržinek, "Kohonen self-organizing map for the traveling salesperson problem," in *Recent Advances in Mechatronics*, 2007.

[30] J. Zhu, H. Ye, L. Yao, and Y. Cai, "Algorithm for solving traveling salesman problem based on self-organizing mapping network," *Journal of Shanghai Jiaotong University (Science)*, early access, 2022, doi: 10.1007/s12204-022-2517-3.

[31] 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 Journal of the Communications Society*, vol. 2, pp. 1000–1014, 2021.

[32] "World tsp," Feb 2022. [Online]. Available: https://www.math.uwaterloo.ca/tsp/world/