IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 27 May 2023, accepted 2 July 2023, date of publication 10 July 2023, date of current version 14 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3293737

RESEARCH ARTICLE

Applying RIS-Based Communication for Collaborative Computing in a Swarm of Drones

DADMEHR RAHBARI[1], MUHAMMAD MAHTAB ALAM[2], (Senior Member, IEEE), YANNICK LE MOULLEC[2], (Senior Member, IEEE), AND MAKSIM JENIHHIN[1], (Member, IEEE)

[1]Center for Trustworthy and Efficient Computing Hardware (TECH), Department of Computer Systems, School of IT, Tallinn University of Technology (TalTech University), 12618 Tallinn, Estonia

[2]Thomas Johann Seebeck Department of Electronics, School of Information Technologies, Tallinn University of Technology, 19086 Tallinn, Estonia

Corresponding author: Dadmehr Rahbari (dadmehr.rahbari@taltech.ee)

This work was supported in part by the European Union through the European Social Fund through the Project Information and Communication Technologies (ICT) Program, in part by the European Commission (EC) Horizon 2020 European Research Area (ERA)-Chair Grant Cognitive Electronics (COEL) under Grant 668995, in part by the Tallinn University of Technology Development Program 2016–2022 under Grant 2014-2020.4.01.16-0032, in part by the Estonian Centre of Excellence in ICT Research (EXCITE), in part by the Estonian Research Council PRG1467 CRASHLESS Project, and in part by the Parrot Franco-Estonian Hubert Curien Partnership SUITED Project.

ABSTRACT A swarm of autonomous and heterogeneous drones has many benefits in various scenarios e.g. search and rescue, disaster management, agriculture, delivery and logistics, mapping and surveying, environmental monitoring, etc. However, the presence of obstacles in the environment poses challenges to communication between drones, including network coverage, received signal power, latency, and power consumption. To improve the drones' communication in real-time scenarios, reconfigurable intelligent surfaces (RIS) can be used. RIS is a promising technology for empowering millimeter-waves and sub-millimeter waves communication. It also can provide improved communication links with significantly higher received signal strength in non-line-of-sight situations, which should be taken into account by drones to decide when and with which other drone(s) to perform computation offloading. To this end, we provide two federated learning-based computation offloading strategies through direct and indirect communications. These approaches are based on an advanced rating technique including some key computation and communication parameters. The core of the algorithm also involves two separate deep learning models that are helpful to efficiently transfer and update the decreased model weights, drones' properties, angle of arrival, and angle of departure. Simulation results show that the efficiency of the proposed approaches are superior to a reference strategy in terms of energy consumption by −32%, latency −18%, throughput +50%, and cost of communication and computation by −35%.

INDEX TERMS Swarm of drones, reconfigurable intelligent surfaces, computation offloading, federated learning.

I. INTRODUCTION

Recently, swarms of drones enabled with aerial reconfigurable intelligent surfaces (RIS) have emerged as a research topic, as illustrated by surveys such as [1] and [2]. This technology can provide reconfigurable, smart, and sustainable wireless communication in different environments [3], [4].

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Liu.

In such a wireless communication system featuring RIS, we use two central definitions as follows. **DC** refers to a **D**irect communication **C**hannel between two drones. Conversely, **CC** refers to an indirect or **C**ascaded **C**hannel between two drones through RIS [5], [6].

Each RIS panel includes several elements that reflect the radio signal; the angle of these elements can be controlled by the phase-shifting process [7] of the RIS controller [8], [9]. Designing an optimized swarm of drones communication

empowered by RIS requires an optimized configuration of parameters such as the size and number of RIS elements. Studies show that the wavelength of RIS elements affects these parameters' values [8]. However, there exist several open issues such as beamforming design [10], [11], channel estimation [12], as well as deployment and movement [13]. In particular, channel estimation is an important sub-topic in our study and some possible solutions include novel channel estimation protocols, overhead reduction by grouping adjacent reflecting elements forming sub-surfaces, and parametric channel estimation for millimeter-wave as a key pillar of ultra-high-speed communications of wireless systems [14].

Apart from the communication challenges posed by a swarm of drones, there is also a significant computational problem that needs to be addressed. Let's consider the local processing capacity of each drone. If a drone is unable to execute its tasks locally, offloading them is a viable solution [15], [16]. However, given that each drone may have several neighbors surrounding it, finding the optimal destination for offloading the tasks can be challenging.

We are thus facing two problems, i.e. 1) communication channel estimation and 2) computation offloading. The first issue is finding the largest received signal power (RSP) among many channels of DC and CC. The second issue is finding the most efficient drones as a destination for computation offloading. Recent papers analyzed such problems, e.g. [17] points to some solutions in different architectures including ground users [18], central servers, road station units, mobile users [19], access points, etc. However, there is a research gap not yet addressed for swarms of drones; indeed, in some applications and target environments, for reasons such as geography, trust, security, performance, and flexibility, it is not possible to place or use a central server. Among the solutions, a machine learning model is one of the promising strategies, and providing such a solution considering a distributed architecture of drones without any centralized servers leads us to the federated learning (FL) concept [20].

FL was first introduced by Google in 2016 [21] as a collaborative learning approach in decentralized edge devices without exchanging private data. The authors of [22] showed the capability of FL in a swarm of drones with one drone as a leader and aggregator of all learning models. We previously provided an FL-based solution for computation offloading in a swarm of drones considering only DC between drones [23]. Investigating the aforementioned studies shows that exploiting the (expected) increased capacity of using RIS in a swarm of aerial drones enabled by FL is an open research opportunity. Therefore, in this paper, we propose a RIS-based aerial FL to improve communication and computation in a swarm of drones that are not restricted by only one leader drone but present a collaboration of drones organized into several clusters in the environment.

To address the above issues in a swarm of drones, this paper proposes two approaches. Let's clarify that the base method is the existing FLR (a rating-based FL approach) with direct channel (FLR-DC) [23]. In [23], the addressed problem was related to the computation offloading in a swarm of drones without optimized communication considerations. Now, in this paper, the first proposed approach is FLR with RIS-based channel (FLR-CC) where there is no DC channel between drones. The second approach proposed in this paper is FLR-based direct and RIS-based channels (FLR-DC-CC), i.e., depending on the swarm efficiency in terms of computation and communication, the drones can communicate together either directly or through RIS. Along with these approaches, the drones need to know which channel (DC or CC) has the largest RSP. Note that when there is no obstacle between the transmitter and receiver, the DC channel is expected to be a better choice and vice-versa. Given that CC or DC-CC provide possibly a large number of additional channels to choose from, we propose and use two deep learning (DL) algorithms as the core of FLR-CC and FLR-DC-CC. The first algorithm (named DL-1) in each drone is used to find the optimal channel between all possible DC and CC channels; its main output is each drone's RSP.

In order to find the optimal computing offloading destination, we provide a second DL algorithm (named DL-2). One of the input parameters of DL-2 is the drone's RSP calculated by DL-1. This technique helps us to create a lightweight DL-2 instead of designing a large DL-2 with many weights. This is also a solution for one of the FL issues i.e., the size of model weights [24]. Thus, with fewer transmitted weights between drones in DC or CC channels, the system will be more efficient in terms of bandwidth, energy consumption, latency, etc. In addition to DL-1 and DL-2 algorithms as two parts of the FLR, the rating technique considering the computation and communication parameters plays an important role in the aggregation process of the FLR and the moving of drones. Since the drones are distributed in the environment without any centralized server, FL makes them aware of the whole network. This is an important challenge of the other research [25] that does not allow a distributed system without a server. Since the FL algorithm needs a server to aggregate the learning model weights, we create several clusters of drones where each cluster head (CH) is an aggregation server.

The main contributions of this work are as follows.

- We provide two FL approaches for computation offloading in a swarm of drones using a rating-based federated learning approach (FLR). The first one uses only RIS-based communication between the drones (FLR-CC) and the second one uses both direct and RIS-based channels (FLR-DC-CC). To estimate the optimal channels for the drones, a deep learning algorithm (DL-1) has been designed to calculate the received signal power (RSP).
- We introduce a realistic and RIS-enabled communication model by means of channel estimation. Moreover, the data transmission size between drones is optimized for lower energy consumption and lower delay. In addition, to find the optimal destination for computation offloading in each drone, a new deep learning algorithm (DL-2) is provided. We use the output of DL-1 as one

of the input parameters of DL-2. Using these two DL algorithms allows for decreasing the size of the learning model's weights that must be transferred between drones.
- We evaluate the performance of the proposed approaches in terms of data complexity, time complexity, and RSP. Analyzing the system based on different metrics e.g. the energy consumption, latency, and the trade-off between them, and also throughput of the computation offloading, shows the superiority of FLR-CC and FLR-DC-CC over FLR-DC [23], and FLR-DC-CC over FLR-CC as well.

The rest of the paper is organized as follows. First, an overview of RIS-assisted drones applications is presented in Section II. The system model is provided in Section III. Next, in Section IV, we present the proposed approach as RIS-based communication for collaborative computing in a swarm of drones. Next in Section V, a performance evaluation demonstrates the effectiveness of the proposed approaches. Finally, in Section VII, a conclusion and suggestions for future work are provided.

II. RELATED WORK

In this section, we present recent related works with RIS-based drone scenarios. Some of them used RIS on a building with a fixed position [26], [27], [28], [29] while some others placed it on the drones [30], [31], [32], [33]. Those works addressed different problems and provided different solutions, as discussed below.

Some researchers in [26] worked on drone placement using RIS in non-orthogonal multiple access networks (NOMA). Their problem was to analyze the communication rate with a mixed-integer non-convex optimization. The method was blocked coordinate descent (BCD) which means coupled optimization variables are divided into several blocks and the optimization variables in each block are iteratively optimized with variables in the other blocks fixed. Evaluation of a joint maximization of drone placement and transmit power in a system including four users, two drones, and one RIS showed 25% better channel power gain than other state-of-the-art approaches.

Researchers in [27] investigated the RIS placement optimization problem in the cellular network. They proposed a coverage maximization algorithm (CMA) to show that RIS should be placed vertically in the direction from the base station (BS) to the RIS. Signal-to-Noise Ratio (SNR) was also studied in [34] for a RIS-based application. The SNR coverage was studied based on a realistic path loss model of RIS elements' size and angle. In particular, the distribution of resultant channel gain was utilized by Rayleigh fading and Gamma distribution.

In [28], a RIS-based drone was used as a relay between ground user equipment (UE) and a BS. This approach increased the symbol error rate (SER) and ergodic capacity, which stems from the elevation angle-dependent end-to-end path loss model.

In [29], a drone and a RIS located on a building helped IoT devices (IoTDs) to gather time-constrained data with better connectivity. The problem of resource scheduling and RIS element configuration (including the passive beamforming problem and phase-shift matrix) was solved by a deep reinforcement learning algorithm (DRL) algorithm. Analyzing the RIS size, drone's energy consumption and number of IoTDs showed better performance of the proposed approach than random walk and stationary drone methods.

Researchers in [30] worked on a RIS-based drone and a RIS on the ground (GIRS) for vehicular networks including several cars and BS. This work was analyzed by the achievable signal rate, power consumption, and signal-to-interference-plus-noise (SINR). The results showed the performance of GIRS is more than RIS-based drones.

The RIS-based scenarios with drones are various; in [31], a drone equipped with RIS has been used as a relay between IoTDs and BS. The problem studied there is the age-of-information (AoI) that is used for the drone's altitude, the communication schedule, and the phases-shift of RIS elements. The results showed the proposed approach based on a DRL and proximal policy optimization (PPO) algorithm is superior to a random walk and greedy methods. The researchers in [32] used a flying drone equipped with RIS as a relay between a ground BS and some victims on ships to enable robust and reliable drone communication. They investigated SNR, drone perturbation, and RIS size by a method based on RIS and flight effects (RiFe). Their method outperformed state-of-the-art solutions.

In another research [33], a swarm of drones equipped with RIS elements was applied for direction of arrival (DOA) estimation and perturbation estimation. It was based on sending data from some airplanes to a central drone through a swarm of RIS-aided drones. Applied atomic norm-based estimation and semi-definite programming (SDP) methods showed a better root mean square error (RMSE) than other benchmarks.

The authors in [35] provided a channel estimation strategy by a DL algorithm through direct and RIS-based paths in downlink transmission of mm-wave massive multiple-input multiple-output systems. Evaluation of this work using some metrics such as SNR, RIS's AOA, and normalized mean square error (NMSE) showed that the proposed approach was superior to others. Also, they proved a reasonable performance up to 0.5% amplitude error in switching.

Surveying the related works shows that drones communication is more efficient when using RIS. However, there is no research on a swarm of drones using RIS with optimal computation offloading. Using FL in centralized and semi-centralized systems by private updating of all devices' models empowers it to be used in a fully distributed system without any central server. Table 1 presents an overview of RIS-based drones' communication approaches; analyzing the pros and cons of existing research motivates us to provide a solution for the important gaps in the communication and computation of a swarm of drones using RIS.

TABLE 1. Overview of RIS-based drones' communication approaches.

Ref.	Arc.	Objectives	Technique	Ch.	Env.	Pros	Cons
[18]	HAP, RIS, BS	Trajectory planning and phase-shifting	RL	C2	Sim.	Maximize the SNR of the BS	Ignoring the DC in all situations / Restricted matrix-based area
[26]	Drones, RIS, UE	Drone placement and employing NOMA	BCD	C3	Sim.	Optimizing the drone placement using RIS and NOMA	Analyzing with few fixed drones
[27]	BS, RIS, UE	RIS placement	CMA	C3	Sim.	Providing an optimal place and direction of RIS than BS	Analyzing with few UEs / Ignoring power and latency
[28]	RIS-drone, UE, BS	End-to-End path loss	SNR analysis	C3	Sim.	Comprehensive analysis of channel capacity, RIS size, SER, and SNR	Not compared with other works
[29]	Drone, RIS, IoTDs	Number of IoTDs	DRL	C3	TF	Improving drone energy consumption	High traffic of IoTDs' data in drone / Very limited area
[30]	RIS, drone, BS, vehicles	Signal rate and power consumption	Numeral analysis	C3	Sim.	RIS efficiency in vehicular network	Non-optimized method and insufficient presentation
[31]	IoTDs, drone, RIS, BS	Drone's altitude, communication schedule, and RIS phases-shift of	DRL & PPO	C2	PyTorch	Minimizing the AoI	Ignoring RIS size analysis
[32]	Drone, RIS, BS, Ships	Robust and reliable drone communication	RiFe	C3	Matlab	Modeling the drone flight effects and analyzing SNR and RIS size	High time complexity
[33]	Airplanes, RISs, drones	DOA	SDP	C2	Matlab	Decreasing RMSE considering the position perturbation	Late convergence and high time complexity
[23]	Drones	Computation offloading	FLR-DC	C1	iFogsim	Optimizing the latency, energy consumption, and throughput	Not considering RIS-based communication
This work	RIS-Drones	Computation offloading in RIS-based swarm of drones	FLR-CC & FLR-DC-CC	C3	Matlab & iFogsim	Low latency, energy efficient offloading, and higher throughput	Challenging with Multi-RIS communication in swarm of drones

Note: Arc.: Architecture; HAP: High altitude platform; RL: Reinforcement learning; Ch.: Channel; C1: Direct channel (DC); C2: RIS-based channel (CC); C3: DC and CC channel; Sim.: The name of the simulator or programming language is not mentioned in that paper; TF: TensorFlow; Env.: Environment.

III. SYSTEM MODEL

This section provides information about the system's objects, parameters, and formulation used in our work. As shown in Fig. 1, we consider a swarm of several drones, each denoted as D_i where $i = \{1, 2, \ldots, I\}$. The properties of each drone include position $q_i = (X_i, Y_i, Z_i)$, CPU capacity, amount of RAM, communication bandwidth, idle power, and active power. After capturing the videos by drones, there will be several modules (or computational tasks) that need resources to be executed. In fact, a module is a set of instructions and data that can be executed locally or be offloaded to other drones. Each drone flies under a minimum distance as D^{p} to avoid collision with another drone, which indicates that

$$\sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2 + (Z_i - Z_j)^2} \leq D^{\text{p}} \qquad (1)$$

where X_i, Y_i, and Z_i are the position of i^{th} drone and X_j, Y_j, and Z_j are the position of j^{th} drone. There are some constraints as follows.

$$Z_{min} \leq Z_i, Z_j \leq Z_{max}, \quad \forall i, j \in \{1, 2, \ldots, I\} \qquad (2)$$

$$||q_k - q_j|| \geq \Delta_{min}, \quad \forall k \neq j \in \{1, 2, \ldots, I\} \qquad (3)$$

where Z_{min}, Z_{max} denotes the allowed range of drones' flying height, and Δ_{min} denotes the minimum inter-drone distance required for collision avoidance. To mimic the heterogeneity of the drones in the swarm, the values of these parameters are different. In the environment area, there is a drone equipped with RIS that includes multiple elements.

This work considers some assumptions as follows.

- Drones are flying in the air and their mission is object detection by capturing videos and tracking. Thus, our meaning of a computation task is an image processing task; path planning and routing are not included.

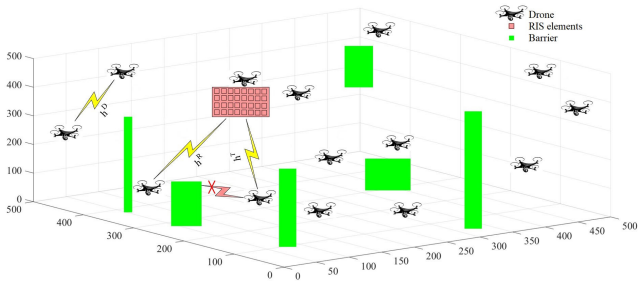

FIGURE 1. System model of a swarm of drones enabled with RIS. For legibility, we show only two types of communication channels: DC (h^D in the figure) when there is no barrier between two drones, and CC (h^T followed by h^R).

- Each drone has enough resources to execute a computation task. The challenge happens when there are many tasks and the drone can not efficiently process them.
- The position of the drone equipped with RIS is considered in the middle of the environment. Thus, it can be in a LOS (Line of sight) or NLOS (Non-line of sight) position. Since the other drones are flying, sometimes a RIS appears in their visible field. If there is no LOS between two drones, the transmitter drone sends the signal to another drone through that RIS.
- Since the system nature is distributed and since there is no server to aggregate the information, a collaborative method is useful. To present an efficient algorithm for computation offloading, FLR with communication modeling is extended in this work.

In the following, we present the equations and descriptions of the RSP, energy consumption, latency, and fairness. All key symbols used in this work are presented in Table 2.

TABLE 2. Symbol definition.

Symbol	Definition
DL	Deep learning
FL	Federated learning
RIS	Reconfigurable intelligent surfaces
AoA	Angle of arrival signal on RIS
AoD	Angle of departure signal on RIS
SNR	Signal-to-Noise Ratio with decibels (dB) unit
LOS	Line of sight
MIPS	Million instructions per second
L	Number of RIS elements
k_1	Number of neighbor devices contributed in the offloading
k_2	Number of clusters
h_k^D	Direct channel from k^{th} drone
h_k^T	Channel between k^{th} drone and RIS
h^R	Channel from RIS
ω_k	Adaptive Gaussian noise
y_k	Received signal power from the k^{th} drone
p_k	Path loss of k^{th} drone
$d_{i,k}$	LOS distance between k^{th} and i^{th} drones
$\overline{h}$	Random scattering component
$d_{i,r}$	Distance between i^{th} drone and RIS
γ_k	Path loss exponent of k^{th} drone
λ	Carrier frequency wavelength
ω_m	Size of the module (or computational task) m
η	Drones' moving rate
$kf_i^{\mathcal{N}}$ and $kf_j^{\mathcal{N}}$	Computation power of i^{th} and j^{th} drones
P_i^{Tx} and P_i^{Rx}	Transmitting and receiving power of i^{th} drone
H and t^h	Altitude and time hovering
D^{f} and t^{f}	Distance and time of the flying
D^{p}	Minimum distance between drones for avoiding collision
F_i	Fairness of offloading from i^{th} drone
D_i	i^{th} drone
$LLC(D_i)$	Local computation capacity (LCC) of i^{th} drone
L_i^{Total}	Total CPU load of i^{th} drone
$L_i^{Current}$	Current CPU load of i^{th} drone
$MIPS_i^{Total}$	Total MIPS of i^{th} drone

A. RECEIVED SIGNAL POWER

The RSP from the k^{th} drone can be calculated [35] by

$$y_k = h_k^D \overline{s} + h_k^T \Theta_k h^R \overline{s} + \omega_k \tag{4}$$

where $h_k^D \overline{s}$ is the direct path and $h_k^T \Theta_k h^R \overline{s}$ is the reflect path. h_k^D is direct from k^{th} drone, h_k^T is the channel between k^{th} drone and RIS, h^R is the channel between RIS and the receiver drone, ω_k is adaptive Gaussian noise (AWGN) at k^{th} drone with zero mean and variance δ^2 [36].

Θ_k is the reflection coefficient defined as follows:

$$\Theta_k = diag\{\beta_1 e^{j\phi_1}, \beta_2 e^{j\phi_2}, \ldots, \beta_n e^{j\phi_n}\} \tag{5}$$

where β_l shows l^{th} element of RIS is turned on or off when it is $(1-\epsilon)$ or $(0+\epsilon)$, respectively. We assume $\epsilon = 0$. Moreover, $\phi_i \in [0, 2\pi)$ is the phase shift of the reflective elements and $j = \sqrt{-1}$. $\overline{s}$ as a data-related parameter from other drones can be expressed by

$$\overline{s} = \sum_{i=1}^{I} \sqrt{\varrho_k} \overline{f}_k s_k \tag{6}$$

where $\overline{f}_k = \frac{f_k}{\|f_k\|_2}$, ϱ_k is the allocated power, and s_k is a data symbol at the k^{th} drone.

Let's present more details about the RSP in what follows.

- **Drone-to-drone communication:**
  h_k^D can be given by

$$h_k^D = \sqrt{p_k d_k^{-\gamma_k}} \overline{h} \tag{7}$$

  where p_k and d_k are the path loss [37] and the direct distance from k^{th} drone, respectively. γ_k is the path loss exponent of k^{th} drone that characterizes the attenuation of the signal power as it propagates through the wireless channel between the transmitting and receiving nodes. This parameter can be as follows.

$$\gamma_k \geq 2, k = \{1, 2, \ldots, I\} \tag{8}$$

  and $\overline{h}$ is the random scattering component with zero mean and variance δ^2 as follows.

$$\overline{h} \sim \mathcal{N}(\mu = 0, \sigma^2 = 1) \tag{9}$$

- **Drone-to-RIS communication:**
  h_k^T can be given by

$$h_k^T = \sqrt{p_{k,r} d_{k,r}^{-\gamma_{k,r}}} \overline{g}_k \tag{10}$$

$$\overline{g}_k = a_k(\varphi_k^{AoA}, V_k^{AOA}) \approx e_k^{\varphi} \tag{11}$$

$$\varphi_k = j\frac{2\pi}{\lambda} u L_x \sin\varphi \sin V_k + L_y \cos V_k \tag{12}$$

  where $d_{k,r}$ is the distance between k^{th} drone and RIS, $\gamma_{k,r}$ is the path loss exponent of k^{th} drone and RIS that can be valued as follows.

$$\gamma_{k,r} \geq 2, k = \{1, 2, \ldots, I\} \tag{13}$$

  u is the antenna separation, AoA is the angle of arrival, ϕ_k^{AoA} and V_k^{AOA} are the corresponding azimuth and elevation AoA of the k^{th} drone-RIS links, λ is the carrier wavelength that equals $\frac{c}{f}$, c is the speed of light that equals $3e^8$, and f is the carrier frequency.
  L_x and L_y are the length and width of a RIS element with the following constraint.

$$0 \leq L_x, L_y < S \tag{14}$$

  where S is the size of a RIS element and also φ_k is restricted by

$$0 \leq \varphi_k < 2\pi \tag{15}$$

- **RIS-to-drone communication:**
  h_k^R can be given by

$$h_k^R = \sqrt{p_{r,k} d_{r,k}^{-\gamma_{r,k}}} \left(\sqrt{\frac{\alpha}{1+\alpha}} \overline{h}_k + \sqrt{\frac{1}{1+\alpha}} h_k'\right) \tag{16}$$

where $\gamma_{r,k}$ is path loss exponent of RIS and k^{th} drone, α is the Rician factor, $\overline{h}_k = a_k^H(\phi_k^{AoD}, V_k^{AoD})$, and h'_k is the indirect component vector with zero mean and variance δ^2.

$$h'_k \sim \mathcal{N}(\mu = 0,\ \sigma^2 = 1) \tag{17}$$

The main problem of the communication part is to estimate the optimized value of h_k^D as the DC and $h_k^T \Theta_k h^R$ as the CC. In order to address this in a swarm of drones, we use FL to propagate the values of *AoA* and *AoD* among drones and calculate the aggregated value of these parameters.

Signal-to-Noise Ratio (SNR): The received SNR at the k^{th} drone [36] can be expressed by

$$SNR_k = \begin{cases} \dfrac{|(h_k^D + h_k^R \Theta_k h^R h_k^T)\rho_k|^2}{\mathcal{N}}, & CC \\ \dfrac{|h_k^D \rho_k|^2}{\mathcal{N}}, & DC \end{cases} \tag{18}$$

where ρ_k^2 is the transmitter power from k^{th} drone, $\mathcal{N}$ is the noise power, CC means k^{th} drone transmits by RIS, and DC means the direct transmission.

Our aim for communication modeling is to estimate the optimal DC and CC channels for each drone. To address this, we apply DL-1 which is fed by a pilot-received signal. This reference signal transmits from the CH to all drones at launching the system, then drones use this for their channel estimation.

B. LATENCY

Depending on their CPU load and total MIPS, the drones can execute their tasks locally or offload them to other neighboring drones; therefore, the latency can be expressed as follows [25].

$$L_i^{\text{Tot}} = Max(\frac{\omega_m}{f_i}, [\frac{\omega_m}{f_j} + \frac{\beta\omega_m}{B_{ij}\log_2(1 + \frac{P_i^{\text{Tx}} d_{ij}^{-\gamma_i} h_i}{G_i})}]) \tag{19}$$

where ω_m is the size of task m, and f_i is the CPU capacity of i^{th} drone, S is the size of data for communication and β is a rate of communication size. Since we transmit a task in its entirety (i.e. without splitting it), $\beta\omega_m$ is the task size, B_{ij} is the bandwidth between D_i and D_j, P_i^{Tx} is transmitting power of i^{th} drone, h_i is the complex Gaussian channel coefficient which follows the complex normal distribution $\mathcal{CN}(0,1)$, and G_i is the additive white Gaussian noise with zero mean and variance δ^2. The G_i unit is dBm (A measure of power spectral density that provides a ratio of the power in one Hertz of bandwidth). Also, d_{ij} is the distance between D_i and D_j. $\gamma_{i,j}$ is the path loss exponent [38] that can be calculated by

$$\gamma_i = \frac{\rho_i}{d_{ij}^2}; \quad i, j = \{1, 2, \ldots, I\} \tag{20}$$

where ρ_i is the channel power of D_i. There are some constraints as follows.

$$h_i \sim \mathcal{CN}(\mu = 0,\ \sigma^2 = 1) \tag{21}$$

Here, the complex Gaussian channel coefficient follows a normal distribution as mentioned.

$$L^{\text{Tot}} \leq L^{Local} \tag{22}$$

This means that the decision to offload computation is taken when the latency of local processing is higher than that of transferring tasks to other drones.

C. ENERGY CONSUMPTION

We consider the energy consumption model for the computation, communication, and moving of drones. The total energy consumption is the sum of these as follows:

$$\begin{aligned} E_i^{\text{Tot}} = & \frac{kf_i^{\mathcal{N}}\omega_m}{f_i} + \sum_{j=1}^{n} kf_j^{\mathcal{N}}\frac{S}{f_j} \\ & + \sum_{j=1}^{n} P_i^{\text{Tx}} \frac{\beta S}{B_{ij}\log_2(1 + \frac{P_i^{\text{Tx}} d_{ij}^{-\gamma_i} h_i}{G_i})} \\ & + \sum_{j=1}^{n} P_j^{\text{Rx}} \frac{\beta\omega_m}{B_{ij}\log_2(1 + \frac{P_i^{\text{Tx}} d_{ij}^{-\gamma_i} h_i}{G_i})} + E_i^{Moving} \end{aligned} \tag{23}$$

where $kf_i^{\mathcal{N}}$ and $kf_j^{\mathcal{N}}$ are the computational power of D_i and D_j, respectively and n is the number of tasks. P_i^{Tx} and P_j^{Rx} are the transmitting and receiving power of D_i and D_j, respectively. The moving energy consumption [39] can be calculated by

$$E_i^{Moving} = (W_1^m H_i + W_2^m)t_i^h + W_3^m t_i^f + W_4^m D_i^f - W_5^m \tag{24}$$

where H and t^h are the altitude and time of hovering, t^f is the flying time, D^f is the flying distance, and $\forall_{j=1}^5 W_j^m$ are coefficients. Some constraints are as follows.

$$k > 0 \tag{25}$$

k is related to the computation power with a positive value.

$$2 \leq \mathcal{N} \in \mathbb{R} \leq 3 \tag{26}$$

As [25], k and $\mathcal{N}$ can be restricted by the mentioned bounds to consider a suitable computation power for i^{th} drone.

$$0 < H_i \leq 50\ m \tag{27}$$

H_i is the altitude of i^{th} drone while hovering.

$$\sum_{i}^{n} t_i^h \leq 1000\ s, n = \text{simulation time} \tag{28}$$

t_i^h is the hovering time of i^{th} drone. It means i^{th} drone can hover throughout the simulation.

$$\sum_{i}^{n} t_i^f \leq 1000\ s, n = \text{simulation time} \tag{29}$$

t_i^f as the flying time can be at last equal to the simulation time.

$$0 \leq D_i^f \leq \sqrt{3 * Area^2} \tag{30}$$

D^f as the flying distance is restricted by the area size in three axes. It means i^{th} drone can stop without any moving during the simulation or fly a maximum $\sqrt{3 * Area^2}$m.

$$E_i^{\text{Tot}} \leq E_i^{Local} \tag{31}$$

A higher local energy consumption causes a drone to offload the computation tasks to other drones. More details about the latency and energy consumption equations and coefficient values can be found in [23].

D. FAIRNESS OF OFFLOADING

We indicate the fairness of the offloading strategy (known as the Jain index in [40]) in a swarm of drones. The fairness of D_i can be calculated as

$$F_i = \frac{(\sum_{j=1}^{k_1} W_e * E_i^{\text{Tot}} + W_l * L_i^{\text{Tot}})^2}{k_1 * \sum_{j=1}^{k_1} (W_e * E_i^{\text{Tot}} + W_l * L_i^{\text{Tot}})^2} \tag{32}$$

$$0 < W_e, W_l < 1 \tag{33}$$

where k_1 is the number of neighbor devices contributed in the offloading, W_e and W_l are weighted coefficients for normalization of the total energy consumption as E_i^{Tot} and the total latency as L_i^{Tot}. The fairness value is in the range $[1/k_1, 1]$. Higher values of fairness correspond to higher performance.

E. PROBLEM DEFINITION

Our work aims at finding the optimal drones for computation offloading. In fact, each drone needs to search for other suitable drones for offloading in a large domain of possible drones through DC and CC channels. Moreover, each drone has some properties such as position, bandwidth B_i, energy consumption E_i^{Tot}, latency L_i^{Tot}, CPU capacity f_i, fairness F_i, and performance $\mathcal{P}_i$. Thus, the offloading decision in j^{th} cluster (O_j) of drones can be expressed by

$$O_j = Max_{i=1}^{n} f(B_i, y_i, f_i, L_i^{\text{Tot}}, E_i^{\text{Tot}}, F_i, \mathcal{P}_i) \tag{34}$$

where n is the number of drones in j^{th} cluster. This function also is useful to understand how suitable is each drone for offloading. The other challenge is the distributed architecture without any central server, which leads us to present a collaborative approach. The problem is constrained as follows.

$$L_i^{\text{Tot}} \leq L^{Local} \tag{35}$$

$$E_i^{\text{Tot}} \leq E^{Local}, \quad i = \{1, 2, \ldots, I\} \tag{36}$$

IV. RIS-BASED COMMUNICATION FOR COLLABORATIVE COMPUTING IN A SWARM OF DRONES

This section presents an FL-based approach in a swarm of drones. Let's explain the **general scheme of FL** for an architecture including some drones denoted $\{D_1, D_2, \ldots, D_N\}$ and an FL server. There are three main phases in FL:

1) The first phase is executed in the FL server, including the initialization of the global model and the learning rate, etc.
2) In the second phase, each D_i trains a model's w_i locally and sends them to the FL server.
3) The third phase includes the aggregation of all collected local models by the FL server which can be given as:

$$W = \frac{1}{N} \sum_{i=1}^{N} w_i \tag{37}$$

where W is the global learning model weight and w_i is the weight of the i^{th} learning model. The second and third phases are repeated until completed or until a specific training accuracy is reached [41].

What follows presents **the proposed approach** to solve the computation offloading in a RIS-based swarm of drones. Figure 2 shows the communication and computation between drones in each cluster. The drones are clustered based on their distance from each other. Each CH is responsible for the aggregation and transmission of the learning model to the neighbors inside the cluster. This solves the need for a server in [25] and the former FLR in [23]. The clusters will be changed after rating and moving the drones. This strategy also helps the real-time applications to run tasks locally or on the edge by other drones.

There are three kinds of signals h_{DC}, h_T, and h_R where h_{DC} is the direct signal (DC) between two drones. Also, the RIS-based channel (CC) includes h_T which is the sending signal from a drone to RIS and the h_R which is the receiving signal from RIS to another drone.

The clusters are numbered 1 to k. As an example, the workflow of Cluster 2 is presented in detail in Fig. 2 and explained in what follows. In this workflow example, the green drone as the CH is placed on the left. There are three other drones in this cluster that are colored yellow, orange, and purple. All steps with the same pre-number (Steps 3.1 and 3.2; 5.1, 5.2, and 5.3; 6.1 and 6.2; 10.1 and 10.2) can be executed in parallel.

Step 1 indicates the video capturing of the environment that is performed by all drones. Then, as per **Step 2**, in order to detect or track the objects, each drone needs to know if it can execute this process locally or offload that to the other drones. Here, we provide Eq. (38) to handle the local computation capacity (LCC) of i^{th} drone as follows.

$$LCC(D_i) = \begin{cases} 1, & L_i^{Total} + L_i^{Current} \leq MIPS_i^{Total} \\ 0, & Otherwise \end{cases} \tag{38}$$

where L_i^{Total} is the total CPU load, $L_i^{Current}$ is the current CPU load, and $MIPS_i^{Total}$ is the total MIPS of i^{th} drone. The value 1 and 0 indicate that the task should be executed locally or offloaded to other drones, respectively.

Two issues that must be dealt with are the channel estimation and the computation offloading. In order to deal with this, two DL algorithms are presented.

- In **Step 3.1**, DL-1 [35] is used for channel estimation. This part helps us to assess how much the RSP of a drone is valuable for communication. This convolutional neural network (CNNs) includes the input layer with a pilot received signal (it is initialized with an identity matrix [35]), three convolutional hidden layers for real,

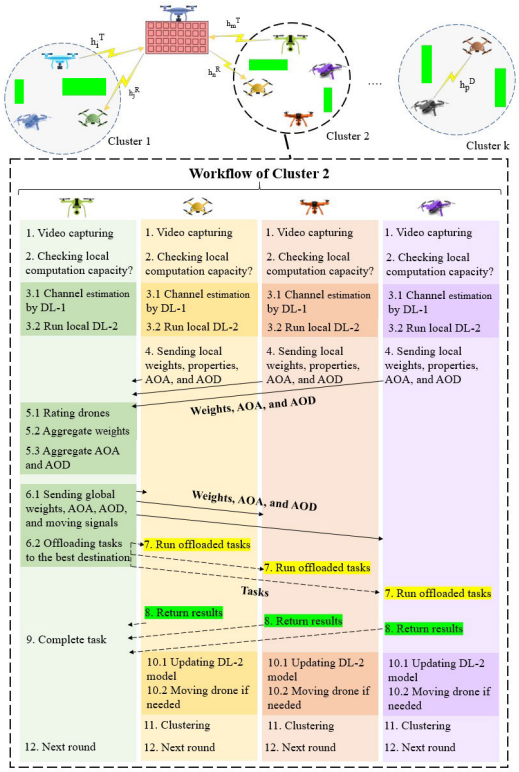

FIGURE 2. Workflow of the proposed approach.

imaginary, and absolute input values, and a set for output layer including two fully connected layers and one regression layer. Each drone executes a DL-1 algorithm locally. The output of this step is the RSP of the current drone. We use the average of this parameter as an input of DL-2.

- In **Step 3.2**, DL-2 is used for the decision making of computation offloading. This multilayer perceptron includes an input layer with $8 * N$ neurons (8 properties are uplink bandwidth, transferred signal power, available MIPS, Jain index, performance, position availability, RSP, and uplink latency for each of N drones), a first hidden layer with 8 neurons, a second hidden layer with N neurons, and finally the output layer with N neurons.

For both DL strategies, 70% of the generated data is used for training, and 30% is used for validation. Using the output of DL-1 as an input of DL-2 helps us in two ways. The first one is to increase bandwidth usage by decreasing the transmitted data between drones in the FL. The second one is to reduce the computation time by decreasing the volume of data for the aggregation process in the FL in the CH.

Indeed, since we want to transmit the DL model's weights between drones, it must not have a large volume for latency and energy efficiency purposes; that is why we propose DL-1 and DL-2 to manage the workflow. Obviously, the volume of DL-2 weights is much less than a large DL model.

Step 4 is related to sending local weights (The weights or gradients between layers in DL-2), properties, AoA, and AoD (WAD) from neighboring drones of the cluster to the CH. Note that using other drones' positions, a RIS can update its reflection matrix and phase shifts. Each drone can send its WAD after completing the DL-1 and DL-2 processes. As shown in Fig. 3, this step is asynchronous, meaning it will take place in different time slots due to the drone's properties.

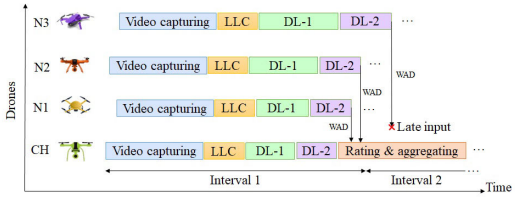

FIGURE 3. Asynchronous aggregation in collaboration between drones in a cluster.

Step 5.1 performs the rating of drones according to communication and computation parameters. We propose an extended rating strategy to evaluate each drone in a swarm. This helps the FL to find the best drone as a destination for offloading. Each drone can rate all its neighboring drones.

The **rating value** is given by

$$Rating_j = \frac{\sum_{i=1}^{(n-1)} w_i.f_j^i}{w_n.L_j} \tag{39}$$

$$0 < \forall_i^n W_i < 1 \tag{40}$$

where f_j^i can be expressed by the bandwidth, total energy consumption (E_i^{Tot}), CPU capacity, fairness, performance, drone's position availability ($\mathcal{A}_i^a$), and total latency (L_i^{Tot}). All w_i are weighted coefficients for the normalization of parameters with real values between 0 and 1.

Drone's position availability (PA) is a parameter that we have created for evaluating the drone's position relative to others. This parameter is calculated according to an average position availability of a drone than the other drones in the environment. In fact, regarding all obstacles, the distance between drones, etc., PA shows the overall position availability. We assign a random real value between 0 and 1 for each drone. The average value of this parameter also shows the dynamic rate of the system. This parameter can be expressed as follows:

$$\mathcal{A}_i^a = \frac{\mathcal{A}_i}{\sum_{j=1}^{N} \mathcal{A}_j} \tag{41}$$

$$0 < \forall_{i,j}, i, j \in [1, n] < 1 \tag{42}$$

where $\mathcal{A}_i$ and $\mathcal{A}_j$ are the position availability of i^{th} and j^{th} drones, respectively. Higher values of $\mathcal{A}_i^a$ mean a better drone's position than barriers and other drones.

The **performance** parameter can be calculated based on the distance between drones and target objects as follows.

$$\mathcal{P}_j = q(1 - \frac{1}{C}\sum_{k=1}^{C}\frac{1}{d'_{jk}}) \tag{43}$$

where d'_{jk} is the Euclidean distance between j^{th} drone and k^{th} object, C is the number of tracked objects, and q as a coefficient value equals 0 if there is no object in the area by D_j, and equals 1 otherwise.

Steps 5.2 and **5.3** can be executed at the same time as Step 5.1. The aggregation of weights helps to create a general learning model for DL-2. Moreover, the aggregation of AoA and AoD supports the channel estimation process that is executed by DL-1. In fact, the CH asynchronously updates its learning model based on the aggregation of its neighboring drones with a higher rating as it is shown in Eq. 44:

$$W_{t+1} = \sum_{i=1}^{N_r} \left(\frac{B_t^i}{\sum_{i \in S} B_t^i} * W_{t+1}^i\right) \tag{44}$$

where W_{t+1} is a new weight and N_r is the number of neighboring drones with higher rating values in the current cluster. B_t^i and B_t are the statistical values of D_i and its neighbors. S is a set of drones with a higher rating value.

Next, in **Step 6.1**, the CH sends the aggregated weights to the neighboring drones of its cluster that had sent their WAD before the interval. The interval or simulation time step refers to the time granularity used to simulate the execution of tasks in a device. **Step 6.2** is related to offloading tasks from the CH to one of three other drones (yellow, orange, or purple drone shown in Fig. 2) if needed. As shown in Fig. 2, there are some dotted lines from CH to other drones which means that only one of these lines will take place for task offloading. Also, let's note that we are explaining the workflow of a round of resource management in a cluster of drones. According to the DL-2 algorithm in each drone, it can decide about its task offloading destination if needed.

In **Step 7**, one of the drones (yellow, orange, or purple drone) that receives the offloaded task can execute it and also return it to the task sender in **Step 8**. In **Step 9**, the drone that offloaded its task can complete it.

In **Step 10.1**, the neighboring drones in the current cluster update their DL-2 model, and also at the same time in **step 10.2**, they can move to another place if their rating value is low. As the application scenario, we place a swarm of drones in the environment, then they will move according to the rating value. We consider a certain distance for drones' movement as 10 meters in X, Y, and Z. In other words, we decide about +10 or −10 to add it to each X, Y, and Z. For example, if the drone's current position is (X,Y,Z), we generate (+10, −10, +10) then the new position is (X+10, Y-10, Z+10).

Finally, after each move, in **Step 11**, the drone executes a clustering process to know which it belongs to what cluster. The i^{th} drone will be placed in a cluster with some other drones if their distance from the i^{th} drone is less than the default radius:

$$Neighbor_i = \forall_{j \in N} d_{ij} \leq R \tag{45}$$

where N is the number of drones, d_{ij} is the distance between D_i and D_j, and R is the neighborhood radius. Note that the maximum number of drones (N_2) in a cluster can be expressed by

$$N_2 = 0.2\, N_1 \tag{46}$$

where N_1 is the total number of drones in the swarm. For example, if N_1 is 50, we have a maximum of 10 drones in each cluster. N_2 helps us to avoid having a single cluster of many drones so that the complexity of the CH can be controlled here.

Doppler Shift: This is the change in frequency of a wave in relation to an observer which is moving relative to the wave source. We handle this fact in our approach after each drone's movement. In order to calculate the drone-to-RIS and RIS-to-drone communications RSP components, we use Eqs. (10) and (16) that include AOA and AOD. As we mentioned in the system model, these two equations are used for calculating the RSP (Eq. (4)).

Step 12 shows the new round that can be repeated by the aforementioned steps from Step 1.

Let's summarize all the steps and related formulations of the proposed approach with the pseudocode shown in Algorithm 1 where D_t represents the t^{th} drone.

Algorithm 1 FLR-DC-CC

```
Place all drones in the area with random positions.
Set each drone in a cluster as Eq. (1).
Run video capturing by drones.
for It=1 to MaxIteration do
    for t=1 to NumberofDrones do
        for All tasks in D_t do
            Check LCC as Eq. (38).
            Estimate the optimal channels as Eq. (4) by DL-1.
            Run local DL-2.
            if D_t is a CH then
                Receive local weights, AoA, and AoD to the neighbors in the cluster.
                Rate all neighboring drones as Eq. (39).
                Aggregate weights as Eq. (44), aggregate AOA and AOD.
                Send global weights, AOA, AOD, and moving signals to all neighboring drones.
            end if
            Update DL-2 model.
            Move D_t if it has a lower rating value.
            Offload task to the best destination.
            Complete task according to offloaded task's result.
            Clustering of moved drones as Eq. (45).
        end for
    end for
end for
```

A. DATA SIZE COMPLEXITY ANALYSIS

First of all, sending local weights, AoA, and AoD from neighbors to the CH is given by

$$(N - k_2)(X + AoA + AoD) \tag{47}$$

TABLE 3. Examples of offloading data size.

Offloading percentage	Data size
10%	400591360 B ≈ 400 MB
25%	1000591360 B ≈ 1 GB
50%	2000591360 B ≈ 2 GB
100%	4000591360 B ≈ 4 GB

where N is the number of drones, k_2 is the number of clusters, X is the size of weights. $AoA = AoD = L$, where L is the number of RIS elements. Moreover, the size of the learning model's weights is calculated as:

$$X = CN * C + C * N + N * N \tag{48}$$

where, for the DL-2 structure, there are two hidden layers with C and N neurons between the input (including $C*N$ neurons where C is a constant equal to 8) and output (including N neurons) of the neural network. Secondly, sending global weights, AoA, AoD, and moving signals from the CH to the neighbors are expressed by Eq. (47).

Finally, the offloading tasks' size to the best destination equals $Y * K$, where Y is the size of the computation task for one drone. Let's assume that the number of drones in each cluster is M, then $k_2 = N/M$. This means that k_2 is a function of N of $O(N)$. The total data transmitted size is $2N(X+2L)+YNP$, where P is the number of drones that need to offload tasks with a maximum of N. After summarizing Eqs. (47) and (48), the offloaded data complexity is:

$$DataComplexity = O(4N^2 + 2N^3 + 4NL + YN^2) \tag{49}$$

For example, let's assume a swarm of 20 drones and 1 RIS with 32 elements, and the size of each computation task including the captured image is 10 MB. As an upper-bound example, if all drones offload 100% of their tasks to the others, the total data transmitted size per interval equals approximately 4 GB. Table 3 gives more examples of offloading percentages and corresponding data sizes.

B. TIME COMPLEXITY ANALYSIS

Here we provide the time complexity of the proposed approach, where the number of drones and tasks are D and M, respectively. The time complexity of line 7 in Algorithm 1 is $O(1)$, that of line 8 is $O(NT * (L_1L_2 + L_2L_3 + L_3L_4 + L_4L_5 + L_5L_6))$, where in DL-1, we have 7 layers, T training examples, and N epochs. For the test phase of DL-2, there are D^2 DC and RIS-based channels. In line 9, for DL-2, we have $O(NT * (L_1L_2 + L_2L_3))$. The time complexity of lines 11, 12, 13, and 14 are $O(1)$, $O(D)$, $O(D)$, and $O(1)$, respectively. In line 16, we have $O(NT * (L_1L_2 + L_2L_3))$. In lines 17, 18, and 19, we have $O(1)$, $O(1)$, and $O(1)$. In line 20, we have D^2. Finally, the overall time complexity of the proposed FLR-DC-CC is $O(D^2(L^2 + M))$. Since the number of layers and weighted connections are constant, we have

$$TimeComplexity = O(D^2M) \tag{50}$$

TABLE 4. System parameters.

Parameter	Value	Parameter	Value
RAM	2 to 6 GB	α, α'	0.2
f_i	60 GHz	B_i	4 to 8 Mb/s
P_i^{Tx}	1.258 W	P_i^{Rx}	1.181 W
d_{ij}	10 to 500 m	d_{ie}	200 to 500 m
R^N	50 m	N^D	10 to 100
k	$1.25 * 10^{-26}$	γ	3
G_i	-100 dBm	h_i	0 or 1
ω_m	1000 KB	η	20%
D^f	10 m	D^p	10 m
Area (X,Y,Z)	(500 m, 500 m, 500 m)	μ	5
RIS position	(250 m, 250 m, 50 m)	L	32
SNR	{-10, 0, 10, 20, 30, 40, 50 dB}	N^R	30
Interval	100 ms	ST	1000 s

Thus, the determinant parameters in the time complexity of our approach are the number of drones (D) and the number of tasks (M) and the time complexity increases faster with an increasing number of drones than with an increasing number of tasks.

V. EVALUATION

In this section, we provide our simulation-based experimental results and evaluate the performance of the proposed offloading strategy. Firstly, we introduce the simulation environment in Subsection V-A and the devices' configuration in Subsection V-B. Secondly, in Subsections V-E to V-K the proposed FLR-CC and FLR-DC-CC offloading strategies for drones are analyzed under different metrics and are compared with the reference FLR-DC method.

A. SIMULATION ENVIRONMENT

This work uses two simulators. All processes related to resource management in a swarm of drones have been performed in the iFogsim [42] simulator as a Java-based library. Note that we have extended the channel estimation based on a most relevant Matlab source code [35] to make it compatible and integrated with the resource management part. To accelerate the simulation, we trained the DL-1 in Matlab and saved it. The main simulator (iFogsim) only requires for the value of RSP from the DL-1's learned model.

B. CONFIGURATIONS

The configuration of devices is as shown in Table 4. The RAM size, the CPU capacity, and the bandwidth of drones are based on [43]. The simulated environment dimensions, the minimum distance between drones, the flying distance D^f (this value is based on the Euclidean distance traveled in three directions x, y, and z.), the transmitting and receiving powers of the drones are provided in [25]. Also, the diversity rate of drones is considered. This means that the drones in the system are heterogeneous. ST is the simulation time. We assume that each drone has sufficient battery to be active during this time. N^R is the number of realizations in the simulation.

C. ANALYSIS OF DATA COMPLEXITY

This section presents a comparison between different types of FLR with direct and indirect paths based on the offloading

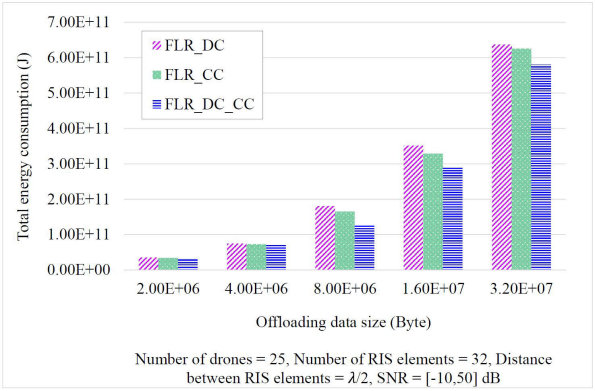

FIGURE 4. Comparison of the total energy consumption vs. data complexity of the FLR-CC and FLR-DC-CC methods against that of the FLR-DC method.

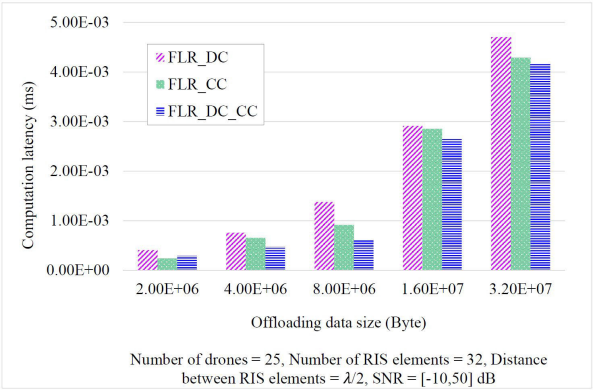

FIGURE 5. Comparison of the computation latency vs. data complexity of the FLR-CC and FLR-DC-CC methods against that of the FLR-DC method.

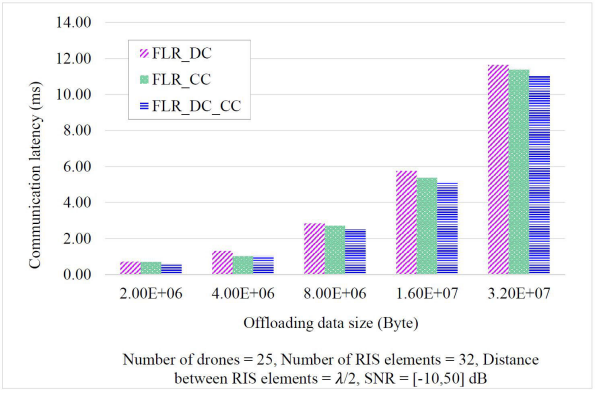

FIGURE 6. Comparison of the communication latency vs. data complexity of the FLR-CC and FLR-DC-CC methods against that of the FLR-DC method.

data size. Fig. 4 indicates that the energy consumption raises by increasing the offloading data size; however, FLR-DC-CC is more efficient than FLR-DC and FLR-CC. This also happens for the computation, communication latencies, and the average total latency as shown in Figs. 5, 6, and 7. Obviously, RIS-based FLR such as FLR-DC-CC helps the system to decrease energy consumption and latency.

According to Fig. 8, FLR-DC-CC causes a more balanced computation offloading in the swarm. There are two important reasons for this result mean rating technique (by considering the computation and communication parameters

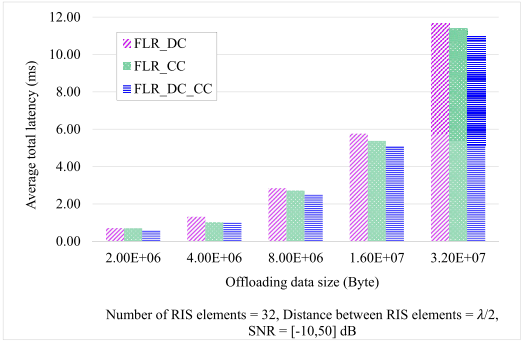

FIGURE 7. Comparison of the total latency vs. data complexity of the FLR-CC and FLR-DC-CC methods against that of the FLR-DC method.

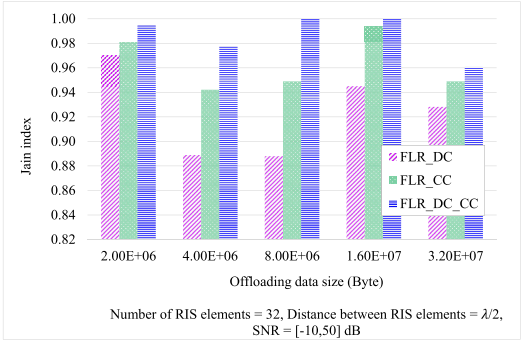

FIGURE 8. Comparison of the fairness vs. data complexity of the FLR-CC and FLR-DC-CC methods against that of the FLR-DC method.

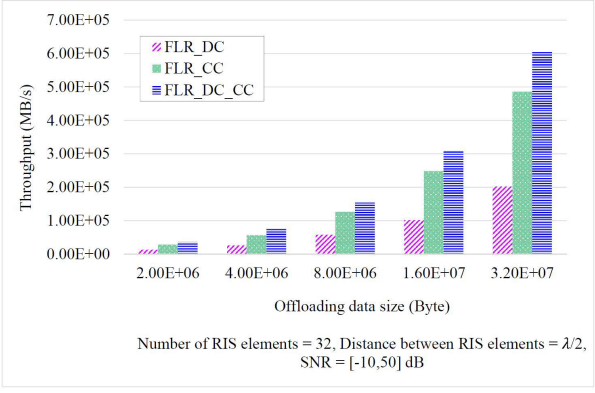

FIGURE 9. Comparison of the throughput vs. data complexity of the FLR-CC and FLR-DC-CC methods against that of the FLR-DC method.

of drones and RIS) and clustering (by an effective distribution of computations regarding energy consumption and latency).

Analyzing the system throughput in Fig. 9 shows that FLR-DC-CC using RIS causes more data can be transmitted in the swarm. In comparison with the inefficient communication on direct paths in FLR-DC, the proposed methods as FLR-DC and FLR-DC-CC present indirect communication with SNR in the range of -10 to 50 dB.

D. ANALYSIS OF TOTAL COMPUTATION AND COMMUNICATION COST PER TASK VS. THE SIZE OF EXECUTED TASKS

We analyze the total computation and communication costs of FLR-CC and FLR-DC-CC methods and compare them with

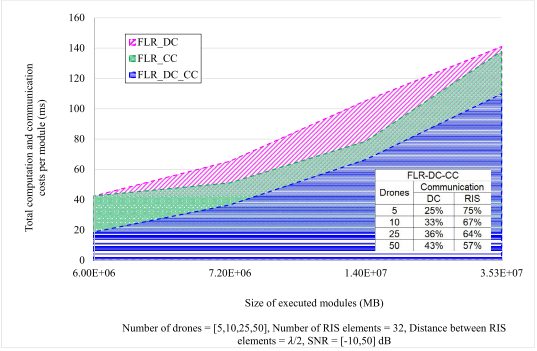

FIGURE 10. Comparison of total computation and communication cost per task vs. the size of executed tasks of the FLR-CC and FLR-DC-CC methods against that of the FLR-DC method.

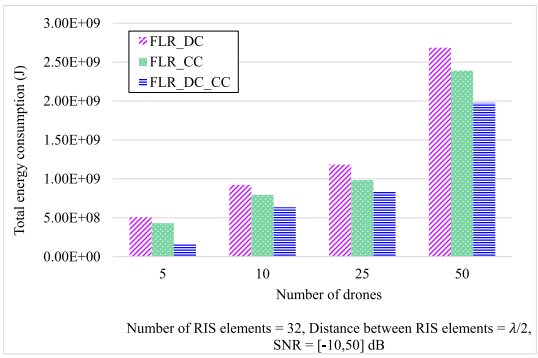

FIGURE 11. Comparison of the total energy consumption of the FLR-CC and FLR-DC-CC methods against that of the FLR-DC method.

FLR-DC. Here, the cost is presented by the task size [44] and execution time. In Fig. 10, the x-axis shows the total size of executed tasks for all drones and the y-axis shows the total computation and communication costs for each task. As shown in the figure, FLR-DC-CC has a lower cost than the other approaches.

There is also a table in this chart related to FLR-DC-CC which shows what percentages of communications are based on DC and RIS channels separately. The table indicates that RIS-based paths have been used more often than direct paths. On one hand, the FLR approach helps drones to make a balanced and efficient computation offloading, and on the other hand, RIS technology supports them for higher received signal strength. These two aspects of the FLR-DC-CC result in a lower-cost communication approach.

E. ANALYSIS OF THE ENERGY CONSUMPTION VS. THE NUMBER OF DRONES

Here, we present a comparison of the total energy consumption of the drones. Figure 11 shows the comparison results of the total energy consumption for the FLR-DC, FLR-CC, and FLR-DC-CC approaches introduced above. The horizontal axis represents the number of drones and the vertical axis represents the energy consumption of the offloading methods.

Here the energy consumption is the sum of the individual energy consumption for computation, communication, hovering, and moving of the drones. According to Fig. 11, FLR-DC-CC has a lower energy consumption than FLR-DC and FLR-CC. It can be seen that when the number of drones is 5, the minimum total energy consumption is $1.62 * 10^8$ J (Joule) for the FLR-DC-CC approach, whereas it is $4.31 * 10^8$ and $5.08 * 10^8$ J for FLR-DC and FLR-CC, respectively.

The reason behind these results is that the FLR-DC-CC approach using DC and CC channels can perform better offloading between drones. In fact, it can find the best destination for task offloading. An effective point in achieving this amount of improvement in energy consumption is the ranking of drones based on their mentioned parameters. The comparison of the methods shows how the use of RIS helps drones to find their offloading destination. Since the drones are distributed in an environment including barriers, using only DC channels is not a good idea for communication, the hybrid use of direct and RIS-based channels has efficient results. The FLR-DC and FLR-CC have higher energy consumption because they used only DC or CC channels for offloading.

Another aspect of the energy consumption analysis is discussed in terms of the number of drones. When there are few drones (1st situation) in the system, the DC channels are more busy compared to when the number of drones is high (2nd situation). In the 1st situation, the FLR-CC has less energy consumption than FLR-DC because in an environment including barriers the drones prefer to use CC channels for communication. On the contrary, in the 2nd situation, there are more DC channels to use by drones for offloading. The hybrid use of DC and CC channels in FLR-DC-CC provides a better understanding of the system and lower energy consumption than the others.

F. ANALYSIS OF THE LATENCY VS. THE NUMBER OF DRONES

Figures 12, 13, and 14 show the comparison of the average computation, communication, and total latencies for the FLR-CC and FLR-DC-CC methods against the FLR-DC method, respectively. We consider this metric because it reflects the impact of direct and RIS-based channels on how long it takes for a drone to perform its real-time task such as tracking an object in the environment, which is a key performance indicator of the application. The individual computation, communication, and total average latency results are presented as follows.

1) AVERAGE COMPUTATION LATENCY

Figure 12 shows the average computation latency of executing tasks in drones. This plot indicates that the proposed approach with the swarm of drones with the FLR-DC-CC approach decreases the average computation latency. The average computation latency for FLR-DC ranges from $2.36 * 10^{-4}$ to $2.93 * 10^{-4}$ ms, FLR-CC ranges from $2.13 * 10^{-4}$ to $2.89 * 10^{-4}$ ms, and FLR-DC-CC range from $1.82 * 10^{-4}$

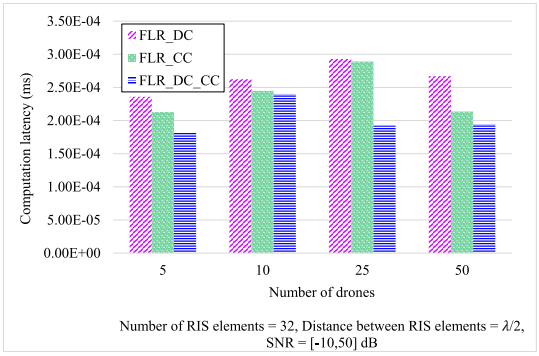

FIGURE 12. Comparison of the computation latency of the FLR-CC and FLR-DC-CC methods against that of the FLR-DC method.

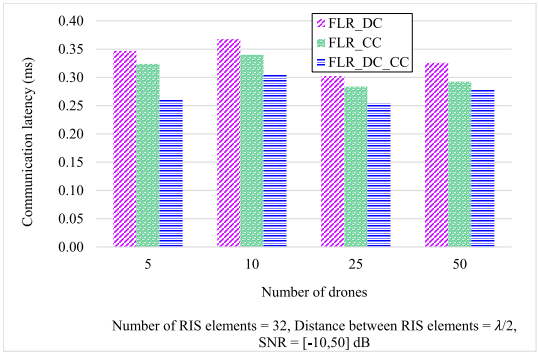

FIGURE 13. Comparison of the communication latency of the FLR-CC and FLR-DC-CC methods against that of the FLR-DC method.

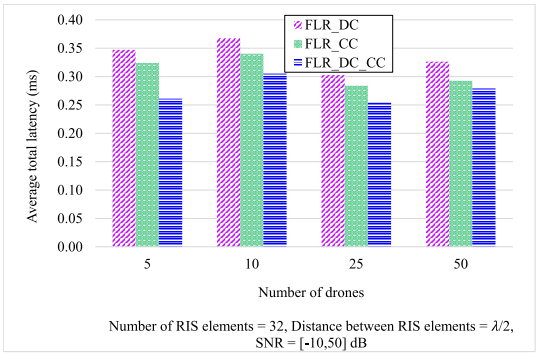

FIGURE 14. Comparison of total latency of the $FLR-CC$ and $FLR-DC_CC$ methods against that of the $FLR-DC$ method.

to $2.40 * 10^{-4}$ ms, respectively. The average computation latency of the FLR-DC-CC is clearly lower than the other methods.

The reason for this result goes back to all rating parameters as they have been mentioned in Eq. (39) and also the role of FLR for optimal use of the DC and CC channels. Especially, an optimal output RSP's value that is calculated by the DL-1 (Channel estimation part of the proposed approach), and using that as an input of the DL-2 was very effective. Moreover, the RIS-based distribution of tasks among a large number of drones yields fair (As Eq. (32), the fairness is based on the energy consumption and latency) resource management, so, almost all drones are equally busy and all computations are executed with cooperation between drones.

2) AVERAGE COMMUNICATION LATENCY VS. THE NUMBER OF DRONES

The aforementioned average communication latency refers to the time it takes between sending a task from a drone and receiving that task in another drone. According to Fig. 13, this value for FLR-DC ranges from 0.30 to 0.37 ms and for FLR-CC from 0.28 to 0.34 ms. The minimum communication latency provided by the FLR strategy ranges from 0.25 to 0.30 ms. Based on this plot we can say that the proposed approach can offload tasks with lower latency than the other methods. As a consequence of using FLR based approach through hybrid DC and CC channels, the waiting time for computation tasks in each drone due to insufficient resources is decreased by offloading them to other drones. In fact, updating the learning models of all drones and also RIS parameters (AoA and AoD) related to them by FLR provides a low latency approach in a swarm of drones.

3) AVERAGE TOTAL LATENCY VS. THE NUMBER OF DRONES

According to Fig. 14, the proposed approach yields the lowest average total latency. The simulation result shows that the average total latency of FLR-DC ranges from 0.30 to 0.37 ms, FLR-CC from 0.28 to 0.34 ms, and the proposed approach from 0.25 to 0.31 ms. It can also be seen that in all cases, the FLR-DC method has the highest latency due to transferring tasks between drones.

G. ANALYSIS OF THE FAIRNESS VS. THE NUMBER OF DRONES

According to the fairness plot of offloading shown in Fig. 15, the minimum value is obtained for FLR-DC which ranges from 89% to 96%, and for FLR-CC which ranges from 90% to 97%. On the other hand, the maximum value is obtained for the FLR-DC-CC which ranges from 91% to 100%. This chart shows that the proposed approach presents the highest distribution of the offloading process. As aforementioned about the energy consumption and latency, the fairness as Eq. (32) is also based on these parameters and the trade-off between them made by the FLR-CC and FLR-DC-CC to be balanced when offloading the drones' computation tasks in a swarm.

H. ANALYSIS OF THE THROUGHPUT VS. THE NUMBER OF DRONES

Analyzing the throughput in Fig. 16 shows that the FLR-DC-CC strategy can transfer larger volumes of data per time than FLR-DC and FLR-CC. In the proposed approach, when the number of drones is 50, the minimum throughput is $3.53 * 10^4$ MB/s for FLR-DC and the maximum is $5.84 * 10^4$ MB/s for FLR-DC-CC. The main reason for this result is the distribution of the RIS-based proposed algorithm. For more clarification, the throughput gives us another view of

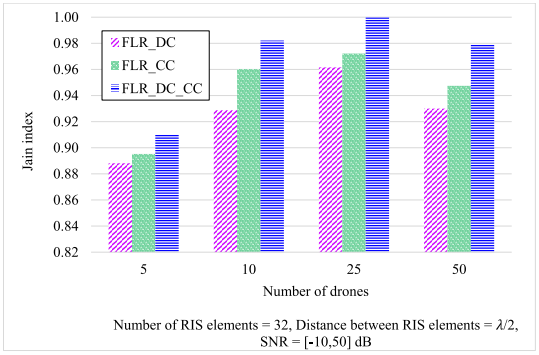

FIGURE 15. Comparison of the fairness in offloading tasks of the FLR-CC and FLR-DC-CC methods against that of the FLR-DC method.

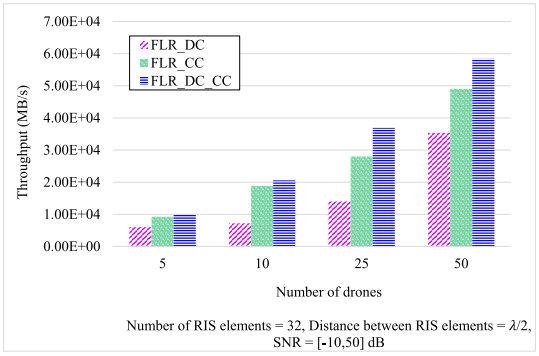

FIGURE 16. Comparison of the throughput of the FLR-CC and FLR-DC-CC methods against that based FLR-DC method.

the performance of the system in terms of delivered tasks to the destination and the size of transmitted data in a swarm of drones. These results reflect the higher data volume in less time in the computation and communication between drones as compared to what the FLR-CC and FLR-DC-CC methods provided. Unlike these two methods, FLR-DC has not a suitable position for drone application in an environment with obstacles because it forces drones to send tasks with a direct link and therefore the need to place them in specific and perhaps inefficient situations.

I. TRADE-OFF BETWEEN ENERGY CONSUMPTION AND THROUGHPUT

Figure 17 shows the trade-off between energy consumption and throughput of the system. It can be seen that for any given energy consumption level, the FLR-DC-CC yields a higher throughput than the other methods. This shows that the hybrid use of DC and CC channels together provides a better use of the available energy.

J. ANALYSIS OF RSP AND RMSE

Figure 18 indicates the average RSP for DC and CC channels. This result was released by different SNR values meaning an area with various noise levels. Obviously, the CC channel caused the drones to receive more signal power than the DC channel.

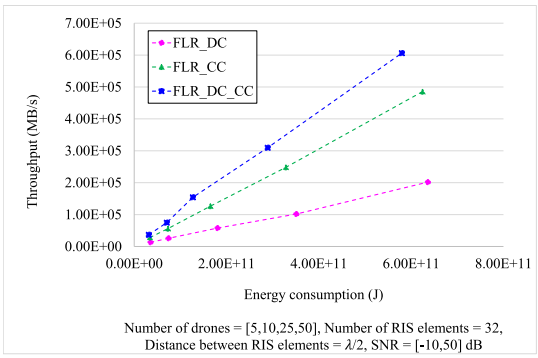

FIGURE 17. Trade-off between the energy consumption and throughput of the FLR-CC and FLR-DC-CC methods against that of the FLR-DC method.

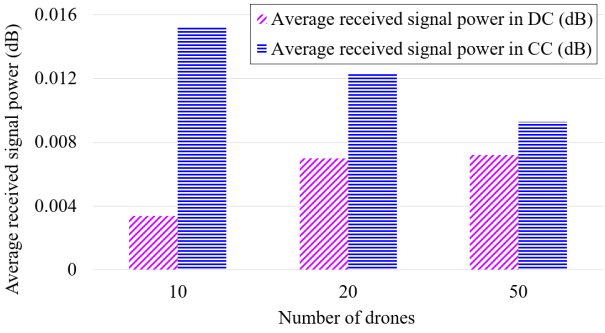

FIGURE 18. Analysis of the average RSP for DC and CC channels.

The channel estimation for CC paths has less RMSE than DC paths. However, using DC channels in a noisy area with obstacles may not be efficiently possible on the contrary having a sustainable channel between drones using CC are more reasonable. Of course, our approach uses both DC and CC channels to empower their capability for continuous communication.

K. SUMMARY OF THE RESULTS

The aforementioned results show the effectiveness of the proposed approaches relative to the reference FLR method with DCs only. In addition to this, we can see how the number of drones affects the ratio of DC and RIS channels as follows.

The average gained channels mean that considering the position of drones relative to barriers and the other drones, how much they used DC or CC channels. The FLR-DC-CC is based on optimized use of the channels and efficient offloading. Figure 20 shows how much this strategy applies direct and indirect (RIS-based) channels in its offloading journey. As can be seen in the figure, with the increase in the number of drones, there are more opportunities to use the DCs for computation offloading in a swarm of drones.

Table 5 summarizes the improvement percentages of the FLR-CC and FLR-DC-CC methods as compared to the FLR-DC method. The rows include total energy

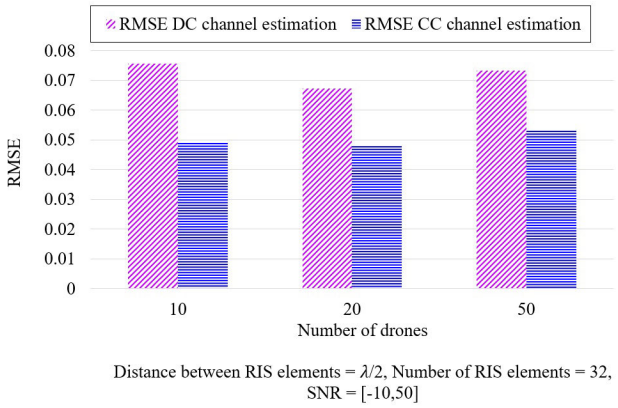

FIGURE 19. Analysis of the RMSE for DC and CC channels.

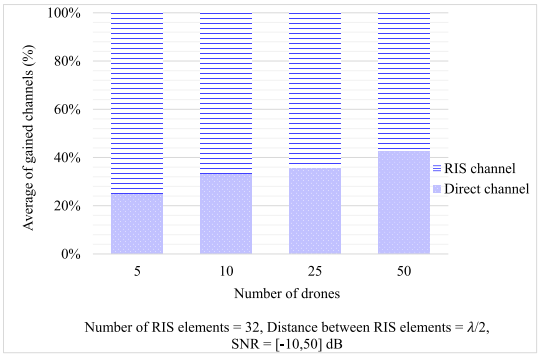

FIGURE 20. Analysis of the average gained channels (DC vs. RIS) for the FLR-CC method as a function of the number of drones in the swarm.

TABLE 5. Improvement percentages of FLR-CC vs. FLR-DC and FLR-DC-CC vs. FLR-DC and vs. FLR-CC.

Parameter	FLR-CC vs. FLR-DC	FLR-DC-CC vs. FLR-DC	FLR-DC-CC vs. FLR-CC
Total energy consumption	-13%	-32%	-21%
Average total latency	-8%	-18%	-11%
Throughput	+40%	+50%	+16%
Total Comm.& Comp. Cost	-12%	-35%	-25%

consumption, average total latency, and throughput values. The presented numbers are calculated based on the average across 5, 10, 25, and 50 drones. The improvement percentage of FLR-CC vs. FLR-DC, FLR-DC-CC vs. FLR-DC, and FLR-DC-CC vs. FLR-CC for total energy consumption are -13%, -32% and -21%, respectively. For total latency, the improvements are -8%, -18%, and -11%, respectively. For the throughput, the improvements are $+40\%$, $+50\%$, and $+16\%$, respectively. Finally, for total communication and computation costs, the improvements are -12%, -35%, and -25%, respectively.

VI. DISCUSSION

Our FLR-based strategy presents a promising solution for addressing the communication challenges within a swarm of drones. By carefully developing and fine-tuning parameters related to RSP, RIS, FL, and the two deep learning models (DL-1 and DL-2), we have devised an optimal approach to enhance various key performance metrics. These metrics include energy consumption, latency, throughput, fairness of computation offloading, RMSE, and the cost of communication and computation. By effectively integrating these processes, we enable drones to cooperatively and efficiently carry out their missions. The outcomes of our approach are derived from the synergistic effects of improved communication and computation efficiencies.

However, it is important to note that our strategy is subject to certain assumptions. These include the flying capabilities of the drones, the utilization of an object detection application involving video capture and tracking, the availability of sufficient resources for computation tasks on each drone, and the presence of heterogeneous drones with varying resource capacities. Additionally, our approach assumes the existence of a drone equipped with a RIS located at the center of the environment.

Further research in the field of communication aspects is necessary to address challenges such as SINR, fading, shadowing, antenna gain, phase shifting, modulation schemes, and multi-RIS communication. The benefits of multi-RIS deployments lie in their ability to actively manipulate the propagation environment i.e. shaping the wireless channels, mitigating signal interference, and enhancing signal quality. It can also effectively amplify the received signal power, increasing the overall coverage and reliability of the system. However, deploying multiple RIS units also introduces certain challenges. Optimally coordinating and controlling multiple RIS units in real-time requires efficient resource allocation and coordination mechanisms. Designing effective algorithms and protocols to synchronize the operations of multiple RIS units can be complex. Additionally, managing signal interference and optimizing the phase shifting of each RIS unit in a multi-path environment can be demanding tasks. By investigating these factors, we can gain deeper insights and advance our understanding of optimizing communication in drone swarm systems.

VII. CONCLUSION

This paper presented a novel FL-based approach for addressing channel estimation and computation offloading challenges in a swarm of drones assisted by RIS. The proposed FL framework leverages two DL models to enable drones to learn optimal channel conditions and offload destinations within multiple clusters. By incorporating information such as AoA, AoD, RSP, and available resources, the proposed solution effectively guides the members of each cluster towards high-rated drones considering a fair offloading, thereby improving overall system performance. The experimental results demonstrate the advantages of employing FL in a swarm of drones with DC and CC channels. Compared to the base FLR approach utilizing DC channels alone, the FL-based swarm exhibits reduced

latency, energy consumption, and cost, while simultaneously achieving enhanced fairness and throughput. These findings highlight the potential of FL as a valuable technique for optimizing communication and computation in swarm-based drone systems. Moving forward, our future research endeavors will focus on developing an optimal and intelligent swarm of drones, specifically tailored to various applications, with a particular emphasis on multi-RIS channels. Addressing challenges such as signal conflicts, data fusion, interference signals, and other pertinent issues will be key objectives of our future investigations. Our preliminary investigation indicates that an FL-based methodology can be instrumental in addressing these challenges effectively.

REFERENCES

[1] B. Shang, R. Shafin, and L. Liu, "UAV swarm-enabled aerial reconfigurable intelligent surface (SARIS)," *IEEE Wireless Commun.*, vol. 28, no. 5, pp. 156–163, Oct. 2021.

[2] R. Liu, M. Li, H. Luo, Q. Liu, and A. L. Swindlehurst, "Integrated sensing and communication with reconfigurable intelligent surfaces: Opportunities, applications, and future directions," *IEEE Wireless Commun.*, vol. 30, no. 1, pp. 50–57, Feb. 2023.

[3] S. Hassouna, M. A. Jamshed, J. Rains, J. U. R. Kazim, M. U. Rehman, M. Abualhayja, L. Mohjazi, T. J. Cui, M. Al. Imran, and Q. H. Abbasi, "A survey on reconfigurable intelligent surfaces: Wireless communication perspective," *IET Commun.*, vol. 17, no. 5, pp. 497–537, 2023.

[4] E. C. Strinati, G. C. Alexandropoulos, H. Wymeersch, B. Denis, V. Sciancalepore, R. D'Errico, A. Clemente, D. Phan-Huy, E. De Carvalho, and P. Popovski, "Reconfigurable, intelligent, and sustainable wireless environments for 6G smart connectivity," *IEEE Commun. Mag.*, vol. 59, no. 10, pp. 99–105, Oct. 2021.

[5] Y. Liu, X. Liu, X. Mu, T. Hou, J. Xu, M. Di Renzo, and N. Al-Dhahir, "Reconfigurable intelligent surfaces: Principles and opportunities," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1546–1577, 3rd Quart., 2021.

[6] A. C. Pogaku, D. Do, B. M. Lee, and N. D. Nguyen, "UAV-assisted RIS for future wireless communications: A survey on optimization and performance analysis," *IEEE Access*, vol. 10, pp. 16320–16336, 2022.

[7] M. Misbah, Z. Kaleem, W. Khalid, C. Yuen, and A. Jamalipour, "Phase and 3D placement optimization for rate enhancement in RIS-assisted UAV networks," *IEEE Wireless Commun. Lett.*, early access, Mar. 30, 2023, doi: 10.1109/LWC.2023.3263224.

[8] I. Alamzadeh, G. C. Alexandropoulos, N. Shlezinger, and M. F. Imani, "A reconfigurable intelligent surface with integrated sensing capability," *Sci. Rep.*, vol. 11, no. 1, p. 20737, Oct. 2021.

[9] Z. Jiang, M. Rihan, P. Zhang, L. Huang, Q. Deng, J. Zhang, and E. M. Mohamed, "Intelligent reflecting surface aided dual-function radar and communication system," *IEEE Syst. J.*, vol. 16, no. 1, pp. 475–486, Mar. 2022.

[10] X. He, Y. Cui, and M. M. Tentzeris, "Tile-based massively scalable MIMO and phased arrays for 5G/B5G-enabled smart skins and reconfigurable intelligent surfaces," *Sci. Rep.*, vol. 12, no. 1, p. 2741, Feb. 2022.

[11] S. Li, B. Duo, X. Yuan, Y. Liang, and M. Di Renzo, "Reconfigurable intelligent surface assisted UAV communication: Joint trajectory design and passive beamforming," *IEEE Wireless Commun. Lett.*, vol. 9, no. 5, pp. 716–720, May 2020.

[12] B. Zheng, C. You, W. Mei, and R. Zhang, "A survey on channel estimation and practical passive beamforming design for intelligent reflecting surface aided wireless communications," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 2, pp. 1035–1071, 2nd Quart., 2022.

[13] M. Di Renzo, A. Zappone, M. Debbah, M. Alouini, C. Yuen, J. De Rosny, and S. Tretyakov, "Smart radio environments empowered by reconfigurable intelligent surfaces: How it works, state of research, and the road ahead," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 11, pp. 2450–2525, Nov. 2020.

[14] S. Basharat, S. A. Hassan, H. Pervaiz, A. Mahmood, Z. Ding, and M. Gidlund, "Reconfigurable intelligent surfaces: Potentials, applications, and challenges for 6G wireless networks," *IEEE Wireless Commun.*, vol. 28, no. 6, pp. 184–191, Dec. 2021.

[15] W. Wu, F. Zhou, B. Wang, Q. Wu, C. Dong, and R. Qingyang Hu, "Unmanned aerial vehicle swarm-enabled edge computing: Potentials, promising technologies, and challenges," 2022, *arXiv:2201.08517*.

[16] S. Mao, S. He, and J. Wu, "Joint UAV position optimization and resource scheduling in space-air-ground integrated networks with mixed cloud-edge computing," *IEEE Syst. J.*, vol. 15, no. 3, pp. 3992–4002, Sep. 2021.

[17] M. D. Nguyen, L. B. Le, and A. Girard, "UAV placement and resource allocation for intelligent reflecting surface assisted UAV-based wireless networks," *IEEE Commun. Lett.*, vol. 26, no. 5, pp. 1106–1110, May 2022.

[18] N. Gao, S. Jin, X. Li, and M. Matthaiou, "Aerial RIS-assisted high altitude platform communications," *IEEE Wireless Commun. Lett.*, vol. 10, no. 10, pp. 2096–2100, Oct. 2021.

[19] X. Cao, B. Yang, C. Huang, G. C. Alexandropoulos, C. Yuen, Z. Han, H. V. Poor, and L. Hanzo, "Massive access of static and mobile users via reconfigurable intelligent surfaces: Protocol design and performance analysis," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 4, pp. 1253–1269, Apr. 2022.

[20] M. Al-Quraan, L. Mohjazi, L. Bariah, A. Centeno, A. Zoha, S. Muhaidat, M. Debbah, and M. A. Imran, "Edge-native intelligence for 6G communications driven by federated learning: A survey of trends and challenges," 2021, *arXiv:2111.07392*.

[21] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist. (PMLR)*, Feb. 2017, pp. 1273–1282.

[22] Y. Shen, Y. Qu, C. Dong, F. Zhou, and Q. Wu, "Joint training and resource allocation optimization for federated learning in UAV swarm," *IEEE Internet Things J.*, vol. 10, no. 3, pp. 2272–2284, Feb. 2023.

[23] D. Rahbari, M. M. Alam, Y. L. Moullec, and M. Jenihhin, "Fast and fair computation offloading management in a swarm of drones using a rating-based federated learning approach," *IEEE Access*, vol. 9, pp. 113832–113849, 2021.

[24] L. Barbieri, S. Savazzi, M. Brambilla, and M. Nicoli, "Decentralized federated learning for extended sensing in 6G connected vehicles," *Veh. Commun.*, vol. 33, Jan. 2022, Art. no. 100396.

[25] X. Hou, Z. Ren, J. Wang, S. Zheng, W. Cheng, and H. Zhang, "Distributed fog computing for latency and reliability guaranteed swarm of drones," *IEEE Access*, vol. 8, pp. 7117–7130, 2020.

[26] X. Mu, Y. Liu, L. Guo, J. Lin, and H. V. Poor, "Intelligent reflecting surface enhanced multi-UAV NOMA networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 10, pp. 3051–3066, Oct. 2021.

[27] S. Zeng, H. Zhang, B. Di, Z. Han, and L. Song, "Reconfigurable intelligent surface (RIS) assisted wireless coverage extension: RIS orientation and location optimization," *IEEE Commun. Lett.*, vol. 25, no. 1, pp. 269–273, Jan. 2021.

[28] A. Mahmoud, S. Muhaidat, P. C. Sofotasios, I. Abualhaol, O. A. Dobre, and H. Yanikomeroglu, "Intelligent reflecting surfaces assisted UAV communications for IoT networks: Performance analysis," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 3, pp. 1029–1040, Sep. 2021.

[29] A. Al-Hilo, M. Samir, M. Elhattab, C. Assi, and S. Sharafeddine, "RIS-assisted UAV for timely data collection in IoT networks," 2021, *arXiv:2103.17162*.

[30] Y. Cao, S. Xu, J. Liu, and N. Kato, "Toward smart and secure V2X communication in 5G and beyond: A UAV-enabled aerial intelligent reflecting surface solution," *IEEE Veh. Technol. Mag.*, vol. 17, no. 1, pp. 66–73, Mar. 2022.

[31] M. Samir, M. Elhattab, C. Assi, S. Sharafeddine, and A. Ghrayeb, "Optimizing age of information through aerial reconfigurable intelligent surfaces: A deep reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 70, no. 4, pp. 3978–3983, Apr. 2021.

[32] P. Mursia, F. Devoti, V. Sciancalepore, and X. Costa-Pérez, "RISe of flight: RIS-empowered UAV communications for robust and reliable air-to-ground networks," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 1616–1629, 2021.

[33] P. Chen, Z. Chen, B. Zheng, and X. Wang, "Efficient DOA estimation method for reconfigurable intelligent surfaces aided UAV swarm," *IEEE Trans. Signal Process.*, vol. 70, pp. 743–755, 2022.

[34] Z. Cui, K. Guan, J. Zhang, and Z. Zhong, "SNR coverage probability analysis of RIS-aided communication systems," *IEEE Trans. Veh. Technol.*, vol. 70, no. 4, pp. 3914–3919, Apr. 2021.

[35] A. M. Elbir, A. Papazafeiropoulos, P. Kourtessis, and S. Chatzinotas, "Deep channel learning for large intelligent surfaces aided mm-wave massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 9, no. 9, pp. 1447–1451, Sep. 2020.
[36] X. Cao, B. Yang, C. Huang, C. Yuen, M. D. Renzo, D. Niyato, and Z. Han, "Reconfigurable intelligent surface-assisted aerial-terrestrial communications via multi-task learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 10, pp. 3035–3050, Oct. 2021.
[37] S. Zhang, H. Zhang, B. Di, and L. Song, "Cellular UAV-to-X communications: Design and optimization for multi-UAV networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 2, pp. 1346–1359, Feb. 2019.
[38] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for multi-UAV enabled wireless networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 2109–2121, Mar. 2018.
[39] H. V. Abeywickrama, B. A. Jayawickrama, Y. He, and E. Dutkiewicz, "Comprehensive energy consumption model for unmanned aerial vehicles, based on empirical studies of battery performance," *IEEE Access*, vol. 6, pp. 58383–58394, 2018.
[40] S. Ghasemi-Falavarjani, M. Nematbakhsh, and B. S. Ghahfarokhi, "Context-aware multi-objective resource allocation in mobile cloud," *Comput. Elect. Eng.*, vol. 44, pp. 218–240, May 2015.
[41] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031–2063, 3rd Quart., 2020, doi: 10.1109/COMST.2020.2986024.
[42] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, edge and fog computing environments," *Softw., Pract. Exper.*, vol. 47, no. 9, pp. 1275–1296, Jun. 2017.
[43] B. Liu, W. Zhang, W. Chen, H. Huang, and S. Guo, "Online computation offloading and traffic routing for UAV swarms in edge-cloud computing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8777–8791, Aug. 2020.
[44] Y. Tan, J. Wang, J. Liu, and N. Kato, "Blockchain-assisted distributed and lightweight authentication service for industrial unmanned aerial vehicles," *IEEE Internet Things J.*, vol. 9, no. 18, pp. 16928–16940, Sep. 2022.

DADMEHR RAHBARI received the B.Sc. degree in computer engineering from the Iran University of Science and Technology, Iran, in 2007, the M.Sc. degree in artificial intelligence from IAUM, Iran, in 2010, and the Ph.D. degree in information security from the Department of Computer Engineering and IT, University of Qom, Iran, in 2020. From 2008 to 2020, he was a Designer, a Programmer, and a Consultant in software engineering and IT in several companies and also a Lecturer and a Researcher with UAST, IAU, TVU, PNU, and MSRT universities in Iran. From 2020 to 2022, he was a Postdoctoral Researcher with the Communication System Research Group, Thomas Johann Seebeck Department of Electronics, School of IT, Tallinn University of Technology (TalTech University), Estonia. He is currently a Researcher with the Center for Reliable Computing Hardware, Department of Computer Systems, School of IT, TalTech University. His major research interests include resource management, distributed machine learning, reliability, and security in autonomous systems. He has been invited to be a TPC, an editorial board member, and the guest editor of several international conferences and journals.

MUHAMMAD MAHTAB ALAM (Senior Member, IEEE) received the M.Sc. degree in electrical engineering from Aalborg University, Denmark, in 2007, and the Ph.D. degree in signal processing and telecommunication from the INRIA Research Center, University of Rennes I, France, in 2013. He joined the Swedish College of Engineering and Technology, Pakistan, as an Assistant Professor, in 2013. From 2014 to 2016, he did his postdoctoral research with the Qatar Mobility Innovation Center, Qatar. In 2016, he joined as a European Research Area (ERA) Chair and an Associate Professor with the Thomas Johann Seebeck Department of Electronics, Tallinn University of Technology, where he was elected as a Professor in 2018. Since 2019, he has been the Communication Systems Research Group Leader. He has more than 15 years of combined academic and industrial multinational experiences while working in Denmark, Belgium, France, Qatar, and Estonia. He has several leading roles as a PI in multimillion Euros international projects funded by the European Commission (EC) (H2020-ICT-2019-3, 951867, NATO-SPS (G5482), Estonian Research Council (PRG424), and Telia Industrial Grant). He is the author or coauthor of more than 100 research publications. He is also a contributor to two standardization bodies (ETSI SmartBAN and IEEE-GeeenICT-EECH), including the Rapporteur of work item: DTR/SmartBAN-0014 and Applying SmartBAN MAC (TS 103 325). His research interests include wireless communications connectivity, NB-IoT 5G/B5G services, and applications, and low-power wearable networks for SmartHealth.

YANNICK LE MOULLEC (Senior Member, IEEE) received the M.Sc. degree from Université de Rennes I, France, in 1999, and the Ph.D. and HDR (accreditation to supervise research) degrees from Université de Bretagne Sud, France, in 2003 and 2016, respectively. From 2003 to 2013, he was a Postdoctoral Researcher, an Assistant Professor, and an Associate Professor with the Department of Electronic Systems, Aalborg University, Denmark. He joined the Thomas Johann Seebeck Department of Electronics, Tallinn University of Technology, Estonia, first as a Senior Researcher, from 2013 to 2016, and a Professor, since 2017. He has supervised or co-supervised more than 50 M.Sc. students and 11 Ph.D. students. He has been involved in more than 20 projects, including five as PI, co-PI, or co-main applicant; one such notable project was the H2020 COEL ERA-Chair Project, from 2015 to 2019. His research interests include embedded systems, reconfigurable systems, the IoT, and the application thereof. He is a member of the IEEE Sustainable ICT Technical Community and the IEEE Circuits and Systems Society.

MAKSIM JENIHHIN (Member, IEEE) received the M.Sc. and Ph.D. degree in computer engineering from the Tallinn University of Technology, in 2004 and 2008, respectively. He is currently a Professor of computing systems reliability with the Department of Computer Systems, Tallinn University of Technology. He participated in numerous EU research projects in FP6, FP7, H2020, and COST frameworks. He is a Project Coordinator of H2020 MSCA ITN RESCUE (2017–2021)-Interdependent Challenges of Reliability, Security and Quality in Nanoelectronic Systems Design, and the Founder of the Biannual European–Latin American Summer School on Design Test and Reliability. His research interests include nanoelectronics lifetime reliability and manufacturing test topics, deep learning methods and HW architectures, electronic design automation (EDA) tools and methodologies for computing systems modeling, verification, and debugging as well as interference analysis of functional and extra-functional design aspects, such as resilience and security. He has published more than 120 research papers on these topics. He served on the executive committees for a number of IEEE conferences (including the Program Chair for ETS, DDECS, and NORCAS) and the guest editor for journals as well as a PC member for many scientific events.