IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 26 April 2024, accepted 22 May 2024, date of publication 27 May 2024, date of current version 3 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3405566

RESEARCH ARTICLE

Optimizing Task Offloading for Collaborative Unmanned Aerial Vehicles (UAVs) in Fog–Cloud Computing Environments

MOHAMMAD ALDOSSARY

Department of Computer Engineering and Information, College of Engineering, Prince Sattam Bin Abdulaziz University, Wadi Al-Dawasir 11991, Saudi Arabia

e-mail: mm.aldossary@psau.edu.sa

This study is supported via funding from Prince Sattam bin Abdulaziz University project number (PSAU/2024/R/1445).

ABSTRACT Unmanned Aerial Vehicles (UAVs) are used in various applications, including crowd management, crime prevention, accident detection, and rescue operations. However, since UAVs perform their tasks independently, some UAV applications are dynamic and geographically distributed, which may require extensive real-time processing capabilities. Thus, processing UAV data locally can be challenging due to their limited computing capabilities. To overcome such limitations, fog and cloud computing can facilitate UAV application development by providing additional resource capacities when needed. Despite this, designing sophisticated and efficient UAV task offloading strategies that collaborate with fog and cloud technologies considering their service latency and energy consumption, is rarely addressed in the literature. Therefore, a collaborative offloading strategy for UAV applications is presented in this work, leveraging fog and cloud computing advantages and capabilities. This approach aims to minimize UAVs' service latency and energy consumption, as well as provide the required resources and services in real time. In addition, task offloading decisions are formulated using the Mixed-Integer Linear Programming (MILP) model to reduce the energy consumption of the entire UAV-fog-cloud system by optimizing the allocation of computation resources and communication requested by each UAV. The simulation results demonstrate that the proposed strategy can significantly reduce UAV service latency by 15.38%, 35.29%, and 59.26%, as well as decrease overall energy consumption (including processing and networking) by 3.3%, 7.37%, and 12% when compared to alternative standalone strategies (namely UAV, fog, and cloud).

INDEX TERMS Unmanned aerial vehicle (UAV), cloud computing, fog computing, collaborative UAVs, energy-efficiency, task offloading.

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) that have sensors, cameras, memory, and communication devices are becoming the most investigated emerging technologies in different fields, such as military, civilian, and industrial applications. Also, UAVs can play significant roles in many areas, such as providing logistic services, controlling, monitoring, and managing crowds, since they are commercially available and inexpensive. Moreover, UAVs can reduce operational costs and risks, and improve work efficiency (e.g., by reducing human interventions and reaching areas that are difficult to access using manned vehicles) [1]. However, UAVs that perform heavy tasks (e.g., image analysis and video recording) require high network traffic and produce more processing data. This leads to the requirement for more computational resources and communication support [2]. Also, processing UAV data locally is a challenging task due to their limited computing capabilities [3]. In this regard, some computing tasks can be offloaded from the UAVs and processed remotely

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

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

on a cloud server, or fog nodes located at the edge of the networks [4].

Both cloud and fog computing are enabling technologies for operating and developing UAV applications, as well as providing additional resource capacity and network coverage for UAVs. Even though cloud computing is capable of providing efficient computing services, there will be significant communication delays when data is offloaded from local UAV devices to a remote cloud and retrieved data from the cloud to the UAVs [5]. Thus, cloud computing often fails to satisfy the requirements of geo-distributed UAV devices and Internet of Things (IoT) sensors in terms of latency-sensitive applications, mobility support, and location awareness. This results in congested networks, high service latency, and poor Quality of Service (QoS). Therefore, a fog computing paradigm has emerged as an intermediary layer to extend cloud resources and services closer to UAV devices. Fog computing supported by sufficient computing resources can reduce service latency during the UAV offloading process [6]. However, there are some UAV tasks that cannot be performed at fog nodes because of their limited computation resources and storage. Therefore, cloud resources are required to execute these tasks [7].

To alleviate these constraints and obtain processing and communication requirements, heavy computation tasks and latency-sensitive tasks can be offloaded from UAVs and slightly processed on more resourceful platforms (i.e., fog and cloud systems). Where cloud services provide resource-intensive, and scalable resources to meet computation demand, while fog services provide low latency for UAV applications to satisfy the stringent delay requirement (since fog nodes are closer to the UAVs). By combining fog and cloud computing models, different service capabilities can be provided (e.g., increasing processing and storage, and decreasing service latency) while maintaining the UAV's resources (e.g., battery) at a healthy level. Also, the collaboration between UAVs can be considered in order to reduce services' latency and support required resources and services in real time.

Therefore, in this work, a collaborative approach for UAV applications is presented to minimize service latency and support the required resources and services in real time. For example, when one UAV requires extra resources in certain areas, the UAVs will collaborate to provide the needed services such as processing capabilities, low service latency, communication, and data storage. In addition, this approach takes advantage of fog and cloud computing capabilities that dynamically support certain UAV applications at different locations. Accordingly, task offloading decisions using the Mixed-Integer Linear Programming (MILP) model are formulated to reduce the energy consumption of the entire proposed fog-cloud system by optimizing the allocation of computation resources and communication requested by each UAV. The following summarizes the main contributions to this work:

- Develop an efficient architecture to investigate the offloading of UAV applications and services into geo-distributed fog-cloud environments.
- Design a collaborative UAV-Fog-Cloud strategy to efficiently support the required resources for UAVs in real time and reduce service latency.
- Optimize UAV offloading decisions over a fog-cloud architecture, considering the power consumption of processing and networking.
- Evaluate the efficiency and applicability of the proposed collaborative offloading strategy, employing a simulation environment and contrasting the results to other existing strategies.

The rest of this work is organized as follows. The related work is discussed in Section II. In Section III, a collaborative UAV–fog–cloud system is presented along with its contact layers. The mixed-integer linear programming (MILP) model design is introduced in Section IV along with its input parameters to optimize the offloading of UAV applications into the UAV–fog–cloud system. Section V explains the experimental setup and analyses the results. The validation of the MILP model through a heuristic approach is introduced in Section VI. Finally, the conclusion and potential future research directions are presented in Section VII.

II. RELATED WORK

To understand UAV-enabled fog-cloud systems from a technological perspective, it is imperative to review current scientific achievements and their limitations related to service latency, task offloading, and energy consumption. In this section, an overview of these achievements is provided with a comparison summary of closely related work, as shown in Table 1.

A. UAV'S SERVICE LATENCY AND TASK OFFLOADING

By employing Mobile Edge Computing (MEC) servers, UAVs act as intermediaries between ground-based users and MEC nodes using a UAV-assisted model [8]. This model leverages MEC storage and computing capabilities to reduce offloaded task service times for IoT users. Accordingly, this would help UAV task scheduling be jointly optimized. 5G mobile networks are incorporated with UAVs in [9] to improve communication reliability and reduce end-to-end latency. Through the use of UAVs associated with MEC, network management has been achieved efficiently, resulting in lower service times and more effective offloading. Zhou et al. [10] developed an iterative algorithm to optimize task offloading, data transmission, UAV computing capacity, UAV location, and service latency. Furthermore, a task-offloading algorithm (iTOA) has been proposed for UAV-enabled MEC services [11]. In their method, a deep Monte Carlo tree algorithm is used to intelligently perceive the network's environment and determine on offloading decisions. Compared to greedy search and game theory methods,

TABLE 1. A comparison summary of the closely related work.

Reference	Service Latency	Task Offloading	Energy Consumption	Collaborative UAVs	Proposed Solution	Evaluation Methods
[8]	✓	✓	✗	✗	UAV-assisted model that leverages MEC storage and computing capabilities.	Simulation and theoretical analysis
[9]	✓	✗	✗	✗	5G mobile networks incorporated with UAVs.	Simulation and testbed
[10]	✓	✓	✗	✗	Iterative algorithm.	Simulation
[11]	✓	✓	✗	✗	Task-offloading algorithm (iTOA) that uses the deep Monte Carlo tree search to intelligently perceive the network's environment and determine offloading decisions.	Simulation
[12]	✗	✓	✓	✗	UAV-enabled MEC architecture based on the Markov Decision Process (MDP).	Simulation
[13]	✗	✓	✓	✗	Block coordinate descent algorithm and successive convex approximation techniques.	Simulation
[14]	✗	✓	✗	✗	Offloading approach based on the MEC architecture.	Simulation
[15]	✗	✗	✓	✗	GEESE system that integrates cloudlets on multiple UAVs.	Simulation
[16]	✗	✗	✓	✗	UAV-based MEC model using Time Division Multiple Access (TDMA) with optimized task partitioning.	Simulation
[17]	✗	✗	✓	✗	Successive convex approximation and Lagrangian duality method.	Simulation
[18]	✗	✗	✓	✗	UAV-assisted MEC system that minimizes its energy consumption by employing k-means clustering.	Simulation
[19]	✗	✗	✓	✗	UAV-enabled MEC system that minimizes energy consumption by scheduling UAV computation resources for both UAV and ground-based users.	Simulation
[20]	✗	✗	✓	✗	Optimization method for UAV-enabled MEC that combines two layers of optimization with a removal agent and differential evolution algorithm.	Simulation
[21]	✗	✗	✓	✗	Optimize communications and computing resource allocation.	Simulation
[22]	✓	✗	✓	✓	Collaborative multi-UAV-assisted MEC system with optimized offloading decisions.	Simulation
[23], [24]	✗	✓	✗	✓	Partial offloading of tasks between a cloud server and several edge nodes for collaborative execution.	Simulation
[25]	✓	✗	✓	✗	Delay-based workload distribution model between local edge nodes, nearby edge nodes, and cloud servers.	Simulation
[26]	✗	✓	✓	✓	Collaborative task offloading algorithm considering energy consumption and battery power.	Simulation
[27]	✓	✗	✗	✓	Multilayered computing strategy named Joint Task Offloading and Service Caching (JTOSC).	Simulation
[29]	✓	✗	✗	✓	Genetic trajectory planning algorithm to operate IoT devices using multiple UAVs.	Simulation
[31]	✓	✓	✓	✗	A graph-based approach for computation-intensive task scheduling in air-ground integrated vehicular network (AGVN), addressing challenges related to service latency, task offloading, and energy consumption.	Simulation
[30]	✓	✓	✓	✗	A futures-enabled resource trading mechanism in edge computing-assisted UAV networks, with a focus on service latency, task offloading, and energy consumption.	Simulation
[32]	✓	✓	✓	✓	An intelligent mobile crowdsensing using edge-assisted UAV networks to improve service latency, optimize energy consumption, and enhance task offloading efficiency.	Simulation
[33]	✓	✓	✓	✓	A cooperative UAVs for dynamic task offloading and resource allocation to minimize service latency and energy consumption.	Simulation
Proposed	✓	✓	✓	✓	Collaborative offloading strategy for UAV applications to enable smooth resource integration between local UAV devices and services, and leverage fog and cloud computing advantages and capabilities using mixed-integrated linear programming (MILP).	Simulation and mathematical modelling

this method performs better than the other methods in terms of service latency.

B. ENERGY CONSUMPTION (PROCESSING AND NETWORKING)

To optimize task offloading and energy consumption, a UAV-enabled MEC architecture was developed by relying on the Markov Decision Process (MDP) [12]. In their study, UAVs were considered as intelligent mobiles. For optimizing data offloading based on energy consumption, a block coordinate descent algorithm and successive convex approximation techniques were presented in [13]. This technique enhanced energy consumption and task offloading by considering a single UAV. Due to UAVs' limited processing capacity, an offloading approach based on the MEC architecture has been introduced in [14]. In their approach, IoT devices are used to generate data from ground-based users that is then sent by UAVs to MEC servers located on a private network for processing.

Using an autonomous delivery network, the energy consumption of UAV-enabled MECs was analyzed in [1]. Also, MEC-based scheduling and task-offloading methods were specifically addressed by a computational management system. Their integrated solution combines static task offloading with dynamic resource scheduling. According to experimental results, the system is able to handle multiple UAV applications with less energy consumption. The GEESE system was proposed to integrate cloudlets on multiple UAVs in order to provide compute services at the edge of a network [15]. Researchers examined the performance of their proposed system to determine the impact of offloading computing tasks on energy efficiency. Du et al. [16] presented a UAV-based MEC model using Time Division Multiple Access (TDMA). Based on their model, task partitioning was optimized to reduce UAV energy consumption.

In an attempt to reduce computational complexity associated with UAV-assisted MEC, the successive convex approximation and lagrangian duality method were introduced in [17]. This approach minimizes the overall energy consumption, which include (computation, communication, and UAV energy). Furthermore, a UAV-assisted MEC system that provides computing services to IoT devices through the use of edge servers, is presented in [18]. This proposed system minimizes its energy consumption by employing k-means clustering. Also, a UAV-enabled MEC system was proposed by Hu et al. [19], employing UAVs as flying MEC nodes. This approach minimized energy consumption by scheduling UAV computation resources for both UAV and ground-based users.

Moreover, an optimization method for UAV-enabled MEC was developed in [20], which combined two layers of optimization. With the help of a removal agent, a differential evolution algorithm was proposed, and power consumption was successfully optimized with this approach. Li et al. [21] addressed the problem of IoT devices' energy consumption in UAV-enabled MEC networks. In their approach, optimizing communications and computing resource allocation have improved energy and offloading requirements.

C. COLLABORATIVE UAVS' TASK OFFLOADING FOR OPTIMIZING SERVICE LATENCY AND ENERGY CONSUMPTION

Recently, the use of UAVs as computational and communication platforms has attracted the attention of many researchers. Some recent works support the idea of collaborative UAVs, for example, a collaborative multi-UAV-assisted MEC system is proposed in [22]. As part of their approach, they studied issues related to service latency and energy minimization. Also, they have optimized offloading decisions, considering each UAV's computation resources and communication requests. Moreover, Chen et al. [23] and Ren et al. [24] introduced offloading methods to partially offload tasks between a cloud server and several edge nodes for collaborative execution. These methods improve service performance and use communication and computing resources more effectively. Further, Guo et al. [25] presented a delay-based workload distribution model. It obtains workload distribution between local edge nodes, nearby edge nodes, and cloud servers to reduce energy consumption and task delay for IoT-edge-cloud systems. In addition, He et al. [26] proposed collaborative task offloading, considering energy consumption and battery power. They have developed an algorithm for collaborative task offloading to handle delay-sensitive tasks effectively. According to Liu et al. [27] a dynamic caching strategy was used to improve collaborative task offloading in MEC. To effectively reduce service delays, they proposed a multilayered computing strategy named Joint Task Offloading and Service Caching (JTOSC). As shown in the simulation results, the proposed strategy performs better in terms of service cache rate, offloading delay, and load balance than existing methods.

Cooperative task assignment is considered an NP-hard problem. In previous studies, the cooperative task assignment problem was formulated using mixed-integrated linear programming (MILP) [28], which achieved an optimal assignment solution. Rahbari et al. [2] proposed a rating method for swarm drones based on a federated learning strategy. The rating method continually computes drones' offloading events, considering current properties (e.g., energy consumption and communication latency). A novel system for MEC has been introduced in [29]. They have presented a genetic trajectory planning algorithm to operate IoT devices, using multiple UAVs. Accordingly, traditional optimization methods are used to estimate the number of UAVs and their constructs to reduce flying distances and hence their communication latency and energy consumption. Liwang et al. [30] proposed a futures-enabled resource trading mechanism designed to facilitate fast and efficient resource allocation in edge computing-assisted UAV networks. The mechanism leveraged futures contracts to enable proactive resource trading between UAVs and edge computing servers. This approach aimed to anticipate future resource demands and efficiently allocate resources to minimize service latency, optimize task offloading decisions, and reduce energy consumption. Additionally, the authors in [31] introduced a graph-represented approach for computation-intensive task scheduling, leveraging the integration of air and ground vehicles in vehicular network (AGVN). The solution involved formulating the task scheduling problem as a graph-based optimization problem, where nodes represent tasks and edges represent communication links between UAVs and vehicles. This approach aimed to minimize service latency and energy consumption while ensuring efficient task offloading and collaboration among UAVs. The authors in [32] proposed a novel approach that leverages edge-assisted UAV networks for intelligent mobile crowdsensing. They introduced task state information sharing mechanisms to enhance task scheduling and resource allocation, aiming to reduce service latency and optimize energy consumption. Also, Guo et al. [33] presented a novel solution for optimizing task offloading and resource allocation among multiple UAVs in advanced wireless communication networks, focusing on reducing service latency, optimizing energy consumption, and leveraging collaborative UAV capabilities.

Based on the above review of related work, multiple approaches have been introduced with various objectives, such as reducing service latency, optimizing task offloading, and minimizing energy consumption. Nevertheless, there are still some challenges due to the variable dynamics of networks and resource allocation, resulting in high service latency and energy consumption for performing tasks on UAVs.

It is important to note that the proposed approach shares similarities with the works discussed in [32] and [33] regarding their objectives. However, it offers several significant improvements compared to these approaches, listed as follows:

- **Integration of Fog and Cloud Computing**: While [32] focuses on edge-assisted UAV networks and [33] addresses resource allocation among multiple UAVs in wireless communication networks, the proposed method integrates fog and cloud computing to overcome the limitations of UAVs' computing capabilities and optimize task offloading strategies. This integration allows for additional resource capacities as needed, thereby enhancing the overall efficiency and effectiveness of UAV applications.
- **Collaborative Offloading Strategy**: In contrast to the methods described in [32] and [33], which primarily focus on optimizing task offloading and resource allocation among UAVs, the proposed method introduces a collaborative offloading strategy leveraging fog and cloud computing capabilities. This strategy aims to minimize UAVs' service latency and energy consumption by optimizing the allocation of computation resources (among UAVs, fog nodes, and cloud servers) and communication requested by each UAV.
- **Mixed-Integer Linear Programming (MILP) Model**: While [32] and [33] use different optimization techniques, the proposed method utilizes a MILP model for task offloading decisions. This model enables the reduction of energy consumption of the entire UAV-fog-cloud system by optimizing the allocation of computation resources and communication, thereby enhancing overall system efficiency.

Motivated by these considerations, a collaborative approach is proposed to enable smooth resource integration between local UAV devices and services, leveraging fog and cloud computing advantages and capabilities. Additionally, an optimization model is presented to minimize UAV service latency and energy consumption, considering different traffic loads and processing requirements when placing UAV applications in a UAV-fog-cloud architecture. It also allows offloading heavy tasks remotely to fog and cloud nodes to utilize their powerful services and effectively support UAV applications.

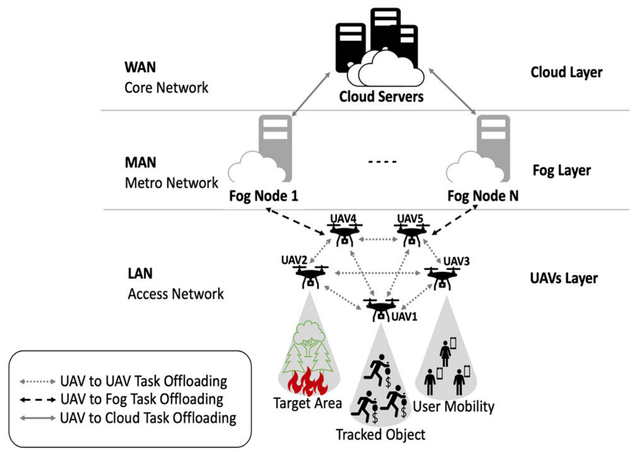

FIGURE 1. The collaborative UAV-fog-cloud architecture.

III. THE PROPOSED SYSTEM ARCHITECTURE

As discussed in sections I and II, some UAV applications may need to be connected to the cloud in order to benefit from advanced services (e.g., auto-scaling resources, powerful processing, and data storage). However, connecting UAV applications to the cloud may have some limitations, as the cloud lacks support for some of the essential requirements of UAV applications such as location awareness and mobility [29]. Also, when a task is submitted to the cloud, this may increase service latency as well as network energy costs [2]. Hence, it is necessary to enhance task offloading strategies in order to efficiently perform UAV applications.

In order to overcome these limitations, fog computing enhances cloud architecture by providing distributed platforms located at the edge of the network and closer to geographically distributed UAVs [4]. It also provides monitoring, processing, and communication services to support certain UAV applications. However, since UAVs work independently to perform their applications, there is a need to develop a collaborative approach for UAVs to support real time task response with low latency.

Therefore, a collaborative approach is proposed to incorporate three layers, namely, the UAV layer, the fog layer, and the cloud layer, as illustrated in Fig. 1. These layers interact with each other to efficiently support UAV offloading decisions. In the following, each layer of the proposed system is described:

A. UAV LAYER

This layer provides a set of collaborative UAVs, equipped with processors, storage, and communication devices, and working as a flying fog node. Also, these collaborative UAVs can be allocated and reallocated to various locations where extra resources are needed to execute different UAV applications and services. Therefore, instead of immediately offloading UAV tasks to cloud or fog nodes, there are several features that can be provided by using collaborative UAVs [3], for example:

- **Rapid Deployment**: UAVs can be quickly deployed to support critical and urgent operations at any location (e.g., accident detection and rescue in areas that are difficult for humans to reach).
- **Collaboration**: Multiple UAVs can be allocated to easily support and satisfy specific application requirements.
- **Resource Elasticity**: Multiple UAVs can be equipped with different resource capabilities (e.g., a temporary need for high processing or more storage).

B. CLOUD AND FOG LAYERS

Cloud computing is a fundamental enabler for the development of UAV applications. It provides on-demand services and large-scale computing resources (e.g., processing, storage, and networking) to meet the requirements of UAVs. However, cloud computing is a centralized design, which inspired researchers to establish distributed services as a cloud extension. Therefore, Cisco introduced the fog computing paradigm in 2014 [34], which expands the usage of cloud resources to the edge of the network (proximity to users) to lower service latency for critical applications (e.g., driverless cars and UAV applications) [35]. Also, fog computing is widely distributed (geographically) and can provide networking, computing, and storage services among cloud data centers and end devices [28].

In light of the limited computing resources of UAV devices along with the massive amount of data generated by UAV applications, it is suggested that tasks requiring substantial computational resources will be performed on computing systems that have adequate computing resources (such as fog or cloud systems).

Hence, both cloud and fog computing are models that deliver computational resources, where fog nodes provide partial computing with low service latency as well as location awareness to optimize the placement of UAV applications, while cloud servers provide substantial computing capability with energy-efficient processing and elastic resources [4]. These models will overcome processing capability issues and service response time (e.g., real time analysis) of UAV applications.

C. TELECOMMUNICATION NETWORKS

The telecommunication network can be divided into three layers [28], including core, metro, and access network layers.

The core network, which usually refers to the Wide Area Network (WAN), is the backbone infrastructure of a telecommunication network, which provides interconnection of large areas (e.g., different cities). In addition, the core network extensively uses the Internet Protocol (IP) via Wavelength Division Multiplexing (WDM) for its large capacity, scalability, and high bandwidth.

The metropolitan region is usually covered by a Metro Area Network (MAN), which has a direct connection to the WAN network. Also, the MAN network infrastructure is operated by metro ethernet technology. This allows connectivity between the WAN network and UAVs that are located in the access network.

The access network indicates a Local Area Network (LAN) that allows end-users to connect to the Internet from various locations. In this work, Passive Optical Networks (PONs) are used, and they are also recommended in the LAN network as a leading choice [36].

IV. MILP MODEL DESIGN

The MILP model is a mathematical optimization technique aiming to find a maximum or minimum solution subject to linear constraints. MILP variables contain both integer and non-integer values, and the studies demonstrated MILP is efficient in optimizing telecommunications for both cloud and fog applications. For instance, researchers in [33] and [34] devised models to enhance the energy efficiency of the WAN network. Similarly, [36] concentrated on formulating an energy-efficient model to optimize application placement within cloud and fog architectures. This model considers diverse CPU workloads along with varying download rates.

This section investigates the effectiveness of offloading UAV applications over a fog-cloud architecture using a cooperative approach based on the MILP optimization model. Additionally, this approach takes into account the three telecommunication network layers, where (LAN) is performed with UAVs, (MAN) with fog nodes, and (WAN) in the cloud.

Furthermore, the proposed system architecture is described in terms of parameters and variables (see Tables 2 and 3). UAV, fog, and cloud layers are included in the system architecture, as previously mentioned. In addition, a mathematical model (MILP) is presented to optimize UAV applications' offloading over a fog-cloud architecture, considering service latency and energy consumption of processing and networking.

A. UAVS, FOG, AND CLOUD LAYERS

Tables 2 and 3 describe the input parameters and variables for UAV, fog, and cloud layers. These tables also represent the UAV applications that will be offloaded onto fog nodes or cloud server, along with their communication (network traffic) and computation power consumption.

The UAV-fog-cloud system architecture consumes power in the following ways:

1) UAV and the access network layer (UAV + Access), shown below:

$$\left(\sum_{s\in N} \mathrm{UAV}_s^{(\mathrm{number})}\mathrm{UAV}^{(\mathrm{power})}\right)$$

TABLE 2. UAV, fog, and cloud layers (computing and networking input parameters) [37].

Parameter	Description	Value
$UAV_s^{(number)}$	Number of UAVs distributed in a geographical location s.	100 UAVs in each city (60% low processing requirements, 30% medium processing requirements, 10% high processing requirements), as listed in Table IV.
$UAV^{(power)}$	Power consumption of a single UAV.	63 Watts
$GW^{(number)}$	Maximum number of connected UAVs per gateway.	3 in each area
$GW^{(power)}$	Power consumption of the gateway.	30 Watts
U_i	Data rate of each UAV i distributed within a single geographical location.	100 Kbps
$OLT^{(power)}$	Power consumption of Optical Line Terminal (OLT) devices.	1.842 kilowatt
$ONU^{(power)}$	Power consumption of Optical Network Unit (ONU) terminals.	5 Watts
$UI_{i,s,d}$	Offloaded data from UAV i in node *s* to either UAV, fog, or cloud located in node d.	$\sum_{d\in N} UI_{i,s,d} = UAV_s^{(number)} U_i$
$OLT^{(bitrate)}$	Traffic rate (bitrate) of an OLT.	1.280 Tbps
$ONU^{(bitrate)}$	Traffic rate (bitrate) of an ONU.	2.4 Gbps
$PUE^{(network)}$	Network Power Usage Effectiveness (PUE) of the networks used.	1.5
$NP^{(cloud)}$	Power consumption per bit for internal networking within the cloud layer.	2.48 Watts/Gbps
$NP^{(fog)}$	Power consumption per bit for internal networking within the fog layer.	2.57 Watts/Gbps
$PCS^{(power)}$	Cloud server power consumption.	630 Watts
$PFS^{(power)}$	Fog node power consumption.	126 Watts
$CS^{(MIPS)}$	Number of MIPS of each cloud node.	18000 MIPS
$FS^{(MIPS)}$	Number of MIPS of each fog node.	3600 MIPS
$PCMIPS^{(cloud)}$	Cloud server power consumption per MIPS operation.	$PCMIPS^{(cloud)} = \frac{PCS^{(power)}}{CS^{(MIPS)}} = \frac{630\ Watts}{18000\ MIPS}$
$PCMIPS^{(fog)}$	Fog node power consumption per MIPS operation.	$PCMIPS^{(fog)} = \frac{PFS^{(power)}}{FS^{(MIPS)}} = \frac{126\ Watts}{3600\ MIPS}$
$PUE^{(cloud)}$	PUE of the cloud layer.	1.1
$PUE^{(fog)}$	PUE of the fog layer.	1.9
$MR^{(power)}$	Power consumption per metro router.	30 Watts
$MS^{(power)}$	Power consumption per metro switch.	470 Watts
$MR^{(bitrate)}$	Traffic rate (bitrate) per metro router.	40 Gbps
$MS^{(bitrate)}$	Traffic rate (bitrate) per metro switch.	0.5 Tbps
$\mathbb{r}^{(power)}$	Router port power consumption in the core network.	37.1 Watts
$\mathfrak{t}^{(power)}$	Transponder power consumption in the core network.	129 Watts
$\mathbb{E}^{(power)}$	Amplifier power consumption in the core network.	11 Watts
$\mathcal{S}_d^{(power)}$	Optical switch power consumption in the core network.	85 Watts
$\mathbb{B}$	Router port bandwidth (bitrate).	40 Gbps

$$
\begin{aligned}
&+\left(\sum\nolimits_{s\in N} GW_s^{(number)} GW^{(power)}\right) \\
&\quad + PUE^{(network)}\left(\sum\nolimits_{s\in N} ONU_s^{(number)} ONU^{(power)}\right) \\
&\quad + \left(\sum\nolimits_{s\in N} OLT_s^{(number)} OLT^{(power)}\right) \qquad (1)
\end{aligned}
$$

2) Fog computing and the metro area network layer (Fog + Metro), shown below:

$$
\begin{aligned}
&PUE^{(fog)}\left(\sum\nolimits_{s\in N} MIPS_{i,s}^{(fog)} PCMIPS^{(fog)}\right. \\
&\quad \left. + \sum\nolimits_{s\in N} NP^{(fog)} TU_{s,d}\right) \\
&\quad + PUE^{(network)}\left(\left(MR_s^{(number)} MR_s^{(power)}\right)\right. \\
&\quad \left. + \left(MS_s^{(number)} MS_s^{(power)}\right)\right) \qquad (2)
\end{aligned}
$$

3) Cloud computing and the core network layer (Cloud + Core), shown below:

$$
\begin{aligned}
&PUE^{(cloud)}\left(\sum\nolimits_{s\in N} MIPS_{i,s}^{(cloud)} PCMIPS^{(cloud)}\right. \\
&\quad \left. + \sum\nolimits_{s\in N} NP^{(cloud)} TU_{s,d}\right) \\
&\quad + PUE^{(network)}\left(\sum\nolimits_{d\in N} \mathbb{r}^{(power)} \mathbb{r}_d\right. \\
&\quad + \sum\nolimits_{m\in N} \sum\nolimits_{n\in Nm_m: n\neq m} \sum\nolimits_{s\in N} \sum\nolimits_{d\in N: s\neq d} \Gamma_{m,n}^{s,d} \mathfrak{t}^{(power)} \\
&\quad + \sum\nolimits_{m\in N} \sum\nolimits_{n\in Nm_m: n\neq m} \mathbb{E}^{(power)} \mathbb{F}_{m,n} \mathbb{A}_{m,n} \\
&\quad \left. + \sum\nolimits_{d\in N} \mathcal{S}_d^{(power)}\right) \qquad (3)
\end{aligned}
$$

The power consumption of UAV (i.e., UAVs and gateway devices), fog, and cloud layers is calculated using functions (1, 2, and 3), considering both the processing nodes and the internal networking (access, metro, and core networks) components, along with the Power Usage Effectiveness (PUE) factors (switch devices, amplifiers, transponders, and router ports). Access, metro, and core networks' power consumption is also defined in functions (1, 2, and 3), involving ONU terminals, OLT devices, and networking PUE.

The proposed MILP model aims to minimize the power consumption of the entire UAV-fog-cloud system. Thus, function (4) yields the overall power consumption of the UAV-fog-cloud architecture by summing the energy usage across various processing and communication layers presented in functions (1 – 3) as follows:

$$\underset{Power}{Minimize}\left[(\mathrm{UAV}+\mathrm{Access})+(\mathrm{Fog}+\mathrm{Metro})+(\mathrm{Cloud}+\mathrm{Core})\right]$$

Subject to (s.t.) the following constraints (C1 – C12):

$$\begin{aligned}
&\mathrm{C1}: \sum\nolimits_{s,d\in N} \mathrm{UI}_{i,s,d} = \sum\nolimits_{s,d\in N} \mathrm{T}^{\mathrm{UAV}}_{i,s,d} \forall i \in \mathbb{I}\\
&\left.\begin{array}{l}\mathrm{C2}: \sum_{s\in N} \mathrm{T}^{\mathrm{UAV}}_{i,s,d} \geq \Psi_{i,d}\\ \mathrm{C3}: \sum_{s\in N} \mathrm{T}^{\mathrm{UAV}}_{i,s,d} \leq \omega\Psi_{i,d}\end{array}\right\} \forall d \in N, i \in \mathbb{I}\\
&\left.\begin{array}{l}\mathrm{C4}: \mathbb{L}^{s,d}_{m,n} \geq \Gamma^{s,d}_{m,n}\\ \mathrm{C5}: \mathbb{L}^{s,d}_{m,n} \leq \Gamma^{s,d}_{m,n}\end{array}\right\} \forall s, d, m, n \in \mathbb{c}\mathcal{f}\\
&\mathrm{C6}: \mathrm{MIPS}^{(fog)}_{i,s} = \Psi_{i,d}\mathrm{MIPS}^{(fog)}_{i,s} \forall d \in N, i \in \mathbb{I}\\
&\mathrm{C7}: \mathrm{MIPS}^{(cloud)}_{i,s} = \sum\nolimits_{i\in\mathbb{I}} \mathrm{MIPS}^{(cloud)}_{i,s} \forall d \in N\\
&\mathrm{C8}: \mathrm{TU}_{s,d} = \sum\nolimits_{i\in\mathbb{I}} \mathrm{T}^{\mathrm{UAV}}_{i,s,d} \forall s, d \in \mathbb{c}\mathcal{f}\\
&\mathrm{C9}: \mathbb{r}_d \geq \frac{\sum_{s\in\mathbb{C}} \mathrm{TU}_{s,d}}{\mathbb{B}} \forall d \in \mathbb{c}\mathcal{f}\\
&\mathrm{C10}: \mathrm{ONU}^{(number)}_{s} + \mathrm{OLT}^{(number)}_{s}\\
&\geq \frac{\sum_{i\in\mathbb{i}} \sum_{d\in N} \mathrm{UI}_{i,s,d}}{\mathrm{ONU}^{(bitrate)} + \mathrm{OLT}^{(bitrate)}} \forall s \in N\\
&\mathrm{C11}: \mathrm{MR}^{(number)}_{s} + \mathrm{MS}^{(number)}_{s}\\
&\geq 2\frac{\sum_{i\in\mathbb{i}} \sum_{d\in(\mathbb{f}\cap\mathbb{C})} \mathrm{UI}_{i,s,d}}{\mathrm{MR}^{(bitrate)} + \mathrm{MS}^{(bitrate)}} \forall s \in N\\
&\mathrm{C12}: \mathrm{T}_s = \sum\nolimits_{i\in\mathbb{i}} \sum\nolimits_{d\in N} \mathrm{UI}_{i,s,d} \forall d \in N
\end{aligned} \tag{4}$$

In constraint (C1), all offloaded data from UAVs must be handled either locally by UAV or remotely by fog nodes or cloud server. The $\Psi_{i,d}$ is a binary variable, which is set by the server (node $d \in N$) as $0 = \mathrm{OFF}$ or $1 = \mathrm{ON}$, these values are based on constraints (C2) and (C3), in order to host the offloaded UAV application $i \in \mathbb{I}$ on the appropriate node/server. The data traverses (using a physical communication link $(m, n) \in \mathbb{c}\mathcal{f}$) between nodes $(s, d) \in \mathbb{c}\mathcal{f}$, are also verified in constraints (C4) and (C5). A constraint (C6) denotes the newly processed requests of UAV devices to offload UAV application $i \in \mathbb{I}$ either locally to the UAV layer or remotely to fog or cloud layers. In constraint (C7), the number of processing requests (in either UAV, fog, or cloud layers, where $d \in N$) is calculated. Also, constraint (C8) describes the data traverse between the core network and the metro network (due to UAVs placed in fog nodes and cloud server). The number of router ports is defined in constraint (C9) for the metro and core networks. In the access network, the number of ONU terminals and OLT devices is considered by constraint (C10), while in the metro network, the number of routers and switches is calculated by constraint (C11). Finally, the total data transmission (total amount of transferred data within the communication network) at each node d is calculated by constraint (C12).

V. THE EXPERIMENTAL SETUP, RESULTS, AND DISCUSSIONS

The proposed models are critically evaluated in this section via a simulation-based experiment. Also, a brief description of the environment setup and resources used is presented. This is followed by a detailed discussion of the results.

A. EXPERIMENT SETUP

This section covers performance metrics for evaluating results. To optimize energy consumption, network resource utilization, and service latency, the proposed approach is evaluated using a simulation environment.

Mixed Integer Linear Programming (MILP) can solve complex optimization problems within a set of linear constraints, where only a limited number of variables are required to be integers, while others can have non-integer values. A CPLEX (IBM ILOG) optimization solver is used in the experiment environment to solve the MILP model. A desktop computer running Windows 10 OS has been used for the simulation experiment. It has a 3.4 GHz Intel Core i7 CPU and 16 GB RAM as well as 512 GB SSD storage. Fig. 2 illustrates the European national network used in this work, which models the Euro 28 topology. It consists of (100 UAVs in each city, 28 fog nodes covering all cities, 1 cloud server, and 62 bidirectional links - 600 km in length). In Table 2, the data rate and MIPS required for each UAV task are distributed according to the heterogeneous computing capabilities of UAVs, fog nodes, and cloud server. Furthermore, each UAV executes different computational tasks (e.g., low, medium, and high processing requirements), as shown in Table 4. Based on the input parameters and variables (listed in Tables 2 and 3), simulation experiments are conducted 50 times, and average values are calculated.

B. EXPERIMENT RESULTS AND DISCUSSIONS

This subsection evaluates the performance of the proposed strategy (UAV-Fog-Cloud Execution) along with three different offloading strategies:

- **UAV Execution**: The tasks will be processed locally within the UAV resources or offloaded to another UAV if the workload exceeds the UAV's capabilities.
- **Fog Execution**: The tasks will be offloaded from UAV to any appropriately connected fog nodes.
- **Cloud Execution**: The tasks will be offloaded from UAV or fog nodes to a cloud server for remote processing.
- **UAV-Fog-Cloud Execution (Proposed Strategy)**: The tasks will be handled locally for processing at the UAV layer or offloaded remotely (to other UAVs, fog nodes, or cloud server) based on the requested resources of the

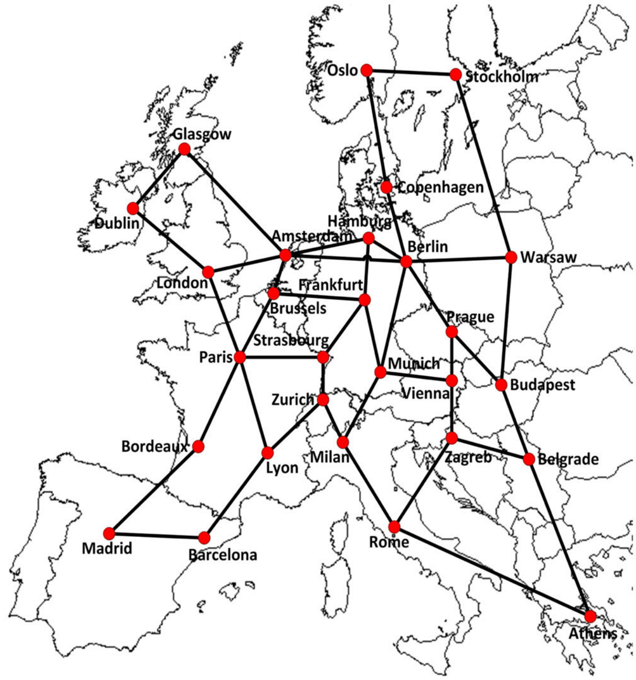

FIGURE 2. The European reference network topology (Euro 28 topology).

UAV applications and the hosted resource capabilities. Using the proposed strategy, the offloading decision can be made to minimize system overhead at the end.

In the following, the results of service latency and energy consumption for processing different UAV tasks relative to the above offloading strategies are discussed.

1) UAV'S SERVICE LATENCY

Fig. 3 presents the service latency versus the number of UAVs for four different offloading strategies (mentioned in section V, subsection B). According to the figure, the service time for all strategies is about the same for a small number of UAVs (e.g., under 20 UAVs). In contrast, the delay for UAV, fog, and cloud strategies increases rapidly with an increasing UAV count (e.g., over 20 UAVs), and the effectiveness of the proposed strategy was superior to UAV, fog, and cloud strategies by 15%, 30.7%, and 55%, respectively. Due to the increase in UAVs, some fog nodes are overloaded while others are underloaded, resulting in poor fog strategy performance. Unlike the proposed strategy, which can autonomously assign and execute tasks, the other offloading strategies do not take into account the multi-level (i.e., UAV, fog, and cloud layers) when executing tasks.

Furthermore, Fig. 4 illustrates the amount of time (service latency) it takes to process computation tasks with different input data sizes. In a comparison of the four strategies, the curves indicate the amount of time and effort needed to implement each one. According to the figure, the data size significantly increases the overall time taken to complete/deliver the service. The proposed strategy has the lowest service latency compared to the other three strategies (i.e., UAV, fog, and cloud) by 15.38%, 35.29%, and 59.26%, respectively. Thus, the proposed strategy can however be adapted to execute computation tasks in the UAV layer, in the fog layer, or in the cloud layer, depending on which location is optimum to perform these tasks.

Fig. 5 shows the execution time of computation tasks using fog nodes with different capabilities. This figure illustrates

TABLE 3. UAV, fog, and cloud layers (computing and networking variables) [37].

Parameter	Description
N	Collection of various nodes within the architecture of the UAV-fog-cloud system.
$\mathbb{c}$	Server at cloud level.
f	Set of nodes at fog level.
i	A number of nodes in UAV layer.
s and d	Source and destination within the architecture of the UAV-fog-cloud system.
$\mathbb{I}$	Set of applications hosting in UAV.
m, n	Source and destination indices of the nodes (m, n) within the UAV-fog-cloud system, where (m, n) are elements of the set N.
N_m	Neighboring node N of node m in the UAV-fog-cloud system.
ω	Large enough positive number.
$\mathbb{A}_{m,n}$	Number of amplifiers present on connected nodes $(m, n) \in \mathbb{c}$. $\mathbb{A}_{m,n} = \left\lfloor \frac{\mathbb{D}_{m,n}}{\mathbb{S}} - 1 \right\rfloor$, $\mathbb{S}$ is the highest length an amplifier can achieve.
$GW_s^{(number)}$	Number of utilized gateways in node s.
$ONU_s^{(number)}$	Number of ONU terminals.
$OLT_s^{(number)}$	Number of OLT devices.
Θ_d	$\begin{cases} \text{Set to "1" if server in node } d \in N \text{ is switched ON} \\ \text{Set to "0" if f server in node } d \in N \text{ is switched OFF} \end{cases}$
$\Psi_{i,d}$	$\begin{cases} \text{Set to "1" if UAV } i \in \mathbb{I} \text{ is placed in a server } d \in N \\ \text{Set to "0", if the condition is not met.} \end{cases}$
$MIPS_{i,s}^{(fog)}$	Processing requirement in MIPS of UAV i placed in either fog s.
$MIPS_{i,s}^{(cloud)}$	Processing requirement in MIPS of UAV i placed in either cloud s.
$T_{i,s,d}^{UAV}$	Data transferred from UAV devices to another UAV i.
$TU_{s,d}$	Data transferred from UAV devices located at node d to either the fog or cloud at node s.
$MR_s^{(number)}$	Total number of routers employed in a metropolitan network at node s.
$MS_s^{(number)}$	Total number of switches employed in a metropolitan network at node s.
$\mathbb{r}_d$	Number of router ports in the core node d belonging to $\mathbb{C}$.
T_s	Total offloaded data in each node s from the set N.
$\mathbb{F}_{m,n}$	Number of fibers on the connection (m, n) belonging to the set $\mathbb{c}$.
$\mathbb{L}_{m,n}^{s,d}$	Offloaded data moves through nodes (s, d) within the set $\mathbb{c}$, transmitting via the physical link (m, n) within set $\mathbb{C}$.
$\mathbb{r}_{m,n}^{s,d}$	$\begin{cases} \text{Set to "1" when the UAV device transmits traffic } (s, d) \\ \text{via the physical link } (m, n) \text{ within the set } \mathbb{c}. \\ \text{Set to "0" if the condition is not met.} \end{cases}$

TABLE 4. **The computational complexity (intensity) of different applications [38].**

Application	Class	MIPS
Automatic Number Plate Reading / Health Monitoring / Sustainable Agriculture.	A	500 MIPS for low processing requirements.
Fire Detection / Video Surveillance.	B	2000 MIPS for medium processing requirements.
Augmented Reality (AR) / Virtual Reality (VR).	C	6000 MIPS for high processing requirements.

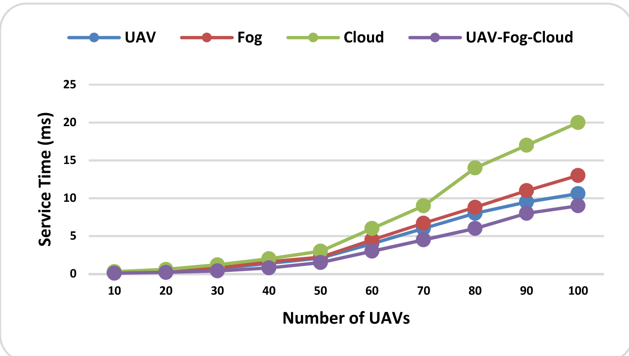

FIGURE 3. **Service time versus the number of UAVs.**

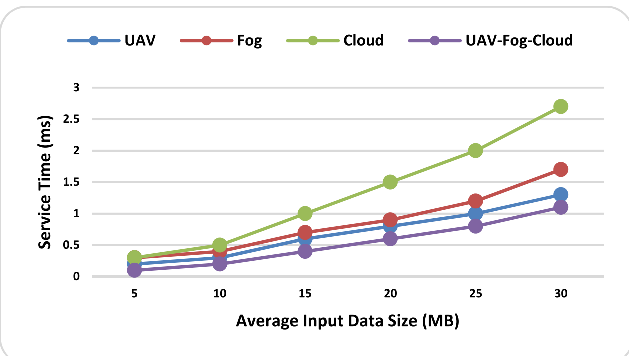

FIGURE 4. **Service time using different input data sizes.**

how UAV and cloud execution strategies are not affected by fog nodes' capabilities, but the service time for fog execution strategy and the proposed strategy decreases as fog nodes' capabilities increase. Also, the proposed strategy performs better than other strategies. To justify that, in the proposed strategy, some UAV tasks are processed locally, while others are offloaded to fog nodes based on the offloading decision. Thus, increasing fog nodes' capabilities means more resources are assigned to UAVs, resulting in shorter execution times for the proposed strategy. In contrast, other standalone strategies (i.e., UAVs and cloud strategies) ignore this benefit.

Finally, Fig. 6 shows the average service time of the four strategies for each type of application (shown in Table 4). The proposed strategy achieved the shortest service latency for all types of applications, while cloud execution had the

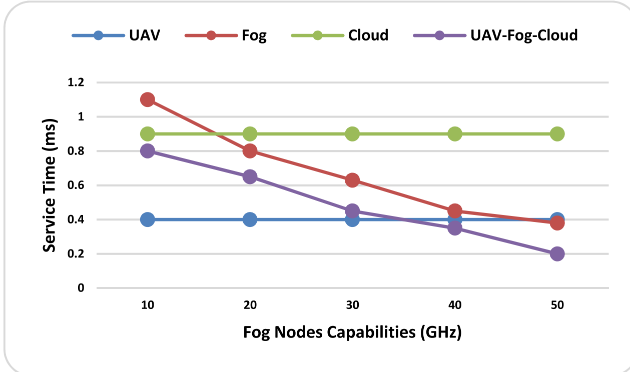

FIGURE 5. **Service time with different capabilities of fog nodes.**

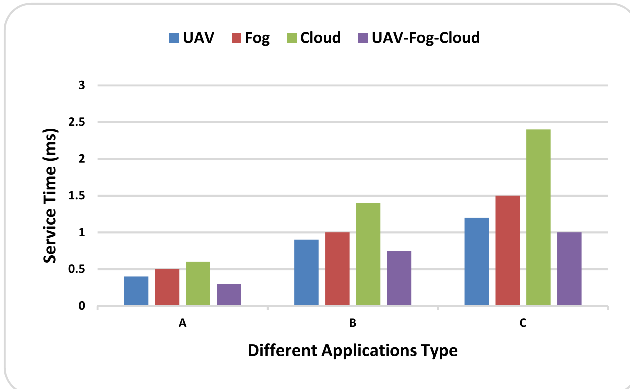

FIGURE 6. **Service time using different applications type.**

longest service latency for all types of applications. Since cloud execution is geographically distant from its users, it takes a long time to execute. Thus, the proposed strategy selects the appropriate location (either in another UAV, fog, or cloud layers) based on the requested resources of the UAV applications.

2) ENERGY CONSUMPTION (PROCESSING AND NETWORKING)

Fig. 7 illustrates the energy consumption of processing tasks using the four strategies with different numbers of UAVs. From this figure, it can be deduced that UAV numbers increase energy consumption exponentially. With a small number of UAVs (e.g., under 40 UAVs), there is no significant energy gap between all strategies. However, this gap is growing with increasing numbers of UAVs, and the proposed strategy achieves the lowest energy consumption compared to other strategies (i.e., UAV, fog, and cloud) by 5.7%, 13.16%, and 17.5%, respectively. It also shows that cloud and fog strategies exceed UAV strategies. Due to the fact that UAVs in all strategies compete for limited communication resources. These resources are used to offload computation tasks to connected fog nodes or cloud server, where most energy is consumed during data transmission. No matter what offloading strategy is used (UAV, fog, cloud, or the pro-

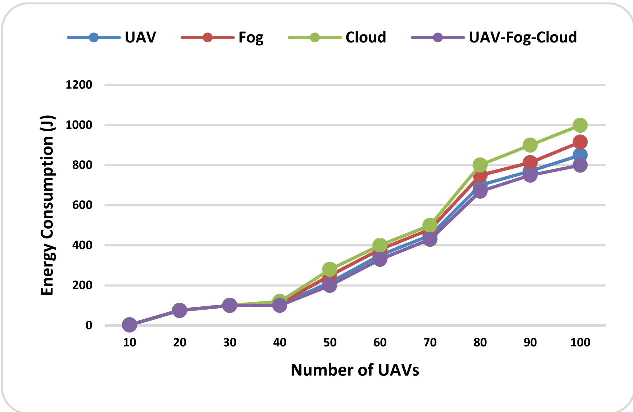

FIGURE 7. Energy consumption versus the number of UAVs.

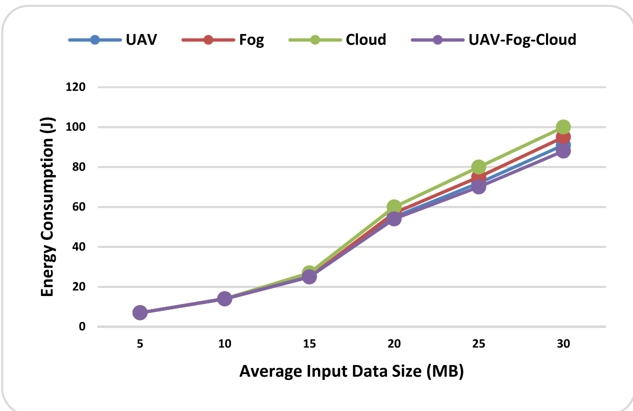

FIGURE 8. Energy consumption using different input data sizes.

posed strategy), communication channel resources compete between UAVs and consume more energy during offloading. Nevertheless, the proposed approach reduces energy consumption significantly compared to cloud and fog strategies.

Likewise, Fig. 8 shows the energy consumption for processing computation tasks using the four strategies based on different input data sizes. According to this figure, with a small data size (less than 15 MB) almost all UAV, fog, and cloud strategies consume the same amount of energy, while the proposed strategy slightly outperforms them. However, when the data size increases, UAV, fog, and cloud strategies consume more energy than the proposed strategy by 3.3%, 7.37%, and 12%, respectively. This is due to the lengthening of communication times and the increasing size of data, which affects total energy consumption. Hence, by smartly adapting the proposed strategy to handle computation tasks, energy consumption can be reduced. In addition, it can be determined the most energy-effective decision (the best location to execute UAV applications).

Figure 9 presents the energy consumption of processing UAV tasks using the four strategies relative to different fog node numbers. This figure shows that the UAV and cloud execution strategies are not affected by the number of fog nodes,

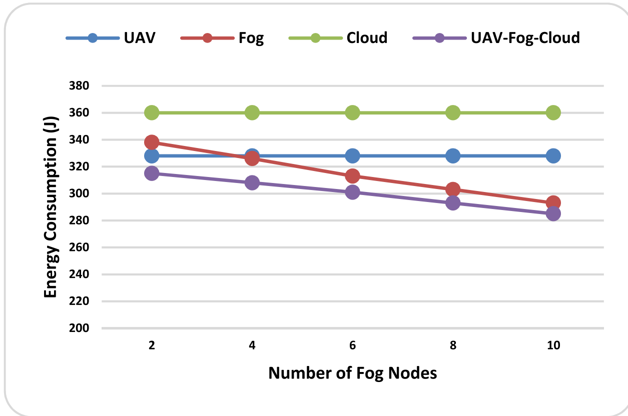

FIGURE 9. Energy consumption with different numbers of fog nodes.

while the other strategies (fog and the proposed strategies) continuously consume less energy when the number of fog nodes increases. Additionally, the proposed strategy enables lower energy consumption and better performance than the other strategies. Due to an increase in fog nodes, UAVs are assigned more resources (in the proposed strategy), which leads to a decrease in energy consumption, while the cloud strategy does not use fog nodes' resources.

VI. THE VALIDATION OF THE MILP MODEL THROUGH A HEURISTIC APPROACH

In this section, an alternative method is introduced to validate the results of UAV application offloading generated by the proposed MILP model. However, addressing the optimization of UAV application offloading utilizing the MILP model to minimize service latency and power consumption is recognized as an NP-hard problem [28]. Due to the multiple possibilities of offloading locations ($L = 29$) in the scenario considered (including 28 fog nodes covering all cities, 1 cloud server, and 62 bidirectional links spanning 600 km), the MILP model faces significant challenges in finding the optimal solution within polynomial time, set as $\left(\sum_{n=1}^{L} \frac{L!}{(L-n)!}\right)$.

To address this issue, a sorted list of nodes across different cities has been generated based on their weights, thereby facilitating the selection of the optimal offloading location for UAV applications. This approach leads to a substantial reduction in the time required for assessing combinations of $\left(\sum_{n=1}^{L} \frac{L!}{(L-n)!}\right)$, thus reducing the time complexity by a factor of 1.5 x 10^{10}. Here, L is designated to represent the number of UAV applications, while n denotes the diverse cloud server and fog node server locations. The heuristic solution for efficient UAV application offloading within a fog-cloud architecture is presented in Algorithm 1.

The algorithm computes the total power consumption (PC) linked to offloading each UAV application $x \in app$ within its designated placement $O \subset$ R, this would be based on the resource requirements of each UAV application type. Initially, a multi-hop heuristic approach is employed by the algorithm

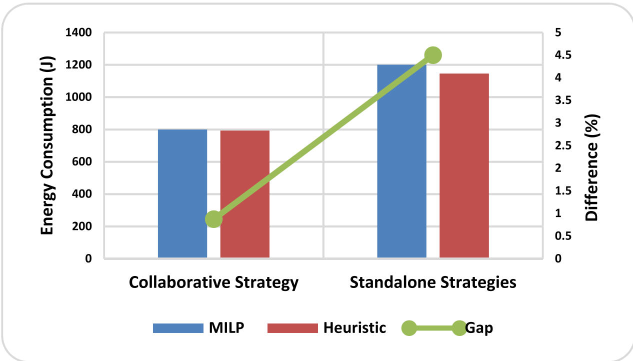

FIGURE 10. Difference between the MILP model and heuristic method.

to distribute UAV applications through the access and core networks to target nodes d and s, thereby estimating the power consumption of these networks. Subsequently, the algorithm determined the best location (O') to host UAV applications and estimates the total power consumption (PC) of the fog-cloud architecture.

Algorithm 1 Optimizing the Offloading of UAV Applications

$PFS^{(power)}$: Fog node power consumption (f) of offloading UAV application *app* into designated location O.
$PCS^{(power)}$: Cloud server power consumption (c) of offloading UAV application *app* into designated location O.
Input: The sorted list of nodes (R).
Output: Optimizing the total power consumption (PC).
Optimizing the offloading decision of UAV application (O').

for each (Type of UAV application $x \in app$) **do**
  for each (Offloading $n \subset v$)**do**
    for each (Node $d \in s$)**do**
      for each (Fog/Cloud $fc \in n$)**do**
        $PC = (PFS^{(power)} R_d PUE^{(fog)})(PCS^{(power)} R_s PUE^{(cloud)})$
      end for
    end for
    $PC_{app,v} = \left(\sum_{s \in N} PC_{app,fc,v}\right) + LAN_{app,f} + MAN_{app,c}$
  end for
  $PC_{app} = \text{Min}\ \{PC_{app,v}\}$
  $O' = O$
end for
Calculate $PC = \sum_{x \in app} PC_{app}$

Additionally, the heuristic method has been assessed using an HP PC operating Windows 11. The PC is equipped with an 8th Generation of Intel Core i7-12700F processor operating at 4.60 GHz, 512GB SSD storage, and 16GB DDR4 RAM. Employing the same European national topology (Euro 28 network) as the MILP model, the heuristic approach produced results comparable to the MILP model within a 7-second evaluation timeframe. As depicted in Fig. 10, the maximum difference between the heuristic method and the MILP model in terms of total power consumption was 4.5%, while similar savings were achieved.

VII. CONCLUSION AND FUTURE WORKS

Most computation offloading approaches for UAV applications are based on a single-level architecture. Therefore, a collaborative UAVs approach assisted with a multi-level fog-cloud system is presented in this work. Also, service latency and energy consumption issues are investigated by optimizing offloading decisions, communications, and computing resource allocations over collaborative UAVs as well as fog-cloud environments. This helped to find the optimal location for the placement of UAV tasks for processing. Additionally, mathematical modeling is provided to demonstrate the effectiveness of the proposed approach by comparing the results with other existing strategies (e.g., local and remote executions). From the results, it is clear that the proposed strategy (UAV-Fog-Cloud execution) is capable of reducing UAV's service latency by (15.38%, 35.29%, and 59.26%), and overall energy consumption of both processing and networking by (3.3%, 7.37%, and 12%) compared to other standalone strategies (i.e., UAV, fog, and cloud), respectively. A part of future work would include artificial intelligence and machine learning methodologies such as deep learning and reinforcement learning to handle the complexity of fog-cloud computing systems for enhancing UAVs' trajectories and task offloading decisions.

ACKNOWLEDGMENT

The author would like to thank the Deanship of Scientific Research, Prince Sattam Bin Abdulaziz University, Al-Kharj, Saudi Arabia, for providing research resources and equipment.

REFERENCES

[1] J. Almutairi, M. Aldossary, H. A. Alharbi, B. A. Yosuf, and J. M. H. Elmirghani, "Delay-optimal task offloading for UAV-enabled edge-cloud computing systems," *IEEE Access*, vol. 10, pp. 51575–51586, 2022, doi: 10.1109/ACCESS.2022.3174127.

[2] 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, doi: 10.1109/ACCESS.2021.3104117.

[3] N. Mohamed, J. Al-Jaroodi, I. Jawhar, H. Noura, and S. Mahmoud, "UAVFog: A UAV-based fog computing for Internet of Things," in *Proc. IEEE SmartWorld, Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People Smart City Innov. (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, Aug. 2017, pp. 1–8, doi: 10.1109/UIC-ATC.2017.8397657.

[4] F. H. Rahman, S. H. S. Newaz, T.-W. Au, W. S. Suhaili, M. A. P. Mahmud, and G. M. Lee, "EnTruVe: ENergy and TRUst-aware virtual machine allocation in vehicle fog computing for catering applications in 5G," *Future Gener. Comput. Syst.*, vol. 126, pp. 196–210, Jan. 2022, doi: 10.1016/j.future.2021.07.036.

[5] E. Mustafa, J. Shuja, K. Bilal, S. Mustafa, T. Maqsood, F. Rehman, and A. U. R. Khan, "Reinforcement learning for intelligent online computation offloading in wireless powered edge networks," *Cluster Comput.*, vol. 26, no. 2, pp. 1053–1062, Apr. 2023, doi: 10.1007/s10586-022-03700-5.

[6] M. Maray and J. Shuja, "Computation offloading in mobile cloud computing and mobile edge computing: Survey, taxonomy, and open issues," *Mobile Inf. Syst.*, vol. 2022, pp. 1–17, Jun. 2022, doi: 10.1155/2022/1121822.

[7] E. Mustafa, J. Shuja, S. K. uz Zaman, A. I. Jehangiri, S. Din, F. Rehman, S. Mustafa, T. Maqsood, and A. N. Khan, "Joint wireless power transfer and task offloading in mobile edge computing: A survey," *Cluster Comput.*, vol. 25, no. 4, pp. 2429–2448, Aug. 2022, doi: 10.1007/s10586-021-03376-3.

[8] B. Zhang, G. Zhang, S. Ma, K. Yang, and K. Wang, "Efficient multitask scheduling for completion time minimization in UAV-assisted mobile edge computing," *Mobile Inf. Syst.*, vol. 2020, pp. 1–11, Jun. 2020, doi: 10.1155/2020/8791030.

[9] S. Zhu, L. Gui, D. Zhao, N. Cheng, Q. Zhang, and X. Lang, "Learning-based computation offloading approaches in UAVs-assisted edge computing," *IEEE Trans. Veh. Technol.*, vol. 70, no. 1, pp. 928–944, Jan. 2021, doi: 10.1109/TVT.2020.3048938.

[10] Y. Zhou, C. Pan, P. L. Yeoh, K. Wang, M. Elkashlan, B. Vucetic, and Y. Li, "Secure communications for UAV-enabled mobile edge computing systems," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 376–388, Jan. 2020, doi: 10.1109/TCOMM.2019.2947921.

[11] J. Chen, S. Chen, S. Luo, Q. Wang, B. Cao, and X. Li, "An intelligent task offloading algorithm (iTOA) for UAV edge computing network," *Digit. Commun. Netw.*, vol. 6, no. 4, pp. 433–443, Nov. 2020, doi: 10.1016/j.dcan.2020.04.008.

[12] W. Ouyang, Z. Chen, J. Wu, G. Yu, and H. Zhang, "Dynamic task migration combining energy efficiency and load balancing optimization in three-tier UAV-enabled mobile edge computing system," *Electronics*, vol. 10, no. 2, p. 190, Jan. 2021, doi: 10.3390/electronics10020190.

[13] H. Guo and J. Liu, "UAV-enhanced intelligent offloading for Internet of Things at the edge," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2737–2746, Apr. 2020, doi: 10.1109/TII.2019.2954944.

[14] J. Liu, A. Zhou, C. Liu, T. Zhang, L. Qi, S. Wang, and R. Buyya, "Reliability-enhanced task offloading in mobile edge computing environments," *IEEE Internet Things J.*, vol. 9, no. 13, pp. 10382–10396, Jul. 2022, doi: 10.1109/JIOT.2021.3115807.

[15] M. Liyanage, F. Dar, R. Sharma, and H. Flores, "GEESE: Edge computing enabled by UAVs," *Pervas. Mobile Comput.*, vol. 72, Apr. 2021, Art. no. 101340, doi: 10.1016/j.pmcj.2021.101340.

[16] Y. Du, K. Yang, K. Wang, G. Zhang, Y. Zhao, and D. Chen, "Joint resources and workflow scheduling in UAV-enabled wirelessly-powered MEC for IoT systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 10, pp. 10187–10200, Oct. 2019, doi: 10.1109/TVT.2019.2935877.

[17] T. Zhang, Y. Xu, J. Loo, D. Yang, and L. Xiao, "Joint computation and communication design for UAV-assisted mobile edge computing in IoT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5505–5516, Aug. 2020, doi: 10.1109/TII.2019.2948406.

[18] Y. Liu, K. Xiong, Q. Ni, P. Fan, and K. B. Letaief, "UAV-assisted wireless powered cooperative mobile edge computing: Joint offloading, CPU control, and trajectory optimization," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 2777–2790, Apr. 2020, doi: 10.1109/JIOT.2019.2958975.

[19] X. Hu, K.-K. Wong, K. Yang, and Z. Zheng, "UAV-assisted relaying and edge computing: Scheduling and trajectory optimization," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4738–4752, Oct. 2019, doi: 10.1109/TWC.2019.2928539.

[20] Y. Wang, Z.-Y. Ru, K. Wang, and P.-Q. Huang, "Joint deployment and task scheduling optimization for large-scale mobile users in multi-UAV-enabled mobile edge computing," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3984–3997, Sep. 2020, doi: 10.1109/TCYB.2019.2935466.

[21] W.-T. Li, M. Zhao, Y.-H. Wu, J.-J. Yu, L.-Y. Bao, H. Yang, and D. Liu, "Collaborative offloading for UAV-enabled time-sensitive MEC networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2021, no. 1, p. 1, Dec. 2021, doi: 10.1186/s13638-020-01861-8.

[22] Y. K. Tun, T. N. Dang, K. Kim, M. Alsenwi, W. Saad, and C. S. Hong, "Collaboration in the sky: A distributed framework for task offloading and resource allocation in multi-access edge computing," *IEEE Internet Things J.*, vol. 9, no. 23, pp. 24221–24235, Dec. 2022.

[23] Y. Chen, Z. Li, B. Yang, K. Nai, and K. Li, "A Stackelberg game approach to multiple resources allocation and pricing in mobile edge computing," *Future Gener. Comput. Syst.*, vol. 108, pp. 273–287, Jul. 2020, doi: 10.1016/j.future.2020.02.045.

[24] J. Ren, G. Yu, Y. He, and G. Y. Li, "Collaborative cloud and edge computing for latency minimization," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 5031–5044, May 2019, doi: 10.1109/TVT.2019.2904244.

[25] M. Guo, L. Li, and Q. Guan, "Energy-efficient and delay-guaranteed workload allocation in IoT-edge-cloud computing systems," *IEEE Access*, vol. 7, pp. 78685–78697, 2019, doi: 10.1109/ACCESS.2019.2922992.

[26] C. He, R. Wang, and Z. Tan, "Energy-aware collaborative computation offloading over mobile edge computation empowered fiber-wireless access networks," *IEEE Access*, vol. 8, pp. 24662–24674, 2020, doi: 10.1109/ACCESS.2020.2969214.

[27] X. Liu, X. Zhao, G. Liu, F. Huang, T. Huang, and Y. Wu, "Collaborative task offloading and service caching strategy for mobile edge computing," *Sensors*, vol. 22, no. 18, p. 6760, Sep. 2022, doi: 10.3390/s22186760.

[28] M. Aldossary and H. A. Alharbi, "Towards a green approach for minimizing carbon emissions in fog-cloud architecture," *IEEE Access*, vol. 9, pp. 131720–131732, 2021, doi: 10.1109/ACCESS.2021.3114514.

[29] M. Asim, W. K. Mashwani, S. B. Belhaouari, and S. Hassan, "A novel genetic trajectory planning algorithm with variable population size for multi-UAV-assisted mobile edge computing system," *IEEE Access*, vol. 9, pp. 125569–125579, 2021, doi: 10.1109/ACCESS.2021.3111318.

[30] M. Liwang, Z. Gao, and X. Wang, "Let's trade in the future! A futures-enabled fast resource trading mechanism in edge computing-assisted UAV networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 11, pp. 3252–3270, Nov. 2021, doi: 10.1109/JSAC.2021.3088657.

[31] M. Liwang, Z. Gao, S. Hosseinalipour, Y. Su, X. Wang, and H. Dai, "Graph-represented computation-intensive task scheduling over air-ground integrated vehicular networks," *IEEE Trans. Services Comput.*, vol. 16, no. 5, pp. 3397–3411, Sep. 2023, doi: 10.1109/TSC.2023.3270169.

[32] L. Deng, W. Gong, M. Liwang, L. Li, B. Zhang, and C. Li, "Towards intelligent mobile crowdsensing with task state information sharing over edge-assisted UAV networks," *IEEE Trans. Veh. Technol.*, pp. 1–16, Feb. 2024, doi: 10.1109/TVT.2024.3369089.

[33] H. Guo, Y. Wang, J. Liu, and C. Liu, "Multi-UAV cooperative task offloading and resource allocation in 5G advanced and beyond," *IEEE Trans. Wireless Commun.*, vol. 23, no. 1, pp. 347–359, Jan. 2024, doi: 10.1109/TWC.2023.3277801.

[34] H.-J. Cha, H.-K. Yang, and Y.-J. Song, "A study on the design of fog computing architecture using sensor networks," *Sensors*, vol. 18, no. 11, p. 3633, Oct. 2018, doi: 10.3390/s18113633.

[35] A. Yousefpour, G. Ishigaki, and J. P. Jue, "Fog computing: Towards minimizing delay in the Internet of Things," in *Proc. IEEE Int. Conf. Edge Comput. (EDGE)*, Jun. 2017, pp. 17–24.

[36] M. Aldossary and H. A. Alharbi, "An eco-friendly approach for reducing carbon emissions in cloud data centers," *Comput., Mater. Continua*, vol. 72, no. 2, pp. 3175–3193, 2022, doi: 10.32604/cmc.2022.026041.

[37] M. Aldossary, "Multi-layer fog-cloud architecture for optimizing the placement of IoT applications in smart cities," *Comput., Mater. Continua*, vol. 75, no. 1, pp. 633–649, 2023, doi: 10.32604/cmc.2023.035414.

[38] H. A. Alharbi, M. Aldossary, J. Almutairi, and I. A. Elgendy, "Energy-aware and secure task offloading for multi-tier edge-cloud computing systems," *Sensors*, vol. 23, no. 6, p. 3254, Mar. 2023, doi: 10.3390/s23063254.

MOHAMMAD ALDOSSARY received the B.Sc. degree (Hons.) in computer from King Saud University, Riyadh, Saudi Arabia, in 2009, the M.Sc. degree in computer science from Southern Polytechnic State University, Georgia, USA, in 2013, and the Ph.D. degree in computer science from the University of Leeds, U.K., in 2019. He is currently an Associate Professor with the Department of Computer Engineering and Information, Faculty of Engineering College, Prince Sattam Bin Abdulaziz University (PSAU), Saudi Arabia. His main research interests include distributed systems, including cloud, fog, and edge computing, and the Internet of Things (IoT) technologies, such as the Internet of Medical Things (IoMT), the industrial Internet of Things (IIoT), artificial intelligence (AI), machine learning (ML), digital twin, metaverse, smart cities, unmanned aerial vehicles (UAVs), smart agriculture, system architectures, resource management, and energy efficiency. He holds the honor of being a Founding Member of the Artificial Intelligence Governance Association (AIGA), Saudi Arabia, and serves as a Board Member for Saudi Internet of Things Association, Saudi Arabia.