This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3174127, IEEE Access

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Delay-Optimal Task Offloading for UAV-Enabled Edge-Cloud Computing Systems

JABER ALMUTAIRI[1], MOHAMMAD ALDOSSARY[2], HATEM A. ALHARBI[3], (Member, IEEE), BARZAN A. YOSUF[4], and JAAFAR M. H. ELMIRGHANI[4], (Fellow, IEEE)

[1]Department of Computer Science, College of Computer Science and Engineering, Taibah University, Madinah 42353, Saudi Arabia
[2]Department of Computer Science, College of Arts and Science, Prince Sattam bin Abdulaziz University, Al-Kharj 16278, Saudi Arabia
[3]Department of Computer Engineering, College of Computer Science and Engineering, Taibah University, Madinah 42353, Saudi Arabia
[4]School of Electronic and Electrical Engineering, University of Leeds, Leeds LS2 9JT, U.K

Corresponding author: Mohammad Aldossary (mm.aldossary@psau.edu.sa).

ABSTRACT

The emergence of delay-sensitive and computationally-intensive mobile applications and services pose a significant challenge for Unmanned Aerial Vehicles (UAVs) devices due to the scarcity in their resources such as computational power and battery lifetime. Mobile cloud computing has been introduced as a promising solution to overcome these limitations through task offloading. However, high-latency and security issues are considered the main challenges of this paradigm. Subsequently, the edge-cloud computing paradigm has been introduced and widely used to help to mitigate these issues. Nevertheless, the current task offloading models permit UAVs to execute their intensive tasks at the connected edge server, which leads to excessive loads due to the large number of UAVs and thereby increases the delay. Therefore, in this paper, we propose a delay-optimal task offloading approach for multi-tier edge-cloud computing in a multi-user environment. The problem is formulated as an optimization model using Integer Linear Programming (ILP) techniques to minimize the total service time of UAVs. Simulation results demonstrate that the proposed approach not only saves the service time by 33.5% and 55% for edge and cloud execution policies respectively, but also scales well for a large number of UAVs.

INDEX TERMS Computation offloading, edge-cloud computing, mobile edge computing, optimization, Internet of Things, unmanned aerial vehicles, linear programming.

I. INTRODUCTION

Nowadays, the Internet of Things (IoT) is fully embraced in virtually every aspect of our lives, owing to the advances made in Unmanned Aerial Vehicles (UAVs), sensors, and communication technologies. Such advancements have aided in the proliferation of complex IoT applications, which can generate and process a massive amount of data [1]. However, UAVs have limited resources onboard such as battery and computational power that restricts the execution of such types of applications on these devices [2], [3].

To alleviate these limitations and achieve the required communication and processing delay, intensive computations can be transmitted and remotely processed at more resourceful devices via the computation offloading concept [4], [5]. Consequently, two types of computation offloading models are proposed namely binary offloading [6] and partial offloading [7]. In the case of binary offloading, the computation task is either executed locally at the UAV or offloaded and remotely executed at the edge server, whereas, in partial offloading, the computation task is divided into two parts, one for local computing and the other one for remote execution.

The Mobile Cloud Computing (MCC) paradigm has been introduced as a prominent solution, in which the highly intensive task will be offloaded and remotely executed at a centralized cloud server [8], [9]. However, high-latency and security issues are determined as the main challenges of this paradigm. Subsequently, the cloudlet concept is introduced to address the latency problem, in which the computational and storage capabilities of the cloud can be provided within the radio range of UAVs' Wi-Fi access points [10]. However, the scaling and limited acceptability (Wi-Fi range) issues are considered the main challenges of the cloudlet approach.

As a result, edge-cloud computing paradigms are considered viable and promising solutions to provide flexible processing, storage, and services capabilities, while reducing battery consumption and service latency [11]–[17].

Recently, researchers have proposed and developed computation offloading models and approaches with different objectives to address the limitations of these devices [18]. For example, a new trajectory scheduling and stochastic offloading model was proposed for UAV-enabled MEC, in which minimizing the total energy consumption was the main objective [19]. Whereas, minimizing the delay and energy consumption of computation offloading in UAV-enabled MEC system was considered in [20]. Moreover, the traffic monitoring problem was considered in [21] using UAVs, in which UAVs were able to offload and share the execution of their computation task among nearby nodes. Regarding the scheduling of UAV, Trotta *et al.* [22] proposed a mathematical framework for optimizing the hops' number for multi-UAVs which is feasible for such average energy consumption of UAVs, device-to-device communications, and video recording. This framework is based on mixed-integer linear programming approaches, which aim to maximize the lifespan of a UAV fleet while monitoring a set of points. Whereas Song *et al.* [23] studied the problem of continuous operation for mobile targets, in which a mixed-integer linear programming is formulated to orchestrate UAVs' operations, missions, and logistics facilities as well as allocate the charging slots properly for energy replenishment. However, the main focus of the study was minimizing the computation, communication and energy costs.

Most of these models are concerned with a single-user Mobile Edge Computing (MEC) environment, whereas few works have addressed the multi-user scenarios [24], [25]. In addition, for a large scale of UAVs, the connected edge servers are limited by computational capacity, which may lead to long processing delays and thereby render these devices insufficient for real-time applications. Moreover, obtaining the optimal solution in a multi-user wireless edge cloud computing environment is a non-trivial problem [26]. Therefore, in this work, we proposed a delay optimal task offloading approach for edge-cloud computing systems in a multi-user environment. Specifically, we formulated the task offloading problem and resource allocation for the multi-user environment as an Integer Linear Programming (ILP) optimization framework with the objective of minimizing the total service time for UAVs applications. In addition, simulation-based tests are conducted to evaluate the performance of our proposed models. The main contributions of this work are summarized as follows:

- Offloading models are designed to determine the optimal computation and communication offloading decisions for UAV-enabled multi-tier edge-cloud systems, in multi-user environments.
- The offloading problem is formulated as an ILP optimization, which jointly considers task offloading and resource allocation in order to minimize the total service time of UAVs.
- Insights obtained from the ILP model are used to develop a delay-optimal task offloading algorithm that is improved system efficiency and can be used for real-time implementations.
- Simulation results proved that the proposed approach not only reduces the service time by 33.5%, 55% and 10.8% with respect to edge execution, cloud execution and task offloading execution policy in [27], respectively but can scale well for a large number of UAVs.

The rest of the paper is organized as follows. A thorough discussion of the related work is presented in Section II. In Section III, we describe our system model and the problem formulation. Afterwards, Section IV introduces the design of the proposed algorithm to derive the problem solution. Then, in Section V, the experimental setup, results, and discussions are presented. Finally, Section VI concludes the paper and discusses future work.

II. OFFLOADING APPROACHES IN EDGE-CLOUD ENVIRONMENT: STATE-OF-THE-ART

Recently, several optimization models and system architectures based on UAV-MEC have been proposed to mitigate the limitation of UAVs and address the challenges using the computation offloading method. Most of these models and approaches have been proposed for a single-tier environment [28]–[30], whereas a few studies have addressed a multi-tier environment with and without a remote cloud [31]. In this section, we provide an overview of these approaches for single and multi-tier environments with a comparison summary of the closely related work, as shown in Table 1.

A. SINGLE-TIER ENVIRONMENT

A novel sequential offloading scheme for UAV-enabled single-tier edge computing system was proposed in [28], in which the highly intensive tasks are offloaded and remotely executed at the edge server to decrease the energy overhead and the execution delay. Specifically, a sequential offloading decision game is modeled, where a drone, base station, and edge server cooperatively work together by sharing the available computational resources for optimizing the energy consumption and delay. Meanwhile, in [29], a new generic offloading scheme was proposed based on UAVs in a MEC environment. Besides, a non-cooperative theoretical game with N players is formulated as an optimization problem, in which three different policies are utilized (i.e., local execution, edge server execution, base station execution) to model the decision. Furthermore, the main goal of this study is to combine energy overhead, computation, and communication delays, while minimizing the communication cost. The simulation results demonstrated that the proposed scheme of [28] was able to minimize the response time by an average of 20.5% and 39% relative to the linear prediction and fully local computing schemes, respectively. Whereas, the simulation results demonstrated that the proposed scheme of [29] could achieve a better global utility by about 19%, 53% and 71% relative to edge server, base station and drone, respectively.

In addition, [32] and [33] proposed new energy and efficient solutions for offloading the computation tasks in UAV-enabled edge computing systems. Specifically, [32] proposed an energy-efficient computation offloading technique with a physical layer of security to protect the system from eavesdroppers, in which a set of security issues have been formulated as different problems which are then transformed into convex ones. Afterward, the optimal solutions are derived. On their part, Ouahouah *et al.* explored the mobility and the diversity of IoT across UAVs in [33]. Then, they introduced two efficient formulations using integer linear programming to minimize the energy consumption and reduce the response time, respectively.

Similar to the enumerated efforts, Valentino *et al.* [34] proposed an opportunistic offloading scheme for UAV clustering networks, in which the UAV with a high-intensity task can execute their tasks at a nearby UAV cluster. Additionally, a new shallow neural network prediction module is introduced to determine the offloading decision based on the response time. Their simulation results reported that the proposed scheme could minimize the response time by 20.5% and 39% relative to the linear prediction and fully local computing schemes, respectively.

The continuous operation of a UAV in solving problems that involve the processing of incoming video and audio streams has high energy costs, thereby reducing the degree of autonomy and lifetime of the UAV. To solve this problem, Callegaro *et al.* [35] proposed an IoT infrastructure deployed in a city to provide communication between UAVs and boundary servers, thereby transferring the task from the UAV to more powerful devices and reducing the energy needed to complete this task. In their contribution, Qu *et al.* [30] proposed a novel framework for UAV video analytics that utilizes edge computation offloading to coordinate large video datasets for intelligent processing. Subsequently, heuristic-based and reinforcement learning-based offloading policies have been presented for UAV-enabled edge computing, where minimizing the computation costs and latency are the main objectives. Finally, the results of [30] verified that the proposed scheme can improve the time and achieve better scheduling make-span by about 15% relative to the full offloading scheme.

Recently, a UAV-enabled ultra-reliable low-latency offloading problem was investigated for IoT networks in [36]. Specifically, the failure rate of UAV and tasks redundancy was formulated as a non-convex and mixed-integer optimization problem with the aim of maximizing the incoming IoT request rates. In addition, this problem was divided into two sub-problems. The first one considers long term task offloading, where the positions of UAVs are optimized. Whereas, the second sub-problem considers the decision of offloading and resource allocation, where tasks responses are optimized. Further, a new hybrid deep-learning-based offloading platform is proposed in [37], in which ground vehicles, ground base stations, and UAVs are considered and optimized, and all the mobile users can offload their intensive tasks with the goal of minimizing the energy consumption. In addition, a large-scale path-loss fuzzy c-means-based algorithm is proposed to locate the ground vehicles and UAVs. Then, the U-based particle swarm optimization-based algorithm is designed to solve the formulated problem and derive high-quality data set, which is used later by Deep Neural Network (DNN) model to make the task admission and resource allocation decision in real-time.

B. MULTI-TIER ENVIRONMENT

To perform complex tasks at UAVs, Ateya *et al.* [38] considered a traffic offloading algorithm for UAVs that is based on two methods: air traffic unloading, based on the transfer of computing tasks to neighboring UAVs, and task unloading to cloud servers from cloud edge devices connected to ground stations. The method that uploads to the edge cloud server is the main one in the system. The proposed method takes into account energy consumption and delay. A device is selected for traffic offloading and a method of unloading. The work discusses the use of a multilevel MEC system. The ground network segment consists of a second and third level, containing a mini-cloud and a micro-cloud, respectively. The air segment - the first level of traffic unloading - is represented by several closely spaced UAVs. Zero-level traffic offload for local task execution is captured by UAV sources. Meanwhile, Wang *et al.* [31] utilized the high controllability and flexibility of the UAVs to offer their computation resources for mobile users through computation offloading. Besides, a mixed-integer offloading problem is formulated to decrease the total overhead of time and energy. Further, an efficient algorithm based on Q-Learning is proposed to solve this problem and derive the offloading decision. Finally, simulation results proved that the proposed architecture in [31] reduce the time by a 5% relative to the traditional two-layer network architecture.

Recently, machine learning techniques have been exploited for addressing the complexity of distributed computing systems such as multi-UAV with the multi-tier environment, in which [39] and [27] proposed an efficient distributed framework for computing offloading. More specifically, [39] introduced a new online distributed machine learning framework for the network-aware multi-UAV-enabled systems in which the swarm networks have been integrated. Besides, federated learning is utilized to produce personalized local models. Whereas in [27], a new efficient framework is introduced for offloading the intensive tasks from UAVs to edge servers, where minimizing latency and energy cost is the main goal. Moreover, two algorithms based on trajectory schedule and code parameter design techniques were developed to solve this problem in an efficient manner. Finally, simulation results in [39] verified that the presented method in of [39] retains about 25% energy consumption's enhancement against the baseline. Further, a new efficient and distributed intelligent resource scheduling framework is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3174127, IEEE Access

proposed in [40], to reduce the sum latency of tasks and energy consumption for all users. In addition, task offloading, recourse allocation, and transmission power are jointly considered in the formulated model.

It is clear from the above review of related work, various offloading models and approaches have been introduced with different objectives for single and multi-UAV edge computing environments. However, there are still some challenges due to the variable dynamics of wireless networks, thereby leading to an increase in the time spent on the task at UAV. In addition, due to the large queues that occur on the boundary servers when processing several tasks, the total time task transfer to the boundary server and task processing time may exceed the time spent analyzing the task locally on the UAV. Besides, few studies address multi-tier with a multi-UAV edge computing system. This motivates our work in this article for proposing a delay-optimal approach for UAV-enabled multi-tier with multi-UAV edge-cloud systems Moreover, we present the optimization process that enables the optimum choice of where the task should be processed. The communication, computational, and energy aspects of the proposed process were considered. Finally, the proposed system is scalable and can support an increase in network traffic without performance degradation.

III. SYSTEM MODEL

In our work, an environment of a multi-tier edge-cloud computing system is considered, as shown in Fig. 1, where the architecture consists of three-tiers: UAV tier, edge tier and cloud tier. In the first tier, the environment has a set of $\mathcal{N} = \{1, 2, \dots, N\}$ UAVs, where each UAV has a computation task that should be processed. The edge tier consists of a set of $\mathcal{M} = \{1, 2, \dots, M\}$ small base stations[1], in which each base station can remotely provide a set of computation and storage capabilities to UAVs over the wireless channel. In addition, there is a backbone router[2] in the middle layer which links the base stations with the cloud tier via an IP core network and is also responsible for controlling and managing these base stations. Finally, in the last tier (i.e., cloud tier), there is a single cloud computing server which can provide a shared pool of virtualized servers, storage, applications, and services. Note that, cloud servers are more powerful than edge servers, but cloud suffers from transmission latency and centralization.

Similar to many previous studies in edge-cloud computing [41], [42], a quasi-static scenario is assumed in our simulation settings in which during computation offloading periods, the UAVs' number will remain unchanged, while this number may be modified across different periods. Moreover, as the computation and communication models play a key role in mobile-edge cloud computing, these models will be further described in the following subsections. Table 2 summarizes the notations that will be used in the models and simulations.

A. COMMUNICATION MODEL

In the communication model, a set of $\mathcal{K} = \{0, 1, 2, \dots, M\}$ servers is considered that can provide computational and storage capabilities, in which 0 denotes the cloud server and $(K \in [1..M])$ denotes one of the edge servers. Additionally, we have a set of $\mathcal{N} = \{1, 2, \dots, N\}$ UAVs, where every UAV i has a computationally intensive task that needs to be carefully offloaded and assigned to a server node for processing after which the output is returned. Our goal is to minimize the total service time of UAVs by allocating the computation tasks of UAVs to the appropriate server node.

In this work, we assume that all the tasks are computationally, which make them incompatible with being executed locally on the UAV devices. Therefore, the process for the task offloading is considered as follows. Firstly, the computation and communication requirements for each task are sent to the central control manager through the base stations. Then, the central control manager utilizes this information to derive the best execution place for each computation task, i.e., at the edge or cloud servers.

Let $\alpha_{i,j,k} \in \{0, 1\}$ represent the binary offloading decision of UAV task i associated with sBS j, allocated to be processed at the server k. Specifically, $(\alpha_{i,j,0} = 1)$ indicates that the UAV user i associated with sBS j chooses to execute its computation task remotely at the cloud server, while $(\alpha_{i,j,k} = 1, \forall k \in [1..M])$ indicates that the UAV i at sBS j chooses to offload its computation task remotely at the edge server node. Also, every UAV's task must be processed only once by a single edge server node (i.e. tasks splitting is not permitted) while $\sum_{k=0}^{M} \alpha_{i,j,k} = 1$; thereby, the offloading decision binary indicator is defined as follows:

$$\alpha = \begin{cases} \alpha_{i,j,0} = 1 & \forall_{i,j} \quad Cloud\ Execution \\ \alpha_{i,j,k} = 1 & \forall_{i,j,k} \quad Edge\ Execution \end{cases}$$

Subsequently, if the UAV i chooses to transmit its task to the connected edge node j, then the allocated uplink and downlink data rates for each UAV can be calculated based on the Shannon's law as follows [43]:

$$R_{i,j}^{UL} = B_j^{UL} log_2 \left(1 + \frac{p_{i,j}^T g_j^2}{\omega_j B_j^{UL}}\right) \tag{1}$$

$$R_{i,j}^{DL} = B_j^{DL} log_2 \left(1 + \frac{p_j^T g_j^2}{\omega_j B_j^{DL}}\right) \tag{2}$$

where B_j^{UL} and B_j^{DL} denote the bandwidth of uplink and downlink channels and p_j^T and $p_{i,j}^T$ indicate the transmission power of the edge server j and the corresponding UAV i. Also, g_j denotes the corresponding channel gain and ω_j indicates the noise power density for the channel.

Consequently, we assume that the uplink and downlink data rates are equally shared between UAVs, in case of simultaneously offloading via the wireless channel. In addition,

[1]This paper uses edge server node and base station interchangeably.

[2]Mobile Edge Network is managed using the architecture of Software Defined Network (SDN), in which the backbone router works as a central controller manager.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3174127, IEEE Access

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

Reference	**Objective**	**Proposed Solution**	**UAV**		**Tier Environment**		**Evaluation Methods**
			Single	**Multiple**	**Single**	**Multiple**	
[28]	Minimize Energy & Delay	A Sequential Offloading Game Scheme	✓		✓		Simulation
[29]	Minimize Communication Cost	A Game Theory Based Efficient Computation Offloading Scheme	✓		✓		Simulation
[32]	Minimize Energy	An Energy-Efficient and Secured Offloading Technique	✓		✓		Numerical Validations, Simulation
[33]	Minimize Energy and Response Time	An Efficient Offloading Mechanism		✓	✓		Simulation
[34]	Minimize Response Time	An Opportunistic Compuational Offloading scheme		✓	✓		Simulation
[30]	Minimize Computation Cost and Latency	A Novel Framework for UAV-enabled Video Analytics		✓	✓		Direct Experiment Simulation
[36]	Maximize served incoming offloading requests' rate	An UAV-enabled ultra-reliable low-latency offloading problem		✓	✓		Simulation
[35]	Minimize Delay and Energy Expense	A Framework Enabling Optimal Offloading Decisions	✓		✓		Simulation
[38]	Minimize Energy and Latency	An Energy- and Latency-Aware Hybrid Offloading Algorithm		✓		✓	Simulation
[31]	Minimize Total Overhead	A Multi-UAVs Computation Offloading Scheme		✓		✓	Simulation
[39]	Performance-Energy consumption Tradeoff	A Hierarchical Nested Personalized Federated Learning Approach		✓		✓	Simulation
[27]	Minimize Latency and Energy Cost	A Coded Distributed Computing Framework		✓		✓	Simulation
[37]	Minimize Energy Cconsumption	A Hybrid Deep-Learning-based Offloading Platform		✓	✓		Simulation
[40]	Minimize Task Latency and Energy Consumption	An Efficient and Distributed Intelligent Resource Scheduling Framework		✓		✓	Simulation
Proposed	**Minimize Total Service Time**	**A Delay-Optimal Task Offloading Approach**		✓		✓	**Simulation**

in the cloud execution scenario, the UAVs will transmit and receive the computation task through the connected base station.

In this work, an orthogonal frequency [44] is utilized to mitigate the intracellular interference among simultaneous transmissions of UAVs in the same cell [45], [46].

Furthermore, the total communication time for transmitting and receiving the computation tasks of UAVs can be computed as follows:

$$T_{i,j}^{Comm} = T_{i,j}^{UL} + T_{i,j}^{DL} + \sigma\alpha_{i,j,k} + \delta\alpha_{i,j,0} \tag{3}$$

where σ and δ indicate the task's transmission delay among two edge nodes and between the edge node and the cloud server. While $T_{i,j}^{UL}$ and $T_{i,j}^{DL}$ are denoted the time of transmission and reception to upload and download the UAV's task and its output, which can be expressed as follows:

$$T_{i,j}^{UL} = \sum_{k=0}^{M} \frac{D_{i,j}}{R_{j,i}^{UL}} \alpha_{i,j,k} \tag{4}$$

$$T_{i,j}^{DL} = \sum_{k=0}^{M} \frac{U_{i,j}}{R_{j,i}^{DL}} \alpha_{i,j,k} \tag{5}$$

Moreover, the total communication energy consumption for transmitting and receiving the computation tasks of UAVs can be computed as follows:

$$E_{i,j}^{Comm} = (T_{i,j}^{UL} P_{i,j}^{T}) + (T_{i,j}^{DL} P_{i,j}^{R}) \tag{6}$$

where $P_{i,j}^{R}$ denotes the reception power for the UAVs.

B. COMPUTATION MODEL

In the computation model, N UAVs are considered in our environment, in which every UAV has a computationally-intensive task. In addition, the computation task should be

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3174127, IEEE Access

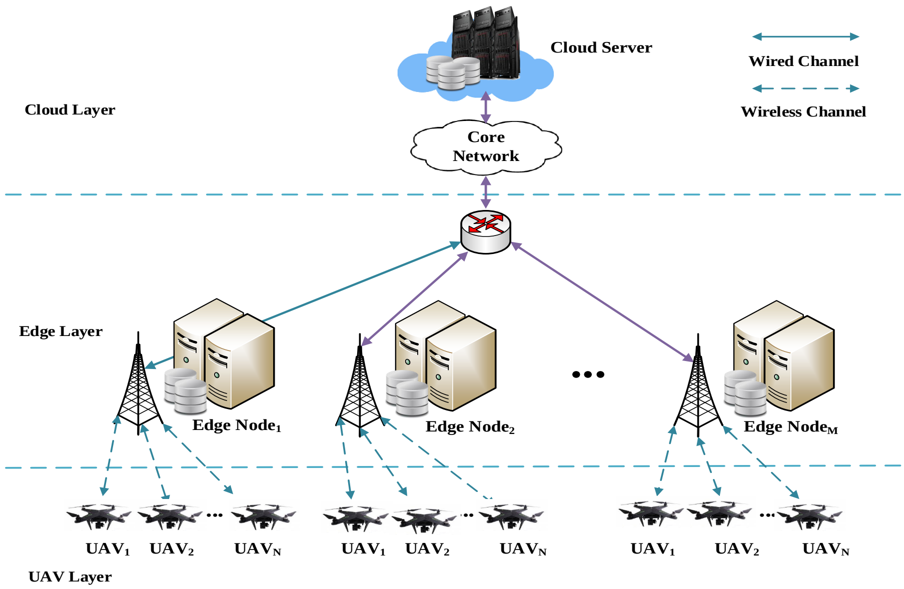

FIGURE 1: A multi-tier edge-cloud computing system.

TABLE 2: Notations summary

Notation	Description
$\mathcal{M}$	The set of base stations (BSs).
$\mathcal{K}$	Servers number.
$\mathcal{N}$	The set of UAVs.
N	Number of UAVs at each server.
j	Denotes the $j-th$ server.
i	Denotes the $i-th$ UAV.
$D_{i,j}$	Data size for computation task of UAV i associated with edge server j.
$C_{i,j}$	CPU cycles to accomplish the task of UAV i associated with edge server j.
$U_{i,j}$	Output Size for computation task of UAV i associated with edge server j.
$\alpha_{i,j,k}$	Offloading decision for computation task of UAV i.
$R_{i,j}^{UL}$	Uplink data rate for UAV i associated with edge server j.
$R_{i,j}^{DL}$	Downlink data rate for UAV i associated with edge server j.
B_j^{UL}	Uplink bandwidth at edge server j.
B_j^{DL}	Downlink bandwidth at edge server j.
$p_{i,j}$	Transmission power of UAV i associated with edge server j.
g_j	Channel gain of edge server j.
$P_{i,j}^{R}$	Reception power for UAV.
ω_j	Noise power density of the channel at edge server j.
σ	Task's transmission delay among two edge nodes.
δ	Task's transmission delay between the edge node and the cloud.
ζ	Energy consumption while the UAV is in an idle state waiting for the results.
$f_{i,j}^{e}$	CPU resource of edge server j assigned to UAV i.
$f_{i,j}^{c}$	CPU resource of cloud server assigned to UAV i associated with edge server j.
F_j	Server computational capability at edge server j.
$T_{i,j}^{UL}$	Offloading time for transferring task of UAV i associated with edge server j.
$T_{i,j}^{DL}$	Downloading time for transferring the task output of task of UAV i associated with edge server j.
$T_{i,j}^{e\_exec}$	Execution time for processing task i at edge node j.
$T_{i,j}^{c\_exec}$	Execution time for processing task i at cloud node j.
$T_{i,j}^{Comm}$	Execution time for communication.
$T_{i,j}^{Comp}$	Execution time for computation.
$E_{i,j}^{Comm}$	Energy Consumption for communication.
$E_{i,j}^{Comp}$	Execution time for computation.
$T_{i,j}^{r}$	Total Service time for processing task remotely.

transmitted and remotely executed at the server node. In this work, the tuple $\{D_{i,j}, C_{i,j}, U_{i,j}\}$ is utilized to represent the requirement for each computation task. Specifically, $D_{i,j}$ denotes the task's input data (in KB) (code and parameters), whereas $C_{i,j}$ and $U_{i,j}$ denote the total number of CPU cycles (in Cycle/Byte), which is required to accomplish the computation task and the output size (result) for the computation task of the UAV i, which is connected to base station j. Given the application's nature, the values of $D_{i,j}$, $C_{i,j}$ and $U_{i,j}$ can be determined. Furthermore, this work is guided by [47] and [48], where the program profiler is used to obtain these values.

In practice, the management between base stations can be achieved by utilizing the Software-Defined Network (SDN) controller technology and the standardized OpenFlow protocol on the backbone router in which SDN has a global view of the network and provides a scalable and flexible structure that is capable of making more efficient and precise network management [49]. As mentioned in section III-A, the computation tasks' requirements of UAVs are transmitted to the central control manager via the connected base stations. After that, the central control manager-based on SDN technology can determine the best offloading policy for allocating the tasks to either the edge or cloud servers. Therefore, there are two main approaches for executing the computation tasks: 1) edge server approach and 2) cloud server approach, which are described in further details in the following subsections.

1) Edge Server Approach

With the edge server approach, the computation tasks will be offloaded and executed remotely on one of the available edge nodes. Therefore, the processing time for executing the computation task of UAV i, associated with BS j, remotely at the edge server node k, can be calculated as follows:

$$T_{i,j}^{e\_exec} = \frac{C_{i,j}}{f_{i,j}^{e}} \tag{7}$$

where $f^{e}_{i,j}$ denotes the edge node's computation capability that is allocated to UAV i.

In this work, the edge servers' capabilities are supposed to be unified and equally shared between UAVs in the case of simultaneous transmission. In addition, during the offloading period, we note that the computational capacity of the edge servers j, (denoted as F_J), should be limited to the following constraint:

$$\sum_{j=1}^{M}\sum_{i=1}^{N}\sum_{k=1}^{M}\alpha_{i,j,k}f^{e}_{i,j} \leq F_J \tag{8}$$

2) Cloud Server Approach

On the other hand, with the cloud server approach, the computation task will be transmitted and processed remotely on the cloud server, and returning the output through the connected base station. Therefore, the processing time for executing the computation task of UAV i, associated with BS j, remotely at the cloud server can be calculated as follows:

$$T^{c\_exec}_{i,j} = \frac{C_{i,j}}{f^{c}_{i,j}} \tag{9}$$

where $f^{c}_{j,i}$ indicates the computational capability of the cloud server, which is assigned to UAV i. In general, the computational capabilities at a cloud server is more powerful than edge servers $f^{e}_{i,j} < f^{c}_{i,j}$.

Consequently, in Eqs.(7) and (9), the total time for processing the computation tasks of UAVs remotely can be calculated as follows:

$$T^{Comp}_{i,j} = T^{c\_exec}_{i,j}\alpha_{i,j,0} + \sum_{k=1}^{M} T^{e\_exec}_{i,j}\alpha_{i,j,k} \tag{10}$$

Moreover, the total energy consumption for completing the UAVs' tasks remotely can be computed as follows:

$$E^{Comp}_{i,j} = \zeta T^{Comp}_{i,j} \tag{11}$$

where ζ indicates the energy consumption, while the UAV is in an idle state waiting for the output.

Finally, in the previous subsections (i.e., communication and computation models), the total service time for executing the computation task of UAV i associated with BS j remotely can be calculated as:

$$T_{i,j} = T^{Comm}_{i,j} + T^{Comp}_{i,j} \tag{12}$$

C. PROBLEM FORMULATION

In this section we introduce the ILP formulation for the delay-optimal offloading problem in the multi-tier edge-cloud computing environment. Moreover, the computation offloading problem is formulated as a single objective optimization problem, which aims to minimize the total service time of UAVs:

$$\begin{aligned}
\min_{\alpha} \quad & \Big[\sum_{j=1}^{M}\sum_{i=1}^{N} T_{i,j}\Big] \\
& \sum_{j=1}^{M}\sum_{i=1}^{N}\sum_{k=1}^{M}\alpha_{i,j,k}f^{e}_{i,j} \leq F_J \quad \forall_{i,j} \quad C1 \\
& \sum_{k=0}^{M}\alpha_{i,j,k} = 1 \quad \forall_{i,j} \quad C2 \\
& \alpha_{i,j,k} \in \{0,1\}, \quad \forall_{i,j} \quad C3
\end{aligned} \tag{13}$$

Where C1 is a capacity constraint which ensures that the computational capabilities of BSs during the offloading period is not violated. While C2 and C3 ensure that each UAV's task is computed only once and the offloading decision variables for the tasks take on discrete binary values, respectively.

The problem's solution can be derived by obtaining the optimal task offloading's values, α^*. In addition, this problem is an integer linear problem, where the objective function of this problem and its constraints are linear. Moreover, this problem is NP-hard and has a non-convex feasible set [50], [51]. Consequently, the binary relaxation approach can be utilized to transform this problem into a convex one, where the α variables are relaxed into real variables, and the new formulation is presented in equation (14) [52], [53]. Furthermore, branch and bound methods can be used to derive the offloading decision in an efficient manner [54], [55].

$$\begin{aligned}
\min_{\alpha} \quad & \Big[\sum_{j=1}^{M}\sum_{i=1}^{N} T_{i,j}\Big] \\
& \sum_{j=1}^{M}\sum_{i=1}^{N}\sum_{k=1}^{M}\alpha_{i,j,k}f^{e}_{i,j} \leq F_J \quad \forall_{i,j} \quad C1 \\
& \sum_{k=0}^{M}\alpha_{i,j,k} = 1 \quad \forall_{i,j} \quad C2 \\
& \alpha_{i,j,k} \in [0,1], \quad \forall_{i,j} \quad C3
\end{aligned} \tag{14}$$

IV. DELAY-OPTIMAL TASK OFFLOADING ALGORITHM

A delay-optimal task offloading algorithm is introduced in this section, in which the processes for deriving the best offloading decision for a multi-UAV and multi-tier-enabled edge-cloud computing system are presented.

To derive the best offloading decision for each computation task, there is a set of parameters that should be obtained from the edge servers and UAVs. Consequently, the proposed optimization model in Eq. (13) used the values of these parameters to derive the offloading solution for each task. Subsequently, the proposed algorithm (i.e., Algorithm 1) describes this process as follows.

First, we initialized the offloading decision for all computation tasks with $\alpha_{i,j,k} = 1$, which implies the task will be assigned to be executed at the connected edge execution. Then, the control manager using SDN technology can iterate over the edge servers and obtains the available computation and storage capabilities as well as the available bandwidth. In addition, each server iterates over each UAV and gathers their computation task's requirements $\{D_{i,j}, C_{i,j}, U_{i,j}, p_i^T, p_j^T\}$ which are then transmitted to the control manager. Subsequently, based on this information, the control manager solves the problem in Eq. (13) and determines the best offloading decision for each computation task α^* (i.e., computation hosted on the edge and/or cloud servers). Finally, each UAV receives the optimal offloading decision, thereby reducing the service time of the entire system.

Algorithm 1 describes the processes involved in the delay-optimal task offloading scheme. Moreover, $\mathcal{O}(MN)$ is the time complexity, in which M and N indicate the edge servers (ESs) and UAVs numbers, respectively.

Algorithm 1 Delay-Optimal Task Offloading Decision

1: **Initialization**: Each computation task i sets the decision with $\alpha_{i,j,k} = 1$
2: **for all** ESs j and at time slot t **do**
3: **Send** the available edge server capabilities and the bandwidth to control manager.
4: **for all** UAV i **do**
5: **Upload** the task requirements, $\{D_{i,j}, C_{i,j}, U_{i,j}, p_i^T, p_j^T\}$, to control manager.
6: **end for**
7: **end for**
8: **Solve** the optimization problem in Eq. (13) and derive the optimal offloading decision values α^* which reduces the total service time.
9: **Send** the optimal offloading decision values to each UAV.

V. EXPERIMENTAL SETUP, RESULTS AND DISCUSSIONS

A simulation-based experiment is presented in this section to critically evaluate the performance of the proposed models. First, the environment setup and resources used are highlighted. Afterward, a comprehensive discussion of the obtained results is provided.

A. EXPERIMENTAL SETUP

The simulation and experiment are performed on a MATLAB-based simulator using a PC running Windows 10 Operating System, with an Intel® Core(TM) i7-4770 CPU - 3.4 GHz and 16 GB memory and 1TB SSD of storage has been used. We consider a mobile network with 5 BSs, in which each BS has a different number of UAVs that are distributed across BSs. The CPU capabilities of edge and cloud nodes are set to 25 GHz and 50 GHz. Moreover, each UAV is associated with a demand that involves a computationally intensive task. The size of input and output data for each task are distributed randomly between the range of [5 MB to 10 MB and 0 MB to 2 MB], respectively. On the other hand, the computation requirements for each task are set to 1900 Cycles/Byte. The transmission and reception power are set to 100 mW and 50 mW, respectively. Furthermore, the channel bandwidth is set to 20 MHz. Moreover, GAMS is considered one of the mathematical programming languages especially suited for optimization problems. Therefore, in our experiment, we programmed the proposed model in GAMS language and used MATLAB as an interface to address the values of parameters to solve the optimization model and derive the best offloading decision for each computation task [56]. The simulation experiments are executed 50 times and the average value is calculated. The other simulation parameters for communication and computation are summarized in Table 3.

TABLE 3: Simulation parameters

Parameter	Value
Number of BSs	5
Edge capabilities	25 GHz
Cloud capabilities	50 GHz
Input data size	$(5, 10)$ MB Uniform-Distribution
Output data size	$(0, 2)$ MB Uniform-Distribution
CPU cycles to accomplish task	1900 Cycle/Byte
Transmission power	100 mW
Reception power	50 mW
System bandwidth	20 MHz

B. EXPERIMENTAL RESULTS AND DISCUSSIONS

This subsection assesses the performance of the proposed model's, in which five different strategies are evaluated which are:

- **Edge Execution:** In this strategy, all the tasks will be offloaded and then remotely processed at the connected edge node.
- **Cloud Execution:** All the tasks will be offloaded in this strategy and then remotely processed at the cloud node.
- **Random Execution:** Within this strategy, all the tasks will be offloaded and then remotely processed randomly at edge or cloud node.
- **Task Offloading (TO) Execution:** In this execution policy, all the tasks will be offloaded and then remotely processed at the appropriate node, based on the proposed model in [27].
- **Model Execution:** Here, all the tasks will be offloaded and then remotely processed at the appropriate node, based on our proposed models.

The service time for executing the UAVs' tasks remotely using the five different strategies versus different numbers of UAVs is presented in Fig. 2. It is shown from the figure that the service time for the strategies is nearly equal for a small number of UAVs (i.e., less than 10). However, the delay for the cloud, edge, and random execution strategies rapidly increases as the number of UAVs increases (i.e., more than 10). Moreover, using the approach in [27] and the proposed model, the service time gradually increases as the number of UAVs increases in comparison to the other approaches and the proposed model's performance was superior throughout. The reason behind the poor performance of the other approaches is that with the increase in the number of UAVs, some BSs are overloaded while others are underloaded. In contrast, the proposed model can smartly select the appropriate place for assigning and executing the tasks, whereas the model in [27], does not consider the cloud server layer in their execution.

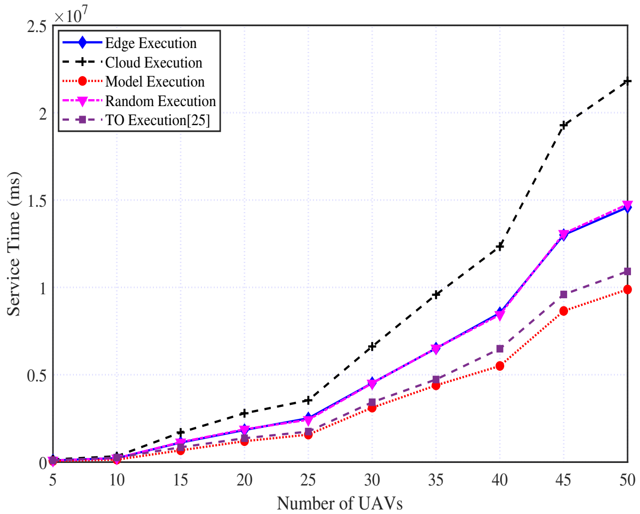

FIGURE 2: Service time versus number of UAVs.

Fig. 3 depicts the energy consumption of processing tasks against different numbers of UAVs. It is observed from the figure that the energy consumption linearly increases as the number of UAVs increases. In addition, it is also noted that, while our proposed model outperforms others, the energy gap between the five strategies is not substantial. This is attributed to the fact that most of the energy is consumed during the transmission of data, where UAVs compete for shared and limited communication resources to offload the computation tasks' requirements to the connected BSs in all strategies.

Furthermore, the total service time and energy consumption for processing the computation tasks against different input data sizes are respectively depicted in Fig 4 and Fig. 5. The four curves denote the time and energy for the four strategies mentioned above. It is seen from the figure that the time and energy consumption are significantly increase as the number of UAVs increase. Moreover, the proposed model has the least consumption of time and energy compared with the other three policies. This can be elucidated as follows. As the data size increases, the communication time increases, which is reflected in the service time and energy consumption. However, the proposed model can be adapted to execute the computation tasks at an optimum location, at either the edge or the cloud node.

Similarly, Figure 6 demonstrates the service time of executing the computation tasks under different capabilities of

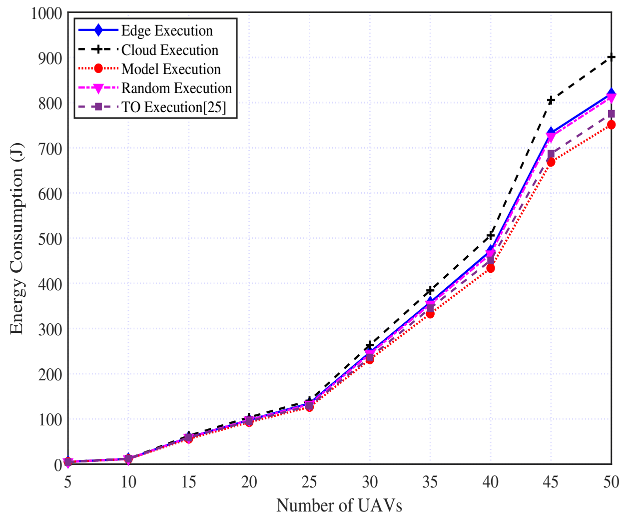

FIGURE 3: Energy consumption versus number of UAVs.

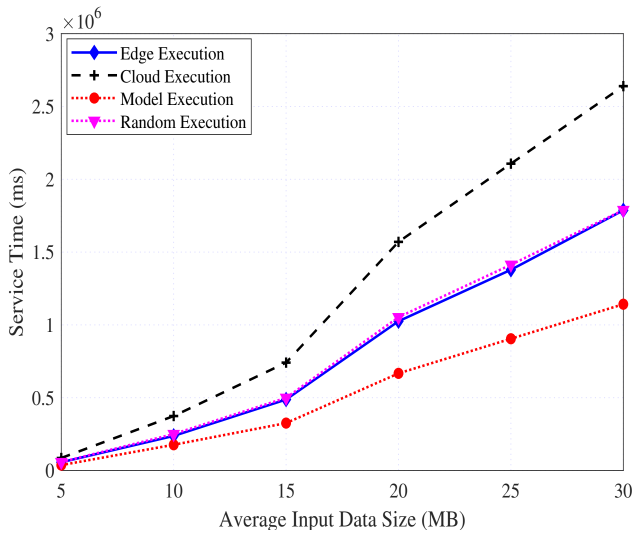

FIGURE 4: Service time versus input data size.

edge servers. It is observed from the figure that the cloud execution policy is not impacted by the edge servers' capabilities, while the service time for edge and the proposed model executions gradually decreases as the capabilities of edge servers increase. Moreover, the proposed model's performance is superior compared to the other solutions. This is due to the shorter service time for computation as the UAVs are allocated more resources at edge servers, whereas the cloud policy does not use the edge servers' capabilities.

Finally, the total service time of the four strategies under five different types of applications (shown in Table 4) is demonstrated in Fig. 7. As can be seen, the service time of cloud execution for all types of applications is longer than the other strategies, while the proposed achieved the shortest service time. This is due to the long latency required for cloud execution because of its geographical distance from the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3174127, IEEE Access

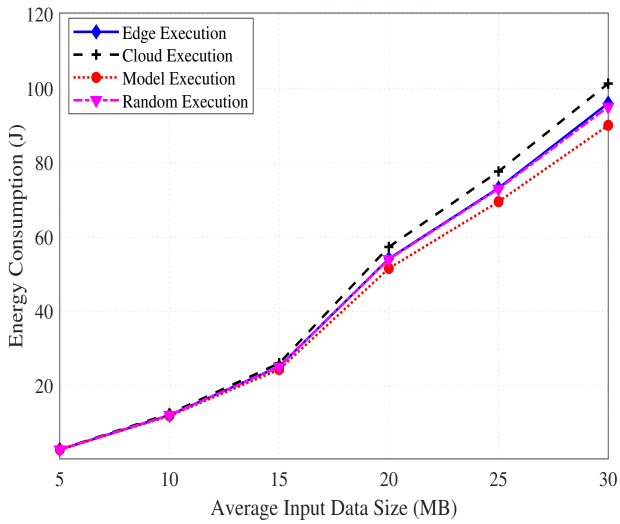

FIGURE 5: Energy consumption versus input data size.

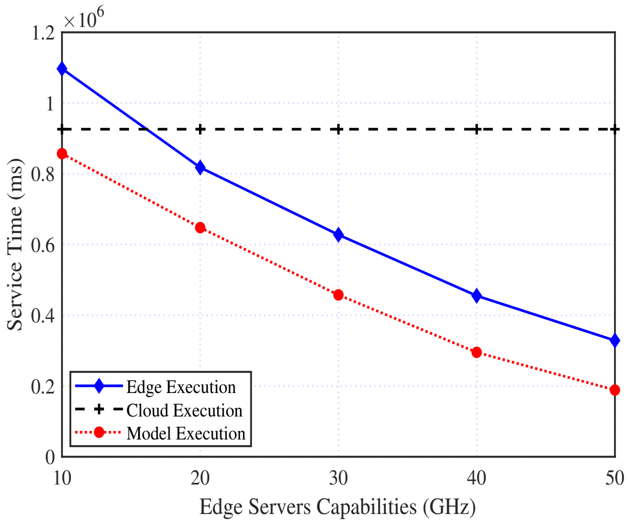

FIGURE 6: Service time versus different capabilities of edge servers.

user. Whereas, the proposed model can select the appropriate location based on the environment settings.

TABLE 4: Computational intensity of different applications [57]

Application	Label	CPU Cycle/Byte
Sustainable Agriculture	A	500
Automatic Number Plate Reading	B	960
Fire Detection	C	1900
Traffic Management	D	5900
Video Surveillance	E	12000

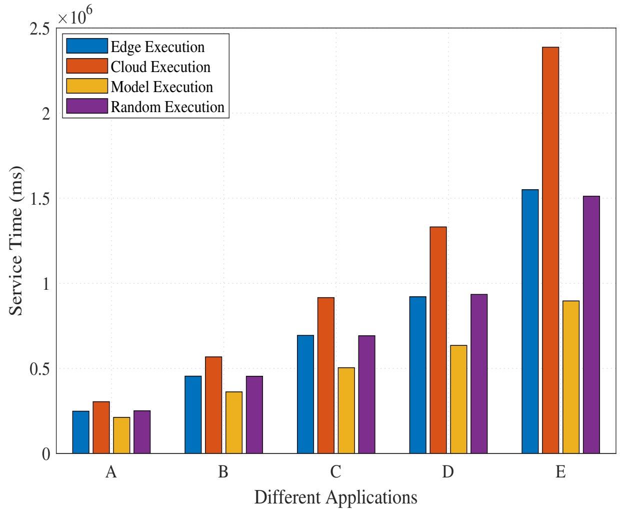

FIGURE 7: Service time versus different applications type.

C. DISCUSSION

This subsection discusses the main contributions of the proposed model, we report the difference in comparison to the other schemes reported in Section II and we discuss the performance enhancement offered by our model.

As listed in Table 1, most of the reported computation offloading models only focus on single tier architectures with multiple or single UAVs. In contrast, the work in this study addresses a multi-tier edge cloud computing environment with multiple UAVs. In addition, we present the optimization framework that enables the optimum choice of where the task should be processed, in which communication, computational, and energy aspects of the proposed process were considered. Finally, simulation results demonstrated that the proposed approach not only reduces the service time by 33.5%, 55% and 10.8% relative to the edge execution, cloud execution and task offloading policy in [27], respectively but can scale well for a large number of UAVs.

VI. CONCLUSION AND FUTURE WORKS

In this paper, a delay-optimal task offloading approach has been proposed for a multi-tier edge-cloud computing system in a multi-user environment, in which the execution of UAVs tasks is efficiently distributed across edge and cloud server nodes to address the load level between edge nodes. In addition, we formulated the task offloading problem as an Integer Linear Programming (ILP) model with the objective of minimizing the total service time of UAVs. Also, a task-offloading algorithm of improved efficiency was designed based on the insights obtained from the ILP model. Finally, numerical and simulation results showed that the proposed approach could reduce the service time by 33.5%, 55% and 10.8% for edge execution, cloud execution and task offloading execution policy in [27], respectively. In addition, the proposed approach can scale well as the number of UAVs or data size increases.

In ongoing and future work, deep learning approaches can be utilized to address the complexity of edge-cloud computing systems, where a neural network can reduce the multiplication operations in model solving and speed up the convergence process.

ACKNOWLEDGMENT

The authors extend their appreciation to the Deputyship for Research and Innovation, Ministry of Education in Saudi Arabia for funding this research work the project number (442/204). Also, the authors would like to extend their appreciation to Taibah University for its supervision support. All data are provided in full in the results section of this paper.

REFERENCES

[1] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, and E. K. Markakis, "A survey on the internet of things (IoT) forensics: challenges, approaches, and open issues," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1191–1221, 2020.

[2] S. Pasricha, R. Ayoub, M. Kishinevsky, S. K. Mandal, and U. Y. Ogras, "A survey on energy management for mobile and IoT devices," *IEEE Design & Test*, vol. 37, no. 5, pp. 7–24, 2020.

[3] M. Khayyat, A. Alshahrani, S. Alharbi, I. Elgendy, A. Paramonov, and A. Koucheryavy, "Multilevel service-provisioning-based autonomous vehicle applications," *Sustainability*, vol. 12, no. 6, p. 2497, 2020.

[4] J. Almutairi and M. Aldossary, "Modeling and analyzing offloading strategies of IoT applications over edge computing and joint clouds," *Symmetry*, vol. 13, no. 3, p. 402, 2021.

[5] H. A. Alharbi and M. Aldossary, "Energy-efficient Edge-Fog-Cloud architecture for IoT-Based smart agriculture environment," *IEEE Access*, vol. 9, pp. 110 480–110 492, 2021.

[6] Z. Wu, B. Li, Z. Fei, Z. Zheng, and Z. Han, "Energy-efficient robust computation offloading for Fog-IoT systems," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4417–4425, 2020.

[7] D. Xu, Q. Li, and H. Zhu, "Energy-saving computation offloading by joint data compression and resource allocation for mobile-edge computing," *IEEE Communications Letters*, vol. 23, no. 4, pp. 704–707, 2019.

[8] A. S. AlAhmad, H. Kahtan, Y. I. Alzoubi, O. Ali, and A. Jaradat, "Mobile cloud computing models security issues: A systematic review," *Journal of Network and Computer Applications*, p. 103152, 2021.

[9] I. A. Elgendy, W.-Z. Zhang, C.-Y. Liu, and C.-H. Hsu, "An efficient and secured framework for mobile cloud computing," *IEEE Transactions on Cloud Computing*, vol. 9, no. 1, pp. 79–87, 2018.

[10] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE pervasive Computing*, vol. 8, no. 4, pp. 14–23, 2009.

[11] Y. Siriwardhana, P. Porambage, M. Liyanage, and M. Ylianttila, "A survey on mobile augmented reality with 5G mobile edge computing: Architectures, applications, and technical aspects," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1160–1192, 2021.

[12] M. Abrar, U. Ajmal, Z. M. Almohaimeed, X. Gui, R. Akram, and R. Masroor, "Energy efficient UAV-Enabled mobile edge computing for IoT devices: A review," *IEEE Access*, 2021.

[13] C. Yan, L. Fu, J. Zhang, and J. Wang, "A comprehensive survey on UAV communication channel modeling," *IEEE Access*, vol. 7, pp. 107 769–107 792, 2019.

[14] J. Wang, C. Jiang, Z. Han, Y. Ren, R. G. Maunder, and L. Hanzo, "Taking drones to the next level: Cooperative distributed unmanned-aerial-vehicular networks for small and mini drones," *Ieee vehIcular technology magazIne*, vol. 12, no. 3, pp. 73–82, 2017.

[15] M. Aldossary, "A hybrid approach for performance and energy-based cost prediction in clouds," *CMC-COMPUTERS MATERIALS & CONTINUA*, vol. 68, no. 3, pp. 3531–3562, 2021.

[16] E. Yaacoub and M.-S. Alouini, "A key 6G challenge and opportunityâĂŤconnecting the base of the pyramid: A survey on rural connectivity," *Proceedings of the IEEE*, vol. 108, no. 4, pp. 533–582, 2020.

[17] F. Jiang, K. Wang, L. Dong, C. Pan, W. Xu, and K. Yang, "Ai driven heterogeneous mec system with uav assistance for dynamic environment: Challenges and solutions," *IEEE Network*, vol. 35, no. 1, pp. 400–408, 2020.

[18] S. Hamdan, M. Ayyash, and S. Almajali, "Edge-computing architectures for internet of things applications: A survey," *Sensors*, vol. 20, no. 22, p. 6441, 2020.

[19] J. Zhang, L. Zhou, Q. Tang, E. C.-H. Ngai, X. Hu, H. Zhao, and J. Wei, "Stochastic computation offloading and trajectory scheduling for UAV-assisted mobile edge computing," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3688–3699, 2018.

[20] M.-A. Messous, H. Sedjelmaci, N. Houari, and S.-M. Senouci, "Computation offloading game for an UAV network in mobile edge computing," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6.

[21] A. Alioua, H.-e. Djeghri, M. E. T. Cherif, S.-M. Senouci, and H. Sedjelmaci, "UAVs for traffic monitoring: A sequential game-based computation offloading/sharing approach," *Computer Networks*, vol. 177, p. 107273, 2020.

[22] A. Trotta, F. D. Andreagiovanni, M. Di Felice, E. Natalizio, and K. R. Chowdhury, "When UAVs ride a bus: Towards energy-efficient city-scale video surveillance," in *Ieee infocom 2018-ieee conference on computer communications*. IEEE, 2018, pp. 1043–1051.

[23] B. D. Song, J. Kim, J. Kim, H. Park, J. R. Morrison, and D. H. Shim, "Persistent UAV service: An improved scheduling formulation and prototypes of system components," *Journal of Intelligent & Robotic Systems*, vol. 74, no. 1, pp. 221–232, 2014.

[24] A. Shakarami, M. Ghobaei-Arani, M. Masdari, and M. Hosseinzadeh, "A survey on the computation offloading approaches in mobile edge/cloud computing environment: a stochastic-based perspective," *Journal of Grid Computing*, vol. 18, no. 4, pp. 639–671, 2020.

[25] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, "A tutorial on UAVs for wireless networks: Applications, challenges, and open problems," *IEEE communications surveys & tutorials*, vol. 21, no. 3, pp. 2334–2360, 2019.

[26] R. Chen, L. Cui, M. Wang, Y. Zhang, K. Yao, Y. Yang, and C. Yao, "Joint computation offloading, channel access and scheduling optimization in UAV swarms: A game-theoretic learning approach," *IEEE Open Journal of the Computer Society*, vol. 2, pp. 308–320, 2021.

[27] Y. Guo, S. Gu, Q. Zhang, N. Zhang, and W. Xiang, "A coded distributed computing framework for task offloading from Multi-UAV to edge servers," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2021, pp. 1–6.

[28] M.-A. Messous, A. Arfaoui, A. Alioua, and S.-M. Senouci, "A sequential game approach for computation-offloading in an UAV network," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–7.

[29] M.-A. Messous, S.-M. Senouci, H. Sedjelmaci, and S. Cherkaoui, "A game theory based efficient computation offloading in an UAV network," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4964–4974, 2019.

[30] C. Qu, P. Calyam, J. Yu, A. Vandanapu, O. Opeoluwa, K. Gao, S. Wang, R. Chastain, and K. Palaniappan, "Dronecoconet: Learning-based edge computation offloading and control networking for drone video analytics," *Future Generation Computer Systems*, vol. 125, pp. 247–262, 2021.

[31] M. Wang, S. Shi, S. Gu, X. Gu, and X. Qin, "Q-learning based computation offloading for multi-UAV-enabled cloud-edge computing networks," *IET Communications*, vol. 14, no. 15, pp. 2481–2490, 2020.

[32] T. Bai, J. Wang, Y. Ren, and L. Hanzo, "Energy-efficient computation offloading for secure UAV-edge-computing systems," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 6074–6087, 2019.

[33] S. Ouahouah, T. Taleb, J. Song, and C. Benzaid, "Efficient offloading mechanism for UAVs-based value added services," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6.

[34] R. Valentino, W.-S. Jung, and Y.-B. Ko, "A design and simulation of the opportunistic computation offloading with learning-based prediction for unmanned aerial vehicle (UAV) clustering networks," *Sensors*, vol. 18, no. 11, p. 3751, 2018.

[35] D. Callegaro and M. Levorato, "Optimal computation offloading in edge-assisted UAV systems," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–6.

[36] E. El Haber, H. A. Alameddine, C. Assi, and S. Sharafeddine, "UAV-aided ultra-reliable low-latency computation offloading in future IoT networks," *IEEE Transactions on Communications*, 2021.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3174127, IEEE Access

[37] F. Jiang, K. Wang, L. Dong, C. Pan, W. Xu, and K. Yang, "Deep-learning-based joint resource scheduling algorithms for hybrid mec networks," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6252–6265, 2019.

[38] A. A. A. Ateya, A. Muthanna, R. Kirichek, M. Hammoudeh, and A. Koucheryavy, "Energy-and latency-aware hybrid offloading algorithm for UAVs," *IEEE Access*, vol. 7, pp. 37 587–37 600, 2019.

[39] S. Wang, S. Hosseinalipour, M. Gorlatova, C. G. Brinton, and M. Chiang, "UAV-assisted online machine learning over multi-tiered networks: A hierarchical nested personalized federated learning approach," *arXiv preprint arXiv:2106.15734*, 2021.

[40] F. Jiang, L. Dong, K. Wang, K. Yang, and C. Pan, "Distributed resource scheduling for large-scale mec systems: A multi-agent ensemble deep reinforcement learning with imitation acceleration," *IEEE Internet of Things Journal*, 2021.

[41] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 974–983, April 2015.

[42] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2015.

[43] L. Lundheim, "On shannon and" shannon's formula"," *Telektronikk*, vol. 98, no. 1, pp. 20–29, 2002.

[44] S. Liu and T. Yang, "Delay aware scheduling in UAV-enabled OFDMA mobile edge computing system," *IET Communications*, vol. 14, no. 18, pp. 3203–3211, 2020.

[45] S. Deb and P. Monogioudis, "Learning-based uplink interference management in 4G LTE cellular systems," *IEEE/ACM Transactions on Networking (TON)*, vol. 23, no. 2, pp. 398–411, 2015.

[46] I. A. Elgendy, W. Zhang, Y.-C. Tian, and K. Li, "Resource allocation and computation offloading with data security for mobile edge computing," *Future Generation Computer Systems*, vol. 100, pp. 531–541, 2019.

[47] W. Zhang, Y. Wen, and D. O. Wu, "Collaborative task execution in mobile cloud computing under a stochastic wireless channel," *IEEE Transactions on Wireless Communications*, vol. 14, no. 1, pp. 81–93, 2014.

[48] X. Lyu and H. Tian, "Adaptive receding horizon offloading strategy under dynamic environment," *IEEE Communications Letters*, vol. 20, no. 5, pp. 878–881, 2016.

[49] L. E. Li, Z. M. Mao, and J. Rexford, "Toward software-defined cellular networks," in *2012 European workshop on software defined networking*. IEEE, 2012, pp. 7–12.

[50] L. A. Wolsey and G. L. Nemhauser, *Integer and combinatorial optimization*. John Wiley & Sons, 1999, vol. 55.

[51] L. Chiaraviglio, F. DâĂŹAndreagiovanni, W. Liu, J. A. Gutierrez, N. Blefari-Melazzi, K.-K. R. Choo, and M.-S. Alouini, "Multi-area throughput and energy optimization of uav-aided cellular networks powered by solar panels and grid," *IEEE Transactions on Mobile Computing*, vol. 20, no. 7, pp. 2427–2444, 2020.

[52] M. Anthony, E. Boros, Y. Crama, and A. Gruber, "Quadratic reformulations of nonlinear binary optimization problems," *Mathematical Programming*, vol. 162, no. 1, pp. 115–144, 2017.

[53] I. A. Elgendy, W.-Z. Zhang, Y. Zeng, H. He, Y.-C. Tian, and Y. Yang, "Efficient and secure multi-user multi-task computation offloading for mobile-edge computing in mobile IoT networks," *IEEE Transactions on Network and Service Management*, vol. 17, no. 4, pp. 2410–2422, 2020.

[54] Y. Hao, M. Chen, L. Hu, M. S. Hossain, and A. Ghoneim, "Energy efficient task caching and offloading for mobile edge computing," *IEEE Access*, vol. 6, pp. 11 365–11 373, 2018.

[55] W.-Z. Zhang, I. A. Elgendy, M. Hammad, A. M. Iliyasu, X. Du, M. Guizani, and A. A. Abd El-Latif, "Secure and optimized load balancing for multi-tier IoT and edge-cloud computing systems," *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 8119 – 8132, 2020.

[56] M. Ferris, S. Dirkse, and R. Jain, "MATLAB and GAMS: interfacing optimization and visualization software. computer sciences department, university of wisconsin at madison, madison," 2010.

[57] N. Dilshad, J. Hwang, J. Song, and N. Sung, "Applications and challenges in video surveillance via drone: A brief survey," in *2020 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2020, pp. 728–732.

JABER ALMUTAIRI received his BSc degree in Computer Science from Taibah University, Medina, Saudi Arabia in 2012 and his MSc degree in Advanced Computer Science (Cloud Computing) and PhD from University of Leeds, Leeds, UK in 2016 and 2020. His research interests include resource management and simulation in the edge computing environment and applications of the internet of things. He is currently Assistant professor with the Department of Computer Science, College of Computer Science and Engineering, Taibah University, Madinah, Saudi Arabia.

MOHAMMAD ALDOSSARY received his B.Sc. degree in Computer (Hons.) from King Saud University, Riyadh, Saudi Arabia in 2009 and the M.Sc. degree in Computer Science from Southern Polytechnic State University, Georgia, USA in 2013. He was awarded a Ph.D. degree in Computer Science from the University of Leeds, UK, in 2019. His main research areas focus on Distributed Systems, including Cloud Computing, Fog Computing, Edge Computing, Internet of Things, System Architectures, Resource Management, and Energy Efficiency. Dr Aldossary is currently an assistant professor in the Computer Science department at Prince Sattam bin Abdulaziz University, Saudi Arabia.

HATEM A. ALHARBI received the B.Sc. degree in Computer Engineering (Hons.) from Umm Alqura University, Makkah, Saudi Arabia in 2012, the M.Sc. degree in Digital communication networks (with distinction) in 2015, and the PhD degree in communication networks in 2020, from the University of Leeds, Leeds, UK. Dr. Alharbi is currently an assistant professor in the Computer Engineering Department in the School of Computer Science and Engineering, University of Taibah, Saudi Arabia. His research interests are in energy efficient fog and cloud networks and network economics.

BARZAN A. YOSUF received the BEng (Hons) degree in computer systems engineering, in 2012 and the MSc degree in embedded systems engineering in 2013, from the University of Huddersfield, Huddersfield, UK, and the Ph.D. degree with the School of Electronic and Electrical Engineering, in energy efficient distributed processing for IoT from the University of Leeds, UK, in 2019. His current research interests include IoT, fog and cloud computing energy optimization. He is currently a member of the IEEE Energy Efficient ICT working group, which focuses on developing improved energy efficient standards in areas such as big data processing over core networks, network virtualization and VM placement in the presence of access fog and core clouds.

JAAFAR M. H. ELMIRGHANI (Senior Member, IEEE) received the Ph.D. degree in the synchronization of optical systems and optical receiver design from the University of Huddersfield, U.K., in 1994, and the D.Sc. degree in communication systems and networks from the University of Leeds, U.K., in 2014. From 2000 to 2007, he was the Chair in optical communications with the University of Wales Swansea. He is currently the Director of the School of Electronic and Electrical Engineering, Institute of Communication and Power Networks, University of Leeds. He joined Leeds, in 2007. Prof. Elmirghani is Fellow of the Institute of Engineering and Technology (FIET), Fellow of the Institute of Physics (FIoP), and Director of the Institute of Communication and Power Networks, Leeds. He has provided outstanding leadership in a number of large research projects and was PI of the £5.9m EPSRC Intelligent Energy Aware Networks (INTERNET) Programme Grant, 2010-2016. He is Co-Chair of the IEEE Sustainable ICT initiative, a pan IEEE Societies initiative responsible for Green ICT activities across IEEE. He was awarded the IEEE Comsoc 2005 Hal Sobol award, 2 IEEE Comsoc outstanding service awards (2009, 2015), the 2015 GreenTouch 1000x award, IET Optoelectronics 2016 Premium Award and shared the 2016 Edison Award in the collective disruption category with a team of 6 from GreenTouch for joint work on the GreenMeter. He was awarded the IEEE Comsoc outstanding technical achievement award, 2020 for contributions to energy efficient communications. His work led to 5 IEEE standards with a focus on network virtualisation and energy efficiency, where he currently heads the work group responsible for IEEE P1925.1, IEEE P1926.1, IEEE P1927.1, IEEE P1928.1 and IEEE P1929.1, this resulting in significant impact through industrial and academic uptake. He is PI of the EPSRC £6.6m Terabit Bidirectional Multi-user Optical Wireless System (TOWS) for 6G LiFi, 2019-2024. He was an IEEE Comsoc Distinguished Lecturer 2013-2016. He is an Editor of IEEE Communications Magazine, IEEE Journal of Lightwave Technology, IET Optoelectronics and is Area Editor for the IEEE Journal on Selected Areas in Communications Series on Machine Learning for Communications. He has published over 550 technical papers and has research interests in communication systems and networks energy efficiency, in wireless systems and optical networks.