Received August 17, 2018, accepted September 22, 2018, date of publication September 28, 2018, date of current version October 25, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2872722

INVITED PAPER

Energy-Efficient Deployment of Edge Dataenters for Mobile Clouds in Sustainable IoT

SAMBIT KUMAR MISHRA[1], DEEPAK PUTHAL[2], BIBHUDATTA SAHOO[3], (Member, IEEE), SURAJ SHARMA[4], ZHI XUE[5], AND ALBERT Y. ZOMAYA[6], (Fellow, IEEE)

[1]Department of Computer Science and Engineering, Oriental University, Indore 453555, India
[2]Faculty of Engineering and Information Technologies, University of Technology Sydney, Ultimo, NSW 2007, Australia
[3]Department of Computer Science and Engineering, National Institute of Technology at Rourkela, Rourkela 769008, India
[4]Department of Computer Science and Engineering, International Institute of Information Technology at Bhubaneswar, Bhubaneswar 751003, India
[5]School of Cyber Security, Shanghai Jiao Tong University, Shanghai 200000, China
[6]School of Information Technologies, The University of Sydney, Sydney, NSW 2006, Australia

Corresponding author: Deepak Puthal (deepak.puthal@gmail.com)

ABSTRACT Achieving quick responses with limited energy consumption in mobile cloud computing is an active area of research. The energy consumption increases when a user's request (task) runs in the local mobile device instead of executing in the cloud. Whereas, latency become an issue when the task executes in the cloud environment instead of the mobile device. Therefore, a tradeoff between energy consumption and latency is required in building sustainable Internet of Things (IoT), and for that, we have introduced a middle layer named an edge computing layer to avoid latency in IoT. There are several real-time applications, such as smart city and smart health, where mobile users upload their tasks into the cloud or execute locally. We have intended to minimize the energy consumption of a mobile device as well as the energy consumption of the cloud system while meeting a task's deadline, by offloading the task to the edge datacenter or cloud. This paper proposes an adaptive technique to optimize both parameters, i.e., energy consumption and latency by offloading the task and also by selecting the appropriate virtual machine for the execution of the task. In the proposed technique, if the specified edge datacenter is unable to provide resources, then the user's request will be sent to the cloud system. Finally, the proposed technique is evaluated using a real-world scenario to measure its performance and efficiency. The simulation results show that the total energy consumption and execution time decrease after introducing an edge datacenters as a middle layer.

INDEX TERMS Cloud computing, latency, edge datacenter, energy consumption, mobile computing, task scheduling, IoT.

I. INTRODUCTION

Mobile devices (smartphones, tablets, etc.) have become essential objects in today's world. Billions of global users are connected to the Internet and various information services through a variety of mobile devices that can in turn be used to build IoT infrastructure. With the help of fast 4G and LTE (Long-Term Evolution) networks mobile devices have become the medium to connect clouds to the mobile realm. For more than a decade, cloud computing has been a hot topic for both industry and academia. It provides computing resources, service platforms, and software to users and IT-companies. It is already known that mobile devices generate more traffic than any other network connected devices [1]. It is important to note that mobile device resources, such as, battery life, computing capability, bandwidth, and storage capacity remain constant. Thus, a new gap is emerging between the requirements of a new generation of mobile devices and legacy mobile devices with limited resource capacity. To solve this problem, mobile devices need to take advantage of the ample resources available in cloud systems [2]. To take advantage of cloud services tasks need to be offloaded into a cloud, *i.e.* from mobile devices to cloud systems over wireless channels.

In IoT, mobile devices have become core components for computation and inter-device communication [3]. Due to limited resources smart devices upload their tasks to the datacenter for computation. Smartphones can have a connection with the cloud through an access point or cellular network services. Service-oriented applications (like database servers and web applications), utility computing and virtualization are the backbones of cloud computing services. Clouds provide an ideal solution for mobile computing because of various reasons such as system mobility, communication, and portability.

Today, battery life is one of the prime factors affecting the usability of mobile devices. By improving CPU execution time, screen quality and utilization, battery life can be increased [4], [5]. Thus, we propose an algorithm for task offloading to the edge datacenter and the cloud, which should relief mobile devices from consuming huge amounts of energy by avoiding task runs locally. Some of the local tasks will take longer time to run due to their size and the limited capability of the mobile device and this increases battery consumption. The introduction of an edge datacenter, between the mobile user and the cloud, allows for exploiting customized services that could be running on a nearby virtual machine (VM) (i.e. close to the mobile user). This technology can assist users to overcome the limitations of mobile computing due to latency and low bandwidth [6].

Storage capacity is also a significant barrier for mobile devices. Mobile Cloud Computing (MCC) has developed to overcome this problem and store data in the cloud through the wireless network [7]. One such example is the Amazon Simple Storage Service, which can allow MCC services to offer users more processors to handle their applications.

Task offloading from a mobile device to a cloud might not be the best solution. The offloading process might take longer time in getting service responses. As a result, edge computing can provide the needed resources for mobile devices instead of a cloud datacenter [8]. In this way, we could amortize the latency problem associated with service responses. An edge datacenter constitutes of a small subset of the resources of those in a cloud. The response will be faster if we offload the tasks to an edge datacenter, and also the propagation time will be lesser as compared to offloading them to a cloud [9]. The execution of a large number of tasks through the resources available in the edge datacenter or cloud, or locally by the mobile device can also improve the total energy consumption.

A. PRELIMINARIES

Definition 1: **Mobile Cloud Computing (MCC)** refers to an infrastructure where both the data storage and data processing occur outside the mobile device. Mobile cloud applications move the computing power and data storage away from mobile phones into the cloud for efficient execution.

Definition 2: **Edge Computing** is a service delivery model that can be used to process tasks that that have limited access to resources in a distributed system.

The resource allocation problem for mobile applications can be illustrated through an example. We consider three mobile devices, where two tasks (t_{11}, t_{12}) in the first mobile, three tasks (t_{21}, t_{22}, t_{23}) in the second mobile and one task (t_{31}) in the third mobile. Further, it is assumed that there are two VMs ($vedc_1, vedc_2$) in the edge datacenter and four VMs (vc_1, vc_2, vc_3, vc_4) in the cloud. The energy consumed

TABLE 1. Energy consumption for offloading tasks to VMs in the edge datacenter and cloud.

Task	Energy used to offload a task to an edge datacenter		Energy used to offload task to the cloud				locally
	vc_1	vc_2	vc_1	vc_2	vc_3	vc_4	
t_{11}	3	4	5	3	4	7	9
t_{12}	1	3	6	3	2	1	7
t_{21}	5	3	3	2	6	3	10
t_{22}	4	4	5	7	5	9	12
t_{23}	3	2	5	8	3	7	9
t_{31}	3	4	3	5	4	3	6

TABLE 2. Execution delays for offloading tasks to VMs in an edge datacenter and cloud.

Task	Delay used to offload a task to the edge datacenter		Delay used to offload a task to the cloud			
	vc_1	vc_2	vc_1	vc_2	vc_3	vc_4
t_{11}	2	2	4	3	5	2
t_{12}	2	3	2	3	4	4
t_{21}	3	2	3	4	2	5
t_{22}	3	5	9	7	6	6
t_{23}	4	3	8	6	7	5
t_{31}	1	2	4	3	5	4

for tasks offloading to the edge datacenter VMs as well as to the cloud VMs are given in Table 1. The last column of Table 1 shows the energy consumption for the execution of tasks locally in the mobile devices. Similar to the energy table, the task execution delay table is shown in Table 2. Here, the execution delay is 0, by running tasks locally in mobile devices.

One of the allocation results for mobile tasks to VMs is shown in Table 3. These allocation results are for minimum energy consumption *i.e.* 20, where we are not concerned with the total execution delay *i.e.* 18. If we allocate the task t_{22} to vc_3 instead of vc_1, then the total execution delay is reduced to 15 without affecting the total energy consumption as shown in Table 4. In most of the cases, the energy consumption value decreases due to the tasks execution in an edge datacenter. One of the reasons for this is the communication distance between the mobile device and edge datacenter and between the mobile device and cloud system.

This paper studies the problem of tasks offloading from mobile devices to the edge datacenter or to the cloud for minimizing the energy consumption as well as execution

TABLE 3. An allocation result.

Task	VM	Energy consumed	Delay
t_{11}	$vedc_1$	3	2
t_{12}	$vedc_1$	1	2
t_{21}	$vedc_2$	3	2
t_{22}	vc_1	5	9
t_{23}	$vedc_2$	2	3
t_{31}	Mobile	6	0
Total		20	18

TABLE 4. An allocation result.

Task	VM	Energy consumed	Delay
t_{11}	$vedc_1$	3	2
t_{12}	$vedc_1$	1	2
t_{21}	$vedc_2$	3	2
t_{22}	vc_3	5	6
t_{23}	$vedc_2$	2	3
t_{31}	Mobile	6	0
Total		20	15

delays. There are three cases: (1) whether the task can be offloaded to the edge datacenter or not; (2) whether the task can be offloaded to the cloud or not; (3) if both (1) and (2) fail, then execute the task locally in the mobile device. This paper also explains how to select an appropriate virtual machine in the edge datacenter or cloud according to the specified deadline so that performance metrics are optimized.

B. MOTIVATIONS

User demands gradually migrate popular software to mobile platforms. However, mobile devices are restricted by limited computational capability and battery capacity [10]. Therefore, energy remains a major obstacle for the implementation of reliable and sophisticated mobile applications. Task offloading models can satisfy the requirement of the energy-saving. Offloading of mobile tasks (application) for further computation can prolong battery life and optimize computation capability, network bandwidth, and storage space [10]. In addition to energy consumption of mobile device, cloud energy consumption can be optimized by the carefully balancing the interplay between clouds and mobiles offloading [11]. The Mobile cloud computing (MCC) paradigm combines the strength of clouds and mobile devices which generated a lot of attention [10], [12]. This motivated us to introduce an intermediate layer to deploy edge datacenters to help in building sustainable IoT infrastructure.

C. CONTRIBUTIONS

The contributions of the paper are summarized as follows.

1) The task allocation problem is formulated as an integer linear programming (ILP) problem.
2) A polynomial time heuristic algorithm has been proposed for the allocation and execution of heterogeneous mobile applications with the available resources of an edge datacenter or cloud or locally in the mobile device.
3) Proposed algorithm optimizes the energy consumption and the delays associated with the task. This paper also has analyzed the complexity of the proposed algorithm.
4) The simulations illustrate the effectiveness of the proposed algorithm.

D. ORGANIZATION

The remaining of the paper is organized as follows. Section 2 outlines an overview of some of the related work; Section 3 reviews the system with the formulation of the problem; Section 4 illustrates the proposed work to optimize various parameters in a heterogeneous computing environment. Section 5 presents the simulation results and shows the effectiveness of the proposed algorithm (*EETAMCS*), and Section 6 concludes the work.

II. RELATED WORK

The advancement of cloud computing systems with virtualization techniques offer an adequate way of executing a large number of user tasks, and hence considerably enhance the utilization of computing resources [13]–[17]. MCC is meant to address the computation specifics of latest smart mobile phones based applications [1], [14], [18]. More research work has been proposed for the offloading of tasks to the cloud [6], [19], [20].

When a huge number of tasks are offloaded to the edge datacenter or cloud, resource allocation becomes more challenging since the problem is already NP-hard [21]. Various heuristic approaches have been proposed to solve the allocation problem. Traditionally, a cloud computing system offers pay per use policy for the execution of tasks.

A cloudlet-based mobile cloud computing model is proposed with a focus on determining the extra energy consumption during dynamic wireless communications [9]. Researchers have introduced a cloudlet layer in between the mobile application and the cloud to build a green mobile cloud. Chen *et al.* [22] have developed an online small-cell base station offloading framework to maximize system performance while meeting energy constraints. They have also formulated an equal offloading game among base stations to examine the stability and efficiency loss in terms of cost. Li and Wang [23] explained the support of mobile applications with the help of a mobile cloudlet (or edge datacenter) [7]. Specifically, researchers have investigated the cloudlet capacity, cloudlet nodes lifetime, and resource reachable time for an efficient mobile cloud system [23].

A probabilistic time deadline for each task and compared energy consumption is proposed by the execution of a task locally on the mobile device and the remote execution on a cloud [20]. Authors have expressed the cooperative task execution as a constrained stochastic shortest path problem across an acyclic graph with a restriction of a probabilistic time deadline. Their main aim is to minimize the energy consumption of mobile device while meeting the task deadline, by offloading tasks to cloud. Satyanarayanan *et al.* [6] have designed an architecture through virtualization technology to

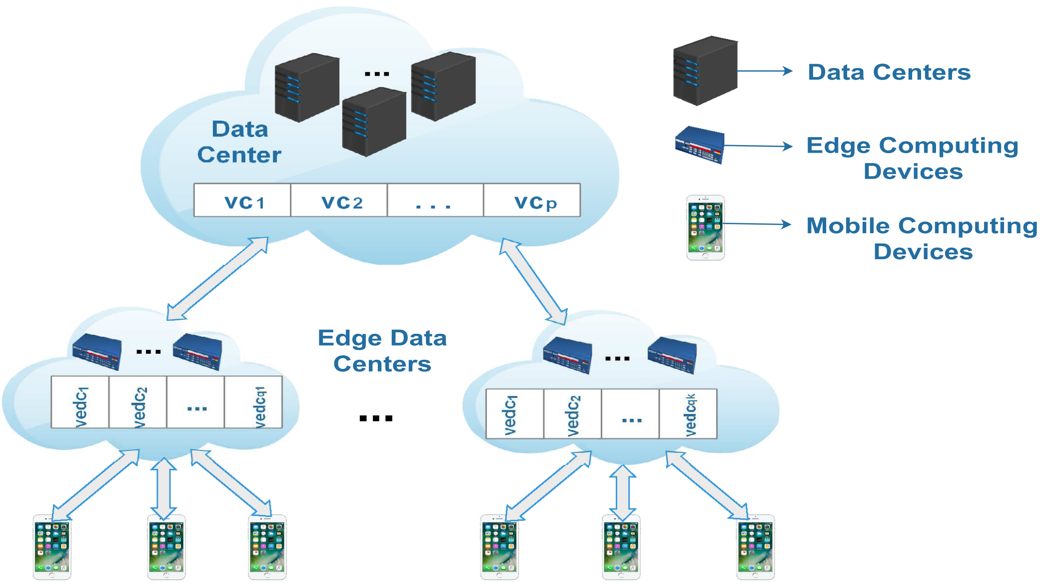


FIGURE 1. Mobile cloud system model with an edge datacenter.

instantiate customized service software quickly on a nearby cloudlet and then utilize that service over a wireless LAN.

Cui *et al.* [21] have expressed the minimum-energy task allocation problem as a 0-1 knapsack problem and shown its NP-hardness. They have proposed a greedy algorithm with a polynomial-time complexity that allocates the task in a centralized mode to conserve the energy of mobile devices. Kwak *et al.* [5] have optimized the CPU and network speeds jointly to reduce the energy consumption of smartphone applications. They have proposed a dynamic speed scaling scheme to adjust the processing and networking rates jointly for smartphones with the help of multiple wireless interfaces, Dynamic Voltage and Frequency Scaling (DVFS), and multi-tasking capabilities. They have shown that their approach can improve 42% of battery usage when compared to the existing schemes.

Previous work that is closely related to ours [13] and inspired how we built our model. Specifically, the work in [13] optimized energy consumption and execution delays by offloading the tasks to the cloud. However, the delay is higher when the task is executed in the cloud as compared to local execution in the mobile device. This paper considers optimizing both energy consumption and the execution delays of mobile applications. By considering the tradeoff between energy consumption and latency, we propose algorithms to decide whether a mobile application should be offloaded to the VMs of the edge datacenter or the VMs of the cloud or executed by the mobile device locally [24].

III. SYSTEM MODEL

In the proposed system in this paper, all mobile devices are connected to a network via an access point; then the access point sends the task to the next platform (edge datacenter). The edge datacenter has a connection with the cloud system as shown in Figure 1. The system model has three layers: (1) the application layer, (2) edge computing layer, and (3) cloud computing layer. In the application layer, the mobile users generate their requests (tasks) for further processing. Tasks are executed if the edge datacenter has available resources. The cloud computing layer has enough resources for the execution of applications. The cloud system, as well as the edge datacenter, have physical resources which are virtualized to support multitasking. The edge computing layer has limited and fewer resources as compared to that of the cloud.

We have considered a set of m mobile devices, denoted by $M = \{M_1, M_2, \ldots, M_m\}$. There are a finite number of input tasks from each mobile device. The input task set of the i^{th} mobile device is represented as $T_i = \{t_{i1}, t_{i2}, \ldots, t_{in_i}\}$. Here, t_{ij} represents the j^{th} input task of i^{th} mobile device, and M_i has n_i number of tasks. The set $D_i = \{d_{i1}, d_{i2}, \ldots, d_{in_i}\}$ represents the deadline of the i^{th} mobile device, where d_{ij} is the deadline of task t_{ij}. The input task from each mobile device can be executed locally on the mobile device or offloaded onto a cloud or offloaded to the edge datacenter to preserve the energy consumption of the mobile device. There are k numbers of edge datacenters. Each edge datacenter has q_i number of VMs, where $1 \leq i \leq k$. The value of q_i is very small when compared to p, and $q = \sum q_i$. The set $VC = \{vc_1, vc_2, \ldots, vc_p\}$ has p number of VMs in the Cloud, while the set $VEDC = \{vedc_1, vedc_2, \ldots, vedc_q\}$ has q number of VMs in the edge datacenter.

This paper considers the allocation of all the mobile applications in three phases. In phase-I, we have tried to allocate each task to the VMs of the edge datacenter. If phase-I fails, then in phase-II we try to allocate the tasks to the VMs of the cloud system. Finally, if both phase-I and phase-II fail, then the task is executed locally in the mobile device.

The energy consumption of mobile device M_i while executing task t_{ij} locally is e_{ij}. The energy consumption is e_{ijk} to execute task t_{ij} on the k^{th} VM of the edge datacenter. Similarly, the energy consumption is e'_{ijk} to execute task t_{ij} on the k^{th} VM of the cloud system. The energy utilized by the mobile devices to execute applications locally is much more than that to offload the applications to the cloud. And also the energy consumed by the cloud system to run a mobile application is more than that needed to run in the edge datacenter.

Assume that DC_{ijk} and $DEDC_{ijk}$ are the delays acquired by offloading task t_{ij} to k^{th} VM of the cloud and the k^{th} VM of the edge datacenter, respectively.

We have defined the two variables X_{ijk}, and Y_{ijk} as follows:

$$X_{ijk} = \begin{cases} 1, & \text{if } t_{ij} \text{ is offloaded to VM } vedc_k \\ 0, & \text{Otherwise} \end{cases}$$

$$Y_{ijk} = \begin{cases} 1, & \text{if } t_{ij} \text{ is offloaded to VM } vc_k \\ 0, & \text{Otherwise} \end{cases}$$

A. PROBLEM FORMULATION

The execution delay of task t_{ij} is 0, if t_{ij} runs locally on the mobile device. DC_{ijk}, $DEDC_{ijk}$ are the delay incurred by offloading task t_{ij} to the k^{th} VM of the cloud and the k^{th} VM of the edge datacenter, respectively. The execution delay of task t_{ij} is:

$$\sum_{k=1}^{q} X_{ijk} \times DEDC_{ijk} + \sum_{k=1}^{p} Y_{ijk} \times DC_{ijk} \tag{1}$$

Then, the total delay of executing all of the tasks can be expressed as follows:

$$\sum_{i=1}^{m} \sum_{j=1}^{n_i} \left[\sum_{k=1}^{q} X_{ijk} \times DEDC_{ijk} + \sum_{k=1}^{p} Y_{ijk} \times DC_{ijk} \right] \tag{2}$$

The energy consumption of a mobile device M_i denoted as E_i incorporates three components: (1) the energy consumed to offload mobile applications to the VMs of the edge datacenter, (2) the energy consumed to offload mobile applications to the VMs in the cloud, and the energy utilized to run some mobile applications locally. We can express it as follows:

$$E_i = \sum_{j=1}^{n_i} \left[\left(1 - \sum_{k=1}^{q} X_{ijk} \times \sum_{l=1}^{p} Y_{ijl} \right) \times e_{ij} + \sum_{k=1}^{q} X_{ijk} \times e'_{ijk} + \sum_{l=1}^{p} Y_{ijl} \times e_{ijl} \right] \tag{3}$$

Therefore, the total energy consumed by all the mobile devices (M) is E estimated by using *Eq.* 3 as shown below.

$$E = \sum_{i=1}^{m} E_i \tag{4}$$

The contribution of this paper focuses on minimizing the energy consumption and execution delay in the cloud environment. Therefore, we have formulated the objective as follows.

$$Minmize \sum_{i=1}^{m} \sum_{j=1}^{n_i} \left[\sum_{k=1}^{q} X_{ijk} \times DEDC_{ijk} + \sum_{k=1}^{p} Y_{ijk} \times DC_{ijk} \right] + \eta \times E \tag{5}$$

Subject to

$$\sum_{l=1}^{q} Y_{ijl} = \{0, 1\}, \quad \forall\, t_{ij} \in T \tag{6}$$

$$\sum_{k=1}^{p} X_{ijk} = \{0, 1\}, \quad \forall\, t_{ij} \in T \tag{7}$$

$$\sum_{j=1}^{n_i} \left[\left(1 - \sum_{k=1}^{q} X_{ijk} \times \sum_{l=1}^{p} Y_{ijl} \right) \times e_{ij} \right] \le E_i \quad \forall\, i,\ 1 \le i \le m \tag{8}$$

Eq. 6 and *Eq.* 7 indicate that a task can be allocated to one VM in the edge datacenter or cloud. Followed by *Eq.* 8 which suggests that the energy employed to run mobile applications locally in mobile device M_i should be less than the available energy for M_i to remain operational.

The objective function *i.e. Eq.* 5 uses η to maintain the trade-off between the energy consumption and delay. In some cases, if the energy consumption of a mobile device is not a critical issue as compared to the delay of the application, then the value of η can be set to a small value or 0. If the η value is 0, the problem becomes one of minimizing delay. Inversely, if energy consumption is a major concern as compared to delay, then the value of η is set to a larger one.

IV. PROPOSED RESOURCE ALLOCATION TECHNIQUE

The resources in an edge datacenter are not as abundant as those in a cloud, due to the limited number of VMs. So, if a large number of tasks is allocated to the edge datacenter in a random manner, then the edge datacenter may run out of resources after a certain time. Particularly, when large numbers of tasks are executed simultaneously, this will lead to the edge datacenter becoming overloaded. This highlights the need to propose solutions for the overloading problem.

In this section, we present an algorithm for the allocation of heterogeneous mobile applications (tasks) to resources. The resource requirements of the mobile applications are always different (i.e. dynamic), which can be estimated by the use of Expected Time to Complete (ETC) matrix. According to the ETC matrix a VM requires different time and energy for the execution of different tasks [25]. In our proposed method, at first, Scheduler-I tries to offload tasks to the edge datacenter. If Scheduler-I fails, then the task is offloaded to the cloud and if Scheduler-II fails again, then the task is executed in the mobile device. So, the task is returned to the mobile device for execution if the task cannot get serviced by the cloud or the edge datacenter. These steps will reduce the energy consumption as well as take into account the execution delays. The proposed method, Energy-Efficient Task

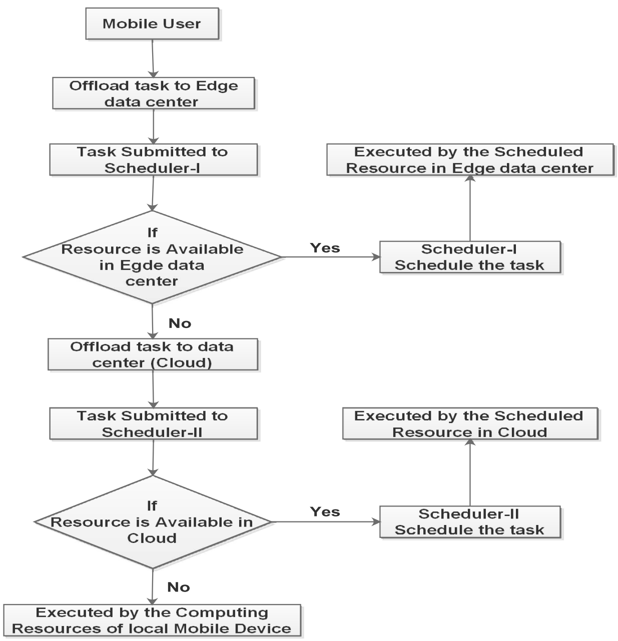


FIGURE 2. **Task execution in a mobile cloud environment.**

Allocation in Mobile Cloud System (EETAMCS), is presented in Algorithm 1. Scheduler-I and Scheduler-II employ Algorithm 2 and Algorithm 3, respectively. A flowchart for the complete process for the execution of a task in a mobile cloud environment is given in Figure 2.

The inputs to Algorithm 1 are the set of mobile devices: $M = \{M_1, M_2, \ldots, M_m\}$, the task set: $T = \{t_{ij}\}, 1 \leq i \leq m, 1 \leq j \leq n$, the deadline of each task: $D = \{d_{ij}\}, 1 \leq i \leq m, 1 \leq j \leq n$, available set of VMs in the cloud: $VC = \{vc_1, vc_2, \ldots, vc_p\}$, available set of VMs in the edge datacenter: $VEDC = \{vedc_1, vedc_2, \ldots, vedc_q\}$. The algorithm undergoes several self-explanatory initializations steps (1-6). If the task can be allocated to a VM in the edge datacenter, then the condition in step (9) of Algorithm 1 becomes true. Then, the *Execute_Edge_Data_Center*() procedure presented in Algorithm 2 is invoked. Algorithm 2 returns the VM location of the edge datacenter to which the task t_{ij} will be allocated. The energy consumption of that VM for the execution of t_{ij} will be computed, i.e, e_{ijl} via *Eq*. 3, and the set of the execution times of all VMs of the edge datacenter will be updated ($ETVEDC$).

The inputs to Algorithm 2 are the task: t_{ij}, the deadline of task: d_{ij}, VM set of the edge datacenter: $VEDC = \{vedc_1, vedc_2, \ldots, vedc_q\}$, execution time of all VMs of the edge datacenter: $ETVEDC = \{ETVEDC_1, ETVEDC_2, \ldots, ETVEDC_q\}$, and ETC matrix. Here, the set AVM_{ij} is initiated to an empty set, which unites all available VMs for the execution of t_{ij} based on the task deadline d_{ij}. The ETC matrix accommodates the expected time to complete task t_{ij} in $vedc_k$, *i.e.* $1 \leq k \leq q$. Algorithm 2 selects the VM which consumes minimum energy among all the VMs of AVM_{ij} and assigned to $vedc_r$. Where, r is the position of the VM and $1 \leq r \leq q$.

Algorithm 1 Energy-Efficient Task Allocation in Mobile Cloud System (*EETAMCS*)

Input: Mobile device set: $M = \{M_1, M_2, \ldots, M_m\}$, Task set: $T = \{t_{ij}\}, 1 \leq i \leq m, 1 \leq j \leq n$, Deadline of task: $D = \{d_{ij}\}, 1 \leq i \leq m, 1 \leq j \leq n$, VM set in Cloud: $VC = \{vc_1, vc_2, \ldots, vc_p\}$, VM set in Edge Datacenter: $VEDC = \{vedc_1, vedc_2, \ldots, vedc_q\}$.

Output: Energy Consumption of all tasks ($Energy$)

1: Initialize Execution time $ETVC_k \leftarrow 0$ ∀ VMs available in the Cloud $k, 1 \leq k \leq p$
2: Initialize Execution time $ETVEDC_k \leftarrow 0$ ∀ VMs available in the edge datacenter $k, 1 \leq k \leq q$
3: Initialize Energy consumption of VMs in Cloud $EVC_k \leftarrow 0\ \forall k, 1 \leq k \leq p$
4: Initialize Energy consumption of VMs in edge datacenter $EVEDC_k \leftarrow 0\ \forall k, 1 \leq k \leq q$
5: Initialize Energy consumption of M_i, $EM_i \leftarrow 0\ \forall i, 1 \leq i \leq m$
6: Initialize $EM_i \leftarrow 0\ \forall i, 1 \leq i \leq m$
7: **for** each mobile $M_i \in M$ do **do**
8:     **for** each task $t_{ij} \in T$ do **do**
9:         **if** $\sum_{l=1}^{q} X_{ijl} = 1$ **then**
10:             $[l, e_{ijl}, ETVEDC_l] = Execute\_Edge\_Data\_Center()$
11:             Execute t_{ij} on l^{th} VM of Edge Datacenter.
12:             Update $EVEDC_l = EVEDC_l + e_{ijl}$
13:         **else**
14:             **if** $\sum_{k=1}^{p} Y_{ijk} = 1$ **then**
15:                 $[k, e'_{ijk}, ETVC_k] = Execute\_Cloud()$
16:                 Execute t_{ij} on k^{th} VM of Cloud.
17:                 Update $EVC_k = ETVC_k + e'_{ijk}$
18:             **else**
19:                 Execute t_{ij} at mobile device M_i locally.
20:                 Update $EM_i = EM_i + e_{ij}$
21:             **end if**
22:         **end if**
23:     **end for**
24: **end for**
25: $EEDC = \sum_{k=1}^{q} EVEDC_k\ \forall vedc_k \in VEDC$
26: $EC = \sum_{k=1}^{q} EVC_k\ \forall vc_k \in VC$
27: $EM = \sum_{i=1}^{m} EM_i\ \forall M_i \in M$
28: Return $Energy = EEDC + EC + EM$

The value of $ETVEDC_r$ is updated and finally returns all values to Algorithm 1. This is followed by allocating task t_{ij} to the l^{th} VM of the edge datacenter for the execution. The update of energy consumption of the l^{th} VM of the edge datacenter is performed as shown in step (12) of Algorithm 1 and then proceed for the allocation of the next mobile task.

If the resource requirements for a task are not available in the edge datacenter, then the condition in step (9) of Algorithm 1 is false. Now, the condition in step (14) of Algorithm 1 is checked, if the task t_{ij} offloaded to the cloud, then the condition becomes true. Then, the *Execute_Cloud*()

Algorithm 2 *Execute_Edge_Data_Center*

Input: Task: t_{ij}, Deadline of task: d_{ij}, VM set: $VEDC = \{vedc_1, vedc_2, \ldots, vedc_q\}$, $ETVEDC = \{ETVEDC_1, ETVEDC_2, \ldots, ETVEDC_q\}$, ETC.

Output: VM position: Loc, Energy Consumption at Edge Datacenter: $Energy$, Updated $ETVEDC$

Initialize $AVM_{ij} \leftarrow \phi$
for each VM $vedc_k \in VEDC$ **do**
  if $\{ETC(vedc_{ijk}) - ETVEDC_k\} \leq d_{ij}$ **then**
    $AVM_{ij} = AVM_{ij} \cup vedc_k$;
  end if
end for
Find $vedc_r = Min\{Energy\_Consumption(AVM_{ij})\}, 1 \leq r \leq q$
Update $ETVEDC_r = ETVEDC_r + ETC_{ijr}$
$Energy = e'_{ijr}$
$Loc = r$
Return Loc, $Energy$, $ETVEDC$

Algorithm 3 *Execute_Cloud*

Input: Task: t_{ij}, Deadline of task: d_{ij}, VM set: $VC = \{vc_1, vc_2, \ldots, vc_p\}$, $ETVC = \{ETVC_1, ETVC_2, \ldots, ETVC_p\}$, ETC.

Output: VM position: Loc, Energy Consumption at Cloud: $Energy$, Updated $ETVC$

Initialize $AVM_{ij} \leftarrow \phi$
for each VM $vc_k \in VC$ **do**
  if $\{ETC(vc_{ijk}) - ETVC_k\} \leq d_{ij}$ **then**
    $AVM_{ij} = AVM_{ij} \cup vc_k$;
  end if
end for
Find $vc_r = Min\{Energy\_Consumption(AVM_{ij})\}, 1 \leq r \leq p$
Update $ETVC_r = ETVC_r + ETC_{ijr}$
$Energy = e'_{ijr}$
$Loc = r$
Return Loc, $Energy$, $ETVC$

procedure presented in Algorithm 3 is called. Algorithm 3 returns the VM location in the cloud to which task t_{ij} will be allocated. The energy consumption of that VM for the execution of t_{ij}, i.e, e'_{ijk}, and the updated set of execution times of all VMs of the cloud ($ETVC$).

The inputs to Algorithm 3 are the same as those of Algorithm 2, *i.e.* task: t_{ij}, the deadline of task: d_{ij}, VMs set of the cloud: $VC = \{vc_1, vc_2, \ldots, vc_p\}$, execution time of all VMs of the cloud: $ETVC = \{ETVC_1, ETVC_2, \ldots, ETVC_p\}$, and ETC matrix. Similarly, here also the set AVM_{ij} is initiated to an empty set, which unites all available VMs for the execution of t_{ij} based on the task deadline d_{ij}. The ETC matrix accommodate the expected time to complete task t_{ij} in vc_k, $1 \leq k \leq p$. Algorithm 3 selects the VM which consumes minimum energy among the VMs of AVM_{ij} and assigns to vc_r. Here, r is the position of the VM and $1 \leq r \leq p$. The $ETVC_r$ value is updated and finally returns all values to Algorithm 1. Followed by, task t_{ij} is allocated to the k^{th} VM of the edge datacenter for execution. The update of energy consumption of the k^{th} VM of the cloud is performed in step (12) of Algorithm 1 and then proceed for the allocation of the next mobile task.

The unavailability of virtual resources in the edge datacenter and cloud for the execution of mobile task t_{ij} results in the local execution in the mobile device M_i. The energy consumption of i^{th} mobile device is updated in step (20) of Algorithm 1. The energy consumption of the edge datacenter ($EEDC$) is the sum of energy consumed by all the VMs of the edge datacenter. Similarly, the energy consumption of the cloud (EC) is the sum of energy consumed by all VMs of the cloud. The energy consumption due to the execution of tasks locally by the mobile devices (EM) is the sum of energy consumption of all individual mobile devices. The total energy consumption is the addition of all three as in step (28) of Algorithm 1.

Lemma 3: The complexity of the proposed algorithm ($EETAMCS$) is $O(mn_ip^2)$, where m is the number of mobile devices, n_i is the number of tasks of the i^{th} mobile device, and p is the number of VMs in the cloud.

Proof: To estimate the complexity of Algorithm 1, we need to analyze the individual steps of the algorithm. The algorithm includes the following steps:

- All the initialization steps of Algorithm 1 from steps (1-6) requires $Maximum\{O(2p) + O(2q) + O(2m)\}$ as time complexity which is linear.
- The maximum time required to evaluate step (9) of Algorithm 1 is $O(mn_iq)$. If the comparison is true, then Algorithm 2 is called and which requires $O(q)$ in time complexity. Therefore, the complexity is $O(mn_iq^2)$.
- Similarly, the maximum time required to evaluate step (14) of Algorithm 1 is $O(mn_ip)$. If the comparison is true, then Algorithm 3 is called and which requires $O(p)$ in time complexity. Therefore, the complexity is $O(mn_ip^2)$.
- The rest of the of Algorithm 1 in steps (25-27) which requires linear time and the last step requires constant time *i.e.* $O(1)$.

From the above analysis, we conclude that the complexity of the algorithm is $O(mn_iq^2 + mn_ip^2)$. Since, $q \leq p$, the complexity is $O(mn_ip^2)$. ■

V. PERFORMANCE EVALUATION

In this section we evaluate the performance of the proposed algorithm by simulating a real world scenario. The simulation of *EETAMCS* is conducted using the CloudSim simulator [26]. In this work, we have studied the impact of task offloading onto the edge datacenter and have considered a network of 20 mobile devices with randomly generated heterogeneous tasks. Also, we assumed a uniform distribution for the generation of energy consumption and delay of each task, when they are executed locally, or on the VMs of edge

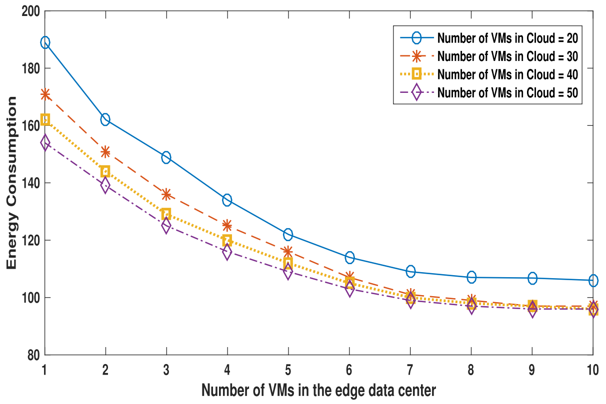


FIGURE 3. Energy consumption in the mobile cloud system by varying the number of VMs in the edge datacenter and cloud for 50 tasks.

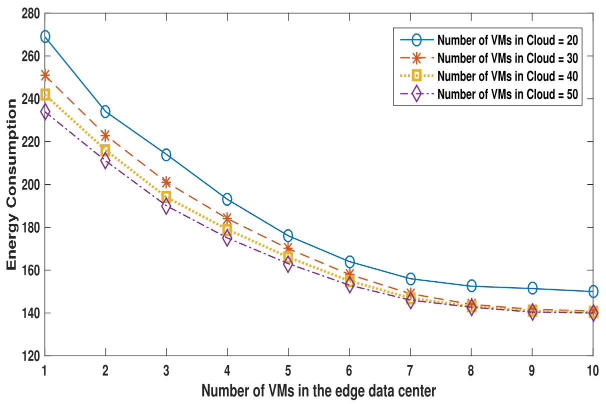


FIGURE 4. Energy consumption in the mobile cloud system by varying the number of VMs in the edge datacenter and cloud for 80 tasks.

datacenter or the VMs of the cloud. To set the number of VMs of the edge datacenter and cloud, we have performed simulations with a different number of tasks, *i.e.*, 50, 80, and 100. The number of VMs in the edge datacenter are less as compared to the number of VMs in the cloud. Here, we have experimented by varying the number of VMs of the edge datacenter from 1 to 10 for different number of VMs in the cloud *i.e.*, 20, 30, 40, and 50. Since we have not considered the energy consumption at the idle state of a VM, the least number of VMs are taken into account. The simulation runs for 10 times and the average results are considered for the performance evaluation.

Figures 3-5 show energy consumption by varying the number of VMs in the edge datacenter and the cloud. It has been observed that the changes of energy consumption are steady after reaching 7 VMs in the edge datacenter and for all four sets of VMs of the cloud. The energy consumption is less when the number of VMs is more, which is concluded from the experiments as shown in Figures 3-5. From the graph, we can set the number of VMs in the cloud at 30 or 40 or 50. Now, we consider another parameter, execution delay, to observe the performance in the last experimental setup. From Figures 6-8, it can be observed that the

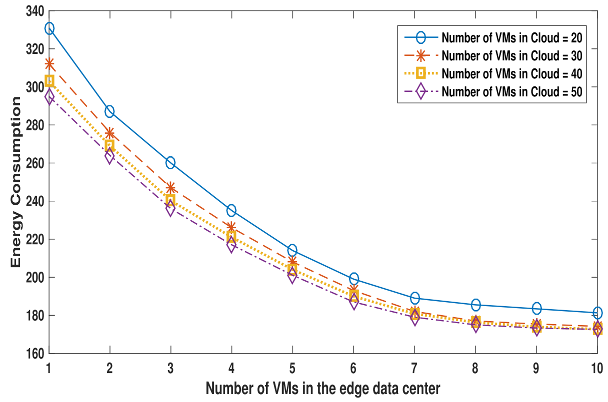


FIGURE 5. Energy consumption in the mobile cloud system by varying the number of VMs in the edge datacenter and cloud for 100 tasks.

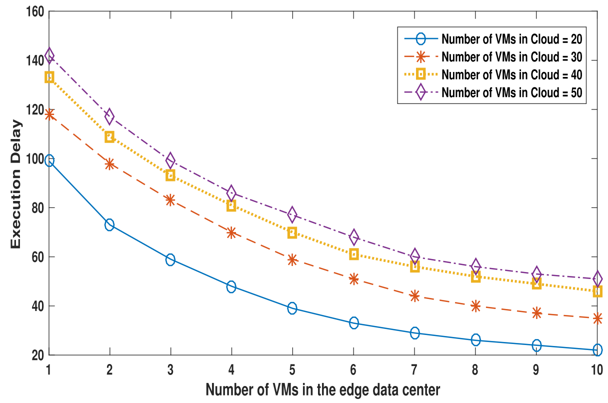


FIGURE 6. Execution delay in the mobile cloud system by varying the number of VMs in the edge datacenter and cloud for 50 tasks.

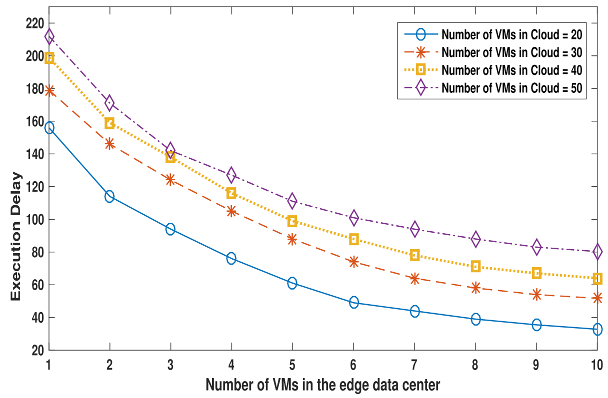


FIGURE 7. Execution delay in the mobile cloud system by varying the number of VMs in the edge datacenter and cloud for 80 tasks.

improvement in delay is consistent after reaching 7 VMs in the edge datacenter. From the results, we can set the number of VMs in the cloud to 30 or 40 or 50. So, from Figures 3-8, we found that the number of VMs in the edge datacenter can be set to at least 7 and the number of VMs in the cloud is set to 30 as we give more priority to energy as compared to execution delays to deliver sustainable IoT infrastructure.

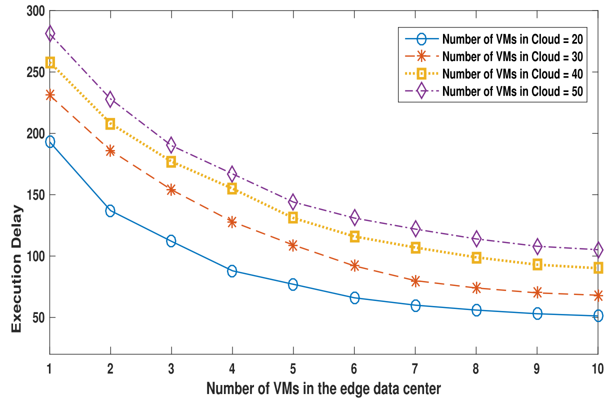


FIGURE 8. Execution delay in the mobile cloud system by varying the number of VMs in the edge datacenter and cloud for 100 tasks.

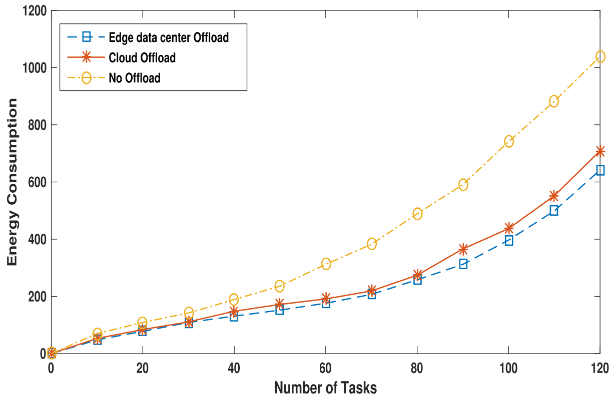


FIGURE 9. Energy consumption versus the number of tasks for 7 VMs in the edge datacenter and 30 VMs in the cloud system.

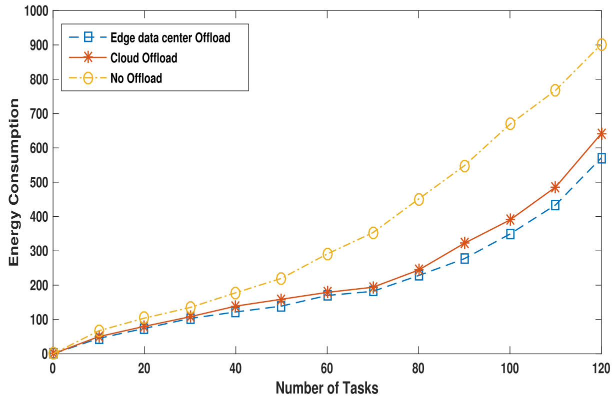


FIGURE 10. Energy consumption versus the number of tasks for 8 VMs in the edge datacenter and 30 VMs in the cloud system.

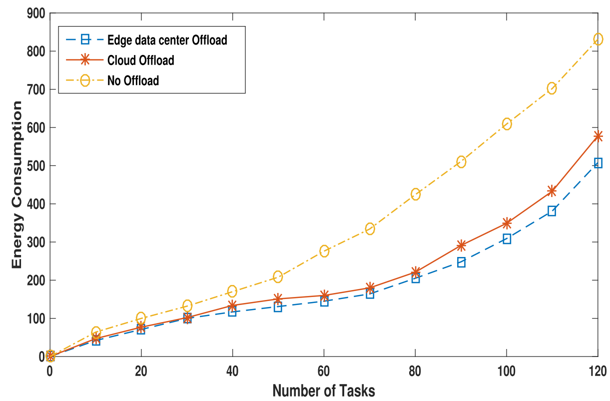


FIGURE 11. Energy consumption versus the number of tasks for 9 VMs in the edge datacenter and 30 VMs in the cloud system.

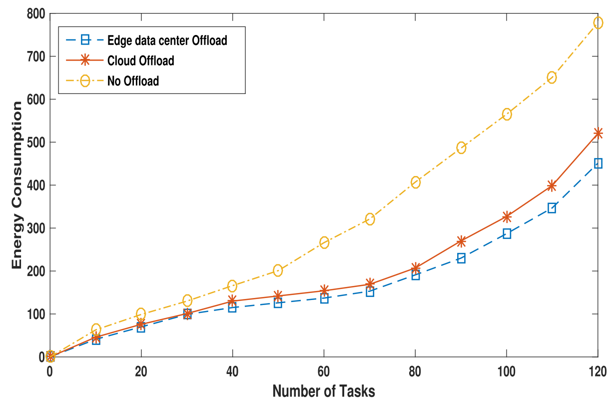


FIGURE 12. Energy consumption versus the number of tasks for 10 VMs in the edge datacenter and 30 VMs in the cloud system.

The performance of *EETAMCS* is evaluated by simulating the energy consumption and the execution delay for No Offload (all tasks are executed locally on the mobile devices), Cloud Offload (all tasks are executed on the VMs in the cloud remotely), and Edge datacenter Offload (all tasks are executed on the VMs of the edge datacenter). For the simulation, we have set the number of virtual machines in the edge datacenter to 7, 8, 9, and 10, the number of VMs in the cloud to 30, and the number of tasks varies from 10 to 120 with an interval of 20.

For cloud offload, if the cloud system does not provide services to a task, then that task is executed locally by the mobile device. Similarly, for the case of an edge datacenter Offload, if the edge datacenter does not provide services to a task, then that task is offloaded to the cloud, and if the cloud is not be able to provide services, then the task is executed locally in the mobile device. It is obvious that the energy consumption of mobile devices is much higher when the tasks are executed locally. However, if the tasks are offloaded to the cloud or the edge datacenter, the energy consumption decreases significantly. This is verified by our experiments as given in Figures 9-12. The total energy consumption of the system is less when the tasks are executed via the edge datacenter as compared to the resources of the cloud. Figure 13 shows the comparison of the execution delay of the edge datacenter offload and the cloud offload. We have ignored the case of no offload because we assumed that the execution delay of locally executed tasks is 0. The performance comparison shows that the estimated execution delay as affected by the number of tasks, when the number of VMs is fixed to 7 in the edge datacenter, and number of VMs is fixed to 30 in the cloud. The comparison graph shows the advantage of edge datacenter offload with reduced propagation delay involved in task offloading to the edge datacenter rather than offloading to the cloud.

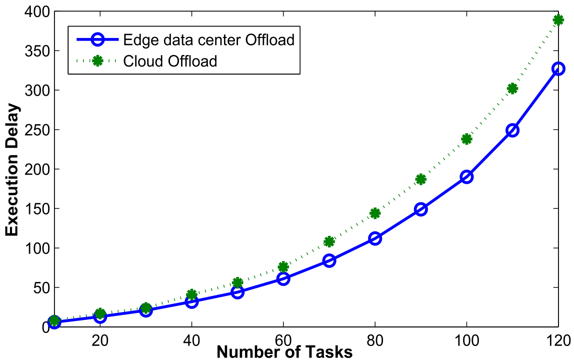


FIGURE 13. Execution delay versus the number of tasks for 7 VMs in the edge datacenter and 30 VMs in the cloud system.

VI. CONCLUSION AND FUTURE DIRECTIONS

This paper investigated system performance issues (energy consumption and execution delays) when offloading tasks from mobile devices to support services for sustainable IoT. We proposed an energy-efficient task allocation in a mobile cloud system (*EETAMCS*) algorithm which also considers execution delays. The algorithm manages to select an appropriate VM for the execution of the task while meeting deadline constraints. It was also proved that the proposed algorithm runs in polynomial time. The mobile cloud system model presented in this paper is based on augmented execution of heterogeneous mobile applications in a cloud environment. In addition to this, we have conducted extensive simulation studies, which demonstrates the performance of the proposed *EETAMCS* for mobile offloading involving an edge datacenter and cloud. This work helps to find an IaaS for different mobile applications by varying the task model.

The proposed technique provides efficient computing services to the mobile cloud user. This can be further modified for various applications. In the future, we aim to apply the proposed approach to other applications that are dynamic in nature and run experiments using real-time testbeds.

REFERENCES

[1] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Future Generat. Comput. Syst.*, vol. 29, no. 1, pp. 84–106, 2013.

[2] M. Shiraz, A. Gani, R. H. Khokhar, and R. Buyya, "A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1294–1313, 3rd Quart., 2013.

[3] S. S. Roy, D. Puthal, S. Sharma, S. P. Mohanty, and A. Y. Zomaya, "Building a Sustainable Internet of Things: Energy-efficient routing using low-power sensors will meet the need," *IEEE Consum. Electron. Mag.*, vol. 7, no. 2, pp. 42–49, Mar. 2018.

[4] D. Puthal, B. P. S. Sahoo, S. Mishra, and S. Swain, "Cloud computing features, issues, and challenges: A big picture," in *Proc. IEEE Int. Conf. Comput. Intell. Netw. (CINE)*, Jan. 2015, pp. 116–123.

[5] J. Kwak, O. Choi, S. Chong, and P. Mohapatra, "Processor-network speed scaling for energy: Delay tradeoff in smartphone applications," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1647–1660, Jun. 2016.

[6] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, Oct./Dec. 2009.

[7] M. Tiwary, D. Puthal, K. S. Sahoo, B. Sahoo, and L. T. Yang, "Response time optimization for cloudlets in mobile edge computing," *J. Parallel Distrib. Comput.*, vol. 119, pp. 81–91, Sep. 2018.

[8] D. Puthal, M. S. Obaidat, P. Nanda, M. Prasad, S. P. Mohanty, and A. Y. Zomaya, "Secure and sustainable load balancing of edge data centers in fog computing," *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 60–65, May 2018.

[9] K. Gai, M. Qiu, H. Zhao, L. Tao, and Z. Zong, "Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing," *J. Netw. Comput. Appl.*, vol. 59, pp. 46–54, Jan. 2016.

[10] L. Zhang, D. Fu, J. Liu, E. C.-H. Ngai, and W. Zhu, "On energy-efficient offloading in mobile cloud for real-time video applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 1, pp. 170–181, Jan. 2017.

[11] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.

[12] "Cisco visual networking index: Global mobile data traffic forecast update, 2016–2021," Cisco, San Jose, CA, USA, White Paper 1454457600805266, Mar. 2017, pp. 1–35.

[13] X. Wang, J. Wang, X. Wang, and X. Chen, "Energy and delay tradeoff for application offloading in mobile cloud computing," *IEEE Syst. J.*, vol. 11, no. 2, pp. 858–867, Jun. 2017.

[14] A. ur Rehman Khan, M. Othman, F. Xia, and A. N. Khan, "Context-aware mobile cloud computing and its challenges," *IEEE Cloud Comput.*, vol. 2, no. 3, pp. 42–49, May/Jun. 2015.

[15] S. K. Mishra, D. Puthal, B. Sahoo, S. K. Jena, and M. S. Obaidat, "An adaptive task allocation technique for green cloud computing," *J. Supercomput.*, vol. 74, no. 1, pp. 370–385, 2018.

[16] C. Yang, M. Yu, F. Hu, Y. Jiang, and Y. Li, "Utilizing cloud computing to address big geospatial data challenges," *Comput., Environ. Urban Syst.*, vol. 61, pp. 120–128, Jan. 2017.

[17] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: Architecture, applications, and approaches," *Wireless Commun. Mobile Comput.*, vol. 13, no. 18, pp. 1587–1611, Dec. 2013.

[18] M. Rawashdeh and A. Alnusair, "Models for multimedia mobile cloud in smart cities," *Multimedia Syst.*, to be published, doi: 10.1007/s00530-017-0552-y.

[19] S. K. Mishra *et al.*, "Energy-efficient VM-placement in cloud data center," *Sustain. Comput., Inform. Syst.*, to be published, doi: 10.1016/j.suscom.2018.01.002.

[20] W. Zhang, Y. Wen, and D. O. Wu, "Collaborative task execution in mobile cloud computing under a stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 14, no. 1, pp. 81–93, Jan. 2015.

[21] Y. Cui *et al.*, "Software defined cooperative offloading for mobile cloudlets," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1746–1760, Jun. 2017.

[22] L. Chen, S. Zhou, and J. Xu. (2017). "Computation peer offloading for energy-constrained mobile edge computing in small-cell networks." [Online]. Available: https://arxiv.org/abs/1703.06058

[23] Y. Li and W. Wang, "Can mobile cloudlets support mobile applications?" in *Proc. IEEE INFOCOM*, Apr. 2014, pp. 1060–1068.

[24] H. El-Sayed *et al.*, "edge of things: The big picture on the integration of edge, IoT and the cloud in a distributed computing environment," *IEEE Access*, vol. 6, pp. 1706–1717, 2018.

[25] S. K. Mishra, D. Puthal, J. J. P. C. Rodrigues, B. Sahoo, and E. Dutkiewicz, "Sustainable service allocation using metaheuristic technique in fog server for industrial applications," *IEEE Trans. Ind. Informat.*, to be published, doi: 10.1109/TII.2018.2791619.

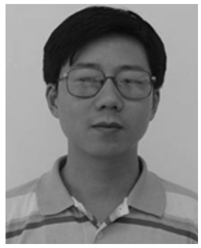
[26] R. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Softw., Pract. Exper.*, vol. 41, no. 1, pp. 23–50, 2011.

SAMBIT KUMAR MISHRA received the M.Tech. and M.Sc. degrees in computer science from Utkal University, India. He has submitted his Ph.D. thesis in computer science and engineering at the National Institute of Technology at Rourkela, Rourkela, India. He is currently an Assistant Professor with the Department of Computer Science and Engineering, Oriental University, Indore, India. His research interests include cloud computing, edge/fog computing, Internet of Things, and wireless sensor networks. He is a member of the IEEE Computer Society, IETE, and CSI.

DEEPAK PUTHAL received the Ph.D. degree in computer science from the University of Technology Sydney (UTS), Australia. He is currently a Lecturer (an Assistant Professor) with the Faculty of Engineering and Information Technologies, UTS. He has authored in several international conferences and journals, including the ACM and IEEE TRANSACTIONS. His research interests include cyber security, Internet of Things, distributed computing, and edge/fog computing. He is a PC chair and a PC member to several IEEE and ACM sponsored conference and symposium. He was a recipient of the 2017 IEEE Distinguished Doctoral Dissertation Award from the IEEE Computer Society and STC on Smart Computing. He served as a Co-Guest Editor of several reputed journals, including *Concurrency and Computation: Practice and Experience*, *Wireless Communications and Mobile Computing*, and *Information Systems Frontier*. He is an Associate Editor of the IEEE TRANSACTION ON BIG DATA, the *International Journal of Communication Systems*, the *IEEE Consumer Electronics Magazine*, the *Internet Technology Letters*, and the *KSII Transactions on Internet and Information Systems*.

BIBHUDATTA SAHOO (M'03) received the M.Tech. and Ph.D. degrees in computer science and engineering from the National Institute of Technology (NIT) at Rourkela, Rourkela, India. He is currently an Associate Professor with the Department of Computer Science and Engineering, NIT Rourkela. His technical interests include data structures and algorithm design, parallel & distributed systems, networks, computational machines, algorithms for VLSI design, performance evaluation methods and modeling techniques distributed computing systems, networking algorithms, and Web engineering. He is a member of ACM.

SURAJ SHARMA received the Ph.D. degree from the National Institute of Technology at Rourkela, Rourkela, India. He is currently an Assistant Professor with the Department of Computer Science and Engineering, International Institute of Information Technology at Bhubaneswar, Bhubaneswar, India. He has authored several journal and international conference papers. His research interests include information security, Internet of Things, and wireless sensor networks.



ZHI XUE is currently a Professor with the Electronic Information and Electrical Engineering Associate Dean, School of Cyber Security, Shanghai Jiao Tong University, China. His research interests include wireless network security, cloud security, cryptography, and cyber threat intelligence.

ALBERT Y. ZOMAYA (M'90–SM'97–F'04) was an Australian Research Council Professorial Fellow from 2010 to 2014. He is currently a Chair Professor of high performance computing and networking with the School of Information Technologies, The University of Sydney. He is also the Director of the Centre for Distributed and High Performance Computing which was established in late 2009. He has published over 500 scientific papers and articles and has authored, co-authored, or edited over 20 books.

Mr. Zomaya served as the Editor-in-Chief for the IEEE TRANSACTIONS ON COMPUTERS (2011–2014). He currently serves as an Associate Editor for 22 leading journals, such as the *ACM Computing Surveys*, the IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, the IEEE TRANSACTIONS ON CLOUD COMPUTING, and the *Journal of Parallel and Distributed Computing*. He is also the Founding Editor-in-Chief of the IEEE TRANSACTIONS ON SUSTAINABLE COMPUTING. He has delivered over 180 keynote addresses, invited seminars, and media briefings and has been actively involved, in a variety of capacities, in the organization of over 700 national and international conferences.

His research interests lie in parallel and distributed computing, networking, and complex systems. He is a Chartered Engineer, a fellow of AAAS and IET (U.K.), and an IEEE Computer Society's Golden Core Member. He was a recipient of the IEEE Technical Committee on Parallel Processing Outstanding Service Award (2011), the IEEE Technical Committee on Scalable Computing Medal for Excellence in Scalable Computing (2011), and the IEEE Computer Society Technical Achievement Award (2014).