Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2021.018480
Article

Tech Science Press

Resource Management and Task Offloading Issues in the Edge–Cloud Environment

Jaber Almutairi[1] and Mohammad Aldossary[2,*]

[1]Department of Computer Science, College of Computer Science and Engineering, Taibah University, Al-Madinah, Saudi Arabia
[2]Department of Computer Science, College of Arts and Science, Prince Sattam bin Abdulaziz University, Al-Kharj, Saudi Arabia
*Corresponding Author: Mohammad Aldossary. Email: mm.aldossary@psau.edu.sa

Received: 10 March 2021; Accepted: 28 April 2021

Abstract: With the increasing number of Internet of Things (IoT) devices connected to the internet, a platform is required to support the enormous amount of data they generate. Since cloud computing is far away from the connected IoT devices, applications that require low-latency, real-time interaction and high quality of service (QoS) may suffer network delay in using the Cloud. Consequently, the concept of edge computing has appeared to complement cloud services, working as an intermediate layer with computation capabilities between the Cloud and IoT devices, to overcome these limitations. Although edge computing is a promising enabler for issues related to latency sensitivity, its deployment produces new challenges. Therefore, this paper presents and discusses the main factors of service latency for IoT applications, considering the impact of application characteristics (e.g., computation and communication demands), as well as resource utilization and heterogeneity at the edge level. Thus, a number of simulation experiments were set up to evaluate the influence of these factors. The outcomes of this research can be used to understand the complex interactions between many factors that affect the overall service time for latency-sensitive applications. Additionally, several open challenges are highlighted, serving as potential directions for future research.

Keywords: Edge-cloud computing; fog computing; resource management; scheduling; task offloading; internet of thing

1 Introduction

The information technology (IT) sector has developed at a massive rate recently: more than 50 billion devices will be connected to the internet in the coming years [1,2,3], the so-called Internet of Things (IoT) era [4]. This immense growth requires platforms to support the increased number of IoT devices (along with various application domains such as autonomous vehicles, the smart city, and industry 4.0), as well as to organize and process the data produced. IoT devices are limited in terms of power and computational capabilities, i.e., the central processing unit (CPU) and memory [5]. This primarily affects the adoption of computer-intensive applications such as augmented reality (AR), online gaming and the processing of video streaming [6].

Cloud computing is one of the main technologies to support this growth by enabling on-demand access to a vast pool of computation resources for service processes and data analytics [7]. However, since the Cloud is far away from IoT devices, there is an enormous quantity of generated data that needs to be transferred and processed in real time. Applications that require low-latency, real-time interaction and high quality of service (QoS) have suffered from network delay when trying to connect with the Cloud [8]. Further, data are increasingly produced at the edge of the network, which makes it more efficient to process the data at the edge level. It is worth mentioning that cloud computing is not always practical for data processing when the data are produced at the edge and require processing in real time.

Therefore, the concept of edge computing has appeared to complement Cloud services. Edge computing refers to an intermediate layer with computation capabilities between the Cloud and IoT devices to fill the latency gaps. Also, edge computing enables better streaming services that are both latency-sensitive and bandwidth-intensive, such as Google Stadia and Netflix. It avoids the uploading/downloading of massive files and allows the pre-processing of offloading tasks, thus helping to minimize the overall service time.

Edge computing is a promising enabler for latency-sensitive applications because of its closeness to IoT devices. Computational resources in edge computing are similar to those of cloud computing, which consists of a pool of servers operated and managed virtually, as well as hosted at the edge of the network. Thus, efficient Edge–Cloud resource management for latency-sensitive applications is essential to fully utilize the capabilities of edge nodes [9].

A number of studies have been conducted on Edge–Cloud resource management to achieve improvements in the overall service time for IoT applications. Some in particular have focused on latency-sensitive applications, while others have focused on resource utilization and energy efficiency as the main objectives. Thus, there is a clear need to consider the characteristics of offloading IoT applications tasks (e.g., computational demand, communication, and latency), as well as resource parameters, such as resource utilization and resource heterogeneity, in order to meet the required service time for the applications and utilize Edge–Cloud resources efficiently. Improving the utilization of Edge–Cloud resources and reducing dependency on the Cloud, thereby reducing the overall cost, is another challenge.

The aim of this research is to investigate the task offloading issues and the impact of computation and communication demands, as well as resource heterogeneity, in the Edge–Cloud system. This will help to facilitate the development of an approach that aims to improve both applications' QoS and system performance. This paper's main contributions are:

- Quantifying the impact of computation and communication demands of offloading tasks on the overall service time within the Edge–Cloud system.
- Quantifying the effectiveness of resource utilization and resource heterogeneity in terms of the overall end-to-end service time within the Edge–Cloud system.
- Evaluating the impact of offloading task decisions on service performance within the Edge–Cloud system, considering computation and communication demands as well as resource utilization and heterogeneity.

The remainder of this paper is organized as follows. Section 2 presents an overview of cloud computing, edge computing and Edge–Cloud computing architecture, with details of its main components and their interactions. The related computing paradigms, including cloudlets, fog computing, and mobile edge computing, along with a detailed comparison of these paradigms, are also presented in Section 3. Section 4 considers the important factors involved in offloading tasks in an Edge–Cloud environment. Section 5 includes the experimental set-up, results, and discussion, and highlights the key findings. A discussion of the open challenges is presented in Section 6. Finally, Section 7 concludes this paper and discusses future work.

2 Overview

2.1 Cloud Computing

Cloud computing is a computing model that has characteristics to support the services of IoT and applications of big data [7]. Cloud service providers offer flexibility and efficiency to their users with services such as infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS), and software-as-a-service (SaaS) [10]. Also, Cloud service providers offer various deployment models, including public, private, hybrid, and community Clouds, to leverage the benefits of cloud computing [11].

However, studies have consistently shown that cloud computing does not efficiently deal well with latency-sensitive applications such as self-driving cars, healthcare applications, and video gaming [12]. Besides, cloud computing may not be considered an efficient computing model for applications that require mobility support and location awareness. Such applications have sharply increased in quantity and will continue to increase the Cloud's load [13]. As mentioned earlier, the Cloud is far away from the user, and the delay in transferring a huge amount of data (e.g., video with high resolution) to be processed into the Cloud and then back to the edge device may not be efficient [14]. Therefore, a challenging critical issue is that cloud computing should have full responsibility to transmit from the Cloud to the end devices [15]. Such a responsibility could contribute to the increased use of energy as a result of transmitting data over multiple hubs from the end devices to the Cloud, as well as performing all the computations in the Cloud [16]. Despite the success of mobile cloud computing (MCC) in improving mobile computing by moving the computational services from mobile users' devices to the Cloud, this technique overheated the network, as the Cloud is far away from mobile users, and this will lead to a long latency [17]. Thus, some sensitive applications cannot work effectively with the Cloud [18]. Moving massive amounts of data from end devices to the Cloud and vice versa is costly in terms of time and energy. Also, it could be infeasible, owing to the growth in the quantity of data and the number of connected devices. Therefore, recent studies have proposed a system model that aims to overcome these challenges, which is discussed in the next section.

2.2 Edge Computing

Cloud computing, mobile computing, and MCC are revolutionary technologies, but they require hosting the services only in the Cloud, which may be impossible for some applications. Therefore, a new approach has emerged, edge computing.

Several papers [19,20,21] describe edge computing as a technology that aims to push computational services from a centralized data center (the Cloud) to the edge of the network in order to reduce latency and provide real-time interaction, as well as supporting the massive growth of devices connected to the internet. OpenEdge [22] defines Edge computing as computation provided by small data centers located closer to IoT devices at the edge of the network. Edge computing provides all the Cloud services (i.e., computation and storage) in a virtualized manner [20]. The Cloud and the Edge complement each other and have nearly the same functionality to provide computing services. Yet, there are some differences, such as location, support mobility, heterogeneity, and scalability to accommodate a vast number of connected devices (Tab. 1 summarizes these differences).

Edge computing has several advantages that improve the process of distributed systems. Edge computing aims to reduce the load in the Cloud, which in turn reduces the latency and produces faster response times because it reduces the movement of data from the end device to the core of the Cloud [14]. Moreover, according to recent studies [23], edge computing could reduce the energy consumption of the Cloud by up to 40%, a significant benefit in light of current concerns about energy consumption [24]. Also, edge computing provides a vast amount of resources to IoT devices, enabling such devices to become smarter by processing complex tasks in a short time. It is difficult for such devices to handle these tasks on their own because of their resource limitations, such as computational power [25]. Furthermore, regarding the massive increase in the number of devices connected to the internet, Edge computing provides scalability to support these devices and deal with their requests closer to them [23]. Moreover, many current applications demand mobile support, such as connected vehicles, transport applications, and healthcare applications [26].

Table 1: Cloud computing *vs.* edge computing

Features	Cloud computing	Edge computing
Computational Capacity	High	Medium to Low
Latency	High	Low
Mobility Supported	Limited	Supported
Location of Servers	Within the Internet	Close to End Devices
Number of Servers	High	Few
Geographical Distribution	Centralized	Decentralized
Requirements for Applications	Intensive Computational and Delay Tolerant	Latency Sensitivity, Mobility, and High QoS

2.3 Edge–Cloud Computing Architecture

Edge computing consists of several edge nodes that are distributed geographically. These edge nodes follow the same concept as cloud computing but are smaller in size, each node having its computational power, storage, and network. Several studies have called these edge nodes “micro clouds” [27,28] or “micro data centers” [29,30]. In an Edge–Cloud computing environment, there are three main layers, as shown in Fig. 1. The lowest layer contains smart end devices that have limited computational power. Devices in this layer have their functions (e.g., healthcare devices, self-driving cars, sensors, and smartphones). These devices are connected to the middle layer, which has edge nodes. Edge nodes are close to the end device and provide the required computational resources to the end device on demand. Edge nodes aim to reduce the latency of IoT applications and dependence on the Cloud. Moreover, edge nodes have limited computational power of the kind that may be required to collaborate with either the Cloud or other edge nodes. As mentioned earlier, the cloud has enormous computational resources, but it is far away from the end-devices, which causes network delays. In such an environment, the Cloud manages edge nodes and provides assistance if the edge nodes require more computational support or applications.

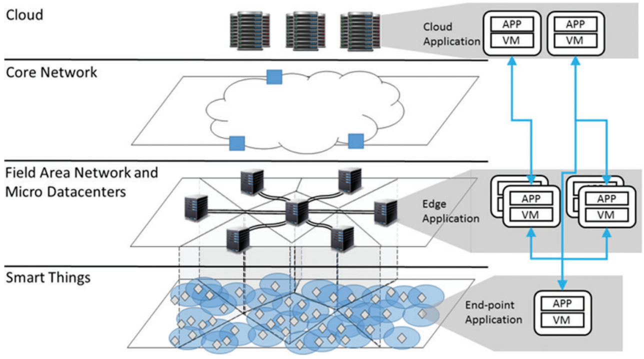


Figure 1: Layers of Edge–Cloud computing architecture [30]

Edge computing is a model that complements the responsibility of the Cloud [31]. Mobile edge computing (MEC), fog computing, and cloudlet are models that have a similar concept [32,33]. Tab. 2 compares these models in terms of architecture and suitable applications [34]. The resources of edge computing can be owned by cloud providers or by any other vendor, such as mobile network providers, university campuses, or coffee shops [35].

Table 2: Comparison of cloudlet, fog computing, and mobile edge computing

Features	Cloudlets/Micro-Cloud	Fog Computing	Mobile Edge Computing
Location	Within the area	Between edge of network and the Cloud	Radio network/Base station
Proximity	Single hop	Single or several hops	Single hop
Devices	Server	Router, access points, server	Servers in the base station
Accessibility	Wi-Fi	Wi-Fi, mobile network, etc.	Mobile network
Application	Suitable for mobile applications that require low latency	IoT	Suitable for applications that require mobility support such as self-driving cars

3 Related Computing Paradigms

3.1 Cloudlets/Micro-Cloud

Cloudlet, first proposed by Carnegie Mellon University, is another model of edge computing, defined by Satyanarayanan as “a trusted, resource-rich computer or cluster of computers that’s well-connected to the Internet and available for use by nearby mobile devices” [36]. In essence, it is a small Cloud that aims to help sensitive applications of users’ mobiles, such as gaming, GPS routing, and internet banking [25]. Cloudlets or Micro-Clouds were devised to push computations from centralized Cloud systems closer to the end-users’ mobile in order to avoid the high latency of an offloading approach [17,33].

The architecture of Cloudlets has three levels. The lowest level is the users’ devices that are connected to the Cloudlet. The middle level connects users’ devices with the Cloud. Thus, users are not required to communicate with the Cloud directly. Cloudlets can communicate with the centralized Cloud for configuration and provisioning [37]. Mobile users can connect to Cloudlets through wireless LAN [36]. Compared to the central Cloud, a Cloudlet is smaller, closer to end-users, saves power and costs, reduces overall latency, and improves the QoS [38,39].

However, the Cloudlets model is not considered scalable in terms of resource provisioning and services. Moreover, it can only be accessed by Wi-Fi, which limits its ability to support other devices that are close to a Wi-Fi area but are not covered by it [40].

3.2 Fog Computing

Fog computing is an extended model of cloud computing, serving the edge of the network. It distributes computing resources such as processing units, storage, and networks in the area between the Cloud and end devices, using the same techniques as the Cloud, such as virtualization and multi-tenancy [41]. Fog computing provides better services to applications and services that do not work effectively with the Cloud. These applications have different attributes (e.g., mobility support and real-time interaction), and different approaches are thus required to work with them [42]. Many applications can benefit at least in part from fog computing, including video conferencing, online gaming, and augmented reality/virtual reality (AR/VR) applications.

The architecture of fog computing consists of four primary levels: the Cloud data center, the core of the network, the edge node, and smart things (IoT) [42]. The intermediate layer, which is the edge node, plays an essential role in supporting the Cloud to reduce the load of computing, storing, and networking and provides the services to end-users with a high QoS. These edge servers are virtualized and can be accessed by connected devices through wired or wireless connections. The edge server connects to the Cloud to collaborate in providing some services [43].

As described earlier, fog computing supports various types of applications, particularly those applications that require real-time analysis and interaction [44]. Thus, the majority of IoT applications are supported by fog computing (e.g., smart home, healthcare, smart factories, and agriculture) [45].

3.3 Mobile Edge Computing

Mobile edge computing (MEC), also known as multi-access edge computing, is defined by the European Telecommunications Standards Institute (ETSI) as a technology that "provides an IT service environment and cloud-computing capabilities at the edge of the mobile network, within the Radio Access Network (RAN) and in close proximity to mobile subscribers" [46].

As previously stated, the requirements of mobile applications are changing, and new network technologies have appeared, such as 5G. Thus, redesigning the network or the way in which it provides services is essential [47]. For example, video streaming in the area of smart cities requires a proper network to carry a huge amount of data. Also, the edge server deals with these data near the source [38]. One main difference between MEC and fog computing is that the former can provide services to connected users without communicating with the Cloud in some applications [48].

Previous works [40,49] list some main characteristics of MEC. The location of the edge server can be accessed within the range of the RAN to provide real-time interaction. These edge servers are distributed geographically to support large systems. Thus, MEC has been recognized as the base for latency-sensitive applications (e.g., video streaming), also providing support for IoT mobility and location awareness (e.g., smart vehicles).

Tab. 2 presents a comparison of the three computing paradigms. The site deployment of Cloudlet and MEC can be at the first single hop; for example, Cloudlet can be located indoors (e.g., in a shopping center, hospital, etc.), and the MEC server can be embedded into the telecoms base station [50]. In contrast, the deployment of fog computing can be anywhere between the IoT devices and the Cloud. The concept of fog computing is used widely in the applications of smart cities and smart grids [41]. MEC is usually used in the area of applications that require mobility, such as autonomous vehicles, and supporting communications vehicle-to-vehicle (V2V), as well as vehicle-to-infrastructure (V2I) [51,52]. From the research perspective, all of the above terms have the same concepts, which push the computational service to the end of the network; however, each vendor (e.g., Cisco and Juniper) argues that its devices (e.g., routers and switches) are the perfect platforms to host Edge–Cloud capabilities. On the other hand, telecoms companies argue that their base stations and 4G/5G can be hosting the Edge–Cloud capabilities [53].

4 Offloading Tasks in the Edge–Cloud Environment: Important Factors

From the application perspective, there are several factors that affect the decision about offloading tasks and the overall service time for IoT applications. For example, latency-sensitive applications could lead to a significant delay in computation and communication, which should be considered in efficiently managing Edge–Cloud resources. The following are some of the main factors.

- **Application characteristics**: This is when there are some tasks that are working jointly, such as direct acyclic graph (DAG). In this case, if the resource manager offloads the tasks in different locations, the communication time will increase [54,55]. Therefore, considering the impact of task dependency could improve the application's QoS.
- **Application tasks variation**: In general, any IoT application consists of several tasks, which vary in their functionality; thus, they will have different demands of the CPU or the amount of transferred data. The place where the task is offloaded will therefore significantly affect the service time.
- **Types of computational resources**: Edge–Cloud resources are heterogeneous, having either different hardware capabilities or different hardware architecture, e.g., graphic processing units (GPUs) and field programmable gate arrays (FPGAs). Therefore, the resource manager needs to assign the tasks to the most appropriate hardware to get the best performance out of the resources.
- **Users' mobility**: Since some IoT applications require mobility support, when the task is offloaded to a local edge node, the IoT device may move to another area covered by a different edge node. This could lead to a significant degradation in service time [56].

In this paper, we particularly focus on the impact of task variations in terms of computational and communication demands of IoT latency-sensitive applications, as well as the offloading of tasks to heterogeneous computational resources.

4.1 Application Characteristics (Computational and Communication)

The computation tasks of IoT applications can be characterized by their needs for computational resources (i.e., CPU and RAM), as well as for communication (e.g., uploading and downloading data). IoT offloaded tasks usually vary in the degree of reliance on such resources between light and heavy. Low computation and communication demands include healthcare applications [57], whereas high computation and communication demands are involved in online video gaming [58]. As depicted in Fig. 2, some tasks require more computation time owing to the intensive processing, and others require more network time owing to the transfer of a massive amount of data.

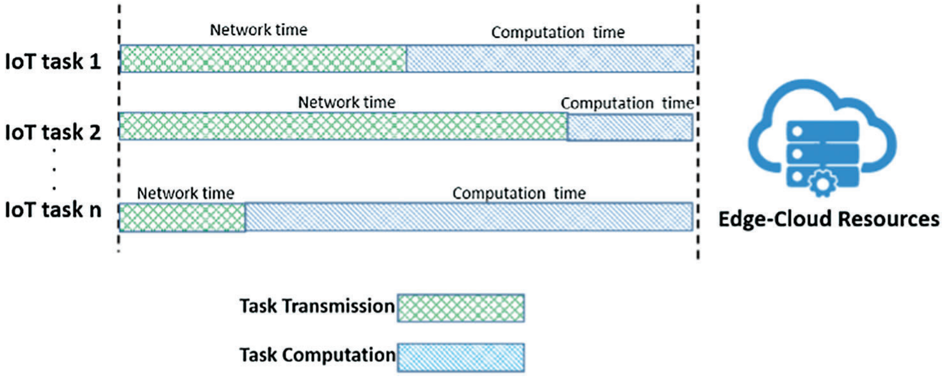


Figure 2: Task variations (computation and communication)

Tasks involved in IoT latency-sensitive applications that require higher computational demands should for preference be processed in the Cloud, since the edge resources are limited, but this also depends on the demands of the transferred data. Conversely, tasks that require moving a large amount of data need to be processed in the Edge to avoid the long latency in the network backhaul.

Basically, the task completion time consists of three essential components, namely computation time, network time, and the site where the task is scheduled (e.g., which server type). The server can be located on the local edge, at other nearby edge nodes close to IoT devices, or in the Cloud.

The computation time of the IoT task depends on the number of instructions, measured as million instructions per second (MIPS), that need to be executed and the processing speed of the hosted resources, e.g., virtual machine (VM). The number of instructions represents the computational volume of an IoT task. As mentioned earlier, IoT tasks can range from a small number of code instructions to a high number, depending on the application. This factor, along with network conditions, determines where the tasks should be offloaded. For example, it is not logical to offload the tasks with vast amounts of data to the Cloud when edge resources are available, because this will increase the overall service time. However, these two factors must be considered together; thus, we need to understand and investigate the impact of each independently. The network time of an IoT task depends on the amount of data to be uploaded and downloaded, as well as the transmission latency between the sender and the receiver. In our case, the sender will be the IoT devices, and the receiver could be (local edge, other collaborative edges or the Cloud). Moreover, for each task, the amount of transferred data can vary based on the IoT application.

4.2 Computational Resource Heterogeneity

In terms of computational resources, both IoT devices and edge servers are heterogeneous. Consequently, for latency-sensitive applications, the computational resource needed to execute the task must be estimated in order to minimize the overall service time [59]. Because of resource heterogeneity, this means there are some servers that are better than others in terms of capabilities, notably in handling the offloading tasks faster (Fig. 3). The quantity of required computational resources varies for each task. Thus, heavy tasks require a powerful machine to process the jobs more quickly.

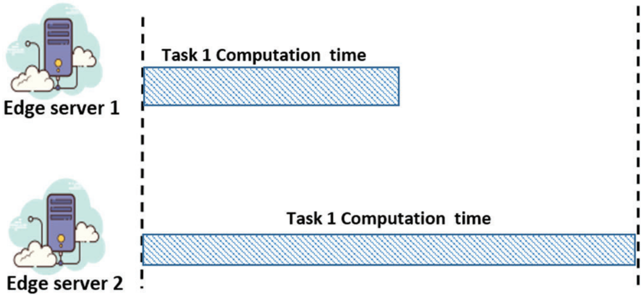


Figure 3: Computation time for heterogeneous resources

Given that a performance method is needed that measures the end-to-end IoT service effectiveness, considering both the computational and communication demands of offloading tasks, the following questions need to be answered:

- How do different applications' characteristics, including computation and communication demands, impact on the overall service time?
- How do different computational resources (e.g., different VM capabilities) impact on the overall service time?

5 Implementation

A number of simulation experiments were conducted on the EdgeCloudSim in order to understand and investigate the impact of computational and communication demands of IoT tasks, as well as the impact of different computational resource capabilities. These experiments used different IoT tasks with a range of demands, e.g., the length of different tasks in MIPS and different quantities of data to be uploaded and downloaded in megabyte (MB).

In order to investigate the impact of resource heterogeneity, two different VMs on EdgeCloudSim were considered. Tab. 3 shows the configurations of the VMs that were considered in the experiments. These configurations are based on Rackspace, which provides a wide range of VM types [60,61]. The first type of VMs has two cores, Intel Xen CPU, and the second type of VMs has four cores.

Table 3: Configurations of the VMs

	CPU core	MIPS	RAM (GB)	Storage (GB)
VM Type 1	2	10000	2000	50000
VM Type 2	4	20000	4000	100000

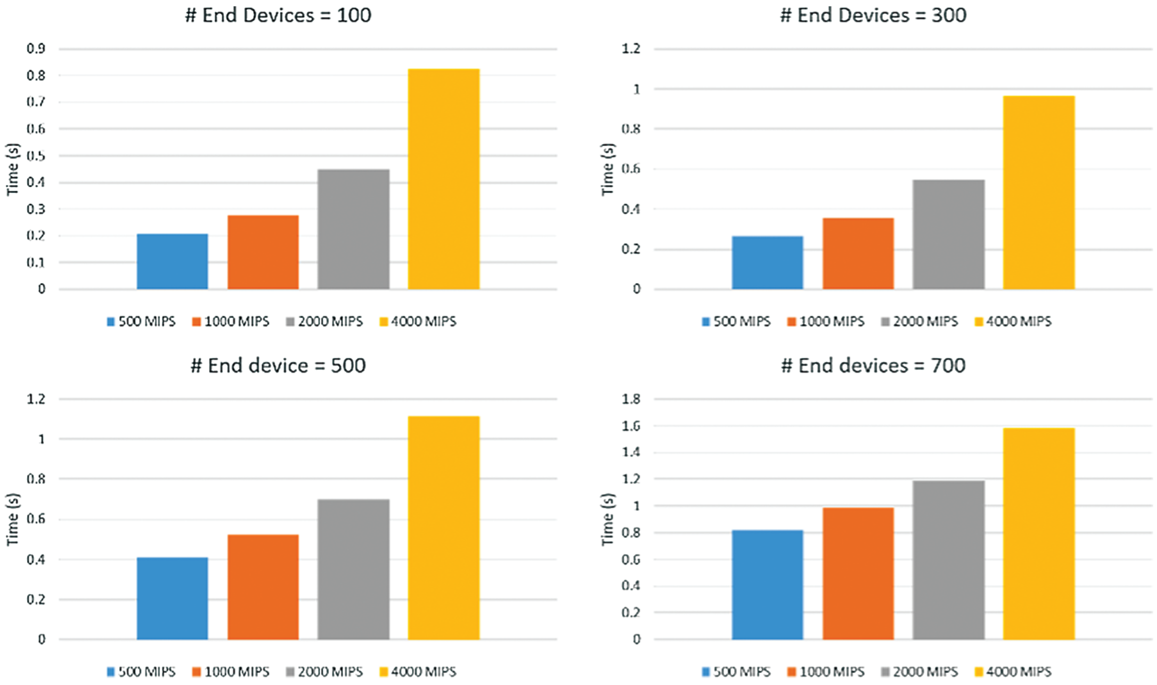
5.1 Experimental Set-Up

The overall aim of the experiments was to investigate and understand the impact of the change in the computational and communication demands of the IoT tasks, as well as the effectiveness of resource heterogeneity in terms of the overall end-to-end service time. Several simulation experiments were conducted using different IoT offloaded tasks.

The simulation key parameters are presented in Tab. 4, including the number of IoT devices, the number of edge nodes, and the number of VMs. To mimic various applications that might be encountered in practice, we defined the configuration of tasks by varying communication bandwidth demand from 0.25 MB to 1 MB (with increasing of 0.25 MB in each stage), and doubling computation demands starting from 500 MIPS and rising to 4000 MIPS. These numbers have been used in similar related work in the literature to represent offloaded tasks [62]. Also, we did a sensitive analysis of the selected parameters similar to the work presented in [63]. First, we maintained task communication as a constant parameter and varied the tasks' computational demand to study the impact of that demand. Then, we increased the communication demand, keeping the computational demand constant, to investigate the communication demand. The impact of computational demands and communication demands are presented in Figs. 4 and 5, respectively. Moreover, we ran the same IoT workload with two different types of VMs, as presented in Tab. 3.

Table 4: Key parameters of the simulation environment

Parameters	Values
Simulation Time	30 minutes
Warm-up Period	3 minutes
Number of Iterations	5
Number of IoT Devices	100–1000
Number of Edge Nodes	3
Number of VM per Edge Server	8

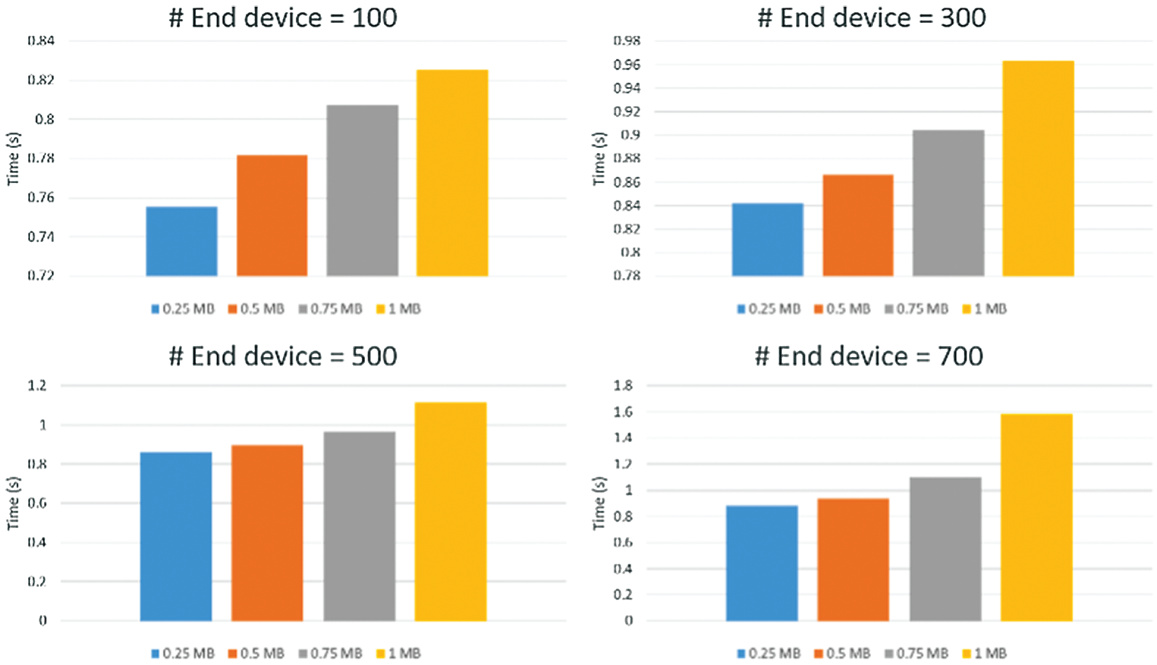

End Devices = 100

End Devices = 300

End device = 500

End devices = 700

Figure 4: The impact of computation demand


End device = 100

End device = 300

End device = 500

End device = 700

Figure 5: The impact of communication demand

5.2 Results and Discussion

This section analyses how the size of the demanded resources (CPU and network) of IoT tasks influences the overall service time for various IoT devices. In order to measure the computational demands, we fixed the amount of communication demand and tried different types of task computation, and vice versa for the task communication demand. Furthermore, we validated the service time performance and utilization for two different types of VMs.

Fig. 4 shows the average service time for offloading tasks with different computational demands (e.g., 500 MIPS, 1K MIPS, 2K MIPS, and 4K MIPS) for a different number of IoT devices. As illustrated in Fig. 4, no matter how many IoT devices there were, the average service time for IoT applications showed a corresponding increase with an increment in their CPU requirements; the fewer the number of end devices, the more obvious was the fluctuation. For example, the end-to-end service time of a 4K MIPS task was about four times that of a task with 500 MIPS when the number of mobile end devices equaled 100, but only nearly twice that for a task when the number was 700. Intuitively, the reason was that computation resources were severely limited, and, when the demand for CPU increased, the time of waiting and processing in the CPU would also rise correspondingly. However, once the number of tasks increased to a certain value, the conflict of CPU resources (means Clock Cycles) would increase slowly, as this would be near to the maximum CPU capacity.

Additionally, Fig. 5 shows the average service time for offloading tasks with different communication demands (e.g., 0.25 MB, 0.5 MB, 0.75 MB, and 1 MB) for a different number of IoT devices. As shown in Fig. 5, when the bandwidth demand of the task varied, the service time only slightly increased because the network bandwidth was not a critical limit for IoT tasks in the present experiments. In other words, the network resource or performance was notably sufficient to handle nearly all IoT tasks. It should be noted that, when the number of end devices is close to 700, the increment becomes obvious, and efficient assignment of network resources will play a meaningful role in the end-to-end service time.

As the VMs are heterogeneous in terms of size, they consequently have different processing times for IoT offloaded tasks, the variation fairly corresponding to their size. In the beginning (Fig. 6), when the number of IoT devices was 100, the average service time of IoT offloaded tasks processed in VM type 1 was double the service time in VM type 2. Further, the experiments revealed a huge increase in service time when the number of IoT devices increased for the VM with low capabilities, while the VM type 2 handled the increase in the number of IoT devices efficiently.

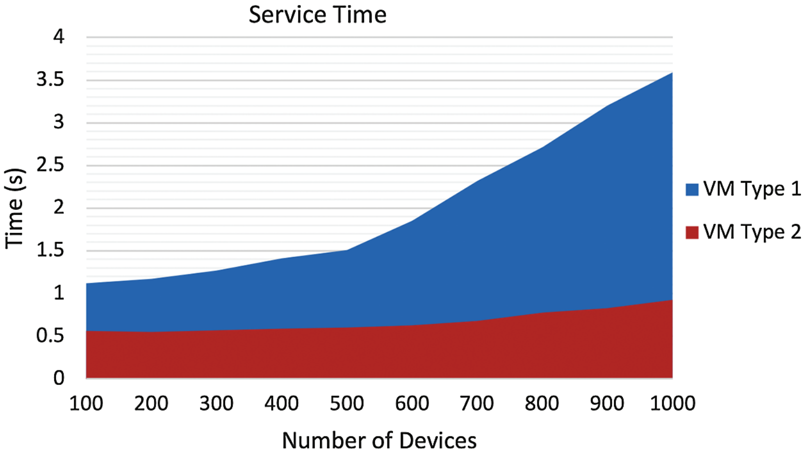


Figure 6: The impact of two different VMs

Since the VMs are heterogeneous, the average utilization of VMs was different. As shown in Fig. 7, when the number of IoT devices was 100, both VM type 1 and VM type 2 had the same utilization level. However, when the number of IoT devices increased to around 1000, the average server utilization of a

VM type 1 was about twice that of a VM type 2. The reason is that VM type 2 has more capability than VM type 1 and can thus process more IoT offloaded tasks with an acceptable level of utilization.

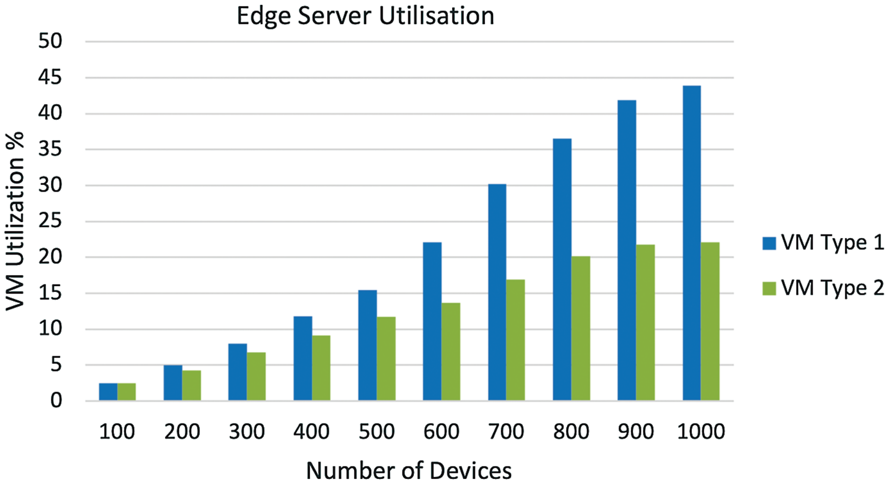


Figure 7: Server utilization of two different VMs

Possible explanations that we can discover when comparing the two results are presented in Figs. 6 and 7, there being a correlation between the processing time of the edge server and the utilization level. Because both tend to have the same level of increase for both VM type 1 and VM type 2, this effect may be due to the rise in the computational load and the IoT devices sharing the same resources. Also, it is possible that, if we increased the load for VM type 2, we would get the same results as for VM type 1.

5.3 Key Findings

In this section, we outline some findings from our simulation-based evaluation, which we hope can be used to improve the efficiency of task offloading and achieving well-balanced resource management in the Edge–Cloud environment. The experiments on EdgeCloudSim showed the impact of tasks' computational demand and communication demand, as well as the effect on overall service time for IoT applications. Some studies [64] emphasize that the central Cloud could overcome any problems of network delay, thanks to its tremendous resources. On the other hand, other studies [65] show that the increased network demand could lead to an exponential delay for some applications. Our simulation results showed that network time had a significant impact on the overall service time; thus, considering this parameter could lead to an improvement in performance. In present experiments, the impact of computational demand affected the overall service time more than the communication demand. For example, when the number of IoT devices was 700, the service time increased by around 0.2 seconds when the amount of computational demand increased from 500 MIPS to 1000 MIPS. In contrast, when the amount of communication demand increased from 0.25 MB to 0.5 MB, the overall service time increased by 0.03 seconds. These results seem to be consistent with other research [66], which found that, when the Edge–Cloud becomes overloaded, there will be degradation in execution performance owing to resources competition and sharing.

Furthermore, the experiments showed that the measured overall service time for the two types of VMs had a clear impact on the performance when the system load was increased. Previous studies on resource heterogeneity have similarly demonstrated that VM diversity results in variation in application performance. Based on experimental results conducted on Amazon EC2, a large VM can enhance the service time performance up to 40%, and, for some specific applications, up to 60% [67]. Moreover, when the server utilization increases, the overall service time is sharply increased. This finding was also reported by [68], showing that, when the amount of computational workload increased, server utilization was increased and the service time performance was affected. This result is expected, and the main purpose of the present simulations was to demonstrate the necessity of considering the resource utilization level in the process of scheduling offloading tasks in order to minimize the overall service time.

6 Open Challenges

As far as offloading tasks in the Edge–Cloud environment is concerned, there are several open challenges remaining, some of which are discussed in this section.

Task dependency: There is a lack of studies on the issue of offloading tasks because they do not consider the dependency of the tasks. To be more precise, allocating tasks (that are dependent on other tasks' results) to different resources in the Edge–Cloud environment could lead to poor QoS for IoT applications. Thus, research is required into the application components and how they interact with one another. Taking this factor into account may help to enhance both overall system performance and to yield application QoS.

Applications require a high degree of mobility: Offloading tasks of applications that require mobility support, such as self-driving cars, crewless aircraft vehicles, and mobile devices, is an open challenge. For example, processing users' tasks of an application, while moving from a covered area to another covered area, could lead to high network latency or process failure [6]. Although several researchers are attempting to tackle this issue, it remains a challenge.

Workload prediction: IoT tasks are dynamically changing and each task's procedure may have different execution times. Also, IoT devices are mobile, and the number of devices may increase in some areas; thus, the workload will be increased for the connected edge node. In turn, the amount of IoT workload will change dynamically over the Edge–Cloud system, which could lead to service performance degradation. Therefore, there is a need for workload prediction modeling, which could help to yield application QoS and maintain the performance of the Edge–Cloud system.

7 Conclusion and Future Work

This paper has presented a comprehensive review of fundamental concepts in resource management and task offloading in the Edge–Cloud environment. It started by defining the core concepts of cloud computing and describing in detail its architecture and deployment models. Mobile cloud computing was also discussed as an extended model of cloud computing. The main idea behind the transformation to edge computing was introduced, along with an explanation and comparison of the related technologies, including fog computing, mobile edge computing, and cloudlets. The work was positioned in the relevant literature, focusing on the main factors of service latency for IoT applications and considering the impact of application characteristics such as computation and communication demands as well as resource utilization and heterogeneity in the Edge–Cloud environment. The paper reported on a number of simulation experiments that were set up to evaluate the influence of these factors on the performance of IoT applications in terms of service time and resource utilization. Finally, numerous open challenges and potential research directions were highlighted.

As a part of future work, we intend to extend our approach by considering some of the presented open challenges of offloading tasks in the Edge–Cloud environment to further enhance the capability of the proposed work.

Acknowledgement: The authors would like to acknowledge the Deanship of Scientific Research, Taibah University, Al-Madinah, Saudi Arabia, for providing research resources and equipment. In addition, the authors would like to thank the Deanship of Scientific Research, Prince Sattam bin Abdulaziz University, Al-Kharj, Saudi Arabia, for supporting this work.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] D. Evans, “The internet of things: How the next evolution of the internet is changing everything,” *CISCO white paper*, 2011.

[2] A. Zaslavsky, C. Perera and D. Georgakopoulos, “Sensing as a service and big data,” in *Proc. of the Int. Conf. on Advances in Cloud Computing*, Bangalore, India, pp. 21–29, 2012.

[3] L. M. Vaquero and L. Rodero-Merino, “Finding your way in the fog: Towards a comprehensive definition of fog computing,” *Computer Communication Review*, vol. 44, no. 5, pp. 27–32, 2014.

[4] J. Gubbi, R. Buyya, S. Marusic and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[5] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali *et al.,* “All one needs to know about fog computing and related edge computing paradigms: a complete survey,” *Journal of Systems Architecture*, vol. 98, pp. 289–330, 2019.

[6] S. Shekhar and A. Gokhale, “Dynamic resource management across cloud-edge resources for performance-sensitive applications,” in *Proc. of the 17th IEEE/ACM Int.. Sym. on Cluster, Cloud and Grid Computing*, Madrid, Spain, pp. 707–710, 2017.

[7] P. Mell and T. Grance, “The NIST Definition of Cloud Computing,” Gaithersburg, MD, 2011.

[8] Y. Sahni, J. Cao, S. Zhang and L. Yang, “Edge mesh: A new paradigm to enable distributed intelligence in internet of things,” *IEEE Access*, vol. 5, pp. 16441–16458, 2017.

[9] R. Mahmud, K. Ramamohanarao and R. Buyya, “Application management in fog computing environments: A taxonomy, review and future directions,” *ACM Computing Surveys*, vol. 53, no. 4, pp. 1–43, 2020.

[10] T. Dillon, C. Wu and E. Chang, “Cloud computing: issues and challenges,” in *Proc. of the Int. Conf. on Advanced Information Networking and Applications*, Perth, WA, Australia, pp. 27–33, 2010.

[11] C. Gong, J. Liu, Q. Zhang, H. Chen and Z. Gong, “The characteristics of cloud computing,” in *Proc. of the Int. Conf. on Parallel Processing Workshops*, San Diego, CA, USA, pp. 275–279, 2010.

[12] Y. Jararweh, A. Doulat, O. Alqudah, E. Ahmed, M. Al-Ayyoub *et al.,* “The future of mobile cloud computing: Integrating cloudlets and mobile edge computing,” in *Proc. of the 23rd Int. Conf. on Telecommunications*, Thessaloniki, Greece, pp. 1–5, 2016.

[13] X. He, Z. Ren, C. Shi and J. Fang, “A novel load balancing strategy of software-defined cloud/fog networking in the internet of vehicles,” *China Communications*, vol. 13, pp. 140–149, 2016.

[14] M. Firdhous, O. Ghazali and S. Hassan, “Fog computing: will it be the future of cloud computing?,” in *Proc. of the Third Int. Conf. on Informatics & Applications*, Kuala Terengganu, Malaysia, pp. 8–15, 2014.

[15] Z. Sanaei, S. Abolfazli, A. Gani and R. Buyya, “Heterogeneity in mobile cloud computing: taxonomy and open challenges,” *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 369–392, 2014.

[16] F. Jalali, K. Hinton, R. Ayre, T. Alpcan and R. S. Tucker, “Fog computing may help to save energy in cloud computing,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1728–1739, 2016.

[17] T. Verbelen, P. Simoens, F. De Turck and B. Dhoedt, “Cloudlets: bringing the cloud to the mobile user,” in *Proc. of the 3rd ACM Workshop on Mobile Cloud Computing and Services*, Lake District, UK, pp. 29–35, 2012.

[18] W. Shi, J. Cao, Q. Zhang, Y. Li and L. Xu, “Edge computing: vision and challenges,” *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[19] P. G. Lopez, A. Montresor, D. Epema, A. Datta, T. Higashino *et al.,* "Edge-centric computing: Vision and challenges," *Computer Communication Review*, vol. 45, no. 5, pp. 37–42, 2015.

[20] M. Satyanarayanan, "The emergence of edge computing," *IEEE Computer Society*, vol. 50, no. 1, pp. 30–39, 2017.

[21] R. Want, B. N. Schilit and S. Jenson, "Enabling the internet of things," *IEEE Computer Society*, vol. 48, no. 1, pp. 28–35, 2015.

[22] OpenEdge, "OpenEdge application development," 2021. [Online]. Available: https://www.progress.com/openedge.

[23] W. Shi and S. Dustdar, "The promise of edge computing," *IEEE Computer Society*, vol. 49, no. 5, pp. 78–81, 2016.

[24] K. Djemame, R. Kavanagh and D. Armstrong, "Energy efficiency support through intra-layer cloud stack adaptation," in *Proc. of the 13th Int. Conf. on Economics of Grids, Clouds, Systems and Services*, Athens, Greece, pp. 129–143, 2017.

[25] T. Yaofeng, D. Zhenjiang and Y. Hongzhang, "Key technologies and application of edge computing," *ZTE Communications*, vol. 15, no. 2, pp. 26–34, 2017.

[26] A. C. Baktir, A. Ozgovde and C. Ersoy, "How can edge computing benefit from software-defined networking: a survey, use cases, and future directions," *IEEE Communications Surveys and Tutorials*, vol. 19, no. 4, pp. 2359–2391, 2017.

[27] Y. Elkhatib, B. Porter, H. B. Ribeiro, M. F. Zhani, J. Qadir *et al.,* "On using micro-clouds to deliver the fog," *IEEE Internet Computing*, vol. 21, no. 2, pp. 8–15, 2017.

[28] S. Wang, K. Chan, R. Urgaonkar, T. He and K. K. Leung, "Emulation-based study of dynamic service placement in mobile micro-clouds," in *Proc. of the IEEE Military Communications Conference*, Tampa, FL, USA, pp. 1046–1051, 2015.

[29] I. Petri, O. Rana, J. Bignell and N. Auluck, "Incentivising resource sharing in edge computing applications," in *Proc. of the 13th Int. Conf. on Economics of Grids, Clouds, Systems, and Services*, Athens, Greece, pp. 204–215, 2017.

[30] J. Xu, B. Palanisamy, H. Ludwig and Q. Wang, "Zenith: utility-aware resource allocation for edge computing," in *Proc. of the IEEE 1st Int. Conf. on Edge Computing*, Honolulu, HI, USA, pp. 47–54, 2017.

[31] R. Deng, R. Lu, C. Lai and T. H. Luan, "Towards power consumption-delay tradeoff by workload allocation in cloud-fog computing," in *Proc. of the IEEE Int. Conf. on Communications-Mobile and Wireless Networking Symposium*, London, UK, pp. 3909–3914, 2015.

[32] W. Hu, Y. Gao, K. Ha, J. Wang, B. Amos *et al.,* "Quantifying the impact of edge computing on mobile applications," in *Proc. of the 7th ACM SIGOPS Asia-Pacific Workshop on Systems*, Hong Kong, China, pp. 1–8, 2016.

[33] B. Varghese, N. Wang, S. Barbhuiya, P. Kilpatrick and D. S. Nikolopoulos, "Challenges and opportunities in edge computing," in *Proc. of the IEEE Int. Conf. on Smart Cloud*, New York, NY, USA, pp. 20–26, 2016.

[34] K. Dolui and S. K. Datta, "Comparison of edge computing implementations: fog computing, cloudlet and mobile edge computing," in *Proc. of the Global Internet of Things Summit*, Geneva, Switzerland, pp. 1–6, 2017.

[35] S. Yi, C. Li and Q. Li, "A survey of fog computing: concepts, applications and issues," in *Proc. of the Int. Sym. on Mobile Ad Hoc Networking and Computing*, Hangzhou, China, pp. 37–42, 2015.

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

[37] A. Bahtovski and M. Gusev, "Cloudlet challenges," in *Proc. of the 24 Int. Sym. on Intelligent Manufacturing and Automation*, Zadar, Croatia, pp. 704–711, 2014.

[38] A. Sathiaseelan, A. Lertsinsrubtave, A. Jagan, P. Baskaran and J. Crowcroft, "Cloudrone: micro clouds in the sky," in *Proc. of the 2nd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, Singapore, Singapore, pp. 41–44, 2016.

[39] B. Varghese and R. Buyya, "Next generation cloud computing: new trends and research directions," *Future Generation Computer Systems*, vol. 79, pp. 849–861, 2018.

[40] Y. Mao, C. You, J. Zhang, K. Huang and K. B. Letaief, "A survey on mobile edge computing: the communication perspective," *IEEE Communications Surveys and Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[41] F. Bonomi, R. Milito, J. Zhu and S. Addepalli, “Fog computing and its role in the internet of things,” in *Proc. of the 1st ACM Mobile Cloud Computing Workshop*, Hong Kong, China, pp. 13–15, 2012.

[42] F. Bonomi, R. Milito, P. Natarajan and J. Zhu, “Fog computing: a platform for internet of things and analytics,” in *Proc. of the Big Data and Internet of Thing*, New York, NY, USA, pp. 169–186, 2014.

[43] T. H. Luan, L. Gao, Z. Li, Y. Xiang and L. Sun, “Fog computing: Focusing on mobile users at the edge,” *eprint arXiv:1502.01815*, pp. 1–11, 2015.

[44] I. Stojmenovic and S. Wen, “The fog computing paradigm: scenarios and security issues,” in *Proc. of the Federated Conference on Computer Science and Information Systems*, Warsaw, Poland, pp. 1–8, 2014.

[45] A. Al-fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, “Internet of things: A survey on enabling technologies, protocols, and applications,” *IEEE Communication Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[46] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher and V. Young, “Mobile edge computing a key technology towards 5G,” *ETSI White Paper*, vol. 11, no. 11, pp. 1–16, 2015.

[47] E. Ahmed and M. H. Rehmani, “Mobile edge computing: opportunities, solutions, and challenges,” *Future Generation Computer Systems*, vol. 70, pp. 59–63, 2016.

[48] R. Mahmud, R. Kotagiri and R. Buyya, “Fog computing: a taxonomy, survey and future directions,” in *Internet of Everything: Algorithms, Methodologies, Technologies and Perspectives*, Singapore: Springer, pp. 103–130, 2018.

[49] M. Al-ayyoub, Y. Jararweh, A. Doulat, O. Alqudah, E. Ahmed *et al.,* “The future of mobile cloud computing: integrating cloudlets and mobile edge computing,” in *Proc. of the 23rd Int. Conf. on Telecommunications*, Thessaloniki, Greece, pp. 760–764, 2016.

[50] R. Roman, J. Lopez and M. Mambo, “Mobile edge computing, fog et al.: a survey and analysis of security threats and challenges,” *Future Generation Computer Systems*, vol. 78, pp. 680–698, 2018.

[51] C. M. Huang, M. S. Chiang, D. T. Dao, W. L. Su, S. Xu *et al.,* “V2V data offloading for cellular network based on the software defined network (SDN) inside mobile edge computing (MEC) architecture,” *IEEE Access*, vol. 6, pp. 17741–17755, 2018.

[52] M. Emara, M. C. Filippou and D. Sabella, “MEC-assisted end-to-end latency evaluations for C-V2X communications,” in *Proc. of the European Conference on Networks and Communications*, Ljubljana, Slovenia, pp. 157–161, 2018.

[53] G. I. Klas, “Fog computing and mobile edge cloud gain momentum open fog consortium, ETSI MEC and cloudlets,” *White Paper*, vol. 1, pp. 1–14, 2015.

[54] G. L. Stavrinides and H. D. Karatza, “A hybrid approach to scheduling real-time IoT workflows in fog and cloud environments,” *Multimedia Tools and Applications*, vol. 78, no. 17, pp. 24639–24655, 2018.

[55] E. Renart, J. Diaz-montes and M. Parashar, “Data-driven stream processing at the edge,” in *Proc. of the IEEE 1st Int. Conf. on Fog and Edge Computing*, Madrid, Spain, pp. 1–10, 2017.

[56] K. Ha, Y. Abe, Z. Chen, W. Hu, B. Amos *et al.,* “Adaptive VM handoff across cloudlets,” *Technical Report-CMU-CS-15-113*, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1–25, 2015.

[57] A. H. Sodhro, Z. Luo, A. K. Sangaiah and S. W. Baik, “Mobile edge computing based QoS optimization in medical healthcare applications,” *International Journal of Information Management*, vol. 45, pp. 308–318, 2019.

[58] S. Choy, B. Wong, G. Simon and C. Rosenberg, “A hybrid edge-cloud architecture for reducing on-demand gaming latency,” *Multimedia Systems*, vol. 20, no. 5, pp. 503–519, 2014.

[59] X. Lyu, H. Tian, L. Jiang, A. Vinel, S. Maharjan *et al.,* “Selective offloading in mobile edge computing for the green internet of things,” *IEEE Network*, vol. 32, no. 1, pp. 54–60, 2018.

[60] V. Scoca, A. Aral, I. Brandic, R. De Nicola and R. B. Uriarte, “Scheduling latency-sensitive applications in edge computing,” in *Proc. of the 8th Int. conf. on Cloud Computing and Services Science*, Funchal, Madeira, Portugal, pp. 158–168, 2018.

[61] M. Aldossary and K. Djemame, “Performance and energy-based cost prediction of virtual machines auto-scaling in clouds,” in *Proc. of the 44th Euromicro Conference on Software Engineering and Advanced Applications*, Prague, Czech Republic, pp. 502–509, 2018.

[62] T. G. Rodrigues, K. Suto, H. Nishiyama and N. Kato, “Hybrid method for minimizing service delay in edge cloud computing through VM migration and transmission power control,” *IEEE Transactions on Computers*, vol. 66, no. 5, pp. 810–819, 2017.

[63] Y. Yin, L. Chen, Y. Xu, J. Wan, H. Zhang *et al.,* “QoS prediction for service recommendation with deep feature learning in edge computing environment,” *Mobile Networks and Applications*, vol. 25, no. 2, pp. 391–401, 2020.

[64] H. Tan, Z. Han, X. Y. Li and F. C. M. Lau, “Online job dispatching and scheduling in edge-clouds,” in *Proc. of the IEEE Conference on Computer Communications*, Atlanta, GA, USA, pp. 1–9, 2017.

[65] R. Beraldi, A. Mtibaa and H. Alnuweiri, “Cooperative load balancing scheme for edge computing resources,” in *Proc. of the 2nd Int. Conf. on Fog and Mobile Edge Computing*, Valencia, Spain, pp. 94–100, 2017.

[66] L. Liu, Z. Chang, X. Guo, S. Mao and T. Ristaniemi, “Multiobjective optimization for computation offloading in fog computing,” *Proc. of the IEEE Symposium on Computers and Communications*, vol. 5, no. 1, pp. 283–294, 2017.

[67] Z. Ou, H. Zhuang, J. K. Nurminen, A. Ylä-Jääski and P. Hui, “Exploiting hardware heterogeneity within the same instance type of amazon EC2,” in *Proc. of the 4th Workshop on Hot Topics in Cloud Computing*, Boston, MA, USA, pp. 4–8, 2012.

[68] T. Tran and D. Pompili, “Joint task offloading and resource allocation for multi-server mobile-edge computing networks,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 856–868, 2019.