IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 August 2022, accepted 7 September 2022, date of publication 12 September 2022, date of current version 20 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3205741

RESEARCH ARTICLE

LBRO: Load Balancing for Resource Optimization in Edge Computing

MUHAMMAD ZIAD NAYYER[1,2], IMRAN RAZA[2], (Member, IEEE), SYED ASAD HUSSAIN[2], (Member, IEEE), MUHAMMAD HASAN JAMAL[2], ZEESHAN GILLANI[2], SOOJUNG HUR[3], AND IMRAN ASHRAF[3]

[1]Department of Computer Science, GIFT University, Gujranwala 52250, Pakistan
[2]Communication and Network Research Center, Department of Computer Science, COMSATS University Islamabad, Lahore Campus, Lahore 54000, Pakistan
[3]Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 38544, South Korea

Corresponding authors: Soojung Hur (sjheo@ynu.ac.kr) and Imran Ashraf (ashrafimran@live.com)

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant NRF-2021R1A6A1A03039493.

ABSTRACT Mobile cloud computing and edge computing-based solutions provide means to offload tasks for resource-limited mobile devices. Mobile cloud computing provides remote cloud solutions while edge computing provides closer proximity-based solutions. Remote cloud solutions suffer from network latency and limited bandwidth challenges due to distance and dependency on the Internet. However, these challenges are addressed by edge-based solutions since the edge node is available in the same network. The use of Internet of Things-based solutions considering future Information Communication Technology infrastructure is on the rise resulting in the massive growth of digital equipment increasing the load at edge devices. Hence, some load balancing mechanism is required at the edge level to avoid resource congestion. The load balancing at the edge must consider the user's preferences about edge resources such as personal computers or mobile devices. A user must declare which resources can be spared for other devices to avoid overprovisioning essential resources. We present Load Balancing for Resource Optimization (LBRO), a collaborative cloudlet platform to address load balancing challenges in edge computing considering users' preferences. A comparative analysis of the proposed approach with the conventional edge-based approach yields that the proposed approach provides significantly improved results in terms of CPU, memory, and disk utilization.

INDEX TERMS Mobile cloud computing, mobile edge computing, fog computing, cloudlet computing, Internet of things, cloud federation.

I. INTRODUCTION

Mobile devices have limited resources including a central processing unit (CPU), memory, energy, and network. Due to the development of resource-intensive applications more resources are required magnifying the resource constraint problem at the mobile end. The cloud computing paradigm offers a resource-rich environment to these mobile devices for resource sharing and load balancing. The concept of virtualization is used to share the resources of a physical machine. Various service-oriented architectures are offered by the cloud computing paradigm namely, Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) [1]. This paper considers the use case of the IaaS model where a task is bundled as a virtual machine (VM) and is placed on a physical server (cloudlet).

The mobile cloud computing (MCC) model is used to offload a compute-intensive task from a mobile device to a cloud environment, thus addressing resource shortage [2]. In the MCC model, a mobile device directly communicates with the remote server using wireless Internet services [3]. However, the challenges of latency, limited bandwidth, and seamless connectivity pose a major hindrance to the usage of this model. Edge-based solutions such as mobile edge

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

computing (MEC), fog computing, and cloudlet computing seem to offer closer proximity-based solutions eliminating these limitations [4], [5]. The cloudlet-based solution is more adaptable as it offers rich computational resources, diversified features, and higher bandwidth without being dependent on specialized equipment as compared to other solutions such as MEC and fog computing [6]. Cloudlet is a mini cloud having rich computing resources and a stable Internet connection available in the same local area network (LAN) to provide services to nearby devices [7]. The cloudlet-based solutions are considered more viable for the Internet of Things (IoT) and smart cities on a bigger scale due to their faster response than MCC [8]. However, the number of devices communicating with the cloudlets has increased lately resulting in more workload which is beyond the capability of the cloudlets. In this situation, the cloudlets forward the exceeded number of requests to a remote cloud to manage the workloads thus mimicking a conventional MCC model voiding the benefits of edge computing [3]. Hence, there is a need to resolve this problem in such a way that a maximum number of requests are entertained without being forwarded to the remote cloud.

Existing cloudlet-based solutions consider the locality of information and closer proximity, and hence are unable to provide an optimized and scalable solution focusing on resource scarcity challenges [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30]. Moreover, existing edge-based solutions do not consider users' resource preferences which are very vital for the node's performance. These preferences include resource priority and percentage to spare for sharing. Especially in edge-based solutions where a single personal computer or mobile device is playing a role in the edge device, the user's preferences are very important to clearly define the extra resources that can be shared to avoid the overprovisioning of essential resources. Every resource in this situation does not hold equal weight for the user and hence every edge device may offer a different capacity of resources to entertain only a specific set of requests that meet the resource requirements. In a broader sense, an optimal solution must consider resource sharing, load balancing, and workload placement considering edge devices' resource conditions.

To address the aforementioned challenge including the availability of global information, scalability, and user preferences, a broker-based centralized federated cloudlet model has been proposed. Broker has the responsibility to manage all the resource information of member cloudlets and use it for resource sharing, load balancing, and placement decisions. The contributions of this paper are as follows:

- Provides the solution for resource shortage at the cloudlet level.
- Provides an edge-level federated solution having more resources to spare than a standalone cloudlet node.
- Provide the user with ease to select essential and extra resources to avoid the overprovisioning problem.

This paper is structured as follows. Section II contains the discussion of important works related to this study. Section III follows the collaborative model, and the design of the proposed LBRO framework while Section IV provides performance analysis. Lastly, the conclusion and future directions are given in Section V.

II. LITERATURE REVIEW

This section provides insight into state-of-the-art edge computing techniques. The related work is reviewed from the perspective of load balancing and resource optimization in an edge federated environment.

A queuing network scheme has been proposed in [9] using a multi-edge and user-based model. This scheme efficiently works on user-to-edge device mapping and edge device placement. The target of this scheme is constantly moving mobile users and the scope is limited to a Metropolitan Area Network (MAN). The scheme proposed in [10] has considered load balancing in the form of a VM to an edge device with adequate resources. The proposed scheme is more focused on total migration time rather than download time. The minimum total time is achieved by adapting to Wide Area Network (WAN) bandwidth and loading on the edge device. The state of the VM on the destination edge device is compared with the source edge device and the difference is calculated to maintain the same state before the source VM can be shut down. Delta encoding scheme is used to calculate the difference that is de-duplicated and compressed before transfer. The user VM is moved in closer proximity to the source edge device to minimize latency. An SDN-based solution named MobiScud has been proposed in [11]. A mini cloud in the core of Radio Access Network (RAN) is established to host users' Virtual Machines (VMs). These VMs assist users to execute compute-intensive tasks and control messages from mobile devices that are monitored by MobiScud to keep the VM moving along user to keep it in closer proximity. MobiScud also optimizes the flow rules for migrating VM to keep the transition phase smooth with less disruption of services to users. However, users tend to use WiFi more often when indoors, and RAN services are thus needed to be adjusted accordingly.

An ad hoc scheme is proposed in [13] that allows peer devices to lend and acquire resources from each other. Two devices in communication are called master and slave devices. A device offering free resources to others is treated as a slave device and the other device borrowing resources is treated as a master device. This cooperative scheme also supports a smaller scale network assuming there is no interruption in task offloading. Another ad hoc scheme is proposed in [14] using short-range radio communication technology to form a peer-to-peer (P2P) network of mobile devices. Mobile devices participating in the proposed scheme are divided into two categories i.e. a computational service provider that has ample resources to offer and a client who requires resources. An opportunistic approach is used by devices to find appropriate peers and lend services. This scheme is

useful for a smaller-scale network with a short span of service requirements. A scheme named DRAP is proposed in [15] that uses middleware between mobile and edge devices. The devices with resources can form a group and will be treated as edge devices whose resources can be acquired by any other mobile device. The operations of DRAP include resource discovery, calculation of unused resources, and control of the role of edge devices and mobile devices. The proposed scheme is very robust as no single device is acting as an edge device and is capable of dynamically reconfiguring itself upon joining or leaving nodes. However, a log is maintained by some buddy nodes to ensure continued services in case of failure. To attract users an incentive-based approach is employed. The scheme proposed in [17] uses a combination of mobile devices, edge devices, and remote cloud for task offloading. A mobile device requiring resources may contact the edge device present in closer proximity or remote cloud for task offloading. In case the service is provided by the edge device available in the vicinity, the latency is minimized, and internet bandwidth is not required. However, in case the services are acquired from the remote cloud, the model simply becomes a standard MCC Model where the task is offloaded to the remote cloud using internet bandwidth with increased latency as compared to the edge device. An edge-based scheme is proposed in [18] that uses predefined VM templates to fulfill user requirements that are received by the edge device and a predefined template is selected that matches closely to the requirements. Furthermore, infrastructure level customizations are performed before use which is reverted upon completion of the task to ensure the steady state of the infrastructure. The use of VMs is to isolate changes at the infrastructure level from the changes at the guest operating system (OS) level. Workload sharing and load balancing among tenant VMs and otherwise are not reported.

A P2P scheme addressing the selfish behavior of the participating devices is proposed in [19]. The proposed scheme introduces a point-based incentive model. There are two kinds of devices participating in the collaboration. A device can earn points by offering free resources to other devices. On the other hand, a device in need of resources can spend the points to acquire resources from other devices. This scheme also employs the concept of social responsibility of the community group built on a pre-trust-based model that ensures that the devices taking part in the collaboration are trusted. A middleware platform is proposed in [20] to optimize the average CPU load for Augmented Reality (AR) applications. The proposed scheme offers software services for AR applications having the capability to deploy or remove any software component at runtime. Only those customized software components are employed that fulfill the need of the applications. This removes extra load from the CPU. The scope of the scheme is limited to AR applications only.

An interactive edge computing application based on infrastructure as a service using three-tier mobile cloud computing architecture has been proposed in [23]. The proposed scheme considers two assumptions, 1) the edge nodes are static and 2) the maximum distance between any two edge nodes is 2 hops. The objective of this scheme is to achieve higher throughput with minimum delay time. The proposed scheme assumes that nodes are static, and the maximum number of cloudlet hops is two for mobility. It has also been reported that if these assumptions are not fulfilled, the proposed scheme will provide poor results as compared to any standard cloud computing solution. So, as a remedy to the challenge of meeting the assumptions a combination of edge-based and cloud-based schemes is recommended. The experimentation is performed only on interactive mobile applications.

A centralized Enterprise Cloud (EC) based scheme is proposed in [24]. All the participating edge devices are registered with EC which maintains complete information of all the edge devices. Moreover, any mobile device requiring resources is also registered with EC. The request for resources first goes to EC which is responsible for allocating an appropriate edge device. The advantage of this scheme is that a mobile device moving away from an edge device may resume the same task on any of the edge devices registered under the same EC thus saving cost, time, and energy. However, this scheme is highly dependent upon internet bandwidth since EC is not a part of the local network. A similar approach using a centralized root server is proposed in [25]. The root server maintains all sorts of information including connected edge devices and services provided by them. A request is forwarded to the root server that routes it to a suitable edge device. A suitable edge device receiving the request from the root server either executes the task by employing its resources or can share resources with other edge devices. It also has the capability to break the task into smaller proportions and distribute it among various edge devices. Being a centralized service model scalability can be an issue if a larger network model is to be considered.

A cloning technique has been proposed in [26] that maintains a clone of the mobile device in the core of RAN. The clone is based on a VM that is kept in closer proximity to the mobile device and is migrated along the mobile device to maintain minimum distance thus reducing latency. However, these frequent migrations may result in increased network traffic between various edge devices maintained in the core of the RAN. SDNs are used to optimize and manage these traffic flows thus improving performance and energy consumption. The implementation of this technique with the existing RANs requires some fundamental changes in the core cellular network. A mesh network-based solution named MeshCloud using Wireless Mesh Networks is proposed in [27]. The fundamental property of a mesh network is high robustness due to multiple paths leading to a single destination. The proposed scheme is highly dynamic as new edge nodes can be added and removed at any time. Mesh topology lacks scalability and is thus not suitable for larger networks.

A novel task offloading scheme is proposed in [28] that caters to DDoS attacks and considers sustainability and security issues of the cloudlet networks. A collaborative task offloading mechanism for mobile cloudlet networks named

TABLE 1. Summary of existing edge computing techniques.

Ref.	Operation architecture support	Load balancing		Optimal selection		Objective
		VMM	CO	RD	OTP	
[9]	Distributed	-	-	✔	-	Cloudlet Placement
[10]	Distributed	✔	-	✔	✔	VM Handoff
[11]	Distributed	✔	-	-	-	VM Migration
[13]	Centralized	-	✔	✔	✔	Load Balancing
[14]	Peer-to-peer	-	✔	✔	✔	Load Balancing
[15]	Distributed	-	-	✔	-	Cloudlet Management
[17]	Centralized	-	✔	✔	✔	Offloading
[18]	Centralized	-	-	✔	✔	VM Based Cloudlet
[19]	Peer-to-peer	-	✔	✔	✔	User Experience
[20]	Peer-to-peer	-	✔	✔	✔	Cyber Foraging
[23]	Distributed	-	-	-	-	Protocol Optimization
[24]	Hybrid	-	✔	-	-	Energy Consumption
[25]	Centralized	-	✔	✔	✔	Inter Cloudlet Communication
[26]	Distributed	✔	-	-	✔	Energy Consumption
[27]	Hybrid	-	-	-	-	Cost Effective Wireless Cloudlet Access
[28]	Centralized	-	✔	✔	✔	Offloading
[29]	Centralized	-	✔	✔	✔	Offloading
[30]	Centralized	-	✔	✔	✔	Offloading

CTOM is proposed in [29]. An online algorithm is proposed in [30] that finds the optimal computation offloading strategy with intertask dependency and adjusts the strategy in real-time when facing dynamic tasks.

Table 1 summarizes the above-cited literature review with respect to their architectures, target areas, salient features, and performance parameters. The following conclusions can be drawn from the above discussion:

- Techniques targeting load balancing are either centralized or P2P and do not consider a hybrid approach.
- The challenge of load balancing in an edge federated environment is not addressed.
- No technique provides a customizable solution where users' preferences for resources are considered.

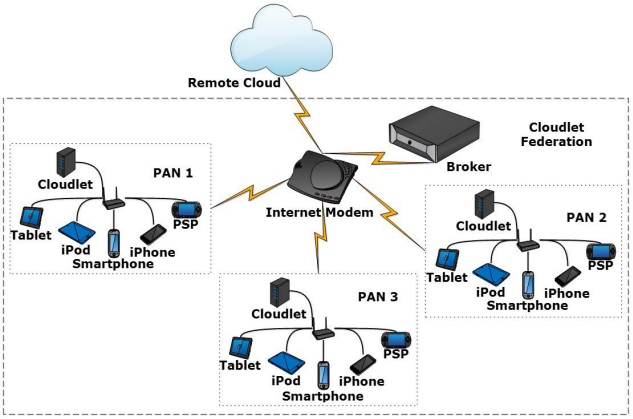


FIGURE 1. Collaborative model.

III. PROPOSED MODEL

This section contains the design of the proposed collaborative model and its architectural details. In addition, the proposed algorithms are discussed in detail.

A. COLLABORATIVE MODEL

The proposed federated cloudlet model shown in Figure 1 provides a solution for resource shortage problems at cloudlets and ensures minimum request forwarding to the remote cloud. The cloudlets may have different owners, and administrative domains, and may also belong to different Cloud Service Providers (CSPs). Every user owning a cloudlet node provides resource preferences to the broker available in the federation. Only the preferred resources considering the priority and percentage are locked for sharing thus providing the cloudlet owner a sense of satisfaction that no resource can be overprovisioned compromising the performance of the owner's tasks and services running on the cloudlet node.

The broker keeps monitoring the resources and maintains an updated state of member cloudlets. When a request arrives at the broker containing the required resources, the broker matches the requirements with all member cloudlets and dispatches the request to the optimal cloudlet having adequate resources with minimum latency, and updates the resource state of that cloudlet. Similarly, when a VM is to be migrated due to load or a more optimal location, the occupied resources are released from the source server and an updated state of resources is maintained.

Besides managing resource information of member cloudlets, the broker performs various operations for cloudlet federation such as cloudlet registration, keeping track of resources, and optimal cloudlet selection. The resources include CPU, memory, storage, and bandwidth, whereas optimal selection includes decisions regarding cloudlet and VM for migration [1].

Existing cloudlet-based models support resource sharing and load balancing based on local knowledge about the other cloudlet nodes (fixed or mobile) in closer proximity, preferably within the same LAN. The proposed cloudlet federation extends the range of closer proximity to MAN and WAN with

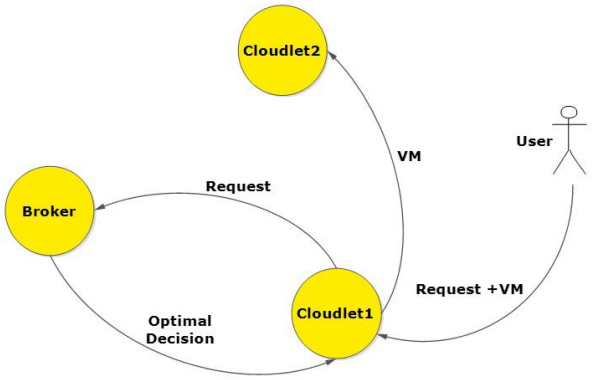


FIGURE 2. LBRO data flow.

added features of user preferences based on resource sharing and load balancing.

B. LOAD-AWARE RESOURCE ALLOCATION

Since all the applications are placed on a cloudlet shared resource based on time thus an increased wait time is observed for the new applications and ultimately the poor resource availability degrades the performance of the overall system and all applications running on it. The resource limitation not only affects the performance but also forces the cloudlet to forward the requests to the remote cloud to manage the load.

The current implementations of cloudlets are only focused on addressing the distance, limited bandwidth, and latency challenges considering only a standalone cloudlet or group of cloudlets at the same location shared via LAN. Our proposed approach offers a load-aware collaborative scenario in which cloudlets share the user-preferred load and resources with peer cloudlets, managed by a centralized broker in the federation that may extend to a MAN or WAN.

Enhancing the scope of federation geographically improves the possibility of getting more cloudlet nodes for resource sharing thus addressing scalability issues. The final decision about optimal cloudlet selection is made considering minimum latency as compared to the remote cloud. A remote cloud is referred to as a conventional cloud that comes into play when the whole federation is out of resources and is treated as a worst-case where the results of proposed and conventional approaches become equal. Figure 2 presents an overview of the information flow between cloudlets and brokers.

The collaborative approach keeps track of resource utilization from the given preferences by the cloudlet owners for all member cloudlets and the one with adequate resources with minimum latency is selected. All the workloads are considered in the form of VMs. In the case of load balancing, a VM is selected for migration based on the required resources to be released. The detailed work of the proposed collaborative approach is presented in the subsequent sections. Table 2 contains the notations used in the paper.

TABLE 2. Notations used in the study.

Notation	Definition
S_f	Open Virtualization Format (OVF) file size
R_L	Resource level
SD	Decision status
S	Cloudlet
SE	Eligible cloudlet
S_o	Optimal cloudlet
R_s	Cloudlet Rank
R_b	Available resource rank
$I_T R$	Total cumulative resource index
I_R	Single resource index
L	Latency
w_T	Total weight
p	Percentage
VM_O	Optimal VM
VM_E	Eligible VM

C. CALCULATION OF RESOURCE UTILIZATION

The problem of finalizing the optimal cloudlet is challenging due to dependency on multiple variables including resources such as CPU, memory, storage, and bandwidth. Weights are assigned to each resource by the owner of the edge device to segregate spare resources for sharing. For example, an owner 'x' wishes to spare 20% of the CPU, 30% of the memory, 10% of his disk storage, and 5% of the bandwidth to take part in the sharing process for some other user 'y' to acquire these resources for the execution of some task. The owner can simply assign weights 2, 3, 1, and 5, respectively to each available resource for sharing.

The resource calculation process can be divided into two phases i.e., resource index calculation and resource level calculation. The resource index calculation phase provides a single cumulative value based on selected resources, their quantities, and assigned weights that are used to rank the cloudlets in the federation. Let $W = \{w_1, w_2, w_3, \ldots, w_n\}$ be set of weights assigned by the owner against each resource, $C = \{s_1, s_2, s_3, \ldots, s_n\}$ be a set of cloudlets, $A = \{b_1, b_2, b_3, \ldots, b_n\}$ and $T = \{r_1, r_2, r_3, \ldots, r_n\}$ represents set of available and total resources at a cloudlet respectively. The value of the total resource index can be calculated using the following equation.

$$I_{TR}(S_i) = \sum_{i=1}^{n} \frac{b_i}{r_i} \times w_i \tag{1}$$

Algorithm 1 is designed for resource index calculation and is presented below.

The resource level calculation phase helps to initiate load balancing on a cloudlet having a critical resource level. There are two levels of resources. One is "normal" and the other is "critical". A cloudlet is considered in a normal state if available resources are above the minimum threshold level, whereas a cloudlet is considered in the critical state if available resources are below the minimum threshold level. The value of the threshold is identified by the resource requirement of the host OS i.e. in our case the minimum

Algorithm 1 Resource Index Calculation

Input: Set of cloudlets, available resource at each cloudlet, resource weights assigned by the administrator, and resource values

Output: Total resource index

Begin:
Let resource index I_{TR} be *NULL*
for each cloudlet s_i **do**
do
for each resource b_i **do**
do
Get weights w assigned by administrator
Get available resource value b_i
Get total resource value r_i
Calculate resource percentage p
$p = \left(\frac{w}{wT}\right)$
Calculate resource index I_R
$I_R = \left(\frac{b_i}{r_i} \times p\right)$
I_{TR} = Accumulate all values of I_R
end for
end forreturn I_{TR} **end:**

recommended resource requirement for UBUNTU 14.0.4 LTE [3], [31] as presented in Table 3.

TABLE 3. Minimum recommended resources for Ubuntu 14.0.4 LTE.

Resource type	Detailed requirement
CPU	1 gigahertz (GHz) x 86 processor
Memory	1 gigabyte (GB)
Storage	5 gigabytes (GB)

Let $U = y_1, y_2, y_3, \ldots, y_n$ be a set of utilized resources, $D = x_1, x_2, x_3, \ldots, x_n$ be set of demanded resources for load balancing, $K = k_1, k_2, k_3, \ldots, k_n$ be set of occupied resources by a VM, and $M = z_1, z_2, z_3, \ldots, z_n$ be set of minimum required resources for host Operating System (OS) of member cloudlets. The resource matrices used throughout the resource level calculation phase are listed in Table 4.

TABLE 4. Resource metrics.

Resource matrix	Detail
ARM$[b_{ij}]$	Contains available resources of member cloudlets
TRM$[r_{ij}]$	Contains total resources of member cloudlets
URM$[y_{ij}]$	Contains utilized resources of member cloudlets
DRM$[x_{ij}]$	Contains demanded resources of member cloudlets
MRM$[z_{ij}]$	Contains minimum required resources for host OS of member cloudlets

The criteria for resource-level calculation are as follows:

$$TRM[r_{ij}] = ARM[b_{ij}] + URM[y_{ij}] + MRM[z_{ij}] \quad (2)$$

$$DRM[x_{ij}] = ARM[b_{ij}] - MRM[z_{ij}] \quad (3)$$

Note: The negative value of the demand resource matrix shows fewer resources than the minimum required resources for the host OS.

$$f(x) = \begin{cases} 1, & if\ ARM < [b_{ij}] < MRM[z_{ij}] \\ \forall & \begin{cases} i = 1, & j = 1, 2, 3, \ldots n \\ i = 2, & j = 1, 2, 3, \ldots n \\ i = 3, & j = 1, 2, 3, \ldots n \\ \vdots & \\ i = n, & j = 1, 2, 3, \ldots n \end{cases} \\ 0, & Otherwise \end{cases} \quad (4)$$

The value of "1" represents a critical state and "0" represents otherwise. Algorithm 2 presents resource level calculation.

Algorithm 2 Resource Level Calculation

Input: Set of cloudlets and available resources at each cloudlet

Output: Resource level

Begin:
Let resource level R_L be *NULL*
for each cloudlet s_i **do**
do
for each resource b_i **do**
do
Calculated resource level R_L
if $b_i \leq (z_i)$ **then**
R_L = "Critical"
else
R_L = "Normal"
end if
end for
end forreturn R_L **end:**

D. OPTIMAL CLOUDLET SELECTION FOR VM PLACEMENT

Two phases of the optimal cloudlet selection include filtration of eligible cloudlets having enough resources to execute the job that is identified by the following condition:

$$f(x) = \begin{cases} 1(Eligible), & if\ R_z[Z_{ij}] \leq R_b[b_{ij}] \\ \forall & \begin{cases} i = 1, & j = 1, 2, 3, \ldots n \\ i = 2, & j = 1, 2, 3, \ldots n \\ i = 3, & j = 1, 2, 3, \ldots n \\ \vdots & \\ i = n, & j = 1, 2, 3, \ldots n \end{cases} \\ 0(Non - eligible), & Otherwise \end{cases} \quad (5)$$

In the second phase, the optimal cloudlet is selected based on the resource index value calculated in algorithm 3. The cloudlet with the maximum resource index value and non-critical resource level is selected as optimal from the eligible cloudlet list. The maximum value of the resource

index indicates that the cloudlet has the maximum available resources in the federation. Let $Q = q_1, q_2, q_3, \ldots, q_n$ represents the set of requests at the broker.

Algorithm 3 Optimal Cloudlet Selection

Input: Requests received by Broker, required resources for each request, available cloudlets and resources at each cloudlet, indexes of cloudlets
Output: Optimal cloudlet

Begin:
Let optimal cloudlet C_o be NULL
for each request q at broker **do**
  check status S_D
  if S_D="Decision Pending" **then**
    for each cloudlet s_i **do**
      for each resource b_i and z_i **do**
        if $b_i \geq z_i$ **then**
          push cloudlet C_E in eligible cloudlet list
        end if
      end for
    end for
  end if
end for
for each cloudlet in eligible cloudlet list **do**
  Get resource level R_L
  if $R_L \neq$ "Critical" and $I_T R = Max$ with min L **then**
    $S_O = S_E$
  end if
end forreturn S_o **end:**

E. OPTIMAL VM SELECTION FOR MIGRATION

In case a cloudlet is in a critical state, a load balancing mechanism is initiated that requires a VM to be migrated from the critical cloudlet to ease up the load. The problem of finalizing the optimal VM is challenging due to dependency on multiple variables such as VM size and required resources. The selection process of optimal VM starts with the filtration of eligible VMs by comparing the utilized resource matrix (URM) with the demand resource matrix (DRM) using the following criteria:

$$f(x) = \begin{cases} 1(Eligible), & if\ URM[u_{ij}] \geq DRM[d_{ij}] \\ \forall & \begin{cases} i = 1, \quad j = 1, 2, 3, \ldots n \\ i = 2, \quad j = 1, 2, 3, \ldots n \\ i = 3, \quad j = 1, 2, 3, \ldots n \\ \vdots \\ i = n, \quad j = 1, 2, 3, \ldots n \end{cases} \\ 0(Non - eligible), & Otherwise \end{cases} \tag{6}$$

The load balancing mechanism is performed offline. A VM is considered eligible if by removing it from the cloudlet, the resource level becomes normal. In the second phase, an optimal VM is selected from the list of eligible VMs. A VM occupying minimum resources and size is selected as it can be migrated in a very short amount of time having a larger solution space of eligible cloudlets. Algorithm 4 presents the mechanism for optimal VM selection for migration.

Algorithm 4 Evaluation of Optimal VM for Migration

Input: Set of the occupied resource by each VM running on the cloudlets, occupied resource indexes, demanded resources list
Output: Optimal VM for migration

Begin:
Let optimal VM VM_o be NULL
for each cloudlet s_i **do**
  for each VM in running VM list **do**
    for each resource k_i **do**
      if removing VM yields to normal level **then**
        push eligible VM VM_E in eligible VM list
      end if
    end for
  end for
end for
for each VM in eligible list **do**
  if $I_{TR} = Min$ **then**
    $VM_O = VM_E$
  end if
end forreturn VM_o
end:

IV. PERFORMANCE EVALUATION

This section discusses performance metrics, experimental setup, and testbed results. In these experiments, we record the resource level of a cloudlet with and without the implementation of our proposed approach and present a comparative analysis. Since we are using cold migration, there is no overhead of memory pre-copying operation consuming more CPU cycles and network bandwidth [32]. An Open Virtualization Appliance (OVA) file is transferred from a heavily loaded cloudlet to a light-loaded cloudlet having maximum available resources.

A. PERFORMANCE METRICS

Several metrics such as CPU, memory, disk utilization, and latency are considered for the evaluation of proposed algorithms. The considered metrics reflect system resources that are independent of application type and nature. The main objective is to launch enough requests that the system is forced to either forward the request to the remote cloud in the case of the conventional cloudlet model or collaborate with peer cloudlets in the case of the proposed approach. Real load in terms of VMs has been used to exhaust the system's resources.

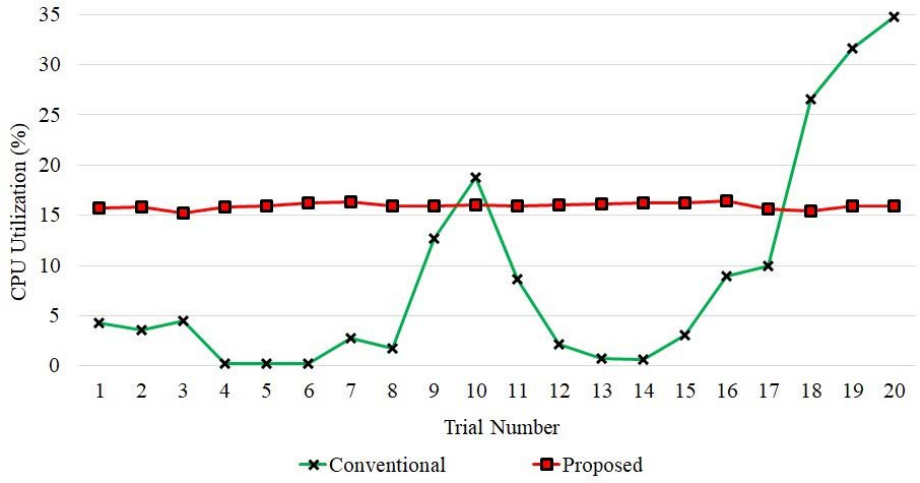


FIGURE 3. CPU utilization.

B. EXPERIMENTAL SETUP

For this study, we establish the setups of conventional and collaborative cloudlet models. The conventional cloudlet model is configured with a single cloudlet, a single client, and a remote cloud. A client's request to execute a task is forwarded to the remote cloud if the required resources exceed the available resources of the cloudlet. The proposed collaborative cloudlet model is configured with three cloudlets, a single client, a broker, and a remote cloud, as shown in Figure 1. In this scenario, the client's request is forwarded to the broker which selects an optimal cloudlet for the execution of the task, considering the load and available resources at a particular cloudlet. For both setups, at the start of the experimentation process, enough requests are launched at the cloudlets to exceed their resource limits. Amazon EC2 instance is used to mimic remote cloud. Cloudlet nodes of both conventional and proposed collaborative models are deployed on VMware ESX 6.0 server. The resources allocated to each cloudlet include a single CPU, 8GB of memory, and 30GB of HDD. The VM taken as a real load to be migrated between cloudlets consist of Tiny Core Linux (TCL). The physical servers on which the virtualization environment is deployed are a part of the data center having the following specifications as presented in Table 5.

TABLE 5. Server specifications.

Server	Gen	CPU	RAM	HDD	NIC	B/w
HP DL360	G6	2.8GHz X5670	64GB	250GB SAS	1Gbps	32Mbps

C. TESTBED RESULTS

All the experiments are conducted in an isolated production environment. Each trial is repeated several times to obtain the values of various resource parameters for both conventional and proposed models.

1) CENTRAL PROCESSING UNIT UTILIZATION

CPU is the primary resource of a computer system. Observations for both models have been shown in Figure 3. These observations are recorded during a peak time, where peak time refers to a system state in which it has the maximum number of requests it can entertain. The results of the proposed model show a stable CPU utilization as compared to the conventional model due to load balancing features while the conventional model suffers ups and downs regarding CPU utilization and an increase is observed while the number of trials is increased.

2) MEMORY UTILIZATION

Memory is considered a critical resource of a computer system. Often this resource creates a bottleneck for the system performance due to continuous read, write, and paging operations. A comparative analysis of the memory utilization trend for both models is shown in Figure 4. The results clearly show an elevated level of memory utilization by the conventional model completely utilizing the memory causing performance degradation. The load balancing feature of the proposed model admits limited requests according to available memory, thus avoiding the critical resource situation.

3) STORAGE UTILIZATION

Storage resource is not considered critical resource now a day due to the availability of larger capacity at a low cost. However, the increased rate of reading and writing requests from storage might cause performance degradation. The results of storage utilization for both models are presented in Figure 5. The results clearly show an elevated level of disk utilization by the conventional model as compared to the proposed model due to load balancing features. The load balancing module admits a limited number of requests according to available memory space. No request is admitted if there is no free memory available avoiding page swapping between

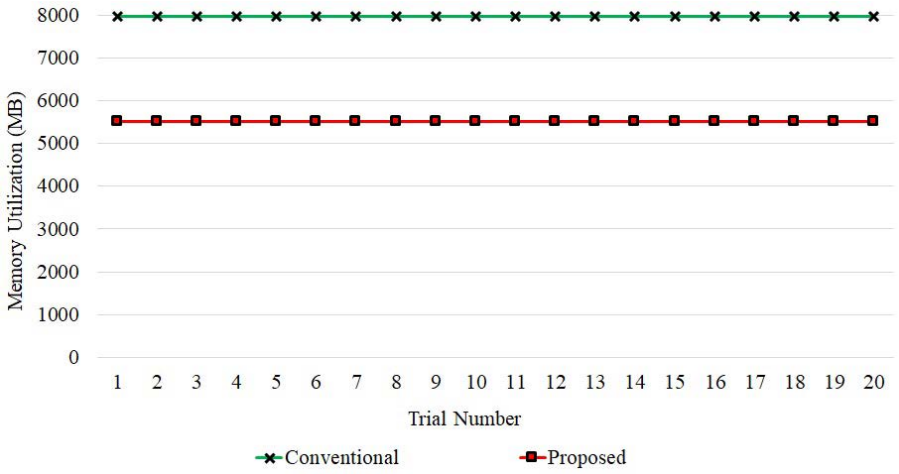


FIGURE 4. Memory utilization.

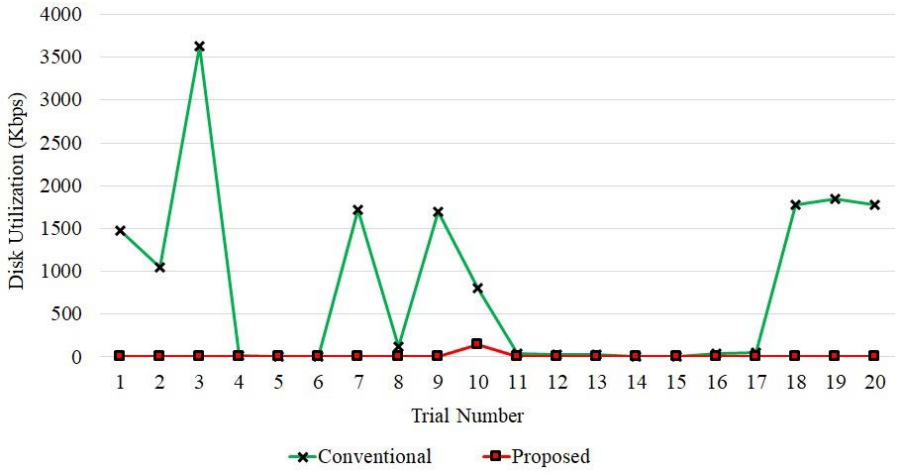


FIGURE 5. Disk utilization.

memory and disk drive. As a result, fewer memory contents are swapped or overwritten. As most of the required contents are already available in memory, it decreases the disk read and writes requests improving performance.

The conclusion drawn from the above experimentation is very clear that the proposed load-aware system performs better than the conventional system in terms of resource preservation. The conventional model ends up overloading the system ignoring the minimum resources required for the host OS compromising the performance of all the applications (VMs) running on a cloudlet. The independent design of the proposed system suggests that it can easily work with any type of application scenario and has the capability to scale.

V. CONCLUSION AND FUTURE WORK

The proposed approach plays an important role in the performance improvement of MCC. The proposed model not only addresses the resource scarcity of cloudlets but also resolves the under-provisioning of resources at peer cloudlets thus maximizing the resource utilization at the cloudlet level. The experimental results show decreased load and stable resource utilization at cloudlets without jeopardizing the performance of applications running on them.

In future work, a software platform that supports resource collaboration is to be developed for commercial purposes considering cost and energy. This model will serve as a base and other parameters such as latency, hop-count, throughput, response time, execution time, and offload time will be incorporated as future enhancements to the presented algorithms. Additionally, instead of manually assigning resource-specific weights to segregate spare resources of a cloudlet for sharing, optimal weights per resource will be dynamically predicted using unsupervised learning methods and neural networks.

REFERENCES

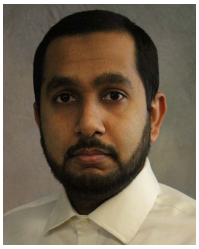
[1] M. Z. Nayyer, I. Raza, and S. A. Hussain, "Revisiting VM performance and optimization challenges for big data," in *Advances in Computers*, vol. 114. Amsterdam, The Netherlands: Elsevier, 2019, pp. 71–112.

[2] W. Liu, W. Gong, W. Du, and C. Zou, "Computation offloading strategy for multi user mobile data streaming applications," in *Proc. 19th Int. Conf. Adv. Commun. Technol. (ICACT)*, Feb. 2017, pp. 111–120.
[3] M. Z. Nayyer, I. Raza, and S. A. Hussain, "A survey of cloudlet-based mobile augmentation approaches for resource optimization," *ACM Comput. Surveys*, vol. 51, no. 5, pp. 1–28, 2018.
[4] K. Bilal, O. Khalid, A. Erbad, and S. U. Khan, "Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers," *Comput. Netw.*, vol. 130, pp. 94–120, Jan. 2018.
[5] S. Yi, C. Li, and Q. Li, "A survey of fog computing: Concepts, applications and issues," in *Proc. Workshop Mobile Big Data*, 2015, pp. 37–42.
[6] K. Dolui and S. K. Datta, "Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing," in *Proc. Global Internet Things Summit (GIoTS)*, Jun. 2017, pp. 1–6.
[7] 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.
[8] D. P. Abreu, K. Velasquez, M. Curado, and E. Monteiro, "A resilient Internet of Things architecture for smart cities," *Ann. Telecommun.*, vol. 72, pp. 19–30, Feb. 2017.
[9] M. Jia, J. Cao, and W. Liang, "Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks," *IEEE Trans. Cloud Comput.*, vol. 5, no. 4, pp. 725–737, Oct./Dec. 2015.
[10] K. Ha, Y. Abe, Z. Chen, W. Hu, B. Amos, P. Pillai, and M. Satyanarayanan, "Adaptive VM handoff across cloudlets," Dept. Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-CS-15-113, Jun. 2015.
[11] K. Wang, M. Shen, J. Cho, A. Banerjee, J. Van Der Merwe, and K. Webb, "MobiScud: A fast moving personal cloud in the mobile network," in *Proc. 5th Workshop All Things Cellular, Oper., Appl. Challenges*, 2015, pp. 19–24.
[12] Y. Wu and L. Ying, "A cloudlet-based multi-lateral resource exchange framework for mobile users," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2015, pp. 927–935.
[13] X. Guo, L. Liu, Z. Chang, and T. Ristaniemi, "Data offloading and task allocation for cloudlet-assisted ad hoc mobile clouds," *Wireless Netw.*, vol. 24, no. 1, pp. 79–88, Jan. 2018.
[14] M. Chen, Y. Hao, Y. Li, C.-F. Lai, and D. Wu, "On the computation offloading at ad hoc cloudlet: Architecture and service modes," *IEEE Commun. Mag.*, vol. 53, no. 6, pp. 18–24, Jun. 2015.
[15] R. Agarwal and A. Nayak, "DRAP: A decentralized public resourced cloudlet for ad-hoc networks," in *Proc. IEEE 4th Int. Conf. Cloud Netw. (CloudNet)*, Oct. 2015, pp. 309–314.
[16] L. Tang, X. Chen, and S. He, "When social network meets mobile cloud: A social group utility approach for optimizing computation offloading in cloudlet," *IEEE Access*, vol. 4, pp. 5868–5879, 2016.
[17] B. Zhou, A. V. Dastjerdi, R. N. Calheiros, S. N. Srirama, and R. Buyya, "A context sensitive offloading scheme for mobile cloud computing service," in *Proc. IEEE 8th Int. Conf. Cloud Comput.*, Jul. 2015, pp. 869–876.
[18] 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.
[19] H. Flores, R. Sharma, D. Ferreira, V. Kostakos, J. Manner, S. Tarkoma, P. Hui, and Y. Li, "Social-aware hybrid mobile offloading," *Pervasive Mobile Comput.*, vol. 36, pp. 25–43, Apr. 2017.
[20] S. Bohez, J. De Turck, T. Verbelen, P. Simoens, and B. Dhoedt, "Mobile, collaborative augmented reality using cloudlets," in *Proc. Int. Conf. MOBILe Wireless MiddleWARE, Operating Syst., Appl.*, Nov. 2013, pp. 45–54.
[21] S. Moon and Y. Lim, "Task migration with partitioning for load balancing in collaborative edge computing," *Appl. Sci.*, vol. 12, no. 3, p. 1168, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/3/1168
[22] R. K. Nayak and G. Srinivasarao, "A greedy load balancing strategy with optimal constraints for edge computing in industrial cloud environment," in *Innovations in Computer Science and Engineering*, H. S. Saini, R. Sayal, A. Govardhan, and R. Buyya, Eds. Singapore: Springer, 2022, pp. 31–38.
[23] D. Fesehaye, Y. Gao, K. Nahrstedt, and G. Wang, "Impact of cloudlets on interactive mobile cloud applications," in *Proc. IEEE 16th Int. Enterprise Distrib. Object Comput. Conf.*, Sep. 2012, pp. 123–132.
[24] Y. Jararweh, L. Tawalbeh, F. Ababneh, A. Khreishah, and F. Dosari, "Scalable cloudlet-based mobile computing model," *Proc. Comput. Sci.*, vol. 34, pp. 434–441, Aug. 2014.
[25] J. Rawadi, H. Artail, and H. Safa, "Providing local cloud services to mobile devices with inter-cloudlet communication," in *Proc. 17th IEEE Medit. Electrotech. Conf. (MELECON)*, Apr. 2014, pp. 134–138.
[26] X. Sun and N. Ansari, "Green cloudlet network: A distributed green mobile cloud network," *IEEE Netw.*, vol. 31, no. 1, pp. 64–70, Feb. 2017.
[27] K. A. Khan, Q. Wang, C. Grecos, C. Luo, and X. Wang, "MeshCloud: Integrated cloudlet and wireless mesh network for real-time applications," in *Proc. IEEE 20th Int. Conf. Electron., Circuits, Syst. (ICECS)*, Dec. 2013, pp. 317–320.
[28] N. Yang, X. Fan, D. Puthal, X. He, P. Nanda, and S. Guo, "A novel collaborative task offloading scheme for secure and sustainable mobile cloudlet networks," *IEEE Access*, vol. 6, pp. 44175–44189, 2018.
[29] X. Fan, X. He, D. Puthal, S. Chen, C. Xiang, P. Nanda, and X. Rao, "CTOM: Collaborative task offloading mechanism for mobile cloudlet networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.
[30] M. Wang, T. Wu, X. Fan, P. Sun, Y. Qu, and P. Yang, "TPD: Temporal and positional computation offloading with dynamic and dependent tasks," *Wireless Commun. Mobile Comput.*, 2021, pp. 1–15, Nov. 2021.
[31] M. Dawson, B. DeWalt, and S. Cleveland, "The case for UBUNTU Linux operating system performance and usability for use in higher education in a virtualized environment," in *Proc. Southern Assoc. Inf. Syst. Conf. (SAIS)*, St. Augustine, FL, USA, Mar. 2016.
[32] W. Voorsluys, J. Broberg, S. Venugopal, and R. Buyya, "Cost of virtual machine live migration in clouds: A performance evaluation," in *Proc. IEEE Int. Conf. Cloud Comput.*, Bangalore, India, Sep. 2009, pp. 254–265.

MUHAMMAD ZIAD NAYYER received the M.S. degree in computer science from the Government College University (GCU), Lahore, Pakistan, in 2011, and the Ph.D. degree in computer science from COMSATS University Islamabad, Lahore Campus. He is currently working as an Assistant Professor with the Department of Computer Science, GIFT University, Gujranwala, Pakistan. He is an Active Member of the Advanced Communication Networks Laboratory. He has numerous publications on his account, including impact factor journal publications and book chapters. His research interests include cloud computing, VM migration, mobile cloud computing, cloud federation, mobile edge computing, fog computing, and cloudlet computing.

IMRAN RAZA (Member, IEEE) received the B.S. degree in CS and the M.Phil. degree in computer science from Pakistan. He has been working as an Assistant Professor with the Department of Computer Science, COMSATS University Islamabad, Lahore Campus, since 2003. He has authored and coauthored more than 40 journals and conference papers. His research interests include cloud computing, mobile edge computing, SDN, NFV, wireless sensor networks, MANETS, QoS issue in networks, and routing protocols. He has been actively involved in simulating CERN O2/FLP upgrades. He has supervised and co-supervised many funded projects related to ICT in Healthcare. He has been a member of ACM.

SYED ASAD HUSSAIN (Member, IEEE) received the master's degree from Punjab University Lahore, Pakistan, the master's degree from the University of Wales, Cardiff, U.K., and the Ph.D. degree from Queen's University Belfast, U.K. He has been working as a Professor, since 2010, and the Dean Faculty of Information Sciences and Technology at COMSATS University Islamabad, Pakistan, since 2015. He has worked as the Head of the Computer Science Department, COMSATS University Islamabad, Lahore Campus, from 2008 to 2010, and from 2011 to 2017. He has been leading the Communications Networks Research Group, Department of Computer Science, COMSATS University Islamabad, Lahore Campus, since 2005. He was funded for his Ph.D. degree by Nortel Networks U.K. He was awarded Prestigious Endeavour Research Fellowship by Australian Government for his post doctorate at The University of Sydney, Australia, in 2010. He has taught at Queen's University Belfast, U.K., Lahore University of Management Sciences (LUMS), and University of the Punjab, Pakistan. He is currently supervising the Ph.D. students at COMSATS University and split-site Ph.D. students at Lancaster University, U.K., in the areas of cloud computing and cybersecurity. He is actively involved in collaborative research with CERN in Switzerland and different universities of the world, such as Cardiff University, U.K., Lancaster University, U.K., Dalhousie University, Canada, and Charles Sturt University, Australia. He is supervising funded projects as a Principal Investigator in the field of healthcare systems. He has authored and coauthored more than 85 journals and conference papers and has written a book titled *Active and Programmable Networks for Adaptive Architectures and Services* (Taylor and Francis, USA). He regularly reviews international journals and conferences papers.



MUHAMMAD HASAN JAMAL received the B.S. degree in computer and information engineering from International Islamic University Malaysia, in 2005, the M.S. degree in computer engineering from the University of Engineering and Technology Lahore, Lahore, Pakistan, in 2008, and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 2015. From 2006 to 2010, he was a Research Associate and a Senior Research Associate with the Al-Khawarizmi Institute of Computer Science, UET Lahore. He was a Graduate Technical Summer Intern at the Sandia National Laboratory, Albuquerque, NM, in summer 2015. Since 2016, he has been an Assistant Professor with the Department of Computer Science, COMSATS University Islamabad, Lahore Campus. His research interests include parallel and distributed systems, scientific computing, data analytics, and NLP. He was a recipient of the Fulbright Scholarship for his Ph.D. studies.

ZEESHAN GILLANI received the B.S. degree in computer science from Brunel University, U.K., in 2006, the M.S. degree in bioinformatics from the Kings College London, U.K., in 2009, and the Ph.D. degree in bioinformatics from Zhejiang University, China, in 2016. From 2010 to 2012, he was an Assistant Professor with the University of Lahore. Since 2016, he has been an Assistant Professor with the Department of Computer Science, COMSATS University Islamabad (Lahore Campus), Pakistan. His research interests include bioinformatics, data mining, and performance evaluation.

SOOJUNG HUR received the B.S. degree from Daegu University, Gyeongbuk, South Korea, in 2001, the M.S. degree in electrical engineering from the San Diego State University of San Diego, in 2004, and the M.S. and Ph.D. degrees in information and communication engineering from Yeungnam University, South Korea, in 2007 and 2012, respectively. She is currently working as a Research Professor at the Mobile Communication Laboratory, Yeungnam University. Her current research interests include the performance of mobile communication, indoor/outdoor location, and unnamed vehicle.

IMRAN ASHRAF received the M.S. degree in computer science from the Blekinge Institute of Technology, Karlskrona, Sweden, in 2010, and the Ph.D. degree in information and communication engineering from Yeungnam University, Gyeongsan, South Korea, in 2018. He has worked as a Postdoctoral Fellow at Yeungnam University, where he is currently working as an Assistant Professor at the Information and Communication Engineering Department. His research interests include indoor positioning and localization, indoor location-based services in wireless communication, smart sensors (LIDAR) for smart cars, and data mining.