IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 6, 2020, accepted September 21, 2020, date of publication September 25, 2020,
date of current version October 9, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.3026875*

# Energy-Aware Task Allocation for Multi-Cloud Networks

**SAMBIT KUMAR MISHRA[1], (Member, IEEE), SONALI MISHRA[2], AHMED ALSAYAT[3], N Z JHANJHI[4], MAMOONA HUMAYUN[5], KSHIRA SAGAR SAHOO[6], (Member, IEEE), AND ASHISH KR. LUHACH[7], (Member, IEEE)**

[1]Department of Computer Science and Engineering, SRM University, Amaravati 522502, India
[2]Department of Computer Science and Engineering, Siksha 'O' Anusandhan (Deemed to be) University, Bhubaneswar 751030, India
[3]College of Computer and Information Sciences, Jouf University, Al-Jouf 2014, Saudi Arabia
[4]School of Computer Science and Engineering (SCE), Taylor's University, Subang Jaya 47500, Malaysia
[5]Department of Information Systems, College of Computer and Information Sciences, Jouf University, Al-Jouf 2014, Saudi Arabia
[6]Department of Information Technology, VNRVJIET, Hyderabad 500090, India
[7]Department of Electrical and Communication Engineering, The PNG University of Technology, Lae 411, Papua New Guinea

Corresponding author: N Z Jhanjhi (noorzaman.jhanjhi@taylors.edu.my)

**ABSTRACT** In recent years, the growth rate of Cloud computing technology is increasing exponentially, mainly for its extraordinary services with expanding computation power, the possibility of massive storage, and all other services with the maintained quality of services (QoSs). The task allocation is one of the best solutions to improve different performance parameters in the cloud, but when multiple heterogeneous clouds come into the picture, the allocation problem becomes more challenging. This research work proposed a resource-based task allocation algorithm. The same is implemented and analyzed to understand the improved performance of the heterogeneous multi-cloud network. The proposed task allocation algorithm (Energy-aware Task Allocation in Multi-Cloud Networks (*ETAMCN*)) minimizes the overall energy consumption and also reduces the makespan. The results show that the makespan is approximately overlapped for different tasks and does not show a significant difference. However, the average energy consumption improved through *ETAMCN* is approximately 14%, 6.3%, and 2.8% in opposed to the random allocation algorithm, Cloud Z-Score Normalization (*CZSN*) algorithm, and multi-objective scheduling algorithm with Fuzzy resource utilization (FR-MOS), respectively. An observation of the average SLA-violation of ETAMCN for different scenarios is performed.

**INDEX TERMS** Resource based, energy consumption, makespan, multi-cloud, task scheduling, cloud virtualization.

## I. INTRODUCTION

Cloud computing is a service delivery model that has accumulation with heterogeneous datacenters affording huge resources. The cloud computing platform is deemed to afford high processing and vast memory-intensive services to the users in pay as you use manner [1]. The resources in the cloud computing system are distributed across different locations in the form of large-scale data centers connected in a distributed geographical area. The virtual shared resource of the cloud system provides software, infrastructure, platform, devices, and other services over the Internet [1]. The cloud services are provided by the cloud service provider (CSP). Cloud users are not supposed to own the infrastructure physically; rather, they get the services on rent from a third-party provider. The duty or burden of the CSP is to afford services seamlessly. The CSP guaranteed services called Quality of Service (QoS) to the cloud users to satisfy their demands. These QoS agreements are formally maintained in the form of Service Level Agreements (SLA) [2].

Cloud networking or cloud-based networking is a concept that describes access to share networking resources from a centralized service provider employing WAN or Internet-based technologies [32], [34]. A cloud network exists within or is part of a cloud computing infrastructure. In cloud networking, the computing resources, as well as the network, can be shared for the cloud users. Users can access services available on the multi-cloud without knowing the executing

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

system. It means, the user unaware about the cloud where the task is currently, also the physical host, as well as the Virtual Machine (VM) that execute the task is also unknown to the user. The cloud environment enhances performance metrics, such as latency, bandwidth, QoS, and optimized wireless resources [34].

In recent years, the researchers have focused on introducing scheduling mechanisms in the cloud, focusing on load-balanced provisioning of tasks to produce efficient resource utilization. The management of virtual cloud resources plays a significant role in having an energy-efficient cloud data center [2], [3], [34]. Since the estimation of data and applications is expanding rapidly, it is an exciting task to reduce the energy consumption of data centers. This needs different approaches that can depreciate the overall energy consumption for various applications. There are many responsibilities of the service providers to deliver on-demand services to the user. The most critical responsibility of CSPs is to make sure that the resources required by the end users of the cloud services are allocated properly [1]–[3]. But, when the same allocation problem is considered for multiple clouds (multi-cloud), the scenario becomes more challenging [30]. Including all these, each CSP tries to minimize the energy usage for profit maximization and experiences penalties for the case of failures during execution [4]. This will lead the problem more attractive for the researchers. Hilman *et al.* have distinguished different cloud platforms (such as single cloud, multi-cloud, hybrid cloud, and bare-metal cloud) [30], [36].

The contributions of the article are reviewed as follows.

***A.*** The multi-cloud system model is presented with respect to task allocation. ***B.*** The problem of task allocation is expressed as an integer linear programming (ILP) problem. ***C.*** A heuristic algorithm (*ETAMCN*) has been proposed for the distribution of heterogeneous tasks in the multi-cloud network. ***D.*** The proposed algorithm (*ETAMCN*) optimizes energy utilization and the makespan of the system. ***E.*** The simulations illustrate the effectiveness of the intended algorithm as compared to the Random allocation algorithm, Cloud Z-Score Normalization (*CZSN*) algorithm, and multi-objective scheduling algorithm with Fuzzy resource utilization (FR-MOS). ***F.*** An observation of the average SLA-violation of *ETAMCN* for two scenarios is performed.

The organization the paper is as follows. The preliminaries for our work are presented in Section II as related work. Section III discusses the system model for the multi-cloud network. Section IV exhibits the task allocation algorithm and its description followed by the experimental assessment and analysis in Section V. Finally, the conclusion of the paper, along with future work, is depicted in Section VI.

## II. RELATED WORK

During the last decade, a number of mechanisms have been notified for different cloud computing scenarios. Researchers presented different variations of task scheduling technique for the cloud [1], [2], [5], [33] and multi-cloud environment [3], [6]–[14]. The consideration of virtual resources and performance parameters from the literature, along with the current work, is listed in Table 1. The environment in Table 1 represents the single cloud or multiple cloud (multi-cloud). Juarez *et al.* [5] have presented an energy-aware cloud system for dynamic allocation of tasks. They have used Direct Acyclic Graph (DAG) model to represent the task dependencies [5]. Panda and Jana have proposed three algorithms for task scheduling in the multi-cloud environment [8]. The DAG model is used to represent the task dependencies. They aim to minimize makespan by improving resource utilization [8]. They have compared their results with two existing standard algorithms (Min–Min, and Max-Min). Ardagna *et al.* [6] have proposed resource allocation policies for the multi-cloud system using a game theoretical approach. They aim to solve the cost reduction problem correlated with the allocation of VMs in multiple infrastructures. Ardagna *et al.* have considered the homogeneous VMs in terms of their resource capacity (like CPU and RAM). Although the task allocation is based on certain SLAs, Farokhi has been well investigated the agreement in the cloud system [7]. He has demonstrated the SLA violation level in a multi-cloud environment during service delivery and also noted various challenges. Panda and Jana proposed two SLA-based task scheduling algorithms to improve various parameters in a multi-cloud environment [15], [37]. The expected time to compute matrix is used to propagate the expected time for each task in all virtual machines [16].

Gao *et al.* [17] have expressed a problem for multi-cloud environments that considers the clustering of services and their correlation effects of the CSPs within the cloud or among the clouds. Shi *et al.* [18] have analyzed the system performance as well as the budget control in multi-cloud on a global scale. Tchernykh *et al.* [19] have presented a multi-cloud storage architecture that consolidates multiple systems with various failure detection/recovery tools. They have proposed a multi-objective optimization mechanism to allocate workloads.

A Minimum Completion Time (MCT)-based approach (Max-Min scheduling [20]) first calculates the execution time of all input tasks on each VM. It then selects the row with the longest execution time. In the already selected row, the Max-Min approach selects VM with the shortest execution time and map tasks on that VM. On each scheduling decision, the VM load is updated. Tasks execution is started after mapping all the tasks on the respective VMs. Max-Min based scheduling heuristics prefer larger tasks and penalizing smaller tasks. Max-Average is Max-Min based, enhanced task scheduling heuristic proposed by Maipan-uku *et al.* [21], and complete their scheduling in two phases. In the first phase, the proposed approach uses the Max-Min approach for the calculation of expected completion time for a set of tasks. This approach chooses the largest task and identifies the VM(s) that execute the selected task in Minimum Completion Time (MCT). In case there is more than one candidate VMs that can perform the task with MCT, the proposed approach will assign the task to the resource with the least usage.

To identify the resource with the least usage, it calculates the average completion time of all the resources. However, if the average completion time is smaller as compared to the MCT of the smallest task, then it chooses the largest task and allocates the task to the VM that executes them in minimum time. Otherwise, select the smallest tasks and allocate the task to VM with MCT. This scheduling algorithm performs better for smaller datasets while high makespan and degrades throughput for larger tasks. Min-Min [22] scheduling heuristic reduces the makespan as compared to other tasks scheduling heuristics. However, the main issue with the Min-Min algorithm is poor resource utilization, which is one of the critical requirements of CSP. Authors in [23] have proposed an improved load balancing scheduling approach, namely Load Balanced Improved Min-Min (LBIMM). Moreover, this approach also can schedule users' jobs according to their priority, like job execution time and cost. The proposed approach provides a user with the option to select the type of services they need. They have considered makespan, resource utilization, and average completion time of high priority jobs as the performance parameter for the evaluation purpose. Chen *et al.* have proposed a scheduling algorithm to schedule dynamic and multiple tasks with deadlines [29]. Their proposed method manages the count of waiting for tasks on service occurrences to prevent the distribution of uncertainties. Jianmin Li has explained the usability of moldable parallel tasks in almost all areas [35]. He has presented a system model for the scheduling of moldable parallel tasks. In [38], authors proposed a cost model called Proportional-Shared Virtual Energy for IaaS providers based on CPU energy consumption.

The majority of the works as mentioned earlier consider that the entire network is attached to the single cloud that means; all Base Stations (BSs) in the network are connected to the same cloud). This structure seems to be unrealistic in practical conditions because of the distance factor for different BSs in the network [24]. Also, the connection of many BSs to a single cloud rises the computation load.

From the above study as in Table 1, we can perceive that to have an energy-aware multi-cloud network proposed by researchers are mostly following task allocation mechanism with the heterogeneous working environment. Moreover, task priority based on the weight of task properties has an essential capacity for effective task scheduling and enhanced system throughput. This paper proposes a task allocation policy to reduce the energy utilization by the multi-cloud network and also to minimize the execution time with the help of the DAG task model. The algorithm takes into account the task prioritization based on the deadline for each task and inserts each task in the queue based on the priority value. The heterogeneous cloud system performance can be improved by improving the task allocation scheme.

**TABLE 1.** Comparison of resources and parameters considered in the literature.

| Existing Work | Environment | Resources | | | Parameters | | | |
|---|---|---|---|---|---|---|---|---|
| | | CPU | RAM | Bandwidth | Energy | Makespan | Throughput | SLA |
| [1] | Single | ✓ | × | × | ✓ | ✓ | × | × |
| [2] | Single | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| [3] | Multiple | ✓ | ✓ | × | × | ✓ | × | ✓ |
| [4] | Multiple | ✓ | ✓ | ✓ | ✓ | ✓ | × | × |
| [5] | Single | ✓ | ✓ | × | ✓ | ✓ | × | × |
| [6] | Multiple | ✓ | ✓ | ✓ | ✓ | × | ✓ | ✓ |
| [7] | Multiple | ✓ | × | × | × | × | × | ✓ |
| [8] | Multiple | ✓ | ✓ | × | × | ✓ | × | × |
| [10] | Multiple | ✓ | × | × | ✓ | × | × | × |
| [12] | Multiple | ✓ | ✓ | ✓ | ✓ | ✓ | × | ✓ |
| [13] | Multiple | ✓ | ✓ | × | × | ✓ | × | × |
| [14] | Multiple | ✓ | ✓ | × | × | ✓ | × | × |
| [15] | Multiple | ✓ | ✓ | × | × | ✓ | × | ✓ |
| [16] | Single | ✓ | ✓ | × | ✓ | ✓ | × | ✓ |
| [18] | Multiple | ✓ | ✓ | ✓ | ✓ | ✓ | × | × |
| [20] | Single | ✓ | × | × | ✓ | × | × | × |
| [29] | Single | ✓ | × | ✓ | × | ✓ | × | × |
| This Work | Multiple | ✓ | ✓ | × | ✓ | ✓ | × | ✓ |

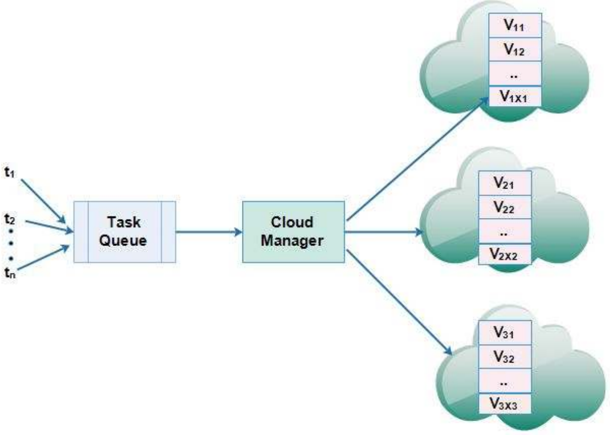

**FIGURE 1.** System model for multi-cloud networks.

## III. MULTI-CLOUD SYSTEM MODEL

The method of allocating virtual resources to the user requests while assuring optimal utilization of the cloud resources is known as task scheduling. The NP-hard nature of his scheduling problem is further becoming more challenging in the heterogeneous environment. Figure 1 shows the system model of multi-cloud networks.

## IV. PROBLEM STATEMENT

The task allocation problem in cloud computing is referenced as the problem of allocation of a large number of users requests assigned to a limited number of VMs. However,

**TABLE 2. Symbol and description.**

| Symbol | Description |
|---|---|
| $C$ | Cloud set |
| $C_k$ | $k^{\text{th}}$ Cloud |
| $d_i$ | Deadline of $\text{i}^{\text{th}}$ task |
| $EC$ | Energy Consumption |
| $ED$ | Set of Edges |
| $E_{pq}$ | Energy consumption of $q^{th}$ VM of $p^{th}$ cloud |
| $ET$ | Execution Time |
| $ETC$ | Expected Time to Complete |
| $L$ | Task length in million instructions |
| $M$ | Makespan |
| $N$ | Number of input tasks |
| $PS_{pq}$ | Processing speed of $q^{th}$ VM of $p^{th}$ cloud |
| $T$ | Task set |
| $TD$ | Task deadline |
| $t_i$ | $i^{\text{th}}$ task |
| $V$ | Set of VM |
| $VM$ | Virtual Machine |
| $VM_{ij}$ | $\text{j}^{\text{th}}$ VM of $\text{i}^{\text{th}}$ Cloud |
| $VM_j$ | $\text{j}^{\text{th}}$ VM |

for this work, we have assumed that the available virtual resource is sufficient for the execution of user requests. The task set $T$ has $n$ input tasks $\{t_1, t_2, \ldots, t_n\}$ and the cloud set $C$ has $k$ clouds $\{C_1, C_2, \ldots, C_k\}$. These $k$ clouds are geographically separated. The $p^{\text{th}}$ cloud has $X_p$ number of virtual machines, $1 \leq p \leq k$. Here, the DAG ($D = (T, ED)$) task model is followed to represent the task dependencies, where $T$ is the task set (called nodes) and $ED$ denotes a set of links (called edges). The link $ED_{i,j} = (t_i, t_j)$ represents the dependency over the task with a priority restriction showing that the execution of the task $t_i$ should be performed before the start $t_j$. An example of a DAG graph with four tasks $\{t_1, t_2, t_3, t_4\}$ is shown in Figure 2. From the figure, it is shown that the task $t_1$ is the entry task and can be executed independently. Task $t_4$ is the exit task to whom no other task depends on execution. The execution of $t_4$ can start only when the tasks $t_2$ and $t_3$ performed, and before the start of execution of $t_2$ and $t_3$, the task $t_1$ should complete its execution. Here, the $PRE\_EXE(t_2)$ and $PRE\_EXE(t_3)$ is $\{t_1\}$. Similarly, the $PRE\_EXE(t_4)$ is $\{t_2, t_3\}$. The symbols, along with their descriptions, are listed in Table 2.

Makespan ($M$) is the total time required by the multi-cloud network to execute all input tasks. To estimate makespan, the Expected Time to Complete ($ETC$) matrix is used. The $ETC$ matrix is shown in equation (2), as shown at the bottom of the next page. It represents the task heterogeneity. It contains the expected time to complete the task when going to execute in a VM. Here, $ETC_{i,pq}$ represents the expected time to complete the $i^{th}$ task in $q^{th}$ VM of $p^{th}$ cloud. The makespan calculation formula is as follows.

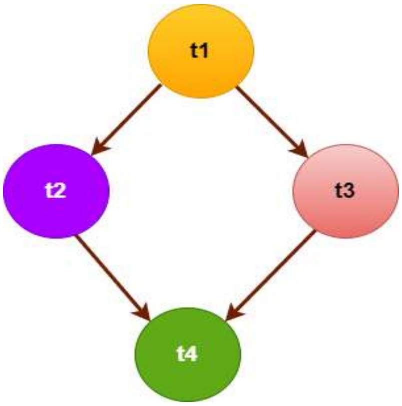

**FIGURE 2. Example of direct acyclic graph (DAG) with four tasks.**

$$M = Max_{p=1,2,\ldots,k}\left\{Max_{q=1,2,\ldots,X_p}\left\{ET_{pq}\right\}\right\} \tag{1}$$

- Here, $ET_{pq}$ represents the execution time of $q^{th}$ VM of $p^{th}$ cloud.

Energy consumption is another critical performance parameter of the multi-cloud system. An Energy Consumption ($EC$) matrix is used for the calculation purpose is shown in equation (4), as shown at the bottom of the next page. $EC_{i,pq}$ represents the energy consumption of $q^{th}$ VM of $p^{th}$ cloud when the VM will execute the $i^{th}$ task. In this work, the active state energy consumption of virtual machine is considered. $E_{pq}$ is the energy consumption of $q^{th}$ VM of $p^{th}$ cloud and is calculated using equation (3) [25], [26]. Cloud applications optimization has been discussed in [27].

$$E_{pq} = 10^{-8} \times \left(PS_{pq}\right)^2 \; Joules/MI \tag{3}$$

Here, $PS_{pq}$ is the processing speed of $q^{th}$ VM of $p^{th}$ cloud in terms of Million Instruction Per Second (MIPS).

The overall energy utilization of the multi-cloud network is the aggregate energy consumption of individual VMs.

## V. PROPOSED TASK ALLOCATION TECHNIQUE IN MULTI-CLOUD NETWORKS

This section confers the precise blueprints of the proposed task allocation algorithm in two phases. In the first phase, a set of virtual machines is selected with the help of the Expected Time to Complete ($ETC$) matrix based on the deadline. In the second phase, a VM is selected, which utilizes less energy by all the available VMs for the task with the help of the Energy Consumption ($EC$) matrix and that VM can have sufficient resources to execute the task.

Figure 3 shows the flow-chart of the *ETAMCN* algorithm. The multi-cloud network will receive the verified tasks, and then, the verified tasks are stored in a queue. Max-Heap data structure is used to store the tasks in the queue. The heap which represents a queue here is based on the priority

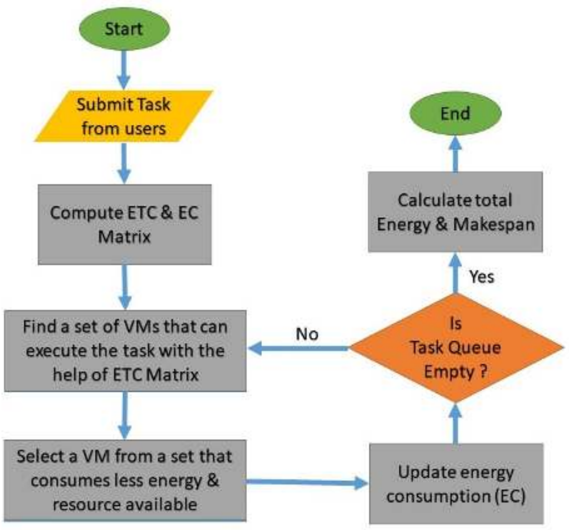
Start
Submit Task from users
Compute ETC & EC Matrix
Find a set of VMs that can execute the task with the help of ETC Matrix
Select a VM from a set that consumes less energy & resource available
Update energy consumption (EC)
Is Task Queue Empty ?
No
Yes
Calculate total Energy & Makespan
End

**FIGURE 3.** Flow-chart of the proposed algorithm.

values. The priority of a task is evaluated from task length divided by task deadline. The cloud manager controls the task queue and also monitors the VMs in the multi-cloud network. For each task, a set of VM is searched those can execute the task maintaining the SLA. The *ETC* matrix is used for searching this set of VMs. Then, with the help of *EC* matrix, a virtual machine is selected that consumes less energy from the set of VMs and resource available for the task execution. The energy consumption and execution time parameters are updated for the input task when the execution of the task is over by the VM with a proper voltage-frequency pair. The CSP frees the virtual resources and also updates the resources in the multi-cloud network.

The priority values are assigned based on the task length ($L_i$: $i^{th}$ task length in Million Instructions) and task deadline ($d_i$), where the priority of $i^{th}$ task is $L_i/d_i$. Based on the priority of tasks, the queue is built. The steps for Energy-aware Task Allocation in Multi-Cloud Networks (*ETAMCN*) are shown in Algorithm-1. Algorithm 1 functions by accepting the set of task ($T$), task deadline ($TD = \{d_1, d_2, \dots, d_n\}$, $d_i$ is the deadline of $i^{th}$ task), *ETC* matrix, *EC* matrix, and a finite set of voltage-frequency (*Vol-Fre*) pair as input. The proposed algorithm (*ETAMCN*) results in the overall energy consumption of the multi-cloud network, makespan value, and the allocation result (*ALL*).

Initially, the execution time ($ET_{pq}$) and energy consumption ($E_{pq}$) of all the VMs of each cloud is set to 0. To execute the $i^{th}$ task, the system will first search for the set of dependent tasks those to be completed before the $i^{th}$ task. So, if *PRE_EXE*($t_i$) exists, then the system will recursively execute those dependent tasks and after that $t_i$ will be executed. Through the recursive execution of dependent tasks, if any task has $\phi$ as the *PRE_EXE* value, then that task needs to be allocated cloud resources (VMs) with the help of Algorithm-2.

The Algorithm-2 finds the set of virtual machines that can execute the task $t_i$. If the deadline is meeting by verifying the ETC matrix for the execution in each VM, then include that VM to the set (see $VM_i$ is the set in Algorithm-2). After getting the set $VM_i$, the system will search for the VM that consumes minimum energy among the set of VMs.

$$\begin{array}{cc}
 & \begin{matrix} \overbrace{V_{11} \quad V_{12} \quad \dots \quad V_{1X_k}}^{C_1} & \quad\dots\quad & \overbrace{V_{k1} \quad V_{k2} \quad \dots \quad V_{kX_k}}^{C_k} \end{matrix} \\
\begin{bmatrix} t_1 \\ t_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ t_n \end{bmatrix} &
\begin{bmatrix}
ETC_{1,11} & ETC_{1,12} & \dots & ETC_{1,1X_1} & \dots & ETC_{1,k1} & ETC_{1,k2} & \dots & ETC_{1,kX_k} \\
ETC_{2,11} & ETC_{2,12} & \dots & ETC_{2,1X_1} & \dots & ETC_{2,k1} & ETC_{2,k2} & \dots & ETC_{2,kX_k} \\
 & & & & \cdot & & & & \\
 & & & & \cdot & & & & \\
 & & & & \cdot & & & & \\
 & & & & \cdot & & & & \\
 & & & & \cdot & & & & \\
ETC_{n,11} & ETC_{n,11} & \dots & ETC_{n,1X_1} & \dots & ETC_{n,k1} & ETC_{n,k2} & \dots & ETC_{n,kX_k}
\end{bmatrix}
\end{array} \qquad (2)$$

$$\begin{array}{cc}
 & \begin{matrix} \overbrace{V_{11} \quad V_{12} \quad \dots \quad V_{1X_k}}^{C_1} & \quad\dots\quad & \overbrace{V_{k1} \quad V_{k2} \quad \dots \quad V_{kX_k}}^{C_k} \end{matrix} \\
\begin{bmatrix} t_1 \\ t_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ t_n \end{bmatrix} &
\begin{bmatrix}
EC_{1,11} & EC_{1,12} & \dots & EC_{1,1X_1} & \dots & EC_{1,k1} & EC_{1,k2} & \dots & EC_{1,kX_k} \\
EC_{2,11} & EC_{2,12} & \dots & EC_{2,1X_1} & \dots & EC_{2,k1} & EC_{2,k2} & \dots & EC_{2,kX_k} \\
 & & & & \dots & & & & \\
 & & & & \dots & & & & \\
 & & & & \dots & & & & \\
 & & & & \dots & & & & \\
 & & & & \dots & & & & \\
EC_{n,11} & EC_{n,11} & \dots & EC_{n,1X_1} & \dots & EC_{n,k1} & EC_{n,k2} & \dots & EC_{n,kX_k}
\end{bmatrix}
\end{array} \qquad (4)$$

**Algorithm 1** Energy-Aware Task Allocation in Multi-Cloud Networks (*ETAMCN*)

```
Input: T, TD, ETC, EC, Vol-Fre;
Output: Energy Consumption ($E$), Makespan ($M$), ALL;
 1: Initialize $ET_{pq} \leftarrow 0$;
 2: Initialize $E_{pq} \leftarrow 0$;
 3: For each task, $t_i$ in T do
 4:     If (PRE_EXE($t_i$) $\neq \phi$)
 5:         For each task, $pt_i$ in PRE_EXE($t_i$) do
 6:     If (PRE_EXE($pt_i$) $== \phi$)
 7:         $V_{pq} = Allocate\_Task(pt_i)$;
 8:         $ALL[i] = V_{pq}$;
 9:         $ET_{pq} = ET_{pq} + ETC_{i,pq}$;
10:         $E_{pq} = E_{pq} + EC_{i,pq}$;
11:     Else
12:         Goto Step-5 to Execute PRE_EXE($pt_i$);
13:     End If
14: End For
15: Else
16:     $V_{pq} = Allocate\_Task(t_i)$;
17:     $ALL[i] = V_{pq}$;
18:     $ET_{pq} = ET_{pq} + ETC_{i,pq}$;
19:         $E_{pq} = E_{pq} + EC_{i,pq}$;
20:     End If;
21: End For
22: $M = Max_{p=1,2,\ldots,k}\{Max_{q=1,2,\ldots,Xp}\{ET_{pq}\}\}$;
23: $E = \sum_{p=1}^{k} \sum_{q=1}^{X_p} E_{pq}$
```

**Algorithm 2** Allocate_Task($t_i$)

```
Input: T, V, TD, ETC, EC;
Output: $V_{pq}$;
 1: $VM_i = \phi$;
 2: For each cloud do
 3: For each virtual machine
 4:     If ($ETC_{ipq} \leq d_i$)
 5:         $VM_i = VM_i \; U \; V_{pq}$;
 6:     End If;
 7: End For;
 8: End For;
 9: $V_{pq} = Min_{p=1,2,\ldots,k}\{Min_{q=1,2,\ldots,Xp}\{EC_{ipq}\}\}\&V_{pq}$
    Є$VM_i$
10: Repeat Step-9 until the $V_{pq}$ is available to execute $t_i$;
```

If the selected VM has not enough resources to execute that task, then that VM to be removed from the set $VM_i$ and once again select another VM with the help of step-9 of Algorithm-2. This algorithm returns the selected virtual machine to Algorithm-1. Then, the resulted values are updated in Algorithm-1.

## VI. PERFORMANCE EVALUATION

This section demonstrates the performance of the suggested task allocation algorithm.

**TABLE 3.** Simulation parameters.

| Parameters | Details |
|---|---|
| Number of Cloud | 16 |
| Number of Hosts | 40 |
| Hypervisor | Xen |
| Speed of VM (MIPS) | 200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000. |
| Main memory size (RAM) of VM | 0.5, 1, 2, 4, 8, 16 in GB |
| Four ranges of input task length (MI) | [3000-5000], [5000-7000], [7000-9000], [9000-11000]. |

### A. SIMULATION SETUP

To evaluate the performance of any scheduling heuristics for cloud computing, different approaches (i.e., experimental, analytical, and simulation) can be used. The performance of the proposed algorithms has been analyzed on the principles of the simulation outcomes produced through the CloudSim simulator [28]. This simulator is also used by researchers for the allocation of tasks in multi-cloud environment with different objectives [7], [31]. The experimental analysis is carried out with a single data center having 16 clouds. There are 40 hosts in total over all the clouds. In this work, at least a single host and at most 5 hosts is placed in a cloud. The overall resource capacity of each cloud is different, and in the same way, each cloud has a different number of physical hosts with heterogeneous resources. For the deployment of virtual machines, 'Xen' acts as the hypervisor for every physical host of the multi-cloud network. Based on the resource, each cloud has several VMs. Virtual machines are heterogeneous in terms of their resources. Each input task has a different resource requirement and considered different task length (in MI) and a different deadline for the completion of the task. Simulation parameters are illustrated in Table 3. Initially, the ETC matrix holds the expected time to compute all tasks in different VMs. Since the dependency among tasks is considered, the ETC matrix is updated by following the DAG graph. There are four ranges of task length considered, as shown in Table 3 for the simulation purpose. For each input task set, 25% of tasks are considered from each task group. The virtual machines are of different processing speed and main memory capacity which is illustrated in Table 3.

## VII. NUMERICAL RESULTS

Here, the results related to a different number of clouds and hosts is not shown because it does not make any change as the ideal state of VM is not considered. The *ETAMCN* algorithm is compared with Random allocation algorithm, Cloud Z-Score Normalization (*CZSN*) algorithm presented in [3], and multi-objective scheduling algorithm with Fuzzy resource utilization (*FR-MOS*) as in [4]. We have run these algorithms ten times for each set of input tasks representing an *ETC* matrix for two scenarios. In scenario-1, the number of user request is varied from 100 to 1000 in the interval of 100 while the number of VM is 80 over the multi-cloud network.

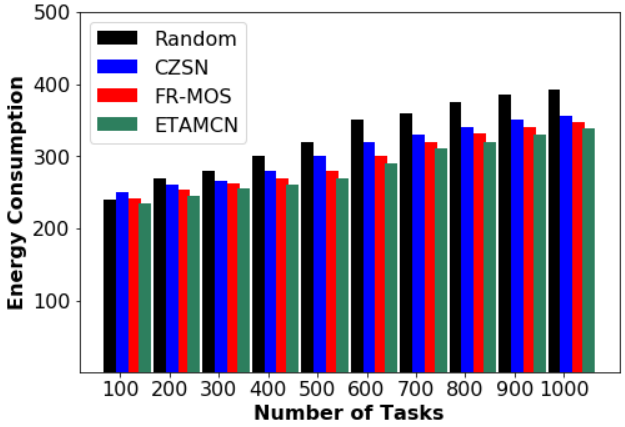

**FIGURE 4.** Energy Consumption Vs the number of Tasks where the number of VM is 80.

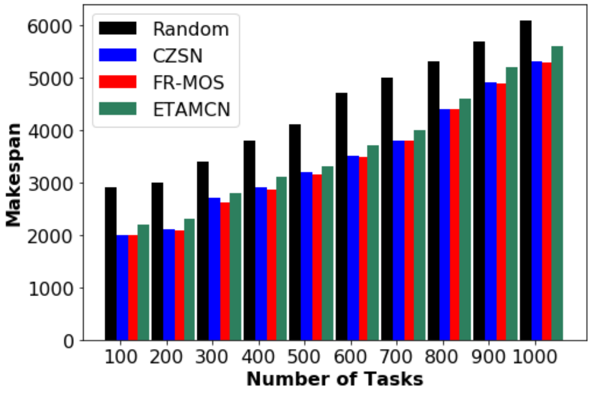

**FIGURE 5.** Makespan Vs the number of Tasks where the number of VM is 80.

In scenario-2, the number of user request is fixed at 500 while the number of VM changes from 20 to 140 in the interval of 20. For both the scenarios, the above mentioned four algorithms have been executed to compute energy consumption and makespan of the multi-cloud network. The results of an average of ten runs are shown in Figure 4 to Figure 7.

The average optimality of the allocation problem is drawn by the Random algorithm. Form Figure 5 and Figure 7, it is understood that our proposed algorithm performs close to *CZSN* and *FR-MOS* algorithms for makespan estimation of the multi-cloud network. Among the four algorithms, around 10.8% and 10% of improved makespan are shown in *FR-MOS* and *CZSN* respectively as compared to the random allocation algorithm. And, approximaly in the range of 1% to 2% better makespan value by *CZSN* and *FR-MOS* as compared to the proposed algorithm (*ETAMCN*), as shown in Figure 5. Similarly, around 8% and 7.5% of improved makespan are shown in *FR-MOS* and *CZSN* respectively as compared to the random allocation algorithm and around 1% to 2% by as compared to the proposed algorithm (*ETAMCN*), as shown

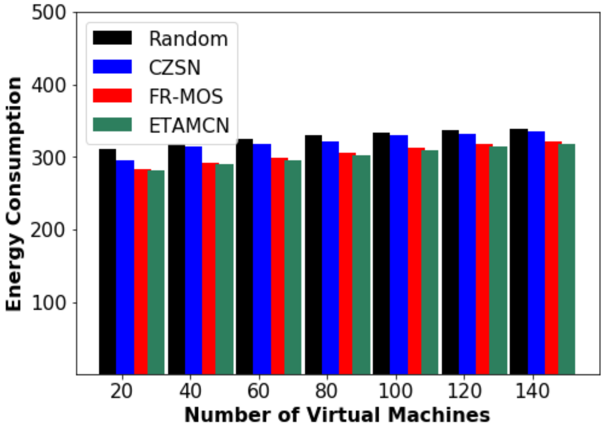

**FIGURE 6.** Energy Consumption Vs the number of VMs where the number of the task is 500.

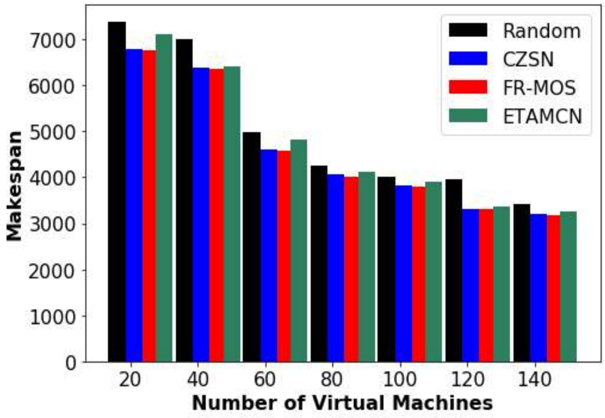

**FIGURE 7.** Makespan Vs the number of VMs where the number of the task is 500.

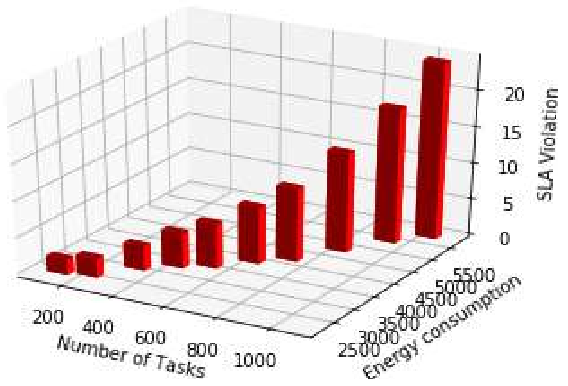

**FIGURE 8.** Energy Consumption Vs SLA Violation when the number of Tasks varies and the number of VM is 80.

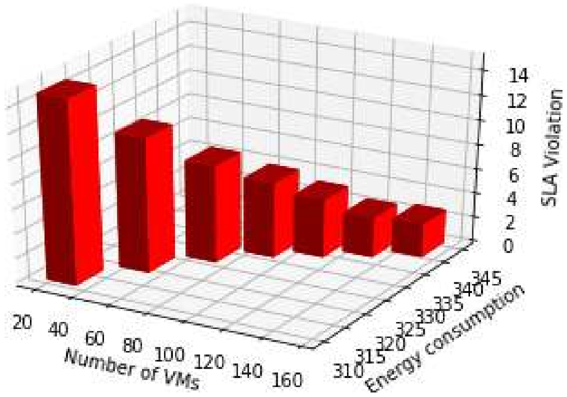

**FIGURE 9.** Energy Consumption Vs SLA Violation when the number of VMs varies and the number of Task is 100.

**TABLE 4.** Simulation summary of energy consumption with respect to energy and number of tasks.

| Scenario-1 | | | Scenario-2 | | |
|---|---|---|---|---|---|
| No. of Tasks | Energy Consumption | SLA Violation | No. of Tasks | Energy Consumption | SLA Violation |
| 200 | 2318 | 2.7 | 20 | 310 | 11 |
| 400 | 3125 | 4.9 | 40 | 318 | 6.2 |
| 600 | 3719 | 7.9 | 60 | 325 | 3.8 |
| 800 | 4607 | 13.8 | 80 | 330 | 3.3 |
| 1000 | 5632 | 24.2 | 100 | 336 | 2.1 |

in Figure 7. This proposed algorithm may not be showing good results in minimizing the makespan, the primary objective over here is to minimize the energy consumption without violating SLA. Among the four algorithms, around 14% of improved energy consumption is shown in ETAMCN as compared to the random allocation algorithm and around 6.3% and 2.8% as compared to the CZSN and FR-MOS respectively as shown in Figure 4. Similarly, around 10% of improved energy consumption is shown in ETAMCN as compared to the random allocation algorithm and around 5% and 1.2% as compared to the CZSN and FR-MOS respectively as shown in Figure 6. The energy parameter reduces more for the ETAMCN algorithm as compared the existing algorithms for both the scenarios as shown in Figure 4 and Figure 6. From the preceding results, we infer that *ETAMCN* delivers more reliable energy consumption and satisfiable makespan for distinct scenarios among the four algorithms.

The Service Level Agreement (SLA) is an essential parameter in the context of cloud services. A model is more apparent if that has less SLA-violation rate. The SLA-violation value depends on ‘‘How many tasks are not getting services from the cloud?’’. In this work, the SLA-violation of the proposed algorithm (ETAMCN) is observed for both Scenario-1 and Scenario-2.

For the SLA-violation concerning energy experimentation, two scenarios have been considered. In Scenario-1, the number of tasks varies, and in Scenario-2 the number of VMs kept fixed. The simulation results of scenario-1 and scenario-2 have depicted in Figure8 and Figure 9 respectively. The simulation summary is also listed in Table-4. From Figure 8 and Figure 9, it is noted that the average SLA-violation rate is lesser for ETAMCN. This is due to the selection of VM from the available cloud resources for each task.

## VIII. CONCLUSION

This research work appraised several renowned state-of-the-art approaches rigorously and considered a broadly used simulation platform, namely Cloudsim, for evaluation comparison. Energy-aware Task Allocation in Multi-Cloud Networks (*ETAMCN*) algorithm by maintaining the SLA for multi-cloud networks is proposed. The algorithm ETAMCN has been illustrated to understand the control flow of task execution. The simulation results on two scenarios and ten synthetic runs have shown that the ETAMCN algorithm outperforms the Random allocation algorithm, Cloud Z-Score Normalization (CZSN), and multi-objective scheduling algorithm with Fuzzy resource utilization (FR-MOS) algorithm regarding overall energy utilization and makespan of the multi-cloud network. The average energy consumption improved through the proposed algorithm (ETAMCN) is around 14%, 6.3%, and 2.8% against the random allocation algorithm, CZSN algorithm, and FR-MOS algorithm, respectively, when the number of input tasks varies. And when the number of VMs varies, the average energy consumption improved through the ETAMCN algorithm is around 10%, 5%, and 1.2% against the random allocation algorithm, CZSN algorithm, FR-MOS algorithm, respectively. However, the proposed algorithm (ETAMCN) has not considered any priority oriented users, such as task execution through reserve resource in the network which will be considered as our future work. The future work also aims to propose a task cum resource-aware scheduling approach that will exploit the nature of the presented workload and efficiently map on the available Cloud resources so that energy consumption will optimize.

## REFERENCES

[1] X. Lin, Y. Wang, Q. Xie, and M. Pedram, ‘‘Task scheduling with dynamic voltage and frequency scaling for energy minimization in the mobile cloud computing environment,’’ *IEEE Trans. Services Comput.*, vol. 8, no. 2, pp. 175–186, Mar. 2015.

[2] S. K. Mishra, D. Puthal, J. J. P. C. Rodrigues, B. Sahoo, and E. Dutkiewicz, ‘‘Sustainable service allocation using a metaheuristic technique in a fog server for industrial applications,’’ *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4497–4506, Oct. 2018.

[3] S. K. Panda and P. K. Jana, ‘‘Normalization-based task scheduling algorithms for heterogeneous multi-cloud environment,’’ *Inf. Syst. Frontiers*, vol. 20, no. 2, pp. 373–399, Apr. 2018.

[4] M. Farid, R. Latip, M. Hussin, and N. A. W. Abdul Hamid, ‘‘Scheduling scientific workflow using multi-objective algorithm with fuzzy resource utilization in multi-cloud environment,’’ *IEEE Access*, vol. 8, pp. 24309–24322, 2020.

[5] F. Juarez, J. Ejarque, and R. M. Badia, ‘‘Dynamic energy-aware scheduling for parallel task-based application in cloud computing,’’ *Future Gener. Comput. Syst.*, vol. 78, pp. 257–271, Jan. 2018.

[6] D. Ardagna, M. Ciavotta, and M. Passacantando, "Generalized Nash equilibria for the service provisioning problem in multi-cloud systems," *IEEE Trans. Services Comput.*, vol. 10, no. 3, pp. 381–395, May 2017.

[7] S. Farokhi, "Towards an SLA-based service allocation in multi-cloud environments," in *Proc. 14th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput.*, May 2014, pp. 591–594.

[8] S. K. Panda and P. K. Jana, "Efficient task scheduling algorithms for heterogeneous multi-cloud environment," *J. Supercomput.*, vol. 71, no. 4, pp. 1505–1533, Apr. 2015.

[9] R. Panuganty, "Apparatus and methods for managing applications in multi-cloud environments," U.S. Patent 9 319 286 B2, Apr. 19, 2016.

[10] B. Pang, F. Hao, Y. Yang, and D. S. Park, "An efficient approach for multi-user multi-cloud service composition in human–land sustainable computational systems," *J. Supercomput.*, vol. 76, pp. 1–18, Jan. 2020.

[11] L. Heilig, E. Lalla-Ruiz, and S. Voß, "Modeling and solving cloud service purchasing in multi-cloud environments," *Expert Syst. Appl.*, vol. 147, Jun. 2020, Art. no. 113165.

[12] A. Souri, A. M. Rahmani, N. J. Navimipour, and R. Rezaei, "A hybrid formal verification approach for QoS-aware multi-cloud service composition," *Cluster Comput.*, pp. 1–18, Nov. 2019.

[13] J. P. A. Neto, D. M. Pianto, and C. G. Ralha, "MULTS: A multi-cloud fault-tolerant architecture to manage transient servers in cloud computing," *J. Syst. Archit.*, vol. 101, Dec. 2019, Art. no. 101651.

[14] K. Indira and M. K. Kavithadevi, "Efficient machine learning model for movie recommender systems using multi-cloud environment," *Mobile Netw. Appl.*, vol. 24, no. 6, pp. 1872–1882, Dec. 2019.

[15] S. K. Panda and P. K. Jana, "SLA-based task scheduling algorithms for heterogeneous multi-cloud environment," *J. Supercomput.*, vol. 73, no. 6, pp. 2730–2762, Jun. 2017.

[16] S. K. Mishra, D. Puthal, B. Sahoo, S. Sharma, Z. Xue, and A. Y. Zomaya, "Energy-efficient deployment of edge dataenters for mobile clouds in sustainable IoT," *IEEE Access*, vol. 6, pp. 56587–56597, 2018.

[17] M. Gao, M. Chen, A. Liu, W. H. Ip, and K. L. Yung, "Optimization of microservice composition based on artificial immune algorithm considering fuzziness and user preference," *IEEE Access*, vol. 8, pp. 26385–26404, 2020.

[18] T. Shi, H. Ma, G. Chen, and S. Hartmann, "Location-aware and budget-constrained service deployment for composite applications in multi-cloud environment," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 8, pp. 1954–1969, Aug. 2020.

[19] A. Tchernykh, M. Babenko, N. Chervyakov, V. Miranda-Lopez, A. Avetisyan, A. Y. Drozdov, R. Rivera-Rodriguez, G. Radchenko, and Z. Du, "Scalable data storage design for non-stationary IoT environment with adaptive security and reliability," *IEEE Internet Things J.*, early access, Mar. 16, 2020, doi: 10.1109/JIOT.2020.2981276.

[20] Y. Mao, X. Chen, and X. Li, "Max–min task scheduling algorithm for load balance in cloud computing," in *Proc. Int. Conf. Comput. Sci. Inf. Technol.* Cham, Switzerland: Springer, 2014, pp. 457–465.

[21] J. Maipan-uku, A. Muhammed, A. Abdullah, and M. Hussin, "Maxaverage: An extended max-min scheduling algorithm for grid computing environtment," *J. Telecommun., Electron. Comput. Eng.*, vol. 8, no. 6, pp. 43–47, 2016.

[22] S. Rehman, N. Javaid, S. Rasheed, K. Hassan, F. Zafar, and M. Naeem, "Min-min scheduling algorithm for efficient resource distribution using cloud and fog in smart buildings," in *Proc. Int. Conf. Broadband Wireless Comput., Commun. Appl.* Cham, Switzerland: Springer, 2018, pp. 15–27.

[23] H. Chen, F. Wang, N. Helian, and G. Akanmu, "User-priority guided min-min scheduling algorithm for load balancing in cloud computing," in *Proc. Nat. Conf. Parallel Comput. Technol. (PARCOMPTECH)*, Feb. 2013, pp. 1–8.

[24] H. Dahrouj, A. Douik, O. Dhifallah, T. Y. Al-Naffouri, and M.-S. Alouini, "Resource allocation in heterogeneous cloud radio access networks: Advances and challenges," *IEEE Wireless Commun.*, vol. 22, no. 3, pp. 66–73, Jun. 2015.

[25] E. Grochowski and M. Annavaram, "Energy per instruction trends in Intel microprocessors," *Technol. Intel. Mag.*, vol. 4, no. 3, pp. 1–8, 2006.

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

[27] D. A. Shafiq, N. Jhanjhi, and A. Abdullah, "Proposing a load balancing algorithm for the optimization of cloud computing applications," in *Proc. 13th Int. Conf. Math., Actuarial Sci., Comput. Sci. Statist. (MACS)*, Karachi, Pakistan, Dec. 2019, pp. 1–6.

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

[29] H. Chen, X. Zhu, G. Liu, and W. Pedrycz, "Uncertainty-aware online scheduling for real-time workflows in cloud service environment," *IEEE Trans. Services Comput.*, early access, Aug. 21, 2018, doi: 10.1109/TSC.2018.2866421.

[30] M. H. Hilman, M. A. Rodriguez, and R. Buyya, "Multiple workflows scheduling in multi-tenant distributed systems: A taxonomy and future directions," *ACM Comput. Surv.*, vol. 53, no. 1, pp. 1–39, May 2020.

[31] R. Entezari-Maleki, S. E. Etesami, N. Ghorbani, A. A. Niaki, L. Sousa, and A. Movaghar, "Modeling and evaluation of service composition in commercial multiclouds using timed colored Petri nets," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 50, no. 3, pp. 947–961, Mar. 2020.

[32] S. K. Mishra, B. Sahoo, K. S. Sahoo, and S. K. Jena, "Metaheuristic approaches to task consolidation problem in the cloud," in *Proc. Resour. Manage. Efficiency Cloud Comput. Environ.* Hershey, PA, USA: IGI Global, 2017, pp. 168–189.

[33] K. Sagar Sahoo, A. Sarkar, S. K. Mishra, B. Sahoo, D. Puthal, M. S. Obaidat, and B. Sadun, "Metaheuristic solutions for solving controller placement problem in SDN-based WAN architecture," in *Proc. 14th Int. Joint Conf. e-Bus. Telecommun.*, 2017, pp. 1–10.

[34] A. Rasheed, P. H. J. Chong, I. H. Ho, X. J. Li, and W. Liu, "An overview of mobile edge computing: Architecture, technology and direction," *Technol. Direction. TIIS*, vol. 13, no. 10, pp. 4849–4864, 2019.

[35] J. Li, "Resource management for moldable parallel tasks supporting slot time in the cloud," *TIIS*, vol. 13, no. 9, pp. 4349–4371, 2019.

[36] S. Sahoo, A. Pattanayak, K. S. Sahoo, B. Sahoo, and A. K. Turuk, "MCSA: A multi-constraint scheduling algorithm for real-time task in virtualized cloud," in *Proc. 15th IEEE India Council Int. Conf. (INDICON)*, Dec. 2018, pp. 1–6.

[37] M. Tiwary, K. S. Sahoo, B. Sahoo, and R. Misra, "CPS: A dynamic and distributed pricing policy in cyber foraging systems for fixed state cloudlets," *Computing*, vol. 99, no. 5, pp. 447–463, May 2017.

[38] M. Hinz, G. P. Koslovski, C. C. Miers, L. L. Pilla, and M. A. Pillon, "A cost model for IaaS clouds based on virtual machine energy consumption," *J. Grid Comput.*, vol. 16, no. 3, pp. 493–512, Sep. 2018.

**SAMBIT KUMAR MISHRA** (Member, IEEE) received the M.Tech. and M.Sc. degrees in computer science from Utkal University, India, and the Ph.D. degree in computer science from the National Institute of Technology, Rourkela, India. He is currently working as an Assistant Professor with the Department of Computer Science and Engineering, SRM University, India. His research interests include cloud computing, edge/fog computing, the Internet of Things, and wireless sensor networks. He has published more than 40 research articles and articles in journals and conference of international repute. He is a member of the IEEE Computer Society, IETE, InSc, and CSI.

**SONALI MISHRA** received the M.Tech. degree in computer science and engineering from VSSUT, Burla, India. She has submitted her Ph.D. thesis in computer science and engineering at the National Institute of Technology at Rourkela, Rourkela, India. She is currently working as an Assistant Professor with the Department of Computer Science and Engineering, SOA University, Bhubaneswar, India. She has published more than 15 research articles and articles in international journal and conference. Her research interests include wireless sensor networks, artificial intelligence, machine learning, and the IoT. She is a member of the IETE and InSc.

**AHMED ALSAYAT** received the D.Sc. degree in computer science from Bowie State University, Bowie, MD, USA. He is currently an Assistant Professor with Jouf University, where he also serves as the Vice-Dean of the Computer and Information Sciences College. His specialties are machine learning, data mining, sentiment analysis, and the Internet of Things IoT. He has published several refereed research publications in this area. He has served in many editorial conferences.

**N Z JHANJHI** has great international exposure in academia, research, administration, and academic quality accreditation. He worked with ILMA University and King Faisal University (KFU) for a decade. He has 20 years of teaching and administrative experience. He has an intensive background of academic quality accreditation in higher education besides scientific research activities, he had worked a decade for academic accreditation and earned ABET accreditation twice for three programs at CCSIT, King Faisal University, Saudi Arabia. He also worked for the National Commission for Academic Accreditation and Assessment (NCAAA), Education Evaluation Commission Higher Education Sector (EECHES) formerly NCAAA Saudi Arabia, for institutional level accreditation. He also worked for the National Computing Education Accreditation Council (NCEAC). He is currently working as Associate Professor with Taylor's University Malaysia.

Dr. Jhanjhi has awarded as top reviewer 1% globally by WoS/ISI (Publons) recently for the year 2019. He has edited/authored more than 13 research books with international reputed publishers, earned several research grants, and a great number of indexed research articles on his credit. He has supervised several postgraduate students, including master's and Ph.D. He is an Associate Editor of IEEE Access, moderator of IEEE TechRxiv, Keynote speaker for several IEEE international conferences globally, External examiner/evaluator for Ph.D. and masters for several universities, a Guest editor of several reputed journals, a member of the Editorial Board of several research journals, and active TPC Member of reputed conferences around the globe.

**MAMOONA HUMAYUN** received the Ph.D. degree in computer architecture from the Harbin Institute of Technology, China. She has 12 years of teaching and administrative experience internationally. She is currently an Active Reviewer for a series of journals. She has supervised various master's and Ph.D. thesis. Her research interests include global software development, requirement engineering, knowledge management, cyber security, and wireless sensor networks.

**KSHIRA SAGAR SAHOO** (Member, IEEE) received the Ph.D. degree in computer science and engineering from the National Institute of Technology, Rourkela, India, in 2019, and the M.Tech. degree in information and communication technology from the Indian Institute of Technology, Kharagpur, India, in 2014. He is currently working as an Assistant Professor with the Department of IT, VNRVJIET, Hyderabad, India. His research interests include future generation network infrastructure, such as software defined networks, edge computing, and SD-IoT. He is a member of IEEE Computer Society and an Associate Member of Institute of Engineers (IE), India. He has published more than 50 articles in reputed journals and tire conferences, including IEEE Transactions on Network and Service Management (TNSM), IEEE Transactions on Intelligent Transportation Systems (TITS), IEEE Internet of Things (IoT), ACM TOMM, FGCS (Elsevier), IEEE System Journal, and ACM Mobicom.

**ASHISH KR. LUHACH** (Member, IEEE) received the Ph.D. degree from the Department of Computer Science, Banasthali University, India. He is currently working as a Senior Lecturer with The Papua New Guinea University of Technology, Papua New Guinea. He has more than a decade of teaching and research experience. He also worked with various reputed universities and also holds administrate experience as well. He has published more than 80 research paper in reputed journals and conferences, which are indexed in various international databases. He is member of CSI, ACM, and IACSIT. He is also edited various special issues in reputed journals and he is Editor/Conference Co-chair for various conferences. He is also editorial board members of various reputed journals.