IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

IEEE RELIABILITY SOCIETY SECTION

Received 27 May 2024, accepted 8 June 2024, date of publication 27 June 2024, date of current version 17 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3420173

RESEARCH ARTICLE

An Adaptive Threshold-Based Modified Artificial Bee Colony Optimization Technique for Virtual Machine Placement in Cloud Datacenters

FATEN KHALID KARIM[1], NITHYA REKHA SIVAKUMAR[1], SAMEER ALSHETEWI[2], AHMED ZOHAIR IBRAHIM[1], AND GEETHA VENKATESAN[3]
[1]Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University (PNU), P. O. Box 84428, Riyadh 11671, Saudi Arabia
[2]General Information Technology Department, The Excellence Services Directorate, Executive Affairs, Ministry of Defense, Riyadh 11564, Saudi Arabia
[3]Department of Computer Science, School of Applied Sciences, REVA University, Bengaluru 560064, India

Corresponding author: Nithya Rekha Sivakumar (NRRaveendiran@pnu.edu.sa)

This work was supported by Princess Nourah Bint Abdulrahman University (PNU), Riyadh, Saudi Arabia, through the Princess Nourah Bint Abdulrahman University Researchers Supporting Project under Grant PNURSP2024R300.

ABSTRACT The usage of cloud computing service platforms are exponentially growing to provide on-demand services for end-users for using advanced technologies. These platform services are achieved through resource virtualization to maximize the resource usage and minimize energy requirements. Energy consumption is a key factor for designing efficient and manageable cloud data centers. Optimal techniques are used for placing virtual machines in physical machines to reduce the energy consumption ratio of physical hosts. This paper proposes a novel efficient virtual machines placement algorithm for a cloud computing environment. This method exploits a modified artificial bee colony optimization algorithm for identifying under-utilized physical machines based on energy consumption and resource allocation charts. An adaptive threshold method is then proposed to select suitable threshold levels for energy consumption to identify under-utilized physical host machines. A comparative analysis with state of art methods is carried out by using the CloudSim 3.0 simulator. Simulation results show the superiority of our method, able to achieve the highest accuracy values of 97.2% for accuracy and of 97.9% for precision rate, thus confirming the efficacy of our approach for virtual machine placement in cloud environments.

INDEX TERMS Adaptive threshold, modified artificial bee colony optimization, VM placement, resource management, virtual service handling, optimization.

I. INTRODUCTION

Cloud computing environment provides many benefits, such as minimizing computation cost involvement, maximizing the resource utilization, on hand service, and high scalability. This computing environment has evolved from the combination of virtualization, distributed, utility-based computing technology to facilitate scalability and availability features. Virtualization is a process of creating a virtual service machine to assign physical resources in an optimized way. Virtualization can be attained by using software known as virtual machine monitor (VMM) or hypervisor, and this VMM is responsible for creating and managing virtual environments. The primary goal of virtual machine (VM) environment is to improve the efficiency in resource utilization, in flexibility cloud environments and to provide an easy way to manage computing resources. The resource virtualization in cloud environment provides the following benefits:

a. Efficient resource utilization. Virtualization achieves better utilization of physical resources and reduces the ideal time to wait for getting the required resource.
b. Scalability and flexibility. Virtual environment is more flexible for scaling the resource requirements depending on demand, and VM can be added or removed virtually.

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

c. Isolation. VM is operated independently and it provides a strong isolation environment.

Cloud environments are designed with basic assumptions about quality of service (QoS), user resource needs, availability of resources, utilization time, energy consumption for each resource, and other computing related factors [1], [2]. The general cloud data center process management and service provisions are discussed in the figure 1.

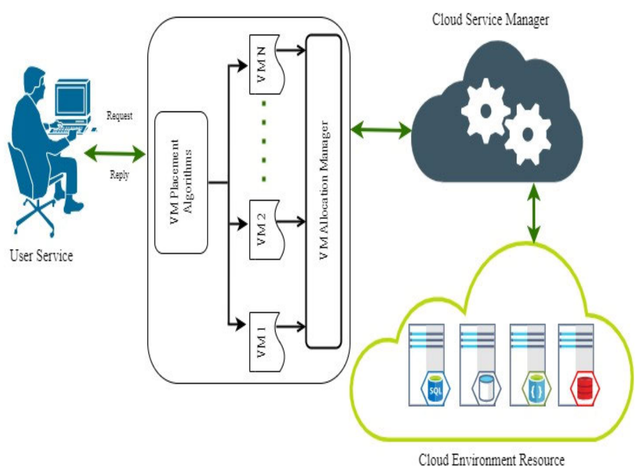

FIGURE 1. Cloud datacenter general working model.

Effective utilization of resources in cloud environment is achieved through an efficient virtual machine placement technique. The VM placement is an important process of identifying suitable physical machines for accommodating different kinds of applications. Each physical machine will maintain a record of resource allocation and energy consumptions chart for selecting a suitable VM for the next placement [3]. Each VM request is processed individually based on the resource (CPU, memory, disk, and bandwidth) request and many VM placement techniques are designed based on the resource availability and requirements. The resource sharing techniques are accepting as many as possible requests from VM, and allocation policies are maintained based on the available VMs. Here, the resources are virtualized to minimize the energy usage ratio and cost.

Energy consumption ratio of a cloud datacenter is a direct factor measured from the number of users' requests and resource needs. Many research articles are published to improve the energy usage ratio based on both the overload and underload of physical machines in cloud datacenters [4]. The main target of a VM placement technique is to achieve QoS and minimize the energy consumption rate through the minimization of the resource utilization [5]. The energy consumption may be achieved by reducing the total power consumed by the computation, information processing, transferring, and storage. In general, VM placement is done based on the holding possibility of VM over physical machine. If the physical machine is ready to accommodate the VM then place it, otherwise find the next optimal physical host. Power consumption can be minimized by decreasing the number of physical machines (PMs) and increasing the number of VMs [6] and activating idle VM for enhancing resource utilization [7]. The tasks can be arranged in a sorted order to be assigned to physical machines to avoid overload or underload in cloud datacenters and this process can reduce energy consumption. The migration process from a virtual machine to another VM and consignment of physical machines to VMs requires additional power and this needs to be take into account [8], [9].

Addressing some of these issues, this paper proposes a novel efficient placement technique for virtualization in cloud service environments. This technique is designed based on adaptive threshold based modified artificial bee colony optimization mechanism. In this method, adaptive threshold measurement is used for detecting the over or under-utilized physical resources in cloud datacenters. Static threshold methods may lead to low efficiency in load balancing and to high energy consumption for resource allocation in physical machines. The proposed method exploits the resource utilization factor with maximum and minimum utilization. The threshold value is measured from the marginal mid-point from maximum and minimum utilization of resources and energy consumption values.

A. RESEARCH GAP

Recently many research articles have been published for VM placement strategy based on energy efficiency using nature inspired optimization techniques. These schemes are still suffers with following limitations,

1. Most of the optimization methods consider throughput and resource usage as the metrics for measuring the energy consumption. These methods are not taking resource ideal time into account for selecting under-utilized physical machine.
2. Nature inspirited optimization techniques are suffers with local optima problems for obtaining optimal result in VM placement
3. Generally, nature inspired optimization techniques require an adaptive global optimization factor to achieve efficient results. Still, some VM placement methods are not working properly for selecting an efficient physical machine due to global and local optimum point selection.

1) RESEARCH CONTRIBUTION

We have made the following contributions in the proposed virtual machine placement technique:

1. The proposed method uses a modified and improved artificial bee colony optimization (IABCO) technique for selecting an optimal physical machine for placing a virtual machine. This method is exclusively designed for sustainable energy usage for physical machine selection and virtual service usage.

2. The proposed method uses an adaptive threshold approach for selecting an optimal threshold level for determining suitable energy values. An efficient selection of threshold value is used to prepare a list of under-utilized physical machines and this will be used as next stage population for VM placement.
3. We use CloudSim 3.0 for implementing the proposed method and the performance evaluation is carried out with different set of VMs and PMs.
4. The VM placement accuracy for the proposed method is measured base on average throughput and average resource ideal time.

The main contribution of the proposed virtual machine placement technique is to address the resource allocation issues in the existing VM placement algorithms. This paper shows a main contribution towards to achieve an efficient VM placement for versatile environment by using nature inspired optimization technique.

The rest of this paper is organized as follows. Section II presents a detailed review on placement of virtual machines over physical machine. A detailed discussion about the methodology and dataset used for conducting experiments about the proposed placement technique is given in Section III. The adaptive threshold-based modified artificial bee colony optimization technique is detailed in Section IV. Section V discusses about the performance evaluation for the proposed VM placement technique, carrying out a comparative analysis with state-of-the-art methods. Finally, Section VI presents the conclusion of our paper.

II. RELATED WORKS

Many applications are designed for massive amounts of information and process handling, such as health, ecommerce, education, and record maintenance. These applications are designed based on the concept of virtualization. This is a crucial section in resource usage and power consumption in cloud service environment. The service request handling in virtualization is considered as a dynamic load balancing and VM migration. Recently, many research articles are published by the researchers in VM placement by using evolutionary and population-based algorithms. The VM placement algorithms are designed in two different scenarios, like static and dynamic [10]. The role of VM placement is to achieve the desired objective from the available physical machines [11]. Cloud environment suffers with security related issues for maintaining confidentiality in customer related information sharing [12]. The dynamic virtualization methods are improving the efficiency of resource utilization, and attackers could not target a single machine [13]. A technique for placing a VM and load assignment is presented in [14]. This method reduces hardware or resource requirements for VMs through a mathematical model. A hybrid optimization algorithm is presented in [15] for resolving issues related to energy consumption in cloud datacenters. This method is designed exploiting random forest and genetic algorithm (GA). The authors have developed an algorithm with the objective of reducing the energy consumption by applying excellent load balancing method.

A technique for placing VM is developed based on utility function for handling placement related issues [10]. This method aims at improving energy utilization and violation of service level agreement (SLA). Dong et al. [16] presented a parallel distributed GA for VM assignment in cloud based service environment. This algorithm is executed on different physical hosts in a distributed and parallel fashion. Liu et al. [17] presented a comprehensive model and articulate the placement problem as a multi-objective-based optimization with constrained function. They designed a strategy by combining VM migration cost, heat recirculation in server side, and energy consumption. Li et al. [18] presented an approach for VM migration by using multi-resource collaboration optimization control. This method uses Gaussian model for estimation of probability in multi-resource utilization.

Yang et al. [19] presented a method for reliable VM placement based on particular integer-based non-linear program and a heuristic technique for solving the issues. An architectural framework as well as the working principles for establishing an efficient energy aware cloud environment is defined in [20]. Zhou et al. [21] proposed redundant placement of VM for improving the dependability of cloud server using optimization approach. They have proposed three different algorithms. The first one is designed to choose a suitable set of VM hosts from the larger set of candidate physical host servers. The next algorithm measures an efficient strategy for selecting main and holdup VMs from the selected hosts. The third algorithm works to solve task-to-VM reallocation problem, which is articulated from the findings from maximum weight matching method.

Hormozi et al. [22] designed a data structure to decrease the difficulty of the fitness computation by using quadratic linear formulation. Peake et al. [23] presented a parallel ant colony optimization technique for VM placement. This method is designed to effectively exploit parallelization technique. Liu et al. [24] presented a VM placement technique based on evolutionary computing method for minimizing the possible numbers of active physical host machines. They have developed an approach by combining ant colony optimization technique with migration local search techniques and exchange order. Marahatta et al. [25] presented an efficient algorithm for VM placement using adaptive thresholding to identify over- and under-utilized host machines for reducing the energy usage rate and service level agreement damages.

Jiang et al. [26] presented a technique for task scheduling based on thresholds for efficient scheduling mechanism for effective utilization of resources. They have proposed a particle swarm based optimization algorithm to select the best set of resource utilization in VM placement. Malik et al. [27] presented a dynamic VM alliance method based on energy consumption. This method migrates VM based on the satisfying constraints for multiple types of resources being overloaded. This method has been designed based on the foraging behavior of artificial bee colony optimization algorithm. Upadhyay et al. [28] presented a framework by using multi-objective GA and Bernoulli simulation. Sudhakar et al. [29] presented an approach for optimal VM placement by combining the Sine-Cosine algorithm and Salp Swarm optimization technique for discrete multi-objective and chaotic functions.

Verma and Bhatt [30] presented an approach for VM alliance method based on energy and temperature. This approach uses two heuristic based energy and temperature aware (HET) and meta-heuristic based FireFly energy and temperature aware (FET) consolidation algorithms. Verma et al. [31] developed a model for placing VM by using decision making process and this decision is taken based on three criteria of placement time, power usage and resource ideal times. This method has been designed to select a suitable PM by satisfying minimum values of these three parameters. The fitness function for this method is developed by using the following three optimization algorithms, particle swarm optimization with levy flight, flower pollination optimization, and a hybrid algorithm that combines these two algorithms.

III. METHODOLOGY

This section discussed about the artificial bee colony optimization algorithm with basic working principles and cloud datacenter resource management methods in detailed way.

A. ARTIFICIAL BEE COLONY ALGORITHM

Artificial Bee Colony (ABC) Algorithm is a nature inspired swam-based meta-heuristic algorithm. This algorithm has been designed based on the inspired behavior of honey bees [1], and has been proposed by Tereshko and Loengarov [2] from the foraging behavior of honey bee colonies. The ABC algorithm contains three important components. The first two components are working as the role of employee and onlooker in the process of foraging bees. These foraging bees are taking the role of searching for rich food sources to collect honey. The third component maintains a set of bees as scouts, carrying a random process for finding the food positions. The solutions from the searching space consist of a set of optimization factors, representing the source position. The participated employed bees are equivalent to the number of food source. The rich food source is known by its fitness value and this value will be associated with its position. The employed bees are taking the accountability for examining the sources of food based on the fitness values and sharing the collected information with onlooker bees for identifying the best solution. The participated employed bees, onlooker bees, and number of solutions in the populations are the same. The following sections discuss about the phases involved in the ABC Algorithm.

Initial Phase: Let $X = \{x_i\}, 1 \leq i \leq SN$ be the initialized population, created randomly in the entire field. The source of food represented as x_{ij} in the initial phase is calculated as follows:

$$x_{ij} = x_j^{MIN} + \alpha \times \left(x_j^{MAX} - x_j^{MIN}\right) \tag{1}$$

Here $1 \leq i \leq SN \; and \; 1 \leq j \leq D$

Employed Bees Phase: This phase is used to generate new solutions V_i by using a random neighborhood searching process over the available population x_i by using the folllwing:

$$V_{ij} = x_{ij} + \alpha \times \left(x_{ij} - x_{kj}\right) \tag{2}$$

Here, k and j are selected randomly from SN (i.e., the number of solution population) and D (i.e, the dimensional vector) and the condition ($k \neq i$). If the V_i produces excellent results than x_i, x_i is replaced with V_i. The counter value will be reset or increased by 1 based on the result acceptance.

Onlooker Phase: This phase applies a probability of selection to find sources for food base fitness ratio. The probability values can be measured as follows:

$$Probability_i = \frac{Fitness_i}{\sum_{i=1}^{SN} Fitness_i} \tag{3}$$

Then, the fitness value for the sources of food source x_i is calculated as follows:

$$Fitness_i = \begin{cases} \dfrac{1}{1 + f\,(x_i)}, & if \; f\,(x_i) \geq 0 \\ 1 + |f\,(x_i)|\,, & Otherview \end{cases} \tag{4}$$

From (3) and (4), it is easy to infer that the source of food with the maximum fitness value has the highest probability to be selected by the bees participating in onlooker lists.

Scout Bee Phase: The onlooker bees that select their food sources randomly are known as scouts. The employed bee's solutions could not be enhanced through maximum attempts of trails (Maximum Limit or Abandonment Criteria) by the ABC algorithm.

B. ENVIRONMENTAL DESIGN FOR CLOUD DATACENTER

The cloud datacenters are logically represented as a physical infrastructure made up of physical machines or hosts. The datacenters consist of as set of physical host machines identified as ($\mathcal{H}_i$, $1 \leq i \leq N$). The host machines are virtualized to generate VMs to accommodate users' defined tasks. The host machines are identified in two modes: if the machine is in a running condition at datacenter, then it is defined as active host; otherwise as inactive host. The active host machine is represented as:

$$\mathcal{H}_i = \left(Avl_{P_i}, Avl_{Mem_i}, Avl_{Stor_i}, Avl_{BW_i}\right).$$

Here, the processing computation cost is defined as P_i, memory and disk access are defined as Mem_i and $Stor_i$, respectively, and the bandwidth requirement is defined as BW_i [36]. The virtual machines are defined as $\left(VM_j, 1 \leq i \leq N\right)$ and each VM requires a certain number of resources from the datacenter and defined as follows:

$$VM_j = \left(Req_{P_i}, Req_{Mem_i}, Req_{Stor_i}, Req_{BW_i}\right).$$

Algorithm 1 Artificial Bee Colony Algorithm

1. Generate Random Initial Populations by using the equation (1)
2. Compute Fitness value for each population $Fitness_{x_i}$, $1 \leq i \leq SN$
3. Initialize the counter $Counter = 1$
4. Do
   a. For each employee bee from E_{bee_i}, $1 \leq i \leq SN$
      i. Compute V_{ij} using equation (2)
      ii. Compute Fitness value $Fitness_{V_i}$
      iii. Apply Greedy Selection Process over x_i, V_i
   b. End For
5. Calculate the probability values $Probability_i$ for the solution x_i using equation (3)
   a. For each onlooker bee from OL_{bee_i}, $1 \leq i \leq SN$
      i. Choose a solution x_i with support of probability value of $Probability_i$
      ii. Create new solution V_j
      iii. Computes its fitness value $Fitness_{V_j}$
      iv. Apply Greedy Selection Process over x_i, V_j
   b. End For
6. If an abandoned solution attained then Substitute it with new set of solution (produced randomly by equation (4))
7. Else No replace
8. Available list of best solutions keep the track and increment the counter
9. While ($Counter \leq Max_{iterations}$)

Each virtual machine has to be allocated to the suitable host based on the adaptive threshold value. The total number of operating statements involved in each physical machine is calculated as follows:

$$Process_{\mathcal{H}_i} = \sum_{j=1}^{n} MIPS_{VM_j} \tag{5}$$

$$Memory_{\mathcal{H}_i} = \sum_{j=1}^{n} MAccess_{VM_j} \tag{6}$$

$$Disk_{\mathcal{H}_i} = \sum_{j=1}^{n} DAccess_{VM_j} \tag{7}$$

$$BW_{\mathcal{H}_i} = \sum_{j=1}^{n} TComm_{VM_j} \tag{8}$$

Here n is the total number of VMs accommodated by the particular physical machine, $MIPS_{VM_j}$ is the total number of instructions involved in VM_j. $MAccess_{VM_j}$means the total number of memory accesses in VM_j, $DAccess_{VM_j}$ represents the total number of disk accesses in VM_j, and $TComm_{VM_j}$ is the total number of communications in VM_j.

The energy consumption for individual operation is defined as $\alpha_P, \alpha_M, \alpha_D,$ andα_{BW}. These values are used to measure the total energy requirement for completing one particular virtual machine. The energy consumption for each VM is calculated as follows:

$$TEng_{VM_j} = \left(\alpha_P.MIPS_{VM_j} + \alpha_M.MAccess_{VM_j} + \alpha_D.DAccess_{VM_j} + \alpha_{BW}.TComm_{VM_j}\right) \tag{9}$$

The total energy consumption for each physical machine $\mathcal{H}_i$ is calculated as:

$$T_{Eng}^{\mathcal{H}_i} = \sum_{j=1}^{m} TEng_{VM_j} \tag{10}$$

$$Avg_{Eng}^{\mathcal{H}} = \frac{\sum_{i=1}^{N} T_{Eng}^{\mathcal{H}_i}}{N} \tag{11}$$

If the total energy consumption is less than the threshold value U_{load}, then the particular physical machine is an under-utilized physical machine. If the value of $T_{Eng}^{\mathcal{H}_i}$ is higher than the threshold O_{load}, then the physical machine $\mathcal{H}_i$ is a PM overload with virtual machines. The U_{load} and O_{load} are calculated as follows:

$$Thres_{U_{load}} = \left[Avg_{Eng}^{\mathcal{H}} - \sigma\right] \tag{12}$$

$$Thres_{O_{load}} = \left[Avg_{Eng}^{\mathcal{H}} + \sigma\right] \tag{13}$$

Here, σ is calculated by:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}\left(T_{Eng}^{\mathcal{H}_i} - Avg_{Eng}^{\mathcal{H}}\right)^2}{N}} \tag{14}$$

IV. PROPOSED VM PLACEMENT METHOD

The working principle for the proposed VM placement method is shown in the figure 2. The proposed VM placement method is developed by using the modified artificial bee colony optimization algorithm. Figure 3 explains the steps involved in the proposed virtual machine placement method by using modified artificial bee optimization algorithm. The existing ABC optimization algorithm is modified for preparing the enriched food source as the suitable physical machine. In the initial setup phase of the modified ABC optimization technique, the algorithm prepares the initial set of population as the available list of physical machines based on the available set of resources. This list will be known as the available food sources for honey bees to collect food from the nearest food source. The fitness function calculates the fitness value for each physical host and selects suitable PM based on fitness value. The physical machines are selected based on the selection criteria. If the selection criteria is not satisfied, then the nearest possible PM will be investigated until all the possible nearest PM are investigated. This modified ABC optimization algorithm returns the final list of suitable list of physical machines. The following algorithm 2 is used to identify the under or over utilization of the physical machines.

The following algorithm 3 explains the working principle of the proposed virtual machine placement algorithm using adaptive threshold based artificial bee colony optimization algorithm.

V. PERFORMANCE AND RESULT ANALYSIS

This section presents a detailed analysis of the performance and results obtained from the experimental setup. The proposed VM placement technique is simulated by using the CloudSim tool [21]. CloudSim is an extensively accepted

Algorithm 2 Initial Stage Population Generation
Input: Set of Physical Machines $\mathcal{H}_i$, $1 \leq i \leq N$
Output: List of Under-utilized Physical machines

1. For each physical machine $\mathcal{H}_i$
2. For each virtual machine VM_j in $\mathcal{H}_i$
   a. Compute total energy consumption for VM_j (equation 9)
3. End For
4. Compute total energy consumption of each $\mathcal{H}_i$ (equation 10)
5. *If* $\left(T_{Eng}^{\mathcal{H}_i} \geq Thres_{O_{load}}\right)$ then $\mathcal{H}_i$ is overload
6. *If* $\left(T_{Eng}^{\mathcal{H}_i} \geq Thres_{U_{load}}\right)$ then $\mathcal{H}_i$ is under-utilized and prepare under-utilized physical machine list as U_{Utiliz}^{PM}
7. End For
8. Prepare a list of under-utilized physical machines as population list

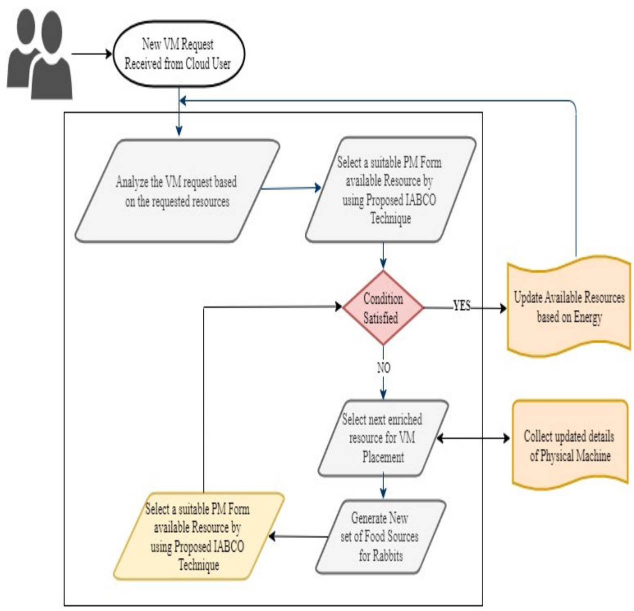

FIGURE 2. Proposed VM placement method working flow diagram.

simulation tool for designing a cloud service computing environment. CloudSim provides an environment to support for handling on-demand resources allocation. It provides a facility for simulating various service loads or workloads on various time slots. CloudSim [22] is a Java 8 platform dependent simulation tool and this tool allows various Cloud Centre services to be modelled and simulated. This simulator helps in supporting various problems related to research, evaluation, and validation of algorithms for different sets of purposes.

The energy consumption rate for each resource is assigned with a cost factor of 0.15, 0.07, 0.05, and 0.05 for processing element, memory access, disk storage, and transfer task, respectively. The proposed VM placement method is evaluated based on the following process: initial set of

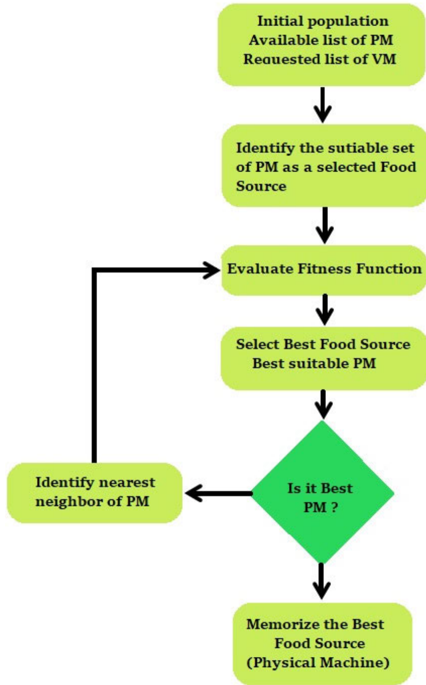

FIGURE 3. Proposed modified ABC method for virtual machine placement technique.

physical machines with standard set of physical resources with four types. The proposed method is implemented by using CloudSim 3.0 and the performance evaluation for the proposed approach is measured based on four standard parameters: precision, recall, F1 score, and accuracy. This performance evaluation is evaluated according to the number of physical machines participating and dynamic request generated as virtual machines. The initial set of VMs is assigned to physical machines based on the procedure discussed in Section III. The success and accuracy of the method is measured based on the virtual machine throughput, energy consumption, and ideal time of resource wastage of a physical machine.

The throughput for each VM is measured as follows:

$$Troughput_{VM_j} = IWT_{VM_j} + \left(Time_F^{VM_j} - Time_S^{VM_j}\right) \tag{15}$$

$$IWT_{VM_j} = \left(Time_S^{VM_j} - Time_{Entry}^{VM_j}\right) \tag{16}$$

$$AvgTput_{\mathcal{H}_i} = \frac{\sum_{j=1}^{m} Troughput_{VM_j}}{m} \tag{17}$$

$$AvgIWT_{\mathcal{H}_i} = \frac{\sum_{j=1}^{m} IWT_{VM_j}}{m} \tag{18}$$

$$Accuracy_{\mathcal{H}_i} = \left(\frac{\sum_j^m \left(AvgTput_{\mathcal{H}_i} - Troughput_{VM_j}\right)}{m}\right) \tag{19}$$

$$Accuracy = \left[\frac{\sum_{i=1}^{N} Accuracy_{\mathcal{H}_i}}{N}\right] \tag{20}$$

Algorithm 3 Proposed AT Based Modified ABC Optimization Algorithm for VM Placement

Input: New VM request
$VM_j^{new} = (Req_{P_i}, Req_{Mem_i}, Req_{Stor_i}, Req_{BW_i})$

Output: Identify suitable physical machine $\mathcal{H}_i$, $1 \leq i \leq N$

1. Identify the required list of resources for virtual machine VM_j^{new}
2. Compute minimum energy requirement for completing virtual machine VM_j^{new} (**equation 9**)
3. Apply algorithm 1 and identify the suitable list of physical machines
4. The suitable set of physical machines is identified by using ***algorithm 2*** and this will be given as an initial set of populations.
5. The suitable list of physical machines are selected based on the availability status of resources in physical machines
6. For each physical machine fitness value is calculated by using **equation (10)** and **(11)**
7. Apply modified ABC optimization algorithm (***Algorithm 3***) for the initial stage of segmentation according to the fitness values.
8. Identify the suitable physical machine as follows,
   a. Do
   b. For each physical machine from $\mathcal{H}_i$, $1 \leq i \leq N$
      i. Compute $T_{Eng}^{\mathcal{H}_i}$ using equation (10)
      ii. Compute new threshold value $Thres_{U_{load}}$ for under-utilized physical machine using equation (12)
      iii. Apply Greedy Selection Process over $Thres_{U_{load}}$, $T_{Eng}^{\mathcal{H}_i}$
   c. Compute energy required rate for virtual machine VM_j by using equation (9)
   d. *If* $\left(Thres_{U_{load}} \geq \left(T_{Eng}^{\mathcal{H}_i} + TEng_{VM_j}\right)\right)$ then
      i) Virtual machine VM_j will be assigned to physical machine$\mathcal{H}_i$.
      ii) Return physical machine$\mathcal{H}_i$
   e. *End if*
   f. While ($Max_{iterations}$)
9. Wait for suitable physical machine

$$Precision_{\mathcal{H}_i} = \sqrt{\frac{\sum_{j=1}^{m}\left(Troughput_{VM_j} - AvgTput_{\mathcal{H}_i}\right)^2}{m}} \tag{21}$$

$$Precision = \frac{\sum_{i=1}^{N} Precision_{\mathcal{H}_i}}{N} \tag{22}$$

The experimental environment is created by using CloudSim 3.0 cloud simulator. We have created three different scenarios: in the first scenario, we have used 30 Physical Machines, in the second scenario 40 Physical Machines, and in the third scenario 60 Physical Machines. Each Physical Machine is designed with the following specifications of 1 TB of storage, 8 GB of RAM and 1 CPU core of 2000 or 3000 MIPS or 16 GB of RAM and 1 CPU core of 2000 or 3000 MIPS. We have used a single variety of Memory and 2 varieties of CPU. The testing environment is designed with 256 user requests with different set of resource requests for each virtual machine. The following methods based on optimization techniques are used for carrying out the comparative analysis: Marahatta et al. [25] (P1), Soltanshahi et al. [37] (P2), Devaraj et al. [38] (P3), Chandran et al. [39] (P4), Rahim et al. [40] (P5), and Vasudevan et al. [41] (P6).

Marahatta et al. [25] proposed a classification-based resource allocation for VM, Soltanshahi et al. [37] presented energy aware resource allocation using krill herd algorithm. Devaraj et al. [38], proposed a hybrid firefly method and enhanced multi-objective partied swarm optimization method for energy efficient load balancing in cloud service computing environment. Chandran et al. [39] proposed GA-based tabu search technique for optimum energy aware allocation for cloud datacenters. Rahim et al. [40] presented an exploiting heuristic algorithm for utilizing power management controllers with renewable energy sources. Vasudevan et al. [41] presented a profile-based resource management using a repairing GA.

We have used the aforementioned optimization algorithms for evaluating the proposed VM Placement technique. The performance evaluation for the proposed algorithm and other three selection algorithms are measured based on the above mentioned three factors, AWT, TT, and ARIT. Table 5, Table 6, and Table 7 provide details about the performance evaluation metrics for 10, 20, and 30 Physical Machines respectively. We have considered 100, 150, and 200 Virtual Machine requests for each testing scenario. The measurements are provided in Mill Seconds.

The following figures clearly explain about the performance evaluation for the proposed selection algorithm compared to Marahatta et al. [25], Soltanshahi et al. [37], Devaraj et al. [38], Chandran et al. [39], Rahim et al. [40], and Vasudevan et al. [41] methods. According to the results, the proposed VM Placement algorithm works well by achieving the highest accuracy and precision. The accuracy is calculated based on the throughput and ideal-waiting-time for each virtual machine. The physical host machine is selected according to the energy consumption rate. Initial stages of physical machines are selected based on under-usage. Then, improved ABC optimization algorithm is used to select a suitable physical host machine. The proposed method considers ideal-waiting-time for measuring the accuracy and precision by using (20) and (22).

The evaluation is conducted for 200, 300, and 400 virtual machines with 20, 40, and 60 physical host machines. Table 2 and figure 5 show that the proposed method achieves 96.2% of accuracy and 96% of precision rate with 20 physical machines. This method shows 98% of accuracy and precision rate for 30 physical machines compared to the existing methods (table 3 and figure 6). Table 4 and figure 7 present that the proposed method achieves 97.7% of accuracy and 97.2%

TABLE 1. Performance evaluation metrics for 20 physical machines.

VM Placement Algorithms	Average Throughput (Sec)			Accuracy (%)			Precision (%)		
	No. of VMs								
	200	300	400	200	300	400	200	300	400
Marahatta et al. [25]	114	127	135	82.3	79.5	83.7	79.4	81.3	84.2
Soltanshahi et al. [37]	107	112	128	84.5	87.2	89.7	82.3	88.1	84.2
Devaraj et al. [38]	111	124	144	83.2	88.4	86.6	84.1	86.7	85.1
Chandran et al. [39]	119	146	167	85.5	86.3	89.4	81.4	79.9	88.5
Rahim et al. [40]	94	112	136	91.3	92.5	93.4	92.4	91.3	94.5
Tian et al. [41]	88	104	135	92.4	93.5	94.5	93.6	94.7	93.7
Proposed Method	**75**	**92**	**116**	**95.8**	**96.2**	**95.7**	**96.3**	**92.4**	**94.9**

TABLE 2. Performance evaluation metrics for 40 physical machines.

VM Placement Algorithms	Average Throughput (Sec)			Accuracy (%)			Precision (%)		
	No. of VMs								
	200	300	400	200	300	400	200	300	400
Marahatta et al. [25]	103	119	133	79.4	76.7	81.4	78.2	77.8	79
Soltanshahi et al. [37]	97	108	121	82.4	88.4	86.4	81.9	85.5	87.2
Devaraj et al. [38]	100	114	130	81.6	89.1	84.1	82.1	87.9	84.1
Chandran et al. [39]	104	112	126	83.7	87.2	91.1	87.4	89.1	90
Rahim et al. [40]	81	98	112	92.5	90.3	92.1	94.6	91	93.9
Tian et al. [41]	76	85	98	93.1	92.3	95.2	92.7	94.4	96.1
Proposed Method	**63**	**78**	**84**	**96.1**	**98.2**	**94.9**	**95.9**	**98**	**97.9**

of precision for 40 physical machines. The overall accuracy and precision for the proposed method is 97.2% and 97.9% respectively.

A. PERFORMANCE EVALUATION USING MACHINE LEARNING TECHNIQUE

This section discussed about the performance evaluation for the proposed method with machine learning optimization technique for VM placement in cloud datacenters. Same dataset and experimental setup is used for conducting the simulation environment for estimating the accuracy and precision with different set of virtual machine and physical resources. The MJPM [43] uses multi-job assignment technique for physical machine selection based on the available list of resources and this technique achieves notable accuracy for increasing PM resources.

Weighted page ranking method [44] uses common weight based PM selection and this weight calculation will not be same for all the times. The accuracy for VM placement is totally depends on weight factor and this will directly affects the resource allocation due to the dynamic assignment.

Butterfly optimization technique and ant colony optimization techniques are nearly producing the excellent result for accuracy of VM placement. Genetic algorithm based VM placement technique [46] achieves only less than 80% of accuracy and this technique is suffers with local optima problem. These local optima problems are addressed by the researchers in many research papers.

TABLE 3. Performance evaluation metrics for 60 physical machines.

VM Placement Algorithms	Average Throughput (Sec)			Accuracy (%)			Precision (%)		
	No. of VMs								
	200	300	400	200	300	400	200	300	400
Marahatta et al. [25]	93	105	119	78.2	80.4	84.2	74.6	78.1	80
Soltanshahi et al. [37]	91	97	109	81.6	87.2	87.5	79.6	82.7	85.3
Devaraj et al. [38]	89	100	117	78.7	85.7	80.4	80.9	86.7	85.5
Chandran et al. [39]	91	103	122	85.5	89.7	92.4	86.6	88.7	93
Rahim et al. [40]	78	105	114	91.8	89.4	93.2	89.7	94	92.7
Tian et al. [41]	69	82	101	94.2	90.9	94	93.1	95.2	94.5
Proposed Method	**53**	**67**	**79**	**95.2**	**97.7**	**96.7**	**96.7**	**97.2**	**95.8**

TABLE 4. Performance evaluation with accuracy of energy based VM placement techniques.

	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Marahatta et al. [25]	84.2	80	76.3	75.5
Soltanshahi et al. [37]	87.5	85.3	78.6	76.5
Devaraj et al. [38]	80.4	85.5	81.4	83.7
Chandran et al. [39]	92.4	93	89.6	91.5
Rahim et al. [40]	93.2	92.7	84.5	85.1
Tian et al. [41]	94	94.5	89.9	86.5
Proposed Method	**96.7**	**95.8**	**93.6**	**92.7**

TABLE 5. Performance evaluation metrics for 20 physical machines.

VM Placement Algorithms	Average Throughput (Sec)			Accuracy (%)			Precision (%)		
	No. of VMs								
	200	300	400	200	300	400	200	300	400
MJPM [43]	108	119	133	76.2	77.4	80.5	74.2	75.4	79.5
Weighted Page Ranking [44]	101	109	121	71.7	75.8	78.7	70.7	73.8	75.7
Butterfly Optimization [42]	107	115	134	81.2	84.5	86.6	82.2	85.5	84.5
Ant Colony Optimization [46]	117	126	139	72.3	74.2	76.4	71.4	73.2	75.9
Genetic Algorithm [45]	192	205	226	69.5	71.5	74.5	70.5	72.4	75.5
Fuzzy Q learning [47]	126	147	155	70.6	72.5	75.6	71.6	73.5	76.7
Proposed Method	**75**	**89**	**97**	**91.8**	**92.4**	**94.7**	**89.8**	**91.5**	**93.6**

TABLE 6. Performance evaluation metrics for 40 physical machines.

	Accuracy (%)			Precision (%)		
	200	**300**	**400**	**200**	**300**	**400**
MJPM [43]	75.2	78.3	82.4	75.3	72.8	78.5
Weighted Page Ranking [44]	73.7	76.7	79.8	72.7	74.5	76.7
Butterfly Optimization [42]	82.3	85.7	88.6	84.3	83.2	85.7
Ant Colony Optimization [46]	73.4	75.5	79.4	72.3	74.3	77.4
Genetic Algorithm [45]	70.5	72.6	75.5	69.5	73.9	76.1
Fuzzy Q learning [47]	74.7	73.5	76.9	72.6	74.5	74.8
Proposed Method	**95.8**	**96.2**	**95.7**	**96.3**	**92.4**	**94.9**

TABLE 7. Performance evaluation metrics for 60 physical machines.

	Accuracy (%)			Precision (%)		
	200	**300**	**400**	**200**	**300**	**400**
MJPM [43]	77.3	80.7	83.7	78.3	81.8	82.6
Weighted Page Ranking [44]	75.7	78.8	82.8	76.7	79.5	82.7
Butterfly Optimization [42]	86.3	82.2	84.6	85.3	86.3	89.7
Ant Colony Optimization [46]	78.4	80.5	82.4	76.3	77.3	80.4
Genetic Algorithm [45]	73.5	76.6	79.5	72.5	74.9	78.1
Fuzzy Q learning [47]	79.7	81.5	83.7	77.6	79.5	81.9
Proposed Method	**95.2**	**97.7**	**96.7**	**96.7**	**97.2**	**95.8**

TABLE 8. Performance evaluation with accuracy of ML based VM placement techniques.

	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
MJPM [43]	83.7	82.6	78.5	74.7
Weighted Page Ranking [44]	82.8	83.7	75.7	78.5
Butterfly Optimization [42]	84.6	89.7	85.4	82.7
Ant Colony Optimization [46]	82.4	80.4	81.5	79.4
Genetic Algorithm [45]	79.5	78.1	74.5	80.1
Fuzzy Q learning [47]	83.7	81.9	79.9	76.5
Proposed Method	**96.7**	**95.8**	**93.6**	**92.7**

Fuzzy Q learning [45] technique produces nearly close to 83% for accuracy for VM placement.

The evaluation is conducted for 100, 200, and 300 virtual machines with 20, 40, and 60 physical host machines. Table 4 and figure 5 show that the proposed method achieves 95.3% of accuracy and 95% of precision rate with 20 physical machines. This method shows 97% of accuracy and precision rate for 30 physical machines compared to the existing methods (table 5 and figure 5). Table 6 and figure 6 present that the proposed method achieves 96.5% of accuracy and 96.3%

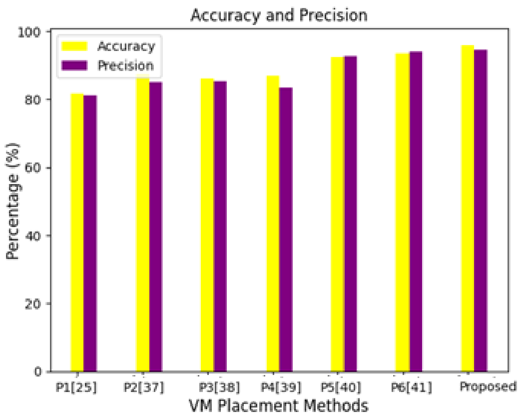

FIGURE 4. Performance evaluation for 20 physical machines.

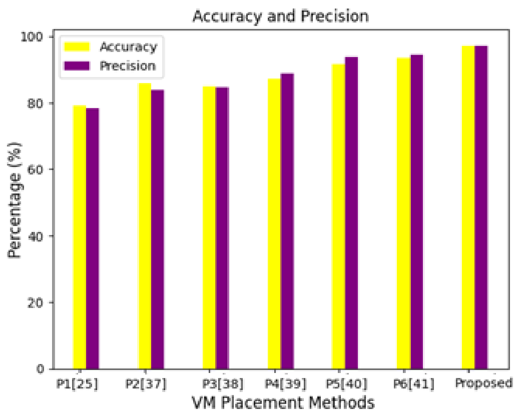

FIGURE 5. Performance evaluation for 40 physical machines.

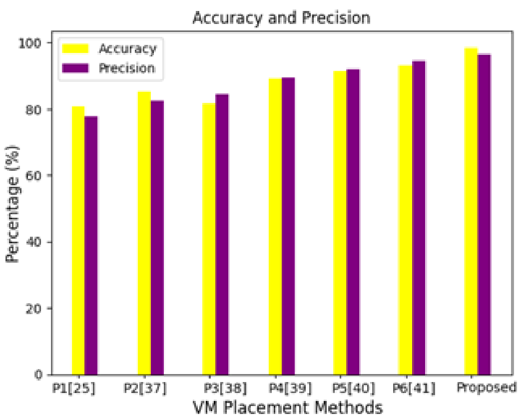

FIGURE 6. Performance evaluation for 60 physical machines.

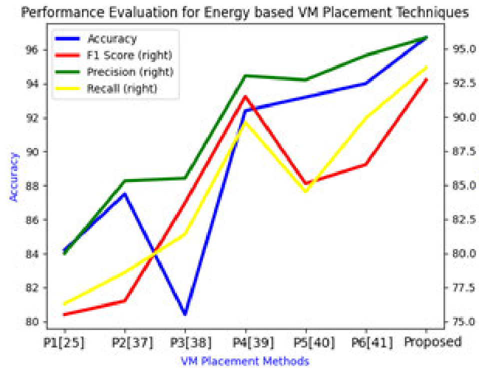

FIGURE 7. Performance evaluation for energy based VM placement techniques.

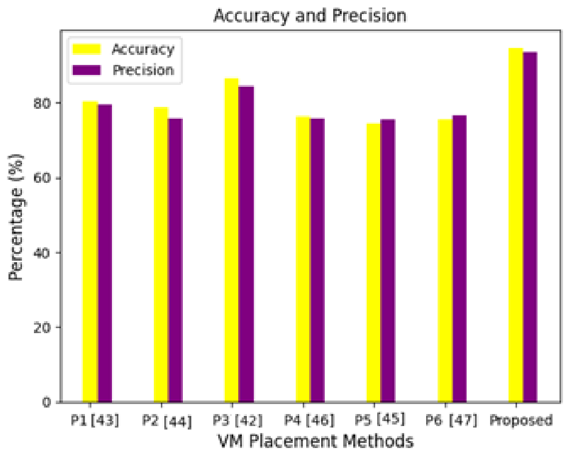

FIGURE 8. Performance evaluation for 20 physical machines for ML techniques.

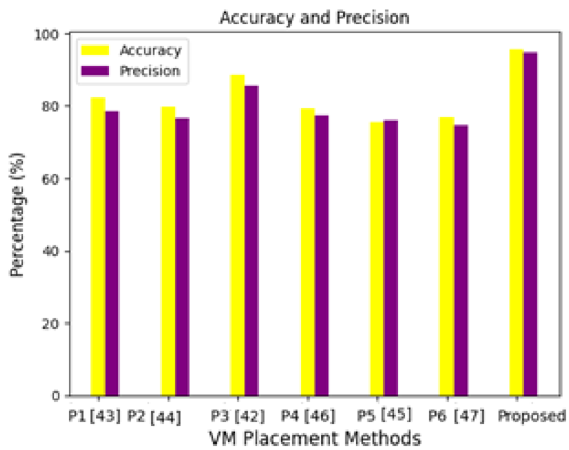

FIGURE 9. Performance evaluation for 40 physical machines for ML techniques.

of precision for 40 physical machines. The overall accuracy and precision for the proposed method is 96.7% and 96.1% respectively.

According to the overall performance evaluation, accuracy for the proposed VM placement technique produces 96.7% accuracy and this accuracy is notably high compare to the other machine learning algorithms. This technique produces 95.8% of precision, 93.6% of recall, and 92.7% of F1 score. This result shows that the proposed method achieves excellent results compare to the other machine learning techniques and table 8 and figure 11 shows the performance evaluation for the proposed technique.

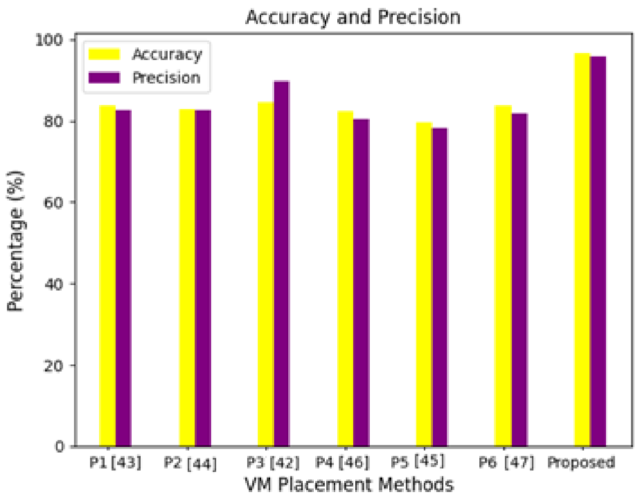

FIGURE 10. **Performance evaluation for 60 physical machines for Ml techniques.**

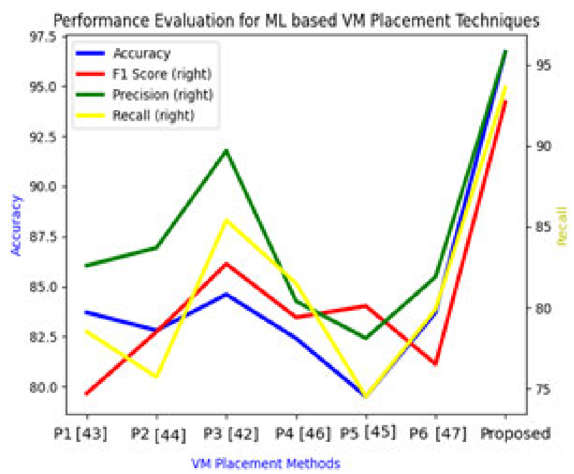

FIGURE 11. **Performance evaluation for ML based VM placement techniques.**

VI. CONCLUSION

This paper has presented a novel energy efficient VM placement technique for selecting suitable physical machine in cloud computing environment. This method uses modified artificial bee colony optimization algorithm for identifying under-utilized physical machines based on energy consumption and resources allocation chart for the existing virtual machines. An adaptive threshold method is used to select suitable threshold levels for energy consumption to identify under-utilized physical host machines. The proposed placement technique is implemented by using CloudSim 3.0 simulator. Compared to existing methods, our approach is able to achieve the highest values of 97.2% for accuracy and 97.9% for precision rate. Still the proposed technique could not measure the exact energy consumption for VM migration process. This limitation is a small barrier for measuring the energy consumption accurately.

Future research will be focused on investigating other nature-inspired optimization method to minimize the energy consumption, that is a key factor for assigning virtual machine in a suitable physical host machine.

REFERENCES

[1] P. D. Bharathi, P. Prakash, and M. V. K. Kiran, "Virtual machine placement strategies in cloud computing," in *Proc. Innov. Power Adv. Comput. Technol. (i-PACT)*, Apr. 2017, pp. 1–7.

[2] A. Muteeh, M. Sardaraz, and M. Tahir, "MrLBA: Multi-resource load balancing algorithm for cloud computing using ant colony optimization," *Cluster Comput.*, vol. 24, no. 4, pp. 3135–3145, Dec. 2021, doi: 10.1007/s10586-021-03322-3.

[3] S. E. Dashti and A. M. Rahmani, "Dynamic VMs placement for energy efficiency by PSO in cloud computing," *J. Experim. Theor. Artif. Intell.*, vol. 28, nos. 1–2, pp. 97–112, Mar. 2016, doi: 10.1080/0952813x.2015.1020519.

[4] A. Choudhary, S. Rana, and K. J. Matahai, "A critical analysis of energy efficient virtual machine placement techniques and its optimization in a cloud computing environment," *Proc. Comput. Sci.*, vol. 78, pp. 132–138, 2016, doi: 10.1016/j.procs.2016.02.022.

[5] L. Liu, M. Zhang, Y. Lin, and L. Qin, "A survey on workflow management and scheduling in cloud computing," in *Proc. 14th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput.*, May 2014, pp. 837–846.

[6] M. Masdari, F. Salehi, M. Jalali, and M. Bidaki, "A survey of PSO-based scheduling algorithms in cloud computing," *J. Netw. Syst. Manage.*, vol. 25, no. 1, pp. 122–158, Jan. 2017, doi: 10.1007/s10922-016-9385-9.

[7] M. Masdari, S. ValiKardan, Z. Shahi, and S. I. Azar, "Towards workflow scheduling in cloud computing: A comprehensive analysis," *J. Netw. Comput. Appl.*, vol. 66, pp. 64–82, May 2016, doi: 10.1016/j.jnca.2016.01.018.

[8] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: State-of-the-art and research challenges," *J. Internet Services Appl.*, vol. 1, no. 1, pp. 7–18, May 2010, doi: 10.1007/s13174-010-0007-6.

[9] M. Muthuswamy and A. M. Ali, "Sustainable supply chain management in the age of machine intelligence: Addressing challenges, capitalizing on opportunities, and shaping the future landscape," *Sustain. Mach. Intell. J.*, vol. 3, pp. 1–14, Jun. 2023.

[10] A. Mosa and N. W. Paton, "Optimizing virtual machine placement for energy and SLA in clouds using utility functions," *J. Cloud Comput.*, vol. 5, no. 1, p. 17, Dec. 2016, doi: 10.1186/s13677-016-0067-7.

[11] Y. Yu and Y. Gao, "Constraint programming-based virtual machines placement algorithm in datacenter," in *Proc. Int. Conf. Intell. Inf. Process.* Cham, Switzerland: Springer, 2012, pp. 295–304.

[12] S. Dörterler, M. Dörterler, and S. Ozdemir, "Multi-objective virtual machine placement optimization for cloud computing," in *Proc. Int. Symp. Netw., Comput. Commun. (ISNCC)*, May 2017, pp. 1–6.

[13] K. Li and J. Nabrzyski, "Virtual machine placement in cloudlet mesh with network topology reconfigurability," in *Proc. IEEE 6th Int. Conf. Cloud Netw. (CloudNet)*, Sep. 2017, pp. 1–7.

[14] W. Wang, Y. Zhao, M. Tornatore, A. Gupta, J. Zhang, and B. Mukherjee, "Virtual machine placement and workload assignment for mobile edge computing," in *Proc. IEEE 6th Int. Conf. Cloud Netw. (CloudNet)*, Sep. 2017, pp. 1–6.

[15] T. S. Kumar, S. D. S. Mustapha, P. Gupta, and R. P. Tripathi, "Hybrid approach for resource allocation in cloud infrastructure using random forest and genetic algorithm," *Sci. Program.*, vol. 2021, pp. 1–10, Oct. 2021.

[16] Y.-S. Dong, G.-C. Xu, and X.-D. Fu, "A distributed parallel genetic algorithm of placement strategy for virtual machines deployment on cloud platform," *Scientific World J.*, vol. 2014, pp. 1–12, 2014, doi: 10.1155/2014/259139.

[17] B. Liu, R. Chen, W. Lin, W. Wu, J. Lin, and K. Li, "Thermal-aware virtual machine placement based on multi-objective optimization," *J. Supercomput.*, vol. 79, no. 11, pp. 12563–12590, Jul. 2023.

[18] Z. Li, M. Pan, and L. Yu, "Multi-resource collaborative optimization for adaptive virtual machine placement," *PeerJ Comput. Sci.*, vol. 8, p. e852, Jan. 2022, doi: 10.7717/peerj-cs.852.

[19] S. Yang, P. Wieder, R. Yahyapour, S. Trajanovski, and X. Fu, "Reliable virtual machine placement and routing in clouds," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 10, pp. 2965–2978, Oct. 2017, doi: 10.1109/TPDS.2017.2693273. https://doi.

[20] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Gener. Comput. Syst.*, vol. 28, no. 5, pp. 755–768, May 2012, doi: 10.1016/j.future.2011.04.017.

[21] A. Zhou, S. Wang, B. Cheng, Z. Zheng, F. Yang, R. N. Chang, M. R. Lyu, and R. Buyya, "Cloud service reliability enhancement via virtual machine placement optimization," *IEEE Trans. Services Comput.*, vol. 10, no. 6, pp. 902–913, Nov. 2017, doi: 10.1109/TSC.2016.2519898. https://doi.org/10.1109/TSC.2016.2519898

[22] E. Hormozi, S. Hu, Z. Ding, Y.-C. Tian, Y.-G. Wang, Z.-G. Yu, and W. Zhang, "Energy-efficient virtual machine placement in data centres via an accelerated genetic algorithm with improved fitness computation," *Energy*, vol. 252, Aug. 2022, Art. no. 123884, doi: 10.1016/j.energy.2022.123884.

[23] J. Peake, M. Amos, N. Costen, G. Masala, and H. Lloyd, "PACO-VMP: Parallel ant colony optimization for virtual machine placement," *Future Gener. Comput. Syst.*, vol. 129, pp. 174–186, 2022, doi: 10.1016/j.future.2021.11.019.

[24] X.-F. Liu, Z.-H. Zhan, J. D. Deng, Y. Li, T. Gu, and J. Zhang, "An energy efficient ant colony system for virtual machine placement in cloud computing," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 113–128, Feb. 2018, doi: 10.1109/TEVC.2016.2623803.

[25] A. Marahatta, S. Pirbhulal, F. Zhang, R. M. Parizi, K. R. Choo, and Z. Liu, "Classification-based and energy-efficient dynamic task scheduling scheme for virtualized cloud data center," *IEEE Trans. Cloud Comput.*, vol. 9, no. 4, pp. 1376–1390, Oct. 2021.

[26] C. Jiang, J. Wu, and Z. Li, "Adaptive thresholds determination for saving cloud energy using three-way decisions," *Cluster Comput.*, vol. 22, no. S4, pp. 8475–8482, Jul. 2019.

[27] N. Malik, M. Sardaraz, M. Tahir, B. Shah, G. Ali, and F. Moreira, "Energy-efficient load balancing algorithm for workflow scheduling in cloud data centers using queuing and thresholds," *Appl. Sci.*, vol. 11, no. 13, p. 5849, Jun. 2021, doi: 10.3390/app11135849.

[28] S. K. Upadhyay, A. Bhattacharya, S. Arya, and T. Singh, "Load optimization in cloud computing using clustering: A survey," *Int. Res. J. Eng. Technol.*, vol. 5, no. 4, pp. 2455–2459, 2018.

[29] C. Sudhakar, R. Jain, and T. Ramesh, "Cloud load balancing–honey bees inspired effective request balancing strategy," in *Proc. Int. Conf. Comput., Power Commun. Technol. (GUCON)*, Sep. 2018, pp. 605–610.

[30] D. Verma and R. Bhatt, "Implementing cloud scheduling algorithm for resource allocation," Jaypee Univ. Inf. Technol. Waknaghat, Himachal Pradesh, India, Tech. Rep., 2019.

[31] N. Verma, V. Sharma, M. Kashyap, and A. Jha, "Heuristic load balancing algorithms in vulnerable cloud computing environment," in *Proc. Int. Conf. Adv. Comput., Commun. Control Netw. (ICACCCN)*, Oct. 2018, pp. 424–429.

[32] Z. Li, C. Yan, L. Yu, and X. Yu, "Energy-aware and multi-resource overload probability constraint-based virtual machine dynamic consolidation method," *Future Gener. Comput. Syst.*, vol. 80, pp. 139–156, Mar. 2018.

[33] M. Riahi and S. Krichen, "A multi-objective decision support framework for virtual machine placement in cloud data centers: A real case study," *J. Supercomput.*, vol. 74, no. 7, pp. 2984–3015, Jul. 2018.

[34] S. Gharehpasha, M. Masdari, and A. Jafarian, "Power efficient virtual machine placement in cloud data centers with a discrete and chaotic hybrid optimization algorithm," *Cluster Comput.*, vol. 24, no. 2, pp. 1293–1315, Jun. 2021.

[35] M. Yavari, A. Ghaffarpour Rahbar, and M. H. Fathi, "Temperature and energy-aware consolidation algorithms in cloud computing," *J. Cloud Comput.*, vol. 8, no. 1, pp. 1–16, Dec. 2019.

[36] S. Mejahed and M. Elshrkawey, "A multi-objective algorithm for virtual machine placement in cloud environments using a hybrid of particle swarm optimization and flower pollination optimization," *PeerJ Comput. Sci.*, vol. 8, p. e834, Jan. 2022, doi: 10.7717/peerj-cs.834.

[37] M. Soltanshahi, R. Asemi, and N. Shafiei, "Energy-aware virtual machines allocation by Krill Herd algorithm in cloud data centers," *Heliyon*, vol. 5, no. 7, Jul. 2019, Art. no. e02066.

[38] A. F. S. Devaraj, M. Elhoseny, S. Dhanasekaran, E. L. Lydia, and K. Shankar, "Hybridization of firefly and improved multi-objective particle swarm optimization algorithm for energy efficient load balancing in cloud computing environments," *J. Parallel Distrib. Comput.*, vol. 142, pp. 36–45, Aug. 2020.

[39] R. Chandran, S. Rakesh Kumar, and N. Gayathri, "Genetic algorithm-based Tabu search for optimal energy-aware allocation of data center resources," *Soft Comput.*, vol. 24, no. 21, pp. 16705–16718, Nov. 2020, doi: 10.1007/s00500-020-05240-9.

[40] S. Rahim, N. Javaid, A. Ahmad, S. A. Khan, Z. A. Khan, N. Alrajeh, and U. Qasim, "Exploiting heuristic algorithms to efficiently utilize energy management controllers with renewable energy sources," *Energy Buildings*, vol. 129, pp. 452–470, Oct. 2016.

[41] M. Vasudevan, Y.-C. Tian, M. Tang, E. Kozan, and X. Zhang, "Energy-efficient application assignment in profile-based data center management through a repairing genetic algorithm," *Appl. Soft Comput.*, vol. 67, pp. 399–408, Jun. 2018.

[42] M. Ghetas, "A multi-objective monarch butterfly algorithm for virtual machine placement in cloud computing," *Neural Comput. Appl.*, vol. 33, pp. 11011–11025, Jan. 2021.

[43] S. M. Sait and K. Shahzada, "Engineering simulated evolution for virtual machine assignment problem," *Appl. Intell.*, vol. 43, no. 2, pp. 296–307, 2015, doi: 10.1007/s10489-014-0634-x.

[44] M. K. Gupta and T. Amgoth, "Resource-aware virtual machine placement algorithm for IaaS cloud," *J. Supercomput.*, vol. 74, no. 1, pp. 122–140, Jan. 2018, doi: 10.1007/s11227-017-2112-9.

[45] J. C. Pérez Gálvez, M. J. Granda, T. López-Guzmán, and J. R. Coronel, "Local gastronomy, culture and tourism sustainable cities: The behavior of the American tourist," *Sustain. Cities Soc.*, vol. 32, pp. 604–612, Jul. 2017, doi: 10.1016/j.scs.2017.04.021.

[46] Y. Gao, H. Guan, Z. Qi, Y. Hou, and L. Liu, "A multi-objective ant colony system algorithm for virtual machine placement in cloud computing," *J. Comput. Syst. Sci.*, vol. 79, no. 8, pp. 1230–1242, Dec. 2013.

[47] S. S. Masoumzadeh and H. Hlavacs, "Integrating VM selection criteria in distributed dynamic VM consolidation using fuzzy Q-learning," in *Proc. 9th Int. Conf. Netw. Service Manage. (CNSM)*, Oct. 2013, pp. 332–338.

FATEN KHALID KARIM received the Ph.D. degree in computing and information technology from Flinders University, Adelaide, SA, Australia. She is currently an Associate Professor with the Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia. Her research interests include cloud computing and information technology. She has published more than 20 peer-reviewed research articles in her field.

NITHYA REKHA SIVAKUMAR received the Ph.D. degree. She was a full-time Research Scholar with Periyar University awarded with "UGC BSR Research Fellowship in Science for Meritorious Students" by the University Grants Commission, New Delhi, Government of India, from 2010 to 2013. She is currently an Associate Professor with the Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University (PNU), Riyadh, Saudi Arabia. She received a Travel Grant from the Department of Science and Technology (DST), Government of India, to go to USA for the Ph.D. research. received a grant from UGC. She has also received a Research Grant for Research Identity Fast-Track Funding Program from the Deputyship for Research and Innovation, the Ministry of Education, and the Researchers Supporting Project Award from PNU, four times. She is supervising several graduate (B.S. and M.S.) students and the Chairperson for the Scientific Research Committee, PNU. She is a Panel of External Examiners for Ph.D. students. More than 25 candidates have been awarded Ph.D. degree. She is the author of more than 50 peer-reviewed articles and five patents. Her research interests include mobile computing, artificial intelligence, the Internet of Things, deep learning, machine learning, wireless networks, network/cyber security, blockchain, VANET, cognitive radio networks, and cloud computing. She received the Best Distinguished Researcher Award from the College of Computer, Qassim Private Colleges, Buraydah, Saudi Arabia, from 2015 to 2016. She is a reviewer in reputed journals, such as IEEE Access, Hindawi, Elsevier, MDPI, Tech Press, and Springer.

SAMEER ALSHETEWI received the Ph.D. degree in computer science from Flinders University, South Australia. He is currently the Manager of E-Transaction Services with the General Information Technology Department, The Excellence Services Directorate, Executive Affairs, Ministry of Defense, Saudi Arabia. His research interests include e-government, t-government, e-transactions, and information technology. He has published several research articles in his field.

AHMED ZOHAIR IBRAHIM received the Ph.D. degree from the Belarusian State University of Informatics and Radioelectronics. He is currently an Assistant Professor with the Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia, where he has been a Faculty Member, since 2009. He was the former Head of the Graphic Design Department, the Director of Communication, Consultation, and a Continuous Learning Office and the Head of the Graphic Design Department, Irbid National University, Jordan, and the Administrator of IT (Network) and Hardware Assembly, Dar Al-Handasah Consultants (Shair and Partners), Jeddah, Saudi Arabia. He has collaborated actively with researchers in several other computer science disciplines, particularly in algorithms and networks. He has already published 12 articles. His research interests include data processing and cryptography.

GEETHA VENKATESAN received the B.Sc. and M.Sc. degrees in computer science from Madras University, Tamil Nadu, and the Ph.D. degree in computer science with specialization in Internet of Things (IoT) from Galgotias University, Uttar Pradesh. She is currently an Assistant Professor with REVA University, Bengaluru. She has eight years of teaching experience from reputed institutions in Bengaluru. She attended many workshops on "Research Writing Skills." Her research interests include computer networks, internet security, and the latest disruptive technologies brought her to narrow down her research to IoT security.