CAAI Transactions on Intelligence Technology

WILEY

ORIGINAL RESEARCH OPEN ACCESS

Energy Efficient VM Selection Using CSOA-VM Model in Cloud Data Centers

Mandeep Singh Devgan[1] | Tajinder Kumar[2] | Purushottam Sharma[3] | Xiaochun Cheng[4] | Shashi Bhushan[5] | Vishal Garg[2]

[1]Department of CSE, Chitkara University Institute of Engineering and Technology, Rajpura, India | [2]Computer Science and Engineering Department, Jai Parkash Mukand Lal Innovative Engineering & Technology Institute, Radaur, India | [3]School of Computer Science & Engineering, Galgotias University, Greater Noida, India | [4]Computer Science Department, Bay Campus Fabian Way, Swansea University, Swansea, UK | [5]Department of Computer Science and Engineering, Amity School of Engineering and Technology, Amity University Punjab, Mohali, India

Correspondence: Purushottam Sharma (purushottam@galgotiasuniversity.edu.in) | Xiaochun Cheng (xiaochun.cheng@swansea.ac.uk)

Received: 27 March 2024 | **Revised:** 10 September 2024 | **Accepted:** 16 October 2024

Funding: This work was supported by Swansea University.

Keywords: cloud computing | data analysis | data protection

ABSTRACT

The cloud data centres evolved with an issue of energy management due to the constant increase in size, complexity and enormous consumption of energy. Energy management is a challenging issue that is critical in cloud data centres and an important concern of research for many researchers. In this paper, we proposed a cuckoo search (CS)-based optimisation technique for the virtual machine (VM) selection and a novel placement algorithm considering the different constraints. The energy consumption model and the simulation model have been implemented for the efficient selection of VM. The proposed model CSOA-VM not only lessens the violations at the service level agreement (SLA) level but also minimises the VM migrations. The proposed model also saves energy and the performance analysis shows that energy consumption obtained is 1.35 kWh, SLA violation is 9.2 and VM migration is about 268. Thus, there is an improvement in energy consumption of about 1.8% and a 2.1% improvement (reduction) in violations of SLA in comparison to existing techniques.

1 | Introduction

In the modern digital world, the services offered by cloud computing have increased exponentially. Cloud computing (CC) is one of the technologies that have dominated the world in recent years [1]. The usage of resources such as hardware, software and other tools by the users and industry has changed the manner and ways CC offers. In essence, CC offers additional services to a wide range of users over the Internet and its sizeable data centres [2]. There are two separate components: front end and back end. The front end addresses the user's need for an application, search engine, connections and other components to utilise cloud-based services, whereas the back end considers the need for the connectivity, servers, hosts and other components of the system. The paradigm of virtualisation emerged in recent decades after an increase in service demand from a variety of users. Additionally, a lot of data centres are created as a result of the virtualisation concept. The US-based cloud network's data centres have used around 3 billion kWh of electricity. In addition, between 2006 and 2013, the data centres' power usage rose dramatically at a rate of 66% *(report from the Lawrence Berkeley National Laboratory)* [3]. Additionally, the increased power use by the cloud platform is comparable to the emissions from the aviation sector, which has added to 3% of global warming. The power used by a cloud environment of typical size is equivalent to the power used by 25,000 houses. Practically speaking, these data centres should offer infinite amounts of computational resources, including CPU, bandwidth, storage space and network. The resources are, however, finite in practice. The needs of the cloud service providers (CSP) and the users have been met through a smart VM algorithm which effectively deploys the virtual machines (VMs) on physical machines (PMs) that lessens the impact of these resources. The CSP seeks to increase revenue while reducing operational expenses like wastage of resources, energy use, thermal emissions and physical infrastructures [4]. Although a cloud customer frequently seeks to cut costs while utilising the greatest services. The reduction of resource wastage is a crucial study topic since it directly affects the cost of running cloud data centres.

Among the cloud services provided, researchers focused on virtual machines (VMs) allocation or consolidation, which are primarily used to allow infrastructure as a service (IaaS) that provides easy configuration [5]. The notion of virtualisation enables the adaptive selection of virtual machines (VMs) in a cloud environment, managing various resources allotted to the actual system (PM). Instead of managing a single VM, the PM serves as a controller for a collection of VMs. These VMs are eventually connected to the physical machines (PMs) of a system using several methods as discussed in the past [6]. Therefore, to manage the resources in the cloud infrastructure more effectively, researchers used optimisation techniques considering the concept of virtualisation that raises hypotheses such as server centralisation and task scheduling [7]. Automation is necessary because the VM selection process is intricate and time-consuming. However, the application may be interrupted if there is mismanagement. By managing the operation of the resources and turning off underused data centres, the virtualisation environment technique implements an energy-aware model [8]. Much of the method significantly reduced energy consumption; it can still be made even more efficient with the help of virtual machine migration (VMM) technology [9]. Even the migration of VMs has a significant effect to reduce the rate of consumption of energy. The relocation of VM is essential to improve execution, adaptability and reasonability. The reason to legitimise the movement of VM in a certain framework is due to the adjustment of load in which VM from overburdened servers has been moved out to less loaded servers [10–12]. In the cloud environment, there is an arrangement of different servers from a resource pool, and every server is represented as (S_j) and the cloud condition for the server resources is (S_j); $j = 1,2......,n$. During the operational mode, in the beginning, having state b, every server has been conveyed about the VMs that are acceptable at every node and it is signified as $S_j = \mathrm{VM}_{j,1}, ..., \mathrm{VM}_{j,b}$. Here, $\mathrm{VM}_{j,b}$ is the VM on the server S_j. Every server is equipped with numerous resources such as network utilisation, CPU utilisation and memory usage. In the cloud, the allotment of VM is a significant task. The user demands a service from the cloud as $\{M_p, p = 1, 2, ..., n\}$. Each administration sort has QoS as a necessity. The user presents the undertaking as $\{M_p, p = 1, 2, ..., n\}$ for VMs in the cloud platform. There are sixfold parameters for each specified task as $\{M_j = \{C_{\mathrm{CPU}}, EC, Sc, Mt, M_{\mathrm{cost}}, and\ R_{\mathrm{cost}}\}$. C_{CPU} is the CPU utilisation, EC is the energy consumption, Sc is the security, Mt is the makespan time, M_{cost} is the migration cost and R_{cost} is the resources cost. The determination of these parameters is recognised as a major problem in the cloud.

Another critical challenge is the efficient and effective control of resources through load balancing and VM consolidation techniques. Load balancing [13] is a part of VM allocation and selection in which the total load is reallocated to the corresponding PM for maximum utilisation of resources with minimum response time [14].

Essentially, there are static and dynamic types of load-balancing techniques. The dynamic method is superior as it does not require any prior knowledge for operation [15]. Moreover, a distributed approach is highly important in PMs because servicing the cloud users by maintaining the SLA is a problematic concern. Therefore, balancing the load reduces the rate of energy consumption, and it is an essential method for optimal efficiency, short response time with increased throughput rate and decreasing the time to complete the task in the cloud framework [16]. Cloud analysts have noticed that nodes of cloud data centres become unbalanced which reduces the level of efficiency. Thus, balancing the load is important for the efficient maintenance of the cloud in an effective manner. But, the main challenge is that minimising energy consumption and maximising the load are two inconsistent and contradictory objectives. The distribution of workload in a fair manner among the computing resources maximises the chances of balancing the load. This increases the consumption of energy, but it is also used to reduce the operational time of the specific task.

Figure 1 shows the architecture of VM selection and migration. This paper effectively addresses this challenge by introducing a novel technique that addresses the problem of energy consumption considering the PMs that are in an active state. EC depends upon the processor and memory usage (MU) in the PMs. The processor used in the PMs is the major factor that consumes a massive amount of energy.

The above depicted system architecture is comprised of three layers user interface layer (UIL), cloud service layer (CSL) and VM management layer (VMML). The layers are described as follows.

1.1 | User Interface Layer (UIL)

The UIL layer comprises the users who submit their requests to the cloud. The users use an interface that can be a phone, personal computer, tablet etc. The user submission takes place via gateways to the cloud network. The users in some cases might get the request submitted by brokers also. Brokers are the mediators of the cloud, and when the user is not able to understand the submission process, they seek brokers. The brokers take their brokerage charge to get the request submitted to the cloud network.

1.2 | Cloud Service Layer (CSL)

The CSL layer manages the process of load balancing including VM allocation and hotspot detection. A hotspot is referred to as a PM that is either underutilised or overutilised in this research document. The physical machine (PM) is used by the Open Cloud to accomplish the incoming tasks. Random access memory (RAM), system facilities and CPU are all contained in the storage device known as PM. Virtual machines (VMs) are typically

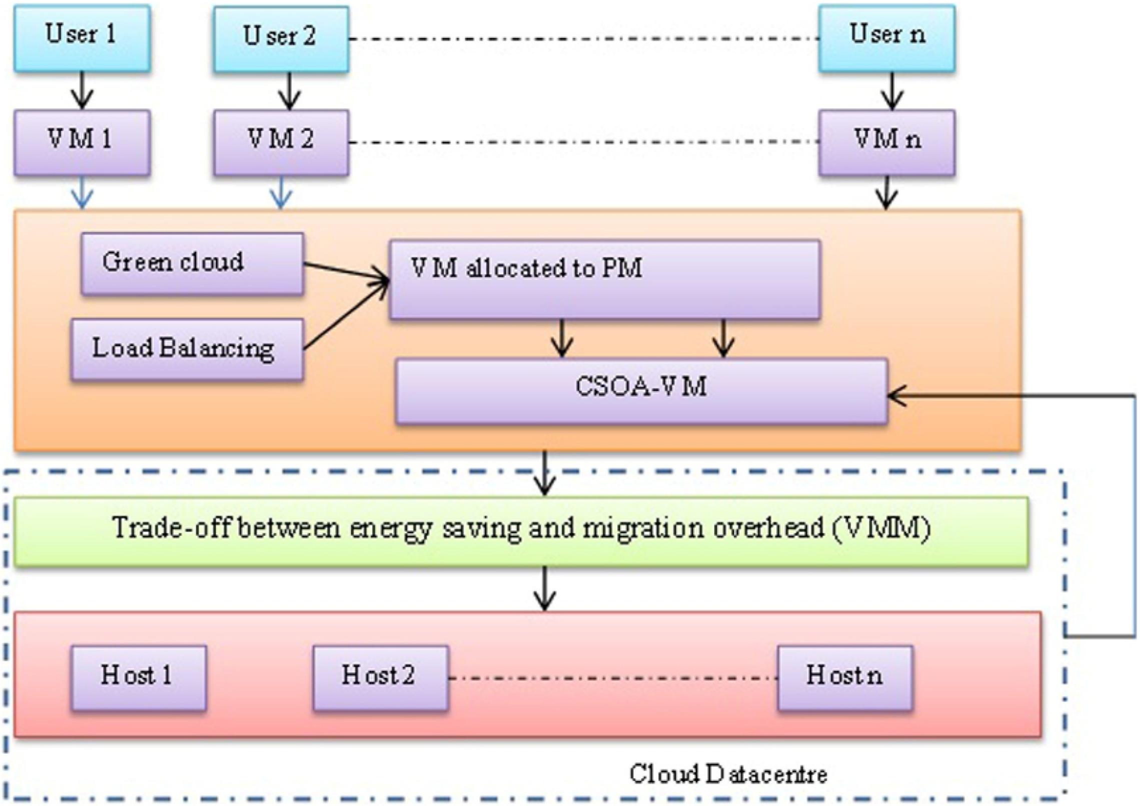

FIGURE 1 | System architecture of VM allocation.

deployed to PMs according to their characteristics to facilitate multiple processors and speedier performance. The distribution procedure is an advanced control system in and of itself. For the selection phase, various computational schemes are introduced, along with the modified best fit decreasing (MBFD) algorithm. It is called the population whenever a PM is deemed to be inappropriate for the given VM and is moved from one PM to the other. The movement method employs a tremendous amount of power, so it is important to minimise transfers to save energy. Migration may occur as a result of a misallocation. Minimisation of migration is a well-known migration form of prevention (MM). Whenever a network operator misses the expected service within the specified timeframe, the SLA is said to be violated, and it is referred to as SLA-V. Overexertion of the load may cause an increase in SLA-V, resulting in increased power consumption. Green computing refers to frameworks that are aimed to minimise power use. In 2012 [17], the MBFD algorithm was used to assign VMs to PMs. The MBFD algorithm relies on a strategy that just considers lowering the number of servers. The algorithm is not only inefficient in terms of power, but it also promotes service level agreement (SLA) violations, which leads to more VM migrations. The VM allocation problem has two components: one is the acceptance of new VM provision orders and the placing of VMs on the host and the latter is the improvement of the present VM allocations. The VM handling is done by the third layer of the framework.

1.3 | VM Management Layer (VMML)

As the name suggests, this layer is used for VM selection from overutilised PMs. All the VMs associated with an underutilised PM are migrated and hence there are no selection policies required to select VMs from the underutilised PM. It is the modified best fit decreasing algorithm architecture. It allocates the VM to the PM based on the CPU utilisation of the VMs. The VMs are sorted based on the decreasing utilisation of the CPU as the high-demand VM will have to be arranged first and the low-demanding VM will have to be arranged later. The hosts are not required to be arranged and they start from 'One'(1) and end up at the last number. Every time a VM is allocated to a host, the PM is reduced with a resource utility that was associated with the PM earlier.

Cloud computing uses the concept of VM consolidation in which the VMs are migrated in a constant manner which is also referred to as live migration. Live migration supports the architecture of green computing as keeping the VM in an idol state with PM, may consume up to 70% of power consumption that could have been attained at the maximum utilisation of the VM over the PM. To migrate the VM from the PM, minimisation of migration (MM) is one of the best policies that could have ever been explored [18–20]. The dual threshold policy utilises two variables 'x' and 't' where x could be the mean, mode or median of the evaluating parameter which is CPU utilisation in the case of the proposed work and t is the threshold margin which should be a finite number. As it has been illustrated in the introduction and the related work section, the cloud computing scheduling architecture incorporates the allocation of the VM to a suitable PM, and further to support elasticity, the data centre migrates the VM from the overutilised and underutilised PMs. The PMs are categorised into three categories based on the utilisation of the CPU and are named overutilised, underutilised and eutralised PMs. The overutilised PMs are the PMs that cross the upper threshold, whereas the underutilised PMs are the PMs that do not cross the lower threshold. Green computing is also managed by this layer which targets reduction in energy consumption. This energy-saving tendency is also clear from Equation (3) as mentioned ahead in this paper. The energy consumption 'EC' model can be developed using the following equation:

$$EC = \int_{t1}^{t2} PC(r(t)) \times dt \tag{1}$$

where EC is the energy consumed by the PM within a timeframe t1 and t2. PC(r(t)) is the consumption of power concerning time 't' and it is computed as follows:

$$PC(r(t)) = d \times \mathrm{Total}_{\mathrm{EC}} + (1 - d) \times \mathrm{Total}_{\mathrm{EC}} \times r(t) \tag{2}$$

In the given Equation (2), $\mathrm{Total}_{\mathrm{EC}}$ is the total consumption of the power when the machine is 100% utilised or exploited, d is the energy consumed by the idle machine and r(t) is the usage rate by the processor ($r(t) \in [0, 1]$) which is determined by the workload that results from the cloud service executed on a PM.

In this paper, we introduce a new VM selection technique called CSOA-VM selection, which migrates VMs from overutilised or underutilised servers while selecting the best VMs from each server to increase server utilisation and boost energy efficiency in virtualisation centres. The proposed technique is based on the brooding mechanism of the cuckoos. The VM selection has been done considering the swarming technique for the best optimal solution. The salient contribution of the paper is as follows:

- A server energy consumption model using the SPECpower_ssj2008 has been developed considering the power consumed by the machines in the data centres.
- A novel optimisation approach using cuckoo search for the placement of VMs aimed to consume less energy and minimisation of migration policy to lessen the number of VM migrations.
- CSOA-VM model: An energy-efficient and violation minimisation (SLA violation-minimisation) model for the selection and placement of VMs in the cloud servers is implemented.

1.4 | Related Work

Keshanchi et al. [21] optimise the task using the scheduling solution. The behavioural modelling technique was employed for efficient results. Further, linear temporal logic was used for the extraction of the qualification of the behavioural model. The outcomes of the proposed model are fruitful in optimising the task that reduces the load on the PMs [21]. Xu et al. [22] solved the problem of allocating the VM to balance the load for multi-dimensional resources. The authors proposed a mathematical model to address this problem in conjunction with the ACO technique. The load imbalance degree concept was also adopted for the selection of PM and outcomes showed that developed ACO works better in comparison to existing ACO optimisation techniques [22]. Liu et al. [23] used the evolutionary algorithm to solve the problem of VM placement by minimising the number of active servers. This can be done by scheduling the servers to save energy and the ant colony system was adopted to take advantage of solving the combinatorial problem [23]. Reddy et al. [24] focused on VM selection for efficient placement using the modified PSO technique. The authors introduced a new VM selection technique for the optimisation of the current location. The usage of memory, utilisation of bandwidth and size of the VM were considered for the new VM selection. The proposed study guarantees the reduction of negative environmental impacts and a lesser number of violations at the service level [24]. Ibrahim et al. [25] developed a power aware approach using the PSO for the selection of VM. The selected VM is migrated to the best location based on the utilisation rate. The authors aimed to reduce the consumption of energy and there was a lesser number of service level violations [25]. Masoudi et al. [24] address the issue of energy management and balancing the load in cloud data centres. The authors developed the two-phase energy-aware scheduling technique to balance the load. The VMM task was handled using the particle swarm optimisation (PSO) technique for the scaling of dynamic voltage. The work is divided into different phases and VMs and PMs were used as inputs to PSO to define the objective function for the load distribution. The proposed work saves energy up to 0.9% and 10% compared with the existing algorithms. The study results were improved but unable to improve the energy efficiency of I/O and optimum allocation for CPU-intensive workloads [14]. M. Xu et al. [26] reviewed the literature focused on VM placement algorithms used to place VMs in hosts in infrastructure clouds and the issues related to load balancing. They also examined the load balancing algorithms aiming at VM placement in cloud [27] data centres along with the classification of the load balancing algorithms. VM allocation and load balancing of multi-dimensional resources are big challenges in cloud computing. In ref. [22], a mathematical model was proposed using the ACO technique. The enhancement of ACO was also proposed using the concept of degree of load imbalance and PM selection probability which performed better than basic ACO. Table 1 represents the comparative analysis of the existing work.

The remainder of the paper is given as follows. Section 2 demonstrates the problem formulation and server energy consumption model. Section 3 describes the research methodology used which included the implementation of the CSOA-VM model for the appropriate selection of VM. Section 4 illustrates the performance measures for experimental outcomes and comparative analysis [33], followed by the conclusion and future scope in Section 5.

2 | Problem Formulation

In the cloud platform, during an operational process, a VM migrates from one place to another. Generally, there are several host machines as shown in Figure 1, and a VM for the process of migration is hosted. At the same time, there is also a change in energy consumption. Thus, it is essential to correctly place the VM and arrange it on accurate host machines in terms of

TABLE 1 | Comparative analysis of the existing work.

References	Methodology	VM allocation and uniformity	Environment	Objectives
Cho et al. 2015 [28]	Hybrid of PSO and ACO	Dynamic and heterogeneous	Private cloud	Minimum count of VM migrations
Thiruvenkadam et al. 2015 [29]	Genetic algorithm	Dynamic and heterogeneous	Private cloud	Minimum count of VM migrations and minimum cost
Ramezani et al. 2015 [16]	PSO	Dynamic and homogeneous	Hybrid cloud	Minimum execution time of the task and minimum transfer time.
Verma et al. 2016 [30]	Best fit decreasing technique	Dynamic and heterogeneous	Private cloud	Minimum downtime of VM
Wang et al. 2016 [31]	Bi-level genetic algorithm	Dynamic and heterogeneous	Hybrid cloud	Minimum EC and minimum load on the network
Reddy et al. 2019 [24]	Modified PSO	Dynamic and heterogeneous	Private cloud	Minimum number of SLA violations, migrations, and energy consumption
Soltanshahi et al. 2019 [32]	Krill herd algorithm	Dynamic and heterogeneous	Hybrid cloud	Minimum number of SLA violations, and energy consumption

optimisation of energy consumption and switching off unused machines. It is assumed that there are 'n' VMs that are executed on 'k' host machines. The main problem is to determine a model for the migration process on the host machines. In addition to reduction in energy consumption, this model also requires the transfer of a large number of virtual machines. The total energy consumption by the data centre is given in Equation (3) [14].

$$EC_t = \min \sum_{i=1}^{n} \sum_{l=1}^{k} EC_i \times \gamma_{il} \tag{3}$$

$$\sum_{i=1}^{n} V_i^{CPU} \times \gamma_{il} < C_l^{\text{cpu}} \tag{4a}$$

$$\sum_{i=1}^{n} V_i^{\text{mem}} \times \gamma_{il} < C_l^{\text{mem}} \tag{4b}$$

$$\sum_{l=1}^{k} V_{il} = 1, i = 1, 2, 3......n \tag{5}$$

In Equation (3), EC_t is the total energy consumption, n is the number of VMs, k is the number of PMs, EC_i is the energy consumption of the i-th VM and γ_{il} is the binary variable indicating whether VM '*i*' is allocated to PM '*l*' *(1 if allocated, 0 otherwise).*

2.1 | Load Balancing Aspect

The energy consumption by machines involved in the execution of some tasks is minimal when all machines are given an equal workload. If any of the machines gets overloaded, then the energy requirement of that machine increases abnormally and the total energy consumption of the system increases. It implies that if energy consumption is to be minimised or if energy consumption is minimal, that means the workload is balanced. Equation (3) reflects this aspect of load balancing.

In Equation (4), V_i^{cpu} is the CPU requirement of the *ith* VM, C_l^{cpu} is the CPU capacity of the l-the PM, V_i^{mem} is the memory requirement of the *ith* VM and C_l^{mem} is the memory capacity of the lth PM. Equation (4a) is called CPU constraint and Equation (4b) is called memory constraint which ensures that the resource requirement of VMs is always less than the resources available in PMs.

2.2 | Load Balancing Aspect

The total CPU and memory consumed by all tasks executing on a machine should not exceed the maximum CPU and memory capacity of that machine. The system advocates such a limiting mechanism on the usage of its resources that reflects the tendency of the system towards load balancing because the system is avoiding overloading of any of its machines. The machine assigned the workload according to its capacity to handle it and hence the state of load balancing is attained. Equations (4a) and 4(b) dictate this load-balancing tendency.

Equation (5), also called assignment constraint, means that each VM '*i*' is assigned to exactly one PM '*j*'. All PMs are not homogeneous, so C_l^{cpu} and C_k^{cpu} are not the same.

2.3 | Load Balancing Aspect

Equation (5) reflects that each task is assigned to exactly one resource or machine; therefore, the tasks are equally distributed among all available resources or machines. When combined with Equation (4), this equation also enables the system to avoid the overloading of any particular resource or machine and hence contributes towards load balancing.

The power consumed by the machines is computed based on the minimum and maximum energy consumed by the host machines. But it is vital to determine how a server consumes energy with different loads. The performance characteristics are generally measured considering different loads and depicted in the SPECpower_ssj2008 benchmark [34]. There are some deviations with different utilisations in energy consumption consumed by the servers. We used the benchmark to measure and calculate the energy consumed by the machines. The given benchmark defined the information about the energy consumed by the servers from the inactive state to the active state at a peak within an interval of 10%. The array is presented in Equation (6) as follows:

$$\mathrm{SP} = (b_0, b_1, \ldots.., b_i, b_{i+1}, \ldots\ldots b_{10}) \tag{6}$$

where $b_0, b_1 \ldots\ldots, b_{10}$ signifies the power consumed by the machines when the PMs are in operational mode within 0%, 10%, 20%...... and 100% utilisation of power, respectively. The estimated power consumed by the PMs is defined in Equation (7).

$$P\big(e_j(ut)\big) = b_i + (b_{i+1} - b_i) * (10 * ut - i) \tag{7}$$

where $i = \mathrm{floor}(10 * ut)$ and $(b_{i+1} - b_i) * (10 * ut - i)$ signify the increase in power consumed by the servers when the memory and CPU utilisation rate increase from $i \; to \; i + 1$. Therefore, the total energy consumed by the data centres is presented in Equation (8) as follows:

$$\mathrm{Total}_{\mathrm{EC}} = \sum_{j=1}^{n} \mathrm{EC}\big(P\big(e_j(ut)\big)\big) \tag{8}$$

3 | Research Methodology

This section describes the implementation of VM selection and placement using the cuckoo search optimisation algorithm. The use of the optimisation technique ensures efficient allocation of resources and migration of VMs in cloud data centres.

3.1 | Cuckoo Search Optimisation Algorithm (CSOA)

The cuckoo search algorithm is a nature-inspired metaheuristic algorithm that was suggested by Yang and Deb in 2009 [35]. Cuckoo species' obligatory brood parasitism behaviour in nature, as well as some birds' Levy flight habits, served as inspiration for this article. Most of the time, the CSA method just uses one parameter which is renowned for being simple to execute in any field. The CSOA algorithm's main goal is to increase the rate of convergence. It can accommodate several criteria involving optimisation and it can also be used with other optimisation methods based on swarms. The CSOA handles the VM selection problem by considering the following three rules:

Each cuckoo laid n number of eggs and each egg was dumped randomly.

$$\mathrm{Egg} = \mathrm{nest(Nt)} + \mathrm{Solution} \tag{9}$$

The next generation use to hold the optimum nest having eggs of higher quality and better solutions.

$$\text{Optimal Solution} = \min f(g), \, or \, \max f(g) \tag{10}$$

$f(g)$ is the objective function for the maximum solution. The set of host nests that can be reached in this situation is constant, and the host can reliably identify the foreign eggs with a probability of either 0 or 1. This allows the host to either remove the foreign egg or empty the nest, after which a new nest can be constructed in a different location. Only the initial and third rules have been amended to impose the multi-objective constraints for the multi-objective CSOA algorithm with k different goals [36].

Each cuckoo lays ‘r’ eggs and then dumps those eggs into the nest chosen randomly.

Egg i, $i \in (1, 2, 3, 4, 5.....r)$, is the rth objective solution. The next generation use to hold the nest having the better or maximum quality of solutions. According to the level of similarity between the eggs, each nest has a certain probability to build a solution of being dumped.

3.2 | Evaluation of Fitness

For the VM selection problem, the first measure for fitness is the use of the number of PMs for selection, in the case of a mathematical point of view. However, the fitness measure depicted by Falkenauer is not practical. This is due to the uneven landscape of the design environment. Using this fitness measure in heuristic algorithms such as cuckoo search optimisation (CSO), the nests in a population have the same fitness solution prematurely. This limitation is taken in mind as a measure of the fitness function which is based on VM rather than employment of PM. Intuitively, consider the fitness measure of a VM to determine the remaining utilisation of space in the machines. A VM that is used to fill this space receives a high fitness value, whereas the VM that fails to cover this space gets a lower fitness function. The given equation shows the fitness measure of a VM.

$$\frac{\mathrm{VM}_i^{\mathrm{Cpu}} + \mathrm{VM}_i^{\mathrm{Memory}}}{\left(T_c - \sum_{k=1,k\neq i}^{n} \mathrm{VM}_k^{\mathrm{Cpu}}\right) + \left(T_M - \sum_{k=1,k\neq i}^{n} \mathrm{VM}_k^{\mathrm{Memory}}\right)} \tag{11}$$

In the given Equation (11), $\mathrm{VM}_i^{\mathrm{Cpu}}$ is the CPU utilisation by ith VM, and $\mathrm{VM}_i^{\mathrm{Memory}}$ is the memory utilisation by ith VM. T_C and T_M are the threshold limits for CPU and memory usage. However, the ith machine is not equal to the kth machine.

Figures 2 and 3 show the fitness of VMs. The given Figure 3 shows that the fitness value of VM1 is 1 and fills the capacity of the server. The fitness of VM1 in Figure 2 is less. The low fitness value shows that VM1 is unable to fill the space. The overall fitness value of the nest is considered as the average fitness value of all VMs placed on the server in the solution space.

Figures 2 and 3 make use of hypothetical data for the sake of simplicity and understanding. The use of three machines Vm1,

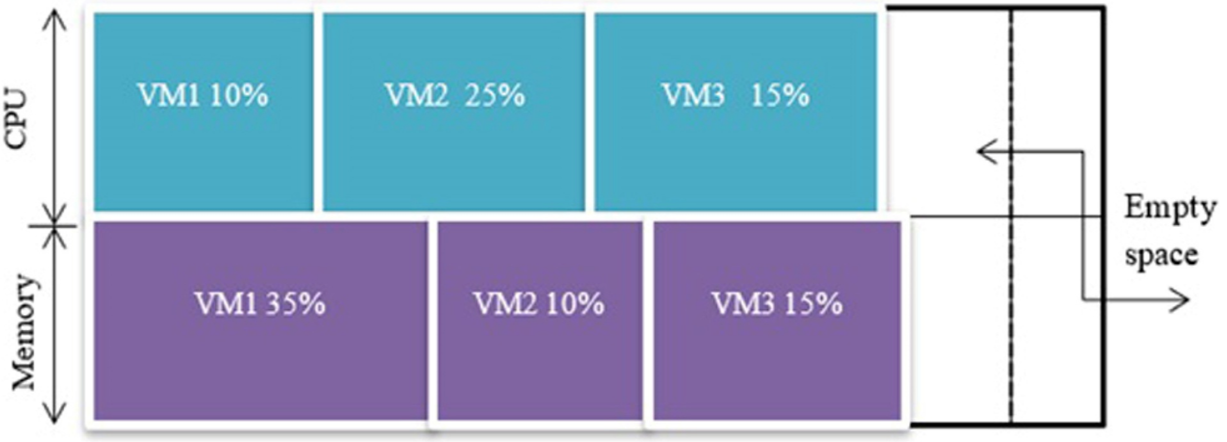

FIGURE 2 | Wastage of resources.

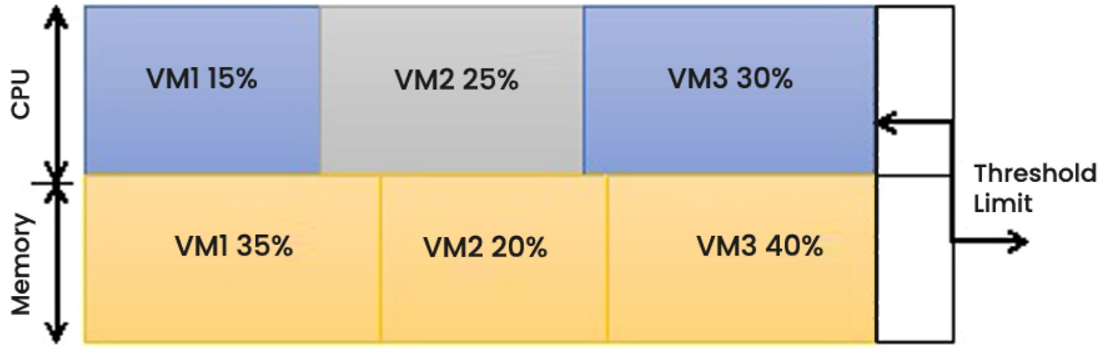

FIGURE 3 | Perfect VM placement on a PM.

Vm2 and Vm3 specific percentages like 10%,15%,25% and 35% helps in clearly showing the effects of different allocation strategies and makes the concept easy to understand. The lack of an allocation strategy may cause resource wastage. Figure 3 shows efficient VM placement, whereas Figure 2 shows inefficient VM placement that causes resource wastage.

3.3 | VM Selection Using CSOA

By transferring VMs from one PM to another PM, live migration enables to respond to high load periods. By dynamically shifting VMs to a small number of hosts while taking into account various criteria such as SLAs, this strategy is frequently used to minimise underutilisation in data centres. Afterwards, to increase energy efficiency, the inactive PM might be put into a sleep state. Between two VMs running, migration occurs. The chosen VM RAM is copied to the target PM once a new configuration file has been created on it. Any memory pages that are altered on the source will be labelled and copied in the meantime. The migration facilitates upkeep without interrupting processes, optimises resource pools and prevents failure. Further, we migrate the VMs to improve productivity and maintain the SLA. For VM selection, we pick servers in one of two extreme states: over- or underutilised. We choose all of the VMs on a server's PM if it is underutilised. Using the suggested selection technique, we choose the VMs for migration if a server is overloaded until the PM utilisation falls below the specified boundary. In this paper, the authors developed a CSOA-VM selection process that takes into account the VM's size, bandwidth use and memory usage. The migration task increases the load on the target and host systems, which has an impact on the data centre's and hosts' performance. As a result, we select the VMs that can be moved from the overused hosts. We developed a cost function that successfully balances memory, throughput and size considerations to do this. This function is also used to select a VM from the list of VMs that can be migrated. The following definitions of how to use the various elements that we thought about are discussed:

3.3.1 | Memory Usage (MU)

A virtual machine's use of memory represents how the VM uses the physical hardware. Memory use often fluctuates slowly while the host utilisation is below the lower threshold and quickly when it is getting close to the upper threshold.

$$VM_i(\text{MU}) = \frac{\text{Allocation of RAM for } VM_i}{\text{Requested RAM by the } VM_i} \tag{12}$$

3.3.2 | Bandwidth Utilisation

VMM involves the transferring of a massive amount of data between the PMs. VMs comprise an application that makes the load transfer among the underlying PMs. For such applications, there is a need to consider the allocation of bandwidth and VM usage as given in the equation.

$$VM_i(\mathrm{BW}) = \frac{\text{Allocation of BW for } VM_i}{\text{Total BW requested by } VM_i} \tag{13}$$

3.3.3 | Size of VM

The data transferred are linked with the cost of migration. The data transfer amount is the factor that has a direct relationship with the consumption of power and migration time when PM has a stable load.

$$VM_i(\text{size}) = \text{VM size } (VM_i) \tag{14}$$

Therefore, considering the concept of memory usage, BW utilisation, and VM size, the cost function for a VM is given as follows:

$$\text{Cost } (VM_i) = x.VM_i(\mathrm{MU}) + y.VM_i(\mathrm{BW}) + z.VM_i(\text{size}) \tag{15}$$

where Cost (VMi) is the total cost of VM_i., and VM_i is the ith VM. VM_i (MU) shows the memory usage of the VM_i , VM_i (BW) is the bandwidth used by VMi and VMi (size) is the size of VM_i. x, y and z are unitless coefficients for scaling each component of the cost equation.

3.4 | Experimental Setup

The experiments to verify the developed algorithm against the existing algorithms have been conducted using MATLAB as an implementation tool and the Anaconda platform with Spyder toolset. The minimum and maximum number of VMs are 50 and 1000, respectively. The supplied load is 10^6 MIPS and the number of simulations per set is 50. The maximum number of simulations is 1000 × 100.

Table 2 shows the experimental settings for VM selection. The selection of VMs using MU, BW and VM size is described in Algorithm 1. The cost of each VM that can be migrated to a physical computer is determined by the CSOA-VM algorithm. It chooses the most-priced VM among those that can be migrated, and then it determines if this decision will result in an extreme consolidation that increases the risk of overutilisation. If not, the algorithm adds this VM to the list of candidates for migration. This procedure goes on until the CPU utilisation falls below the upper utilisation threshold. It is preferable to put the PM into sleep mode if it is underutilised, that is, if utilisation falls underneath the lower limit. To do this, we move every VM to a different physical computer to reduce power consumption while idle. The suggested algorithm dynamically migrates a VM to a target host [37] that offers higher performance if the VM is running on an overloaded physical computer. The chosen VMs are admitted for VM provisioning when the VM selection process utilising the proposed algorithm is completed.

TABLE 2 | Experimental parameter settings.

Maximum number of VMs	1000
Minimum number of VMs	50
Supplied load	10^6MIPS
Total number of simulations per VM set	50
Maximum number of simulations	10,000 ×100
Implementation tool	MATLAB
Toolset	Spyder
Platform	Anaconda

ALGORITHM 1 | Cuckoo search (CSOA-VM) optimisation

1. Input: VM_i // Number of VMs input
2. Initialise the parameters of CS
   VM_{size} // number of eggs representing the number of VMs
   VM_{OT} // other eggs
3. $R = length(VM_i)$ //length of optimised training data
4. Initialise variable:
5. $VM_{optimized} = []$ //initialise the optimised training data variable
6. *For each i in R*
7. $VM_{current} = VM(i)$ // representing selected VM from the number of VMs
8. $Fitness_{objective} = fit(VM_{current}, VM_{threshold})$
9. $\boldsymbol{Optimal\ Solution} = \min \boldsymbol{f}(\boldsymbol{g})[0],\ \boldsymbol{or}\ \max \boldsymbol{f}(\boldsymbol{g})[1]$
10. *Where objective fitness function is denoted as*
   $$Fitness_{objective}\left(F_{objective}\right) = \begin{cases} 1, True \\ 0, False \end{cases}$$
11. $VM_{opt} = CS(Fitness_{objective}, VM_i)$
12. End_{for}

***Return* : $\boldsymbol{VM_{selected}}$** //selected VM for migration

4 | Results

In this study, energy consumption, the number of VM migrations and SLA violations in cloud data centres have been computed. We analysed the proposed CSOA-VM algorithm for the selection of VM in terms of EC and the number of VM migrations. The minimisation of migration policy is used to select the minimum number of VMs. This benefits the data centre by preventing the host from being overutilised.

4.1 | Energy Consumption

The power consumed by the servers and machines is computed to determine the total energy consumption. The energy consumption model is considered to compute the energy. The energy consumption using the number of VMs is shown in Table 3.

4.2 | SLA Violation

The smooth operation of the cloud environment guarantees the SLA in terms of meeting the QoS requirement. The availability of resources, throughput and reliability ensure efficient performance. The SLA violation is computed as follows:

TABLE 3 | Performance metric using the proposed CSOA-VM model.

Number of VMs	Energy consumption (kWh)	SLA violation	VM migrations
100	0.98	8.8	120
200	1.39	9	310
300	2.02	11.02	520
400	2.32	12.01	608
500	2.81	12.32	710
600	3.02	13.21	720
700	3.32	13.43	790
800	3.41	14.23	810
900	4.51	14.56	825
1000	4.62	14.81	838

TABLE 4 | Energy consumption in kWh.

Number of VMs	Proposed CSOA-VM	Kumar and Raza (2015) [7]	Dashti and Rahmani (2015) [8]	Reddy et al. 2019 [24]
50	0.89	0.92	0.92	0.91
100	0.98	1.03	1.20	1.01
150	1.25	1.40	1.60	1.30
200	1.39	1.65	1.75	1.45
250	1.62	1.92	1.97	1.70
300	2.02	2.47	2.50	2.25

$$\text{SLA violation} = \frac{(\text{Resource}_{\text{requested}} - \text{Resource}_{\text{allocated}})}{\text{Resource}_{\text{requested}}} \qquad (16)$$

Typically, SLA is violated if a PM fails to provide the requested resources to the VM. In the proposed study mode, SLA violation is computed for each requested VM. The results are presented in Table 4. The SLA violation remains the same from 50 to 100 VMs and varies with an increase in the number of VMs as given in Table 3.

4.3 | Performance Analysis in Terms of VM Migrations

Excessive migrations of VM in a cloud data centre affect the QoS level under the different peak loads that may increase the consumption of power in a data centre. The comparison of the number of VM migrations with the number of requested VMs is shown in the given table. From the experimental analysis, it is observed that the number of migrations for all existing algorithms is almost the same.

4.4 | Comparative Analysis With the Existing Techniques

Specifically, the energy consumed by the machines depends upon the utilisation of the CPU and memory. The existing techniques use different methods to compute the overall energy consumed by the machines, SLA violation and the number of VM migrations. Table 4 depicts the energy consumed by the machines and is further compared with the existing techniques for validation.

The given table depicts that the average energy consumed by the proposed technique is 1.35 kWh. The average energies con sumed using the state-of-the-art techniques such as [7, 8] and [24] are 1.56 , 1.65 and 1.43 kWh. With the increase in the number of VMs, the energy consumption also increases, and it approaches 2.02 kWh for 300 requested VMs, whereas other techniques proposed by [8] and [24] approach 2.50 and 2.47, respectively. Thus, CSOA-VM is improved by 0.5%, 1.3% and 1.8% in comparison to [7, 8, 24], respectively.

Table 5 shows that the SLA violation using the proposed technique is 8.8 for 50 number of the requested VMs. The violation using the state-of-the-art techniques such as [7, 8] and [24] is almost the same which is 9. With the increase in the number of VMs, the SLA violation also increases, and it approaches 11.02 for 300 requested VMs, whereas other techniques proposed in refs. [8] and [24] approach 12.41 and the average values are 9.8 and 11.26 with an average of about 9.4, respectively. The average SLA violation using the proposed CSOA-VM is 9.2, whereas the existing techniques proposed in ref. [7] are 9.6. Thus, CSOA-VM is improved by 0.2%, 0.6% and 0.4% in comparison to the state-of-the-art techniques.

The given Figure 4 shows the number of VM migrations in comparison to the number of requested VMs. The average migration using the proposed approach for 300 requested VMs is 268, whereas the technique proposed in ref. [24] is 290.

TABLE 5 | SLA violation for different numbers of requested VMs.

Number of VMs	Proposed CSOA-VM	Kumar and Raza (2015) [7]	Dashti and Rahmani (2015) [8]	Reddy et al. 2019 [24]
50	8.8	9	9	9
100	8.8	9.1	9	9
150	8.9	9	9.1	9
200	9	9.22	9.45	9.13
250	9.21	9.75	9.84	9.51
300	11.02	11.89	12.41	11.26

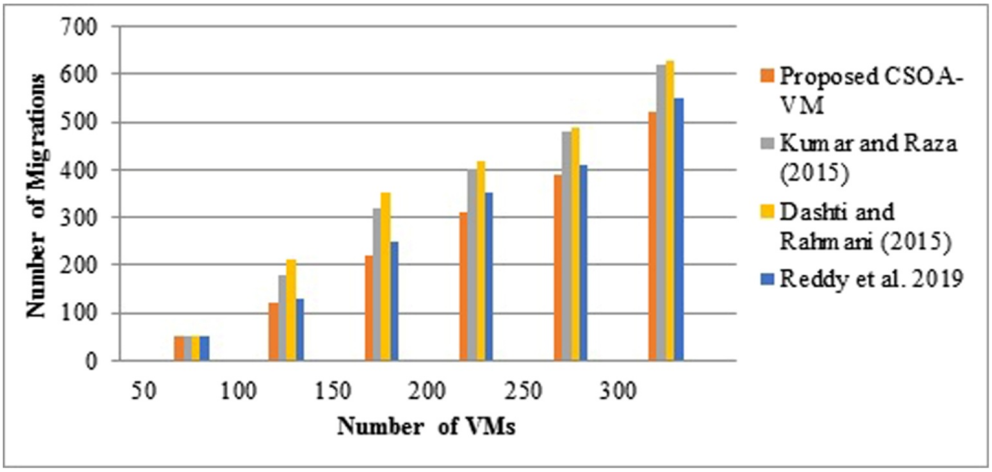

FIGURE 4 | Number of VM migrations.

TABLE 6 | Comparative analysis using the 70:30 ratio for power consumption.

Total PM	Total VM	Supplied load in MIPS	Consumed power CSOA	Consumed power E-MBFD	Consumed power FHCS	Consumed power DCVM
10	50	1000	13.0066	14.4029	14.1588	13.7823
20	100	2000	12.7309	13.2742	13.5328	14.7026
30	150	3000	12.8695	14.7257	14.6468	13.0275
40	200	4000	12.3873	13.0123	12.8843	13.83
50	250	5000	14.3136	15.0304	16.6683	16.5734
60	300	6000	15.0687	15.686	16.8879	16.1978
70	350	7000	15.886	16.1408	16.305	17.137
80	400	8000	15.2866	17.6874	17.6256	17.4066
90	450	9000	14.5838	15.2101	15.8688	15.5795
100	500	10,000	16.2253	16.6925	18.2735	16.728
110	550	11,000	16.4488	16.5583	19.4055	17.6867
120	600	12,000	17.2123	20.2713	19.901	17.366
130	650	13,000	17.5037	19.7325	20.4298	19.4868
140	700	14,000	19.0869	21.2546	19.6149	22.2595
150	750	15,000	18.9773	22.0736	22.2763	20.4748
160	800	16,000	18.757	21.7822	19.8266	20.4936
170	850	17,000	20.2699	20.7425	23.6928	23.5643
180	900	18,000	21.724	24.4292	22.114	23.5748

Similarly, the average migrations by the technique proposed in ref. [8] are 358, whereas the average migration is ref. [7] are 341.6. Thus, there is an improvement of 7.4% in comparison to the technique implemented in ref. [24]. However, the proposed approach outperforms the existing techniques.

To provide a comprehensive analysis, the two scenarios are utilised for the detailed analysis of the proposed work. Firstly, the available dataset is divided into a training set comprising 70% of the data and the remaining 30% is reserved for the testing. Secondly, the volume of the training datasets is increased by 10% and thus 80% of the total data is used as the training data, and 20% of the data is used for the testing of the system. The overall growth in the proposed work is significant enough to be proven better than other state-of-the-art techniques and the illustration is shown in Table 6.

The graph clearly shows that the proposed work consumes less power as compared to the existing works. The average power consumption is the least in the case of the proposed work. Table 7 represents the comparative analysis using the 70:30 ratio for migration count.

The graph clearly shows that the proposed work triggers the least VM migrations as compared to the existing works. Table 8 represents the comparative analysis using the 70:30 ratio for latency in seconds.

The graph visibly displays that the proposed work causes the least latency as compared to the existing works. Table 9 represents the comparative analysis using the 70:30 ratio for SLA violations.

Key performance metrics using a 70:30 distribution are shown in Figure 5a-d, which include power consumption, migration count, latency and SLA violation. These metrics provide information about the system under the study's effectiveness, dependability and compliance with service level agreements (SLAs). The metrics power consumption, migration count, latency and SLA violation indicate how much energy is used, how frequently workloads are migrated and how often service levels fall short of predetermined criteria. Gain a thorough grasp of the system's functionality and optimisation potential by analysing these metrics. Table 10 represents the comparative analysis using the 80:20 ratio for power consumption, whereas Table 11 shows the comparative analysis using the 80:20 ratio for migration count. In Table 12, comparative analysis using the 80:20 ratio for latency is presented, whereas Table 13 represents comparative analysis using the 80:20 ratio for SLA violations.

Figure 6a–d depicts key metrics including power consumption, migration count, latency and SLA violation with an 80:20 distribution. These metrics evaluate energy usage, workload management efficiency, response time and adherence to service level agreements (SLAs), respectively. Power consumption signifies the energy consumed, migration count tracks workload transfers, latency measures response time and SLA violation identifies instances where service level agreements are breached. Analysing these metrics provides insights into

TABLE 7 | Comparative analysis using the 70:30 ratio for migration count.

Total PM	Total VM	Supplied load in MIPS	Migration C S O A	Migration E-MBFD	Migration FHCS	Migration DCVM
10	50	1000	6	7	7	7
20	100	2000	8	9	9	9
30	150	3000	12	14	15	16
40	200	4000	17	19	20	28
50	250	5000	21	23	27	31
60	300	6000	23	25	33	35
70	350	7000	29	32	38	41
80	400	8000	32	38	41	46
90	450	9000	35	40	43	51
100	500	10,000	38	42	45	53
110	550	11,000	46	51	49	53
120	600	12,000	43	59	51	55
130	650	13,000	48	62	57	60
140	700	14,000	55	67	63	62
150	750	15,000	49	71	69	57
160	800	16,000	59	79	73	61
170	850	17,000	61	81	81	74
180	900	18,000	77	86	84	90
190	950	19,000	100	107	105	103
200	1000	20,000	114	122	128	117

TABLE 8 | Comparative analysis using the 70:30 ratio for latency (sec).

Total PM	Total VM	Supplied load in MIPS	Latency CSOA	Latency E-MBFD	Latency FHCS	Latency DCVM
10	50	1000	5.97112	6.95537	6.62958	6.41435
20	100	2000	5.75889	6.13605	6.02863	6.2343
30	150	3000	7.08791	7.8049	7.12481	7.96435
40	200	4000	6.85224	6.91631	7.39933	7.448
50	250	5000	7.45252	8.13447	8.68555	7.48191
60	300	6000	7.65687	8.68814	8.9802	8.30657
70	350	7000	8.19134	8.63677	8.22191	8.97281
80	400	8000	7.67859	7.84655	7.7387	8.86713
90	450	9000	7.31474	7.66225	8.37947	7.6656
100	500	10,000	8.72055	9.39022	10.0149	8.77211
110	550	11,000	8.33476	8.3641	9.32583	8.43287
120	600	12,000	8.01783	8.40005	8.9177	9.31799
130	650	13,000	9.66052	11.1685	10.2539	9.88586
140	700	14,000	9.56439	11.2545	10.5651	9.72171
150	750	15,000	9.30111	10.7312	9.41614	10.5163
160	800	16,000	10.2707	11.4255	10.5928	11.3727
170	850	17,000	9.33814	10.8329	10.8249	9.86843
180	900	18,000	10.4768	11.5572	11.6693	11.3059
190	950	19,000	10.4577	12.171	11.4771	10.6489
200	1000	20,000	10.8216	12.2944	12.3481	11.5522

TABLE 9 | Comparative analysis using the 70:30 ratio for SLA violations.

Total PM	Total VM	Supplied load in MIPS	SLA violations CSOA	SLA violations E-MBFD	SLA violations FHCS	SLA violations DCVM
10	50	1000	8	10	10	10
20	100	2000	8	9	10	10
30	150	3000	9	10	11	11
40	200	4000	10	11	11	12
50	250	5000	10	11	11	12
60	300	6000	11	11	12	13
70	350	7000	11	11	12	13
80	400	8000	11	12	13	13
90	450	9000	12	12	13	14
100	500	10,000	13	13	13	15
110	550	11,000	13	13	14	15
120	600	12,000	14	14	15	15
130	650	13,000	14	14	15	16
140	700	14,000	14	15	15	17
150	750	15,000	14	15	16	17
160	800	16,000	15	15	16	18
170	850	17,000	15	16	17	18
180	900	18,000	15	17	17	18
190	950	19,000	16	17	17	19
200	1000	20,000	16	17	17	20

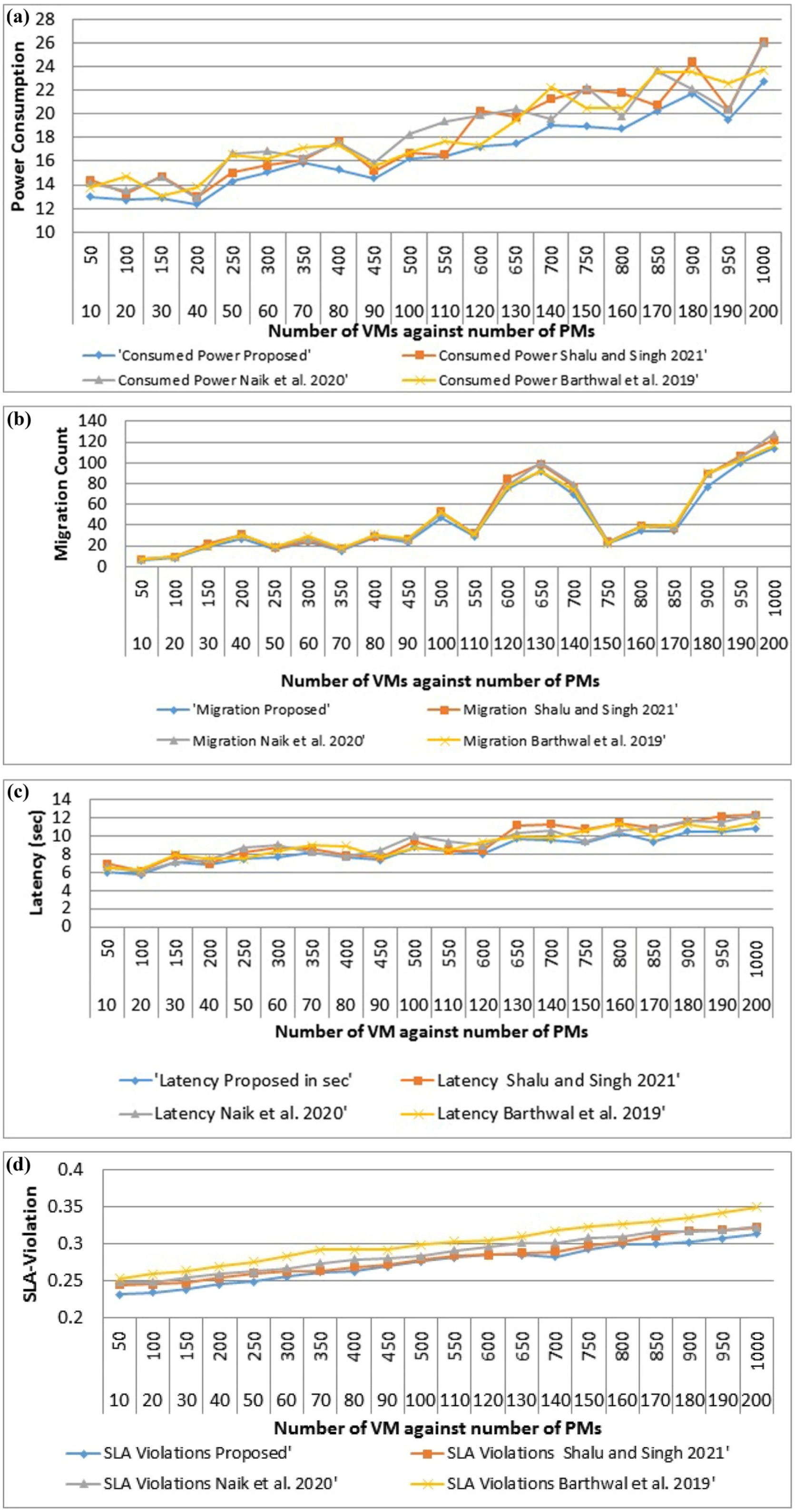

FIGURE 5 | (a) Power consumption using the 70:30 distribution. (b) Migration count using the 70:30 distribution. (c) Latency using the 70:30 distribution. (d) SLA violation using the 70:30 distribution.

system performance and optimisation strategies tailored to the specified workload distribution.

The impacts of the distributions of 80:20 and 70:30 training data on standard parameters are shown in Figure 7a–d. Under these two distributions, it assesses important parameters including dependability, efficiency and performance. This comparison clarifies the effects that different workload distributions have on system performance and resource usage. By examining the variations between the two distributions, one can gain

TABLE 10 | Comparative analysis using the 80:20 ratio for power consumption.

Total PM	Total VM	Supplied load in MIPS	Consumed power CSOA	Consumed power E-MBFD	Consumed power FHCS	Consumed power DCVM
10	50	1000	12.806	13.412	13.192	13.221
20	100	2000	12.363	12.367	12.875	14.409
30	150	3000	11.943	13.815	14.278	12.031
40	200	4000	11.557	12.668	12.507	12.592
50	250	5000	13.072	14.915	16.284	15.746
60	300	6000	14.511	15.573	15.373	15.025
70	350	7000	14.977	16.925	15.413	16.494
80	400	8000	13.975	16.821	17.479	16.74
90	450	9000	13.308	14.709	15.314	14.465
100	500	10,000	15.266	15.348	17.541	15.548
110	550	11,000	15.581	16.337	18.188	17.661
120	600	12,000	16.833	19.102	19.084	17.792
130	650	13,000	16.042	18.795	18.735	17.915
140	700	14,000	18.439	20.258	18.872	20.476
150	750	15,000	17.505	20.833	20.892	18.726
160	800	16,000	18.424	20.3	18.51	19.194
170	850	17,000	18.895	19.902	22.095	23.192
180	900	18,000	20.698	22.803	20.816	21.985
190	950	19,000	19.206	19.656	19.636	21.069
200	1000	20,000	22.339	23.766	24.798	22.349

TABLE 11 | Comparative analysis using the 80:20 ratio for migration count.

Total PM	Total VM	Supplied load in MIPS	Migration CSOA	Migration E-MBFD	Migration FHCS	Migration DCVM
10	50	1000	6	7	7	7
20	100	2000	8	8	8	9
30	150	3000	11	13	12	14
40	200	4000	15	17	16	18
50	250	5000	16	18	17	18
60	300	6000	20	23	26	27
70	350	7000	23	27	29	30
80	400	8000	25	29	27	28
90	450	9000	29	32	33	33
100	500	10,000	31	35	39	37
110	550	11,000	38	42	40	41
120	600	12,000	41	49	46	43
130	650	13,000	46	52	50	49
140	700	14,000	51	59	55	58
150	750	15,000	52	62	62	67
160	800	16,000	59	67	73	79
170	850	17,000	67	74	82	87
180	900	18,000	75	85	95	93
190	950	19,000	91	96	105	107
200	1000	20,000	105	113	124	119

TABLE 12 | Comparative analysis using the 80:20 ratio for latency.

Total PM	Total VM	Supplied load in MIPS	Latency CSOA	Latency E-MBFD	Latency FHCS	Latency DCVM
10	50	1000	5.88737942	6.84608	6.62382	6.31255
20	100	2000	5.70959512	6.12915	6.00724	6.16748
30	150	3000	7.05345654	7.65692	7.08908	7.82561
40	200	4000	6.73083804	6.78563	7.35306	7.39896
50	250	5000	7.41739034	8.05578	8.52272	7.3991
60	300	6000	7.63032666	8.61351	8.89612	8.19748
70	350	7000	8.17369577	8.48905	8.16649	8.82472
80	400	8000	7.67092918	7.81431	7.61978	8.73048
90	450	9000	7.2335839	7.65092	8.27074	7.64829
100	500	10,000	8.65842004	9.32001	9.84184	8.75405
110	550	11,000	8.28503773	8.25786	9.31053	8.36022
120	600	12,000	7.88284902	8.30685	8.76452	9.24646
130	650	13,000	9.55053019	11.0239	10.2358	9.71321
140	700	14,000	9.41531948	11.1161	10.5111	9.52737
150	750	15,000	9.2186724	10.5352	9.36606	10.4905
160	800	16,000	10.0767637	11.3096	10.5017	11.3052
170	850	17,000	9.30227348	10.7345	10.7709	9.85716
180	900	18,000	10.4427839	11.3319	11.5023	11.1812
190	950	19,000	10.3233038	12.0341	11.4101	10.4484
200	1000	20,000	10.7925094	12.2716	12.2941	11.5223

TABLE 13 | Comparative analysis using the 80:20 ratio for SLA violations.

Total PM	Total VM	Supplied load in MIPS	SLA violations CSOA	SLA violations E-MBFD	SLA violations FHCS	SLA violations DCVM
10	50	1000	8	9	9	9
20	100	2000	9	9	10	11
30	150	3000	9	10	11	11
40	200	4000	10	10	11	12
50	250	5000	10	11	12	12
60	300	6000	10	11	12	12
70	350	7000	11	12	12	13
80	400	8000	11	12	13	14
90	450	9000	11	12	13	14
100	500	10,000	11	12	12	13
110	550	11,000	12	13	14	15
120	600	12,000	12	13	14	15
130	650	13,000	13	13	14	15
140	700	14,000	13	14	15	15
150	750	15,000	14	14	15	16
160	800	16,000	14	15	16	17
170	850	17,000	15	16	16	17
180	900	18,000	15	16	17	17
190	950	19,000	16	17	17	18
200	1000	20,000	16	17	17	18

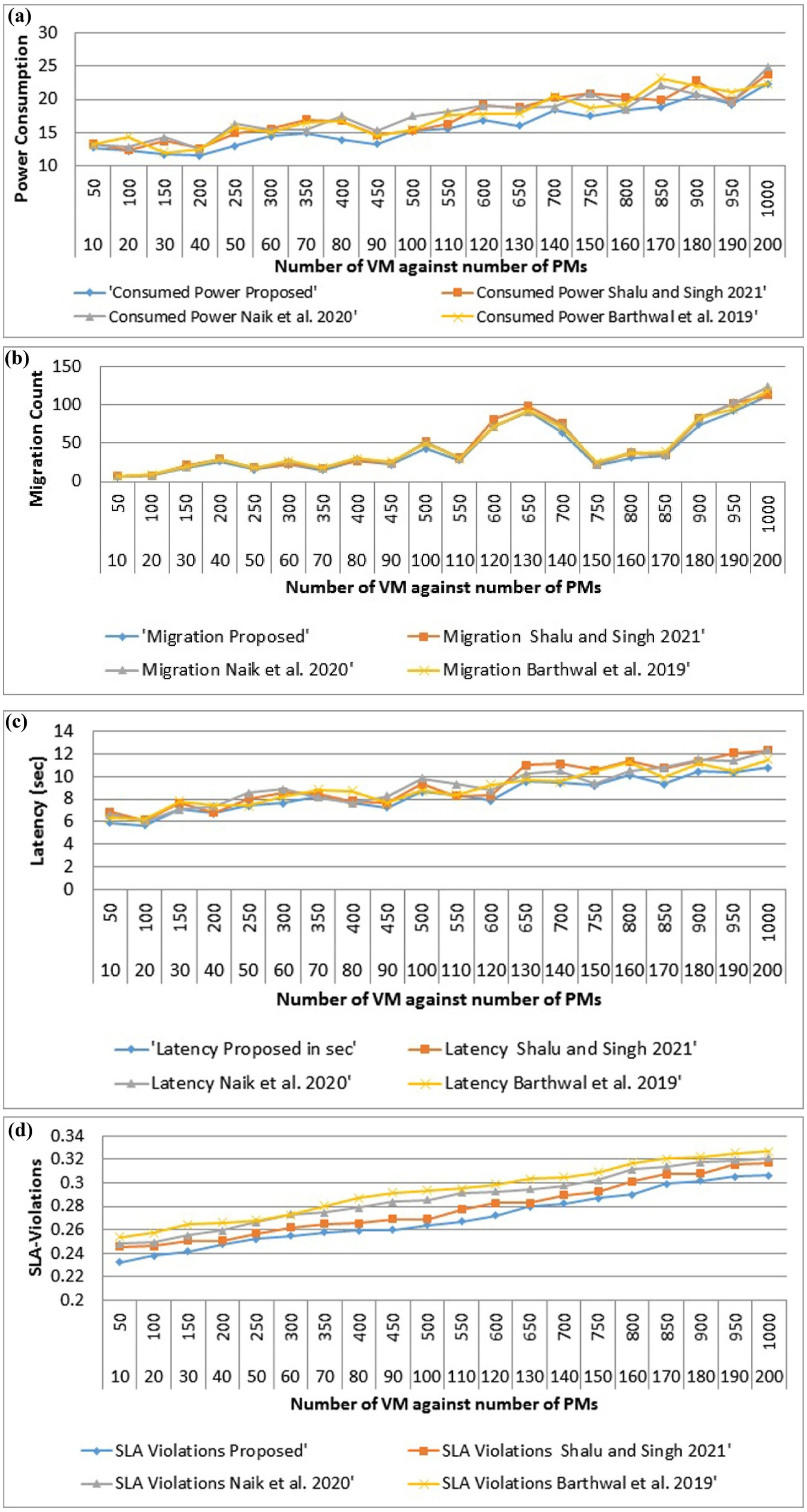

FIGURE 6 | (a) Power consumption using the 80:20 distribution. (b) Migration count using the 80:20 distribution. (c) Latency using the 80:20 distribution. (d) SLA violation using the 80:20 distribution.

knowledge about the best practices for managing workload and how they affect system performance.

The proposed algorithm trains the neural network based on the supplied load to the system, consumed power and total number of migrations that occurred in the list. The proposed algorithm is inspired by the statistical architecture that is represented by Singh and Shalu, Naik and Varun et al. work that combines all the algorithms to represent a model. A new selection behaviour of cuckoo search is proposed and the results have been

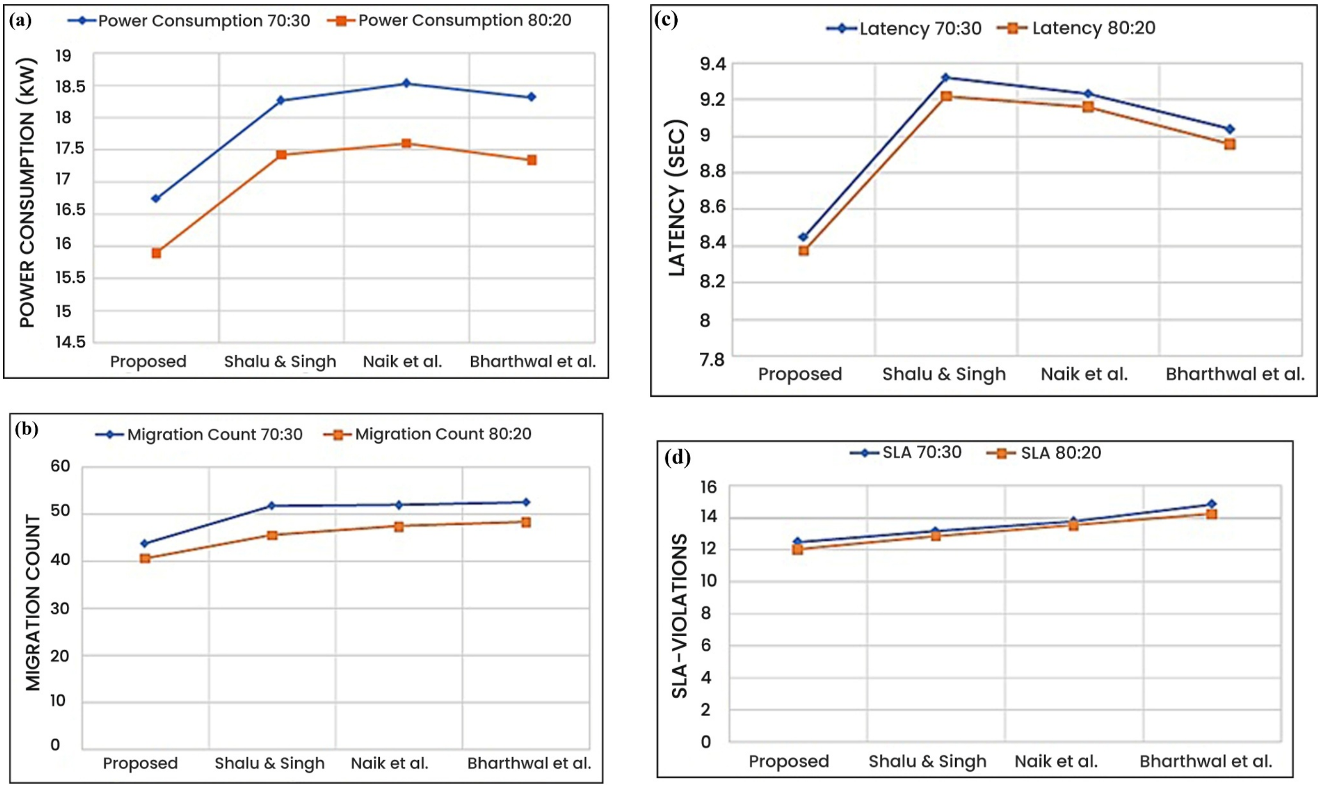

FIGURE 7 | (a) Comparison of 70:30 and 80:20 distributions on power consumption. (b) Comparison of 70:30 and 80:20 distributions on migration count. (c) Comparison of 70:30 and 80:20 distributions on latency. (d) Comparison of 70:30 and 80:20 distributions on SLA violations.

evaluated based on the power consumption, latency and SLA-V with the total number of supplied PMs against its provided VM using variable training to test scenarios.

5 | Conclusion

In the data centre, there are thousands of PMs handling the requests for several VMs. An optimal allocation of the resource strategy helps to improve the utilisation of cloud data centres. In this paper, the proposed CSOA-VM helps to achieve the high utilisation of PMs by decreasing the number of idle machines. The proposed method includes the VM selection method for the efficient selection of VM when the PM is either utilised completely or less utilised. The experimental outcomes demonstrate the combination of the power consumption model and VM selection methods using the cuckoo search technique. The energy consumed by the machines is less in the proposed model and improved by 1.8% in comparison to existing techniques. In the future, we plan to implement the VM selection and placement considering the different workloads and trying to develop an energy-efficient model with more parameters.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

References

1. A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-Aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing," *Future Generation Computer Systems* 28, no. 5 (2012): 755–768, https://doi.org/10.1016/j.future.2011.04.017.

2. M. A. Khoshkholghi, M. N. Derahman, A. Abdullah, S. Subramaniam, and M. Othman, "Energy-Efficient Algorithms for Dynamic Virtual Machine Consolidation in Cloud Data Centers," *IEEE Access* 5 (2017): 10709–10722, https://doi.org/10.1109/ACCESS.2017.2711043.

3. N. S. More and R. B. Ingle, "Energy-Aware VM Migration Using Dragonfly–Crow Optimization and Support Vector Regression Model in Cloud," *International Journal of Modeling, Simulation, and Scientific Computing* 9, no. 6 (December 2018): 1850050, https://doi.org/10.1142/S1793962318500502.

4. A. Tripathi, I. Pathak, and D. P. Vidyarthi, "Modified Dragonfly Algorithm for Optimal Virtual Machine Placement in Cloud Computing," *Journal of Network and Systems Management* 28, no. 4 (May 2020): 1316–1342, https://doi.org/10.1007/S10922-020-09538-9.

5. B. Nazir and others, "QoS-aware VM Placement and Migration for Hybrid Cloud Infrastructure," *Journal of Supercomputing* 74, no. 9 (2018): 4623–4646, https://doi.org/10.1007/s11227-017-2071-1.

6. 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," *Procedia Computer Science* 78 (January 2016): 132–138, https://doi.org/10.1016/J.PROCS.2016.02.022.

7. D. Kumar and Z. Raza, "A PSO Based VM Resource Scheduling Model for Cloud Computing," in *2015 IEEE International Conference on Computational Intelligence \& Communication Technology*, 2015), 213–219.

8. S. E. Dashti and A. M. Rahmani, "Dynamic VMs Placement for Energy Efficiency by PSO in Cloud Computing," *Journal of Experimental &*

Theoretical Artificial Intelligence 28, no. 1–2 (2016): 97–112, https://doi.org/10.1080/0952813x.2015.1020519.

9. S. Singla, P. Sharma, and P. K. Sharma, "Enhanced Security Using Proxy Signcryption Technique for Wireless Mesh Networks," *International Journal of System Assurance Engineering and Management* 14, no. 1 (2023): 474–482, https://doi.org/10.1007/s13198-022-01820-0.

10. M. C. Silva Filho, C. C. Monteiro, P. R. M. Inácio, and M. M. Freire, "Approaches for Optimizing Virtual Machine Placement and Migration in Cloud Environments: A Survey," *Journal of Parallel and Distributed Computing* 111 (2018): 222–250.

11. Z. Wang, D. Sun, G. Xue, S. Qian, G. Li, and M. Li, "Ada-things: An Adaptive Virtual Machine Monitoring and Migration Strategy for Internet of Things Applications," *Journal of Parallel and Distributed Computing* 132 (October 2019): 164–176, https://doi.org/10.1016/J.JPDC.2018.06.009.

12. Y. R. Chen and J. S. Li, "Staggered Approach for Alleviating TCP Incast in Simultaneous Multi-VM Migration," *Computer Communications* 106 (July 2017): 24–32, https://doi.org/10.1016/J.COMCOM.2017.02.015.

13. A. Kaur and B. Kaur, "Load Balancing Optimization Based on Hybrid Heuristic-Metaheuristic Techniques in Cloud Environment," *Journal of King Saud University – Computer and Information Sciences* 34, no. 3 (2022): 813–824: ISSN 1319-1578, https://doi.org/10.1016/j.jksuci.2019.02.010.

14. J. Masoudi, B. Barzegar, and H. Motameni, "Energy-aware Virtual Machine Allocation in DVFS-Enabled Cloud Data Centers," *IEEE Access* 10 (2021): 3617–3630, https://doi.org/10.1109/access.2021.3136827.

15. D. Abdulkareem Shafiq, N. Z. Jhanjhi, and A. Abdullah, "Load Balancing Techniques in Cloud Computing Environment: A Review," *Journal of King Saud University - Computer and Information Sciences* 34, no. 7 (2022): 3910–3933: ISSN 1319-1578, https://doi.org/10.1016/j.jksuci.2021.02.007.

16. F. Ramezani, J. Lu, and F. K. Hussain, "Task-Based System Load Balancing in Cloud Computing Using Particle Swarm Optimization," *International Journal of Parallel Programming* 42, no. 5 (October 2013): 739–754, https://doi.org/10.1007/S10766-013-0275-4.

17. A. Beloglazov and R. Buyya, "Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers," *Concurrency and Computation: Practice and Experience* 24, no. 13 (2012): 1397–1420, https://doi.org/10.1002/cpe.1867.

18. S. Prakash and others, "Role of Virtualization Techniques in Cloud Computing Environment," in *Advances in Computer Communication and Computational Sciences* (Springer, 2019), 439–450.

19. P. Sharma, S. Kapoor, and R. Sharma, "Ransomware Detection, Prevention and Protection in IoT Devices Using ML Techniques Based on Dynamic Analysis Approach," *International Journal of System Assurance Engineering and Management* 14, no. 1 (2023): 287–296, https://doi.org/10.1007/s13198-022-01793-0.

20. A. Zaffar, "Modeling and Forecasting of Sunspots Cycles: An Application of ARMA (P Q)-GARCH (1, 1) Model," (2021): 1–18.

21. B. Keshanchi, A. Souri, and N. J. Navimipour, "An Improved Genetic Algorithm for Task Scheduling in the Cloud Environments Using the Priority Queues: Formal Verification, Simulation, and Statistical Testing," *Journal of Systems and Software* 124 (February 2017): 1–21, https://doi.org/10.1016/J.JSS.2016.07.006.

22. P. Xu, G. He, Z. Li, and Z. Zhang, "An Efficient Load Balancing Algorithm for Virtual Machine Allocation Based on Ant Colony Optimization," *International Journal of Distributed Sensor Networks* 14, no. 12 (December 2018): 2018, https://doi.org/10.1177/1550147718793799/ASSET/IMAGES/LARGE/10.1177_1550147718793799-FIG3.JPEG.

23. H. Zhao, J. Wang, F. Liu, Q. Wang, W. Zhang, and Q. Zheng, "Power-Aware and Performance-Guaranteed Virtual Machine Placement in the Cloud," *IEEE Transactions on Parallel and Distributed Systems* 29, no. 6 (June 2018): 1385–1400, https://doi.org/10.1109/TPDS.2018.2794369.

24. V. Dinesh Reddy, G. R. Gangadharan, and G. Rao, "Energy-Aware Virtual Machine Allocation and Selection in Cloud Data Centers," *Soft Computing* 23, no. 6 (2019): 1917–1932, https://doi.org/10.1007/s00500-017-2905-z.

25. A. Ibrahim, M. Noshy, H. A. Ali, and M. Badawy, "PAPSO: A Power-Aware VM Placement Technique Based on Particle Swarm Optimization," *IEEE Access* 8 (2020): 81747–81764, https://doi.org/10.1109/access.2020.2990828.

26. M. Xu, W. Tian, and R. Buyya, "A Survey on Load Balancing Algorithms for Virtual Machines Placement in Cloud Computing," *Concurrency and Computation: Practice and Experience* 29, no. 12 (June 2017): e4123, https://doi.org/10.1002/cpe.4123.

27. R. Kait, S. Kaur, P. Sharma, C. Ankita, T. Kumar, and X. Cheng, "Fuzzy Logic-Based Trusted Routing Protocol Using Vehicular Cloud Networks for Smart Cities," *Expert Systems* 42, no. 1 (2024): e13561, https://doi.org/10.1111/exsy.13561.

28. K.-M. Cho, P.-W. Tsai, C.-W. Tsai, and C.-S. Yang, "A Hybrid Meta-Heuristic Algorithm for VM Scheduling With Load Balancing in Cloud Computing," *Neural Computing & Applications* 26, no. 6 (2015): 1297–1309, https://doi.org/10.1007/s00521-014-1804-9.

29. T. Thiruvenkadam and P. Kamalakkannan, "Energy Efficient Multi Dimensional Host Load Aware Algorithm for Virtual Machine Placement and Optimization in Cloud Environment," *Indian Journal of Science & Technology* 8, no. 17 (2015): 1–11, https://doi.org/10.17485/ijst/2015/v8i17/59140.

30. M. Verma, G. R. Gangadharan, N. C. Narendra, et al., "Dynamic Resource Demand Prediction and Allocation in Multi-Tenant Service Clouds," *Concurrency and Computation: Practice and Experience* 28, no. 17 (2016): 4429–4442, https://doi.org/10.1002/cpe.3767.

31. X. Wang, Y. Wang, and Y. Cui, "An Energy-Aware Bi-level Optimization Model for Multi-Job Scheduling Problems Under Cloud Computing," *Soft Computing* 20, no. 1 (2016): 303–317, https://doi.org/10.1007/s00500-014-1506-3.

32. M. Soltanshahi, R. Asemi, and N. Shafiei, "Energy-Aware Virtual Machines Allocation by Krill Herd Algorithm in Cloud Data Centers," *Heliyon* 5, no. 7 (2019): e02066, https://doi.org/10.1016/j.heliyon.2019.e02066.

33. R. Sharma, P. Sharma, A. Singh, and V. Srivastava, "An Approach to Optimize Performance of Controller in Networked Control System," *International Journal of System Assurance Engineering and Management* 14, no. 5 (2023): 1639–1646, https://doi.org/10.1007/s13198-023-01967-4.

34. L. D. Gray, A. Kumar, and H. H. Li, "Workload Characterization of the SPECpower-Ssj2008 Benchmark," in *Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, Vol. 5119, 2008), 262–282, https://doi.org/10.1007/978-3-540-69814-2_17/COVER.

35. X.-S. Yang and S. Deb, "Cuckoo Search via Lévy Flights," in *2009 World Congress on Nature \& Biologically Inspired Computing (NaBIC)*, 2009), 210–214.

36. S. H. H. Madni, M. S. A. Latiff, J. Ali, and S. M. Abdulhamid, "Multi-objective-Oriented Cuckoo Search Optimization-Based Resource Scheduling Algorithm for Clouds," *Arabian Journal for Science and Engineering* 44, no. 4 (October 2018): 3585–3602, https://doi.org/10.1007/S13369-018-3602-7.

37. A. Bonkra and P. Dhiman, "IoT Security Challenges in Cloud Environment," in *2021 2nd International Conference on Computational Methods in Science & Technology (ICCMST)*, 2021), 30–34, https://doi.org/10.1109/ICCMST54943.2021.00018.