



# A machine learning model for improving virtual machine migration in cloud computing

Ali Belgacem<sup>1</sup> · Saïd Mahmoudi<sup>2</sup> · Mohamed Amine Ferrag<sup>3</sup>

Accepted: 29 December 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

## Abstract

Cloud Computing is a paradigm allowing access to physical and application resources online via the Internet. These resources are virtualized using virtualization software to make them available to users as a service. Virtual machines (VMs) migration technique provided by virtualization technology impacts the performance of the cloud. It is a significant concern in this environment. When allocating resources, the distribution of VMs is unbalanced, and their movement from one server to another can increase energy consumption and network overhead, necessitating an improvement in VM migrations. This paper addresses the VMs migration issue by applying a machine learning model to reduce the VMs migration number and energy consumption. The proposed algorithm (named VMLM) is based on improving VM's migration process and selection. It has been benchmarked with JVCMMMD and EVSP solutions. The simulation results demonstrate the efficiency of our proposal, which includes two phases the machine learning preparing stage and the VMs migration stage.

**Keywords** Cloud computing · Virtualization · VM migration · Machine learning · Energy consumption

---

✉ Ali Belgacem  
a.belgacem@univ-boumerdes.dz  
Saïd Mahmoudi  
Saïd.MAHMOUDI@umons.ac.be  
Mohamed Amine Ferrag  
mohamed.ferrag@tii.ae

<sup>1</sup> LIMOSE laboratory, Faculty of Sciences, M'hamed Bougara University, Boumerdes, Algeria

<sup>2</sup> Mons University, Mons, Belgium

<sup>3</sup> Technology Innovation Institute, Abu Dhabi, United Arab Emirates

## 1 Introduction

One of the essential contemporary advances in information technology is the emergence of the cloud computing paradigm. A cloud is a collection of hardware and software connected over a network. It has been widely adopted by people, businesses, and large companies. The main idea behind cloud computing is scalability, and virtualization is the critical technology that makes this possible [1]. In its broadest sense, virtualization creates a virtual platform of server operating systems and storage devices within a single physical computer. This allows multiple virtual machines to be provisioned simultaneously. So, virtual machines can share one physical machine. In other words, the virtualization tool simulates hardware resources to create a fully functional virtual machine capable of installing an operating system and associated applications just as one would on a physical machine (PM) [2].

Virtualization technology has changed how data centers are configured and operated by providing new mechanisms for better sharing and controlling data center resources. Specifically, virtual machine migration is an effective management strategy that affects data center performance. It gives the ability to adjust the state of virtual machines according to the required performance while allocating resources, enhance resource usage, adapt to internal failures, reduce power consumption, and improve task scheduling [3]. For example, when a VM crashes due to a bug, the whole system does not crash, allowing the developer to debug the problem. Hence, it helps troubleshoot the issues. In addition, this technique can be done when a physical machine needs to be updated or shut down for maintenance [4].

The VMs migration technique transfers a VM from one physical host to another while the VM is still running. It is specially applied in current data centers to make computing dynamic, flexible, platform-independent, and efficient use and sharing of resources. Also, VM migration within/across data centers happens for power management, load balancing, availability, and reduction of the SLA violation rate [5, 6]. To do so, the VM migration process requires transferring the CPU, memory, network, and disk state. The change in disk state can be avoided by sharing the storage among the hosts participating in the migration process. Then, the VM is suspended on the source machine and resumed on the target machine [4]. The VMs CPU states are also copied, ensuring that the machine is the same in operation and specifications once it resumes at its destination. On the other hand, the VM selection process involves choosing the appropriate VM for migration using specific criteria. It allows for discovering the most appropriate available PM for the selected VMs. Therefore, it is an integral part of VM migration. By the selection rules, we could balance the overloaded physical servers and manage the number of migrations. At the same time, it can lead to the minimization of energy consumption and network overhead [3].

In recent years, the development of machine learning (ML) has made new improvements in applying data-driven methods in computer science. It simulates human learning activities, studies ways to self-improve computers to obtain

further knowledge and skills, identifies existing knowledge, and continuously improves performance and achievement. The success of ML in making significant improvements over some current technologies, for example, in the field of image recognition, encourages us to use it to migrate virtual machines. Machine learning can be divided into supervised, unsupervised, semi-supervised, and reinforcement learning [7]. Supervised machine learning employs labelled datasets to train algorithms and predict future decisions. Its goal is to build a concise model of the distribution of class labels in terms of predictor features [8].

Indeed, before creating an ML model, the data should be processed. Data processing is the problem of processing an item based on the quantitative information of the various features that describe the item. So, first, an item is defined in terms of various properties, and then, data about a set of items of known classes are obtained from multiple sources. The unknown item type is determined using a classifier created with this set of items with known classes. In other words, building any machine learning model should provide it with raw data for training. Then, a classifier is used to assign class labels to test instances where the values of the predictive features are known, but the value of the class label is unknown [8].

Generally, existing VM migration optimization techniques fall under three categories, compression, deduplication, and checkpoint [9]. Also, there are different approaches to VM migration, such as pre-copy and post-copy [9, 10]. However, in this field, current solutions have a complex implementation and focus on optimizing VM selection rules or the VM migration process without achieving an optimal solution. Also, they did not attempt to apply machine learning to virtual machine migration. On the other hand, the increasing use of cloud computing resources makes available recorded data about resource allocation. The latter is the primary factor in building machine learning models. Knowing that, machine learning-based algorithms are essential because they provide insight into system behavior trends and support the development of new solutions. This motivates us to think of a solution based on a learning model to solve the problem facing the cloud environment. Specifically, this paper has the following contribution:

- Propose a solution-based machine learning model to improve virtual machine migration performance in cloud computing. This proposal explicitly aims to enhance virtual machine migration in terms of processes and selection to reduce the number of virtual machine migrations and energy consumption.

The rest of the paper is organized as follows: Section 2 presents related works. Section 3 gives a mathematical equation for the problem under study. Section 4 explains our proposed solution. Section 5 evaluates its efficacy through simulation experiments. Finally, Sect. 6 concludes the paper.

**Table 1** A sample of virtual machine migration optimization researches

References	Problem	Approach	VM migration improvement	
			Selection	Processes
[13]	Improving CPU utilization, VMM time and memory.	PRMA	✓	
[14]	Load balancing	SESA, AWFDP	✓	
[15]	Load balancing, Energy consumption, SLA violation	Power-aware VM selection policy	✓	
[16]	Improve VM migration performance	IOFollow scheme		✓
[18]	Performing the fixed length and variable length block-level deduplication.	Adaptive deduplication strategies		✓
[19]	Decrease the migration cost and the energy consumption	MVC	✓	
[20]	Energy consumption and load balancing	ERVS	✓	
[10]	Network-aware live migration of VMs.	Static adjustment algorithm of live migration	✓	
[23]	Migration time, VM downtime	Ada-Things		✓
[32]	VM packing problem with the multi-resource constraints	Approximation algorithm	✓	
[33]	Reliability and energy consumption	Markov chain model	✓	
[22]	VMs placement, SLA violation	PAPSO	✓	
This paper	Energy consumption, VMs migration performance	Machine learning model	✓	✓

## 2 Literature review

Dynamic resource allocation in cloud computing has been studied from different perspectives [11, 12]. One of the important research trends in this area is the issue of virtual machine migration. It is covered in the literature in various aspects (Table 1). Pal et al. [13] introduced the idea of probabilistic resource cross-cloud live mitigation for efficient use of unoccupied memory and processor. The method proposed in [14] is used to manage VMs in clusters according to the value of their memory and CPU parameters. Then, it processes the collocated virtual machines that share some of their memory pages and are placed on the same physical machine as a group. Then, the migration decision is made based on assessing the whole system. Mandal et al. [15] proposed a new power-sensitive VM selection policy (EVSP) that achieves high energy savings without incurring service level agreement (SLA) violation. Migrated VMs are selected based on current VM usage and resource allocation. In [16], the authors implemented the IOFollow scheme to improve the performance of VM live migration. This work has a significant impact on shared disk bandwidth and system resources.

Other authors have discussed dynamic VM migrations to solve the time-sensitive network problem in cloud computing. The proposed approach is divided into two phases: offline and online scheduling phase [17]. The offline phase is based on a search graph using a tree at a minimum distance to increase reusable rescheduling results. In contrast, the online phase aims to accelerate the rescheduling process according to a heuristic scheduling approach to reuse the results of the offline phase. Tyj et al. [18] improved live migration processes for virtual machines by detecting and removing the zero, similar and redundant memory copies. This mechanism aims to adapt data deduplication to decrease the virtual machine disk image file size, thereby reducing migration time and downtime. However, its impact on cloud computing services has not been studied, and the criteria for selecting virtual machine migration have not been considered. In [19], researchers addressed the performance degradation and service interruption of migrated VMs and VMs consolidation issues. They focused on improving the migration cost caused by VM migration and the remaining runtime of the migrated VM. The study of its impact on energy consumption has shown an accepted decrease.

Cloud VM migration feature is used by Mekala et al. [20] to process data streams for storage generated due to IoT requests. Virtual machines are placed based on overloaded and underloaded hosts and virtual machine types. This is achieved by expecting CPU usage rate and power consumption. Similarly, in [21, 22], they used the virtual machine migration technique to reduce power consumption. In [21], the Markov chain makes it possible to predict the threshold value to make the migration decisions while in [22], the authors relied on a heuristic algorithm to determine the near-optimal placement of the migrated VMs. In the paper [23], the researchers proposed a live migration strategy for IoT applications. This strategy overcomes the constraints related to the generality of the applications and the performance imbalance by selecting the appropriate memory transfer to

obtain an optimal live migration according to the current dirty page rate. VM migration is also discussed in [24] to address the issue of VM consolidation. The authors proposed a new algorithm based on bin-packing heuristic and medium-fit to compromise energy consumption and SLA violation. However, they did not include traffic effects in this study.

The authors in [25] have managed energy consumption by adjusting the VM consolidation. The proposed solution focused on partitioning PMs into different groups based on their workload levels. It then employed a coalitional-game-based VM consolidation algorithm (GCMS) in choosing members from such groups to form effective coalitions. As a result, it has shown few migrations compared to the greedy approach but less than the other approaches. Similarly, in [26], they improved energy consumption by considering the advantage of a joint virtual machine and container migration approach concurrently (JVCMMMD). However, this research focused only on the CPU criteria, not including RAM, network bandwidth, memory, or cooling. In [27], the authors use the VM migration technique to detect internal and external attacks from cloud systems and their migration process. In [28], the authors improved the VM migration performance by searching the migrated VM, composting, and collecting sets of VM migration to achieve trade-offs between energy consumption and physical machine performance. The live migration technique of virtual machines is highlighted in [29] to alleviate network overhead when moving virtual machines to source and destination hosts.

The idea of the work [30] is based on the gravity model. The authors used the theory of gravity on various logical objects to manage the migration of virtual machines to effectively conserve energy consumption, load balancing, and quality of service. Likewise, the main objective of the work [31] was to achieve a good compromise between reducing energy consumption and meeting SLA constraints. The authors of [32] addressed VM consolidation as a packing problem. To maximize the number of packed VMs on a given PM, they apply a heuristic algorithm, thereby reducing network traffic in migration. Moreover, in [33], the researchers used the Markov chain model to keep the system's reliability at a desirable level.

A hybrid optimization algorithm for VM migration was presented in [34]. The cuckoo search algorithm and the particle swarm optimization technique are incorporated to reduce unnecessary migrations. This work still needs further improvement regarding the method for selecting virtual machines. Gupta et al. [35] have introduced a methodology related to energy consumption by limiting the VMs migration and switching off the guests depending on a threshold, thereby enhancing the remaining network bandwidth in the data center with minimal QoS degradation. However, they did not address the problem caused by moving virtual machines. To maintain system load balancing, the authors in [36] proposed a method based on VM selection under of-line. It consists in selecting one or more potential VMs for migration to reduce the resource load of the active servers. From the experiment results, this solution needs further improvement to obtain an optimal solution. In [37], the VMs migration problem is migrated by applying the game theory to ensure both load balance and resource utilization. However, the proposed algorithm has not addressed the VMs migration selection and processes. Also, its impact on energy conception has not been studied. In [38], the

authors present a load-balancing mechanism based on evolutionary computing. They consider the maximum CPU and memory utilization criteria for selecting migrating VMs.

### 3 Problem definition and formulation

The studied problem is described by determining the required notations and formulation. We consider that the servers are distributed and equipped with cloud computing resources, and the VM moves from one place to another. In this work, we address the pre-copy VM migration type. It is a classical approach used by the vendors such as Xen, KVM, and VMware, that try to keep the shorter downtime by transferring fewer data during the stop-and-copy phase [39]. Virtual machine migration needs moving CPU state, memory state, network state, and disk state. During this time, the VM is suspended at the source host and resumed at the target host. Thus, the memory page is copied, ensuring the VM is in the same operation and specifications once it resumes at the destination (Fig. 1).

#### 3.1 VMs migration attributes

VMs migration is affected by the cloud environment state and each VM state during the execution of incoming tasks. So, VM traffic has the following attributes:

- Arrived tasks rate ( $AT$ )
- Load-balancing rate ( $LB$ )
- Captured VM processor utilization ( $CCPU$ )
- Currently allocated memory ( $CAM$ )
- Availability of target host resources ( $ATHR$ )

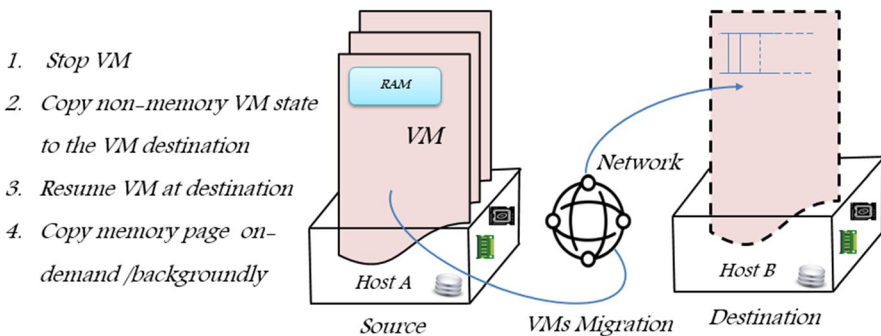


Fig. 1 VM migration process

### 3.2 Energy consumption

The energy consumed during paging depends on the VM's CPU, network bandwidth (BW), and migration time ( $\varphi$ ). Therefore, if the average dissipated power is denoted  $\theta_r$ , the energy consumption during migration time ( $\theta_m$ ) is given by:

$$\theta_m = \theta_r * \varphi \quad (1)$$

VM migration time is calculated as the total time spent in the migration process as follows:

$$\varphi = \frac{\beta}{BW} \quad (2)$$

where  $\beta$  is the size of the disk image in megabytes (MB),  $BW$  is the bandwidth allocated to the migration of the VM in MB/s, and  $\varphi$  is the time predicted in seconds.

### 3.3 Number of VMs migration

The network load ( $v_m$ ) generally reflects the number of VMs migrations ( $\rho$ ). The amount of data transferred during the virtual machine migration affects the network bandwidth. It is calculated according to the size of the original disk data and the availability of network bandwidth (Eq. (3a)).

$$\rho = \sum v_m \quad (3)$$

$$v_m = \beta + (\varphi_m * r_m) \quad (3a)$$

where  $r_m$  is the disk dirty rate.

## 4 Proposed VM migration learning model

The proposed migration strategy is based on two main phases. The first is to create a learning model, and the second is to use this model to move virtual machines (Fig. 3).

### 4.1 Machine learning preparing stage

This section explains the essential steps for creating a training model for the next stage.

#### 4.1.1 Data selection

To construct a machine learning model, we first required a dataset. Data should be collected from authoritative databases such as those provided by google cloud, amazon cloud, etc. It is the process of preparing the raw data about the VMs migration



circumstances and making it suitable for building the machine learning model. This is mandatory to put data in a formatted way. Each VM is characterized by attributes that reflect its state in the cloud environment. These data generally contain noise, and missing values may be unusable and cannot be used directly for the machine learning model.

### 4.1.2 Features choosing

It is the feature extraction process from raw data to enable the application of algorithms. After selecting the data, we must extract the appropriate properties for the expected improvement. This can be formulated by giving the study a set of  $m$  learning items  $\{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ , build a hypothetical classifier  $G : x \rightarrow y$  that maps an item  $x \in X$  to a class label  $y \in Y$ . Briefly, the classifier maps the input data to a specific class.

Therefore, this stage includes the selection of appropriate algorithms to construct the classifier. Each specific algorithm allows VM migration to be optimized in terms of VMs selection or migration process. We analyze the state of the cloud environment by measuring the number of virtual machine migrations ( $\rho_i$ ) and the energy consumption ( $\theta_i$ ) of each algorithm. Then, the average value is calculated as shown in Eq. (4).

$$\text{Avg} = \frac{(\rho_1 + \rho_2 + \dots + \rho_l) + (\theta_1 + \theta_2 + \dots + \theta_l)}{2l} \tag{4}$$

where  $l$  is the number of implemented algorithms in this phase. In this way, rows of data are categorized as follows:

$$\text{Classe}(x_i) = \begin{cases} \text{Good} & \text{if } \varphi, \text{ and } \theta < \text{Avg} \\ \text{Bad} & \text{Otherwise} \end{cases} \tag{5}$$

Table 2 gives an example of the VMs migration classification. It allows knowing if an element is good or bad considering the numerical values of the attributes.

**Table 2** An example of a data set with features

Item	Attributes					Class
	AT	LB	CCPU	CAM	ATHR	Label
$x_1$	0.29	0.448	262	363	360	Bad
$x_2$	0.84	0.195	401	349	223	Bad
$x_3$	0.37	0.216	215	217	267	Good
$x_4$	0.12	0.511	328	379	268	Bad
$x_5$	0.55	0.656	323	297	104	Bad
$x_6$	0.46	0.613	217	224	296	Bad
$x_7$	0.89	0.377	490	277	186	Bad
$x_8$	0.72	0.672	276	240	209	Bad
$x_9$	0.58	0.472	258	118	399	Good
$x_{10}$	0.13	0.913	429	365	241	Bad

### 4.1.3 Learning model (LM) creation

After the classifier is created from training data, its performance is assessed on independent test data (also called validation data) to determine how accurately it ranks unknown instances. Classifier performance is evaluated by a *cross-validation* technique to assess the quality score of a subset of features [40]. Cross-validation provides a more accurate model performance assessment by combining prediction performance measures on different subsets of data. This technique involves reserving a particular sample from the data set on which we are not training the model. Our model will be tested on this sample before finalizing it.

Here, the k-fold method has been used to perform cross-validation. This technique helps reduce variance, avoid overfitting, and gives insight into how the model will behave in practice. In k-fold cross-validation, the data set ( $D$ ) is randomly split into  $k$  "folds"  $D_1, D_2, \dots, D_k$  of approximately similar size. In each trial, one of these folds becomes the testing set, and the rest of the data becomes the training set. In other words, we repeat this process  $K$  times, with each fold being the selected test set once. The rest of the  $k - 1$  subsets constitute the training set for driving the model during each trial (Fig. 2).

Machine learning model evaluation After that, we get an overview of the classification statistics, such as the number of wrong instances. We record the error seen in each prediction. These statistics are restored to a single value (the accuracy  $A_i$ ). Final accuracy  $A$  is the sum of accuracy per round divided by the number of rounds (Eq. (6)).

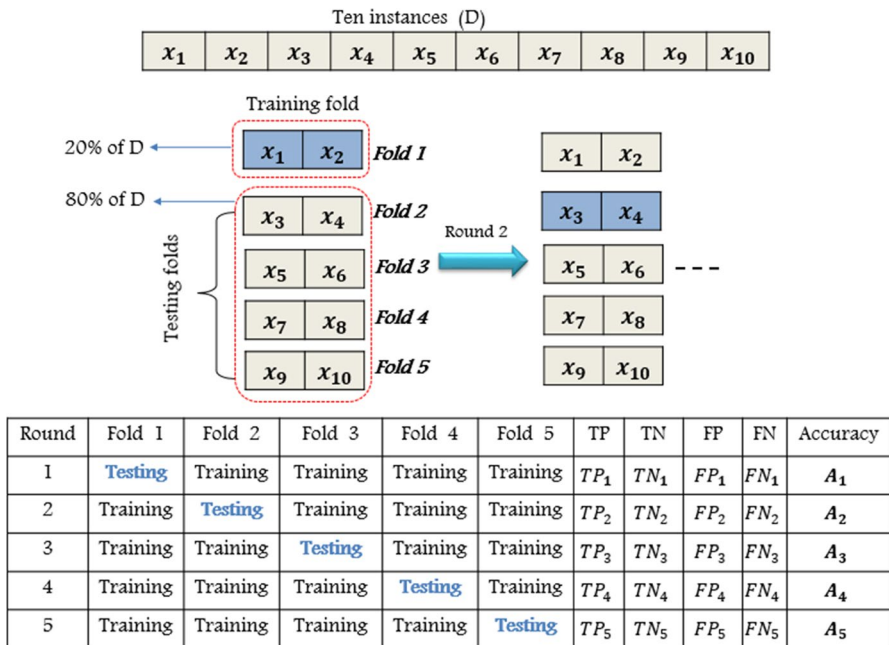


Fig. 2 Learning model validation steps

$$A = \frac{1}{\text{Rounds}} \sum A_i \quad (6)$$

$$A_i = \frac{\text{TN} + \text{TP}}{\text{TN} + \text{TP} + \text{FN} + \text{FP}} \quad (6a)$$

where

- *True positives (TPs)* True positives are the cases where we expect the virtual machine migration to be good when the actual migration gives an improvement.
- *True negatives (TNs)* Cases when we predict the migration as bad when the actual migration does not give an improvement.
- *False positives (FPs)* It is the case when we expect the migration to be good while the actual migration gives no improvement.
- *False negatives (FNs)* It is the case when we expect the migration to be bad while the actual migration gives an improvement.

## 4.2 VM migration stage

The proposed algorithm analyzes the conditions of the migratable virtual machines in each quantum time (we maintain the definition of quantum time ( $\Delta$ ) used in our previous work [41]). Then, it selects the good decisions that achieve the minimum energy consumption and the number of migrations. Thus, the main steps involved in this stage are given as follows:

- *VMs and cloud environment detection* In this stage, the broker gives information about the available resources on the hosts and the overloaded VMs. Additionally, it calculates the arrival rate of requests and the overall cloud environment load. Also, the available hosts to which the virtual machines will be migrated are specified.
- *VMs migration decision-based the created learning model* The learning model evaluates each migration to determine what allows for minimal energy consumption and migration time. More precisely, the VMs that should be migrated from overloaded and underloaded hosts are selected using first stage learning model.
- *Stop criteria* The VM migration-based machine learning model algorithm (VMLM) is called for each quantum time. The program stops if the number of LMLM instances is browsed. If a good decision does not make, the algorithm waits for the new attributes. Otherwise, the iteration is terminated, and the best decision will be taken.

Figure 3 summarizes the main stages of the proposed solution. Algorithm 1 shows the pseudo-code of creating a machine-learning model, while Algorithm 2 presents the proposed solution. The flowchart of the proposed solution is given in Fig. 4.

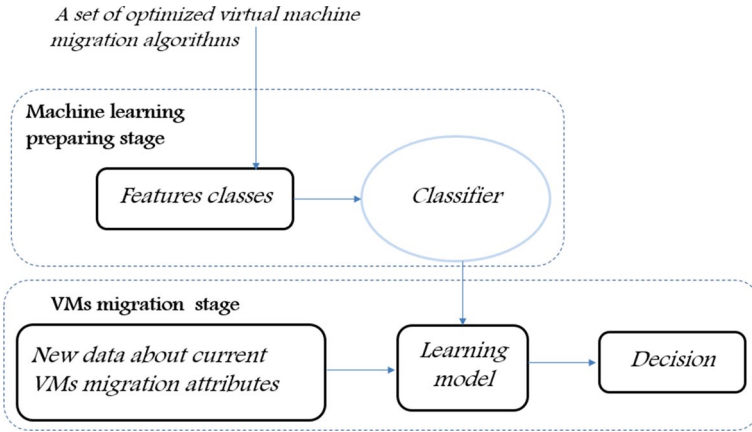


Fig. 3 Proposed solution phases

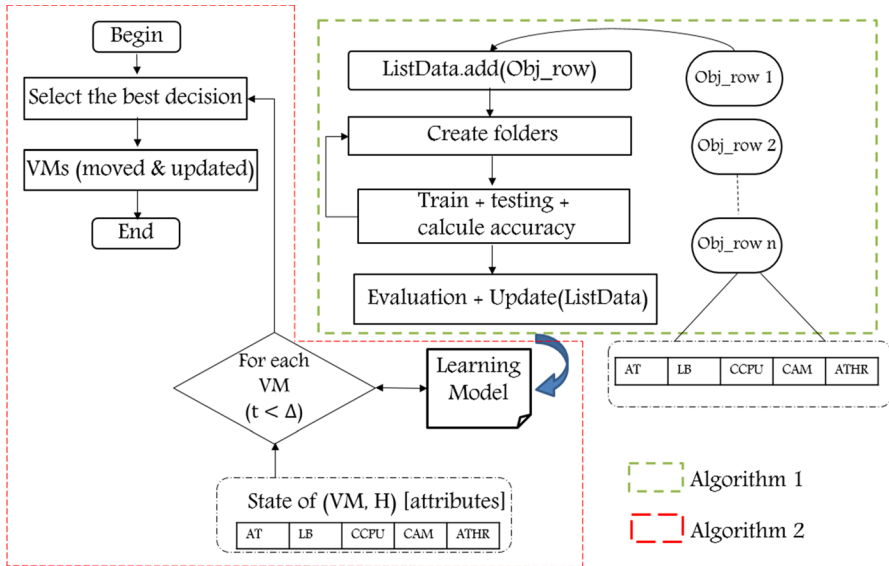


Fig. 4 Schematic diagram of the proposed solution

<p><b>Algorithm 1: Machine learning model</b>  <b>Input:</b> ListData()=[set of obj.row]  <b>Output:</b> Learning model  1: <b>For</b> (i=0; i &lt; round)  2:     <b>For</b>( j=0; j &lt; <math>\frac{1}{20}</math> ListData.size);  3:         raw_number=random();  4:         FoldTrainList().add(ListData().get(raw_number));  5:     <b>EndFor</b>  6:     <b>ForEach</b> (Fold)  7:         Train(FoldTest, FoldTrain),  8:         CalculTP(); CalculTN(); CalculFP(); CalculFN();  9:         CalculAcc();  10:     <b>EndForEach</b>  11: <b>EndFor</b>  12: CalculTotalAcc();  13: Update(ListData);</p>
---

<p><b>Algorithm 2: VMs migration based LM (VMLM)</b>  <b>Input:</b> VMs.attributesList(), ListAvailableHosts(), ListData(), Integer Decision  <b>Output:</b> Best VMs distribution  1: <b>While</b>(t &lt; Δ)  2:     <b>ForEach</b>(VM)  3:         Update(VMs.attributesList(), ListAvailableHost());  //Determines the best decision that returns the min (θ and v)  4:         <b>For</b>(i &lt; ListData.size())  5:             <b>If</b>(Decision &gt; BestDecision(VM, ListData()))  6:                 Decision=BestDecision(VM, ListData());  7:                 VMsListMig(VM,Decision);  8:             <b>EndIf</b>  9:             i++;  10:         <b>EndFor</b>  11:     <b>EndForEach</b>  12:     t ++;  13: <b>EndWhile</b></p>
---

## 5 Performance evaluation

This section presents a simulated cloud environment and experimental setup, and the results are presented and discussed.

### 5.1 Implementation tools

Cloud environment components are simulated using CloudSim 3.0.3. toolkit. We used Python 3 in this work to construct the learning model. It is one of the best

options for data science and machine learning practitioners. The experiments are performed on an Intel (R) Core (TM) i5 3320 M Processor 2.6 GHz, equipped with 4GB RAM, Windows 7 platform, using the Eclipse IDE Luna release 4.4.0.

## 5.2 Cloud environment and setup

Simulations of the proposed solution and comparing methods are performed under the same conditions to establish an objective comparison. Experiments are conducted on data centers containing 800 heterogeneous physical nodes; 400 of them consist of HP ProLiant ML110 G4 servers powered by Intel Xeon3075 clocked at 2660 MHz, and the rest consist of HP ProLiant ML110 G5 servers powered by Intel Xeon3040 clocked at 1860 MHz, each having 4 GB of RAM. For creating these server types in CloudSim, we create an extended “PowerModelSpecPower” class named “ProLiantServers” to implement PowerModel class from the power model package in “cloudbus” residing in CloudSim.

The experiment is conducted for more credibility based on a real system’s workload traces. Thus, the investigations are carried out according to a comparative analysis of the online workload of PlanetLab [42]. PlanetLab workload is a set of CPU usage traces from virtual machines measured over ten random days in March and April 2011. For reading workload traces and returning appropriate values during simulation, we use UtilizationModelPlanetLabInMemory included in CloudSim. We get the CPU utilization of a VM by calling `vm.getTotalUtilizationOfCpuMips(time)`. The performance of the proposed VM migration solution was assessed by comparing it with energy-aware VM selection policy in (EVSP) [15] and multi-criteria migration decision (JVCMMMD) algorithm [26]. The first algorithm showed effective results compared to the existing classic selection algorithms, such as maximum correlation (MC), minimum migration time (MMT) and random selection (RS). The second method shows a significant reduction in energy consumption and the number of virtual machine migrations.

## 5.3 Energy consumption performance

The behavior of the proposed algorithm in terms of energy consumption is tested according to the two important metrics. The first is to analyze the number of active physical servers in each quantum time. The second focuses on calculating the total energy consumption during the experimental period.

As shown in Fig. 5, the number of used physical servers obtained by the proposed algorithm is smaller than other algorithms. Here, the lower the value of the number of active physical servers, the lower the energy consumption of the cloud data center. Similar results are shown in Fig. 6. This means that the proposed solution can be more efficient in energy-saving than JVCMMMD and EVSP algorithms in runtime. The reason lies in: For VM migration, the VMLM algorithm captures the processor utilization, then determines the minimum capacity of the remaining resources that can meet the resource requirements of the VM from the resource availability

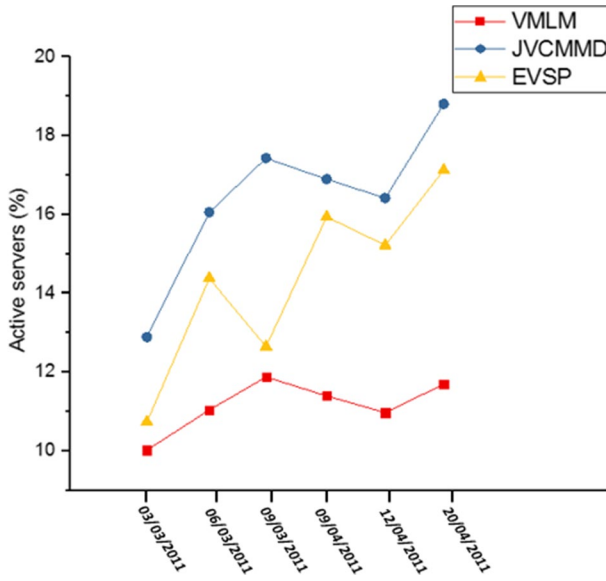


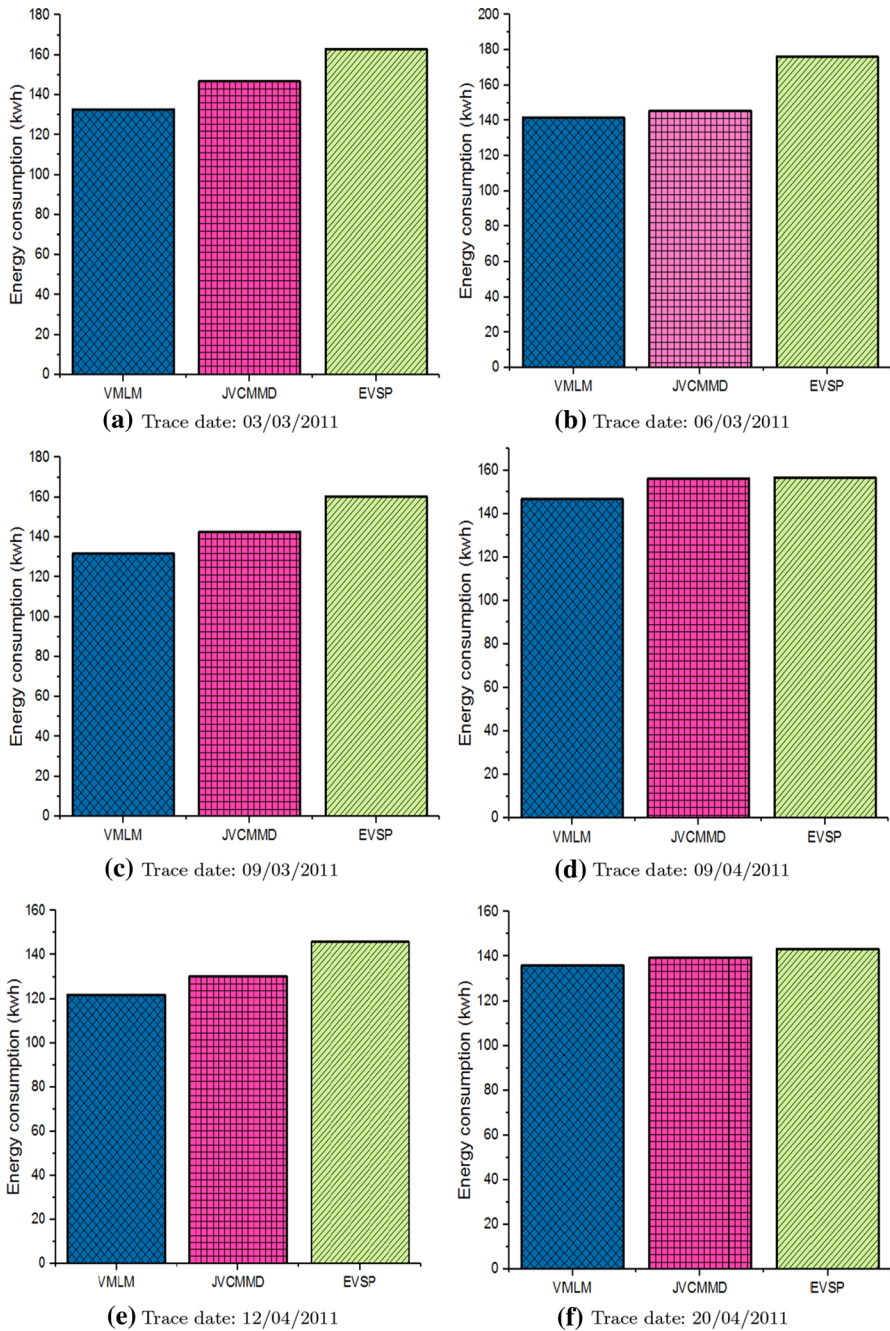
Fig. 5 Number of active servers for each trace date

of the target host. Other algorithms always select the host with the lowest resource usage, negatively affecting the virtual machine migration processes. Thus, energy consumption can be increased.

### 5.4 VM migration

In each quantum time, the number of migrated virtual machines is counted. Figure 7 shows the results obtained by the three comparison algorithms during the entire experiment period. It can be seen that the values obtained by the JVCMMMD algorithm and EVSP algorithm are close to the same. However, the proposed solution performs well than other algorithms. In particular, the total number of virtual machines obtained by the JVCMMMD and EVSP algorithms is 15.4% and 26.7%, respectively, which is greater than that of the VMLM algorithm. This means that the proposed solution algorithm improved the cloud data center’s performance during the VM migration process.

The reasons are given as follows: first, when choosing VMs to migrate, the proposed solution considers the load-balancing rate. If the specified virtual machine load is less than the cloud environment load balancing, then the VMLM algorithm does not migrate and thus can reduce some unnecessary migration; secondly, when VM migration is needed, the proposed VMLM algorithm chooses the virtual machine with the lowest currently allocated memory from the eligible virtual machines. However, JVCMMMD makes a decision on whether containers or virtual machines should be migrated, while virtual machine migration in the EVSP algorithm is associated with system performance degradation, which can increase the

**Fig. 6** Energy consumption



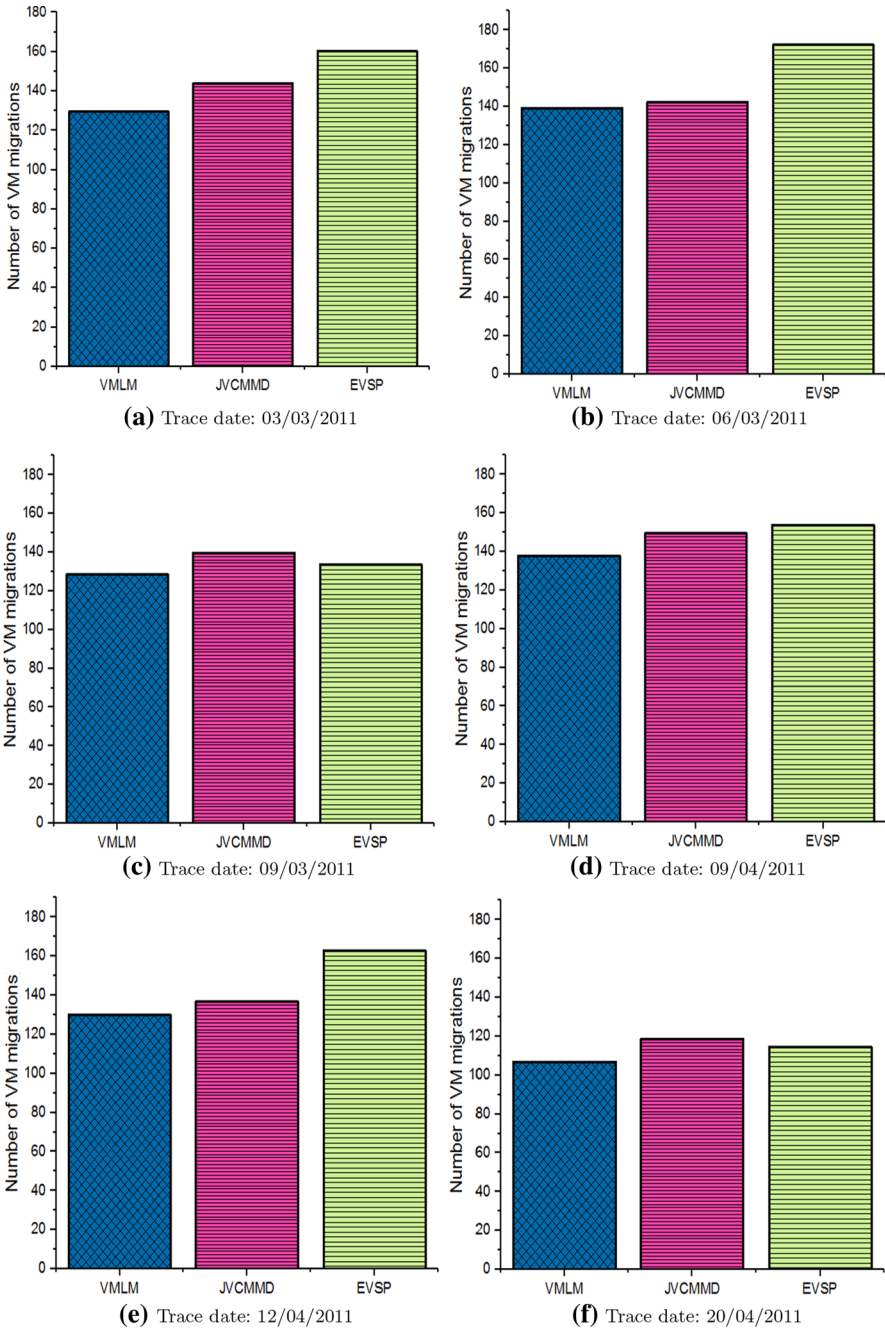
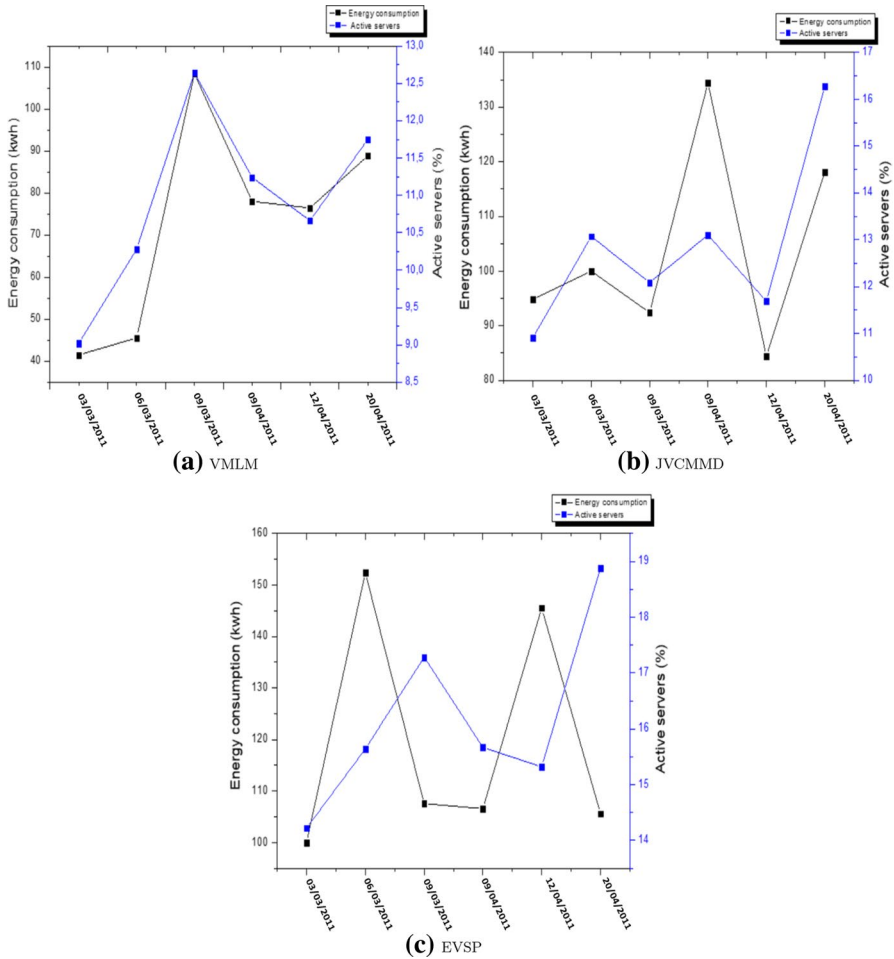


Fig. 7 VMs migration



**Fig. 8** Energy consumption and number of active servers in each trace date

mobility of VMs. Therefore, considering the effects of the load-balancing rate (CCPU) and currently allocated memory (CAM) in selecting the VMs to migrate can effectively avoid unnecessary migration, thereby improving VM migration processes.

The comparative experiments results for energy consumption average and the number of active servers registered on each trace date are shown in Fig. 8. The first observation of the proposed solution curves reveals that they are almost identical. This reflects the proportion between the servers used and the energy consumed. Moreover, there are few servers compared to other solutions due to the excellent distribution of VMs on the hosts. This is not achieved by JVCMMMD and EVSP algorithms (Fig. 8b and 8c). More precisely, migrated virtual machines are effectively selected with minimal energy consumption. Knowing that fewer migrated VMs means less performance degradation

**Table 3** Statistics results for energy consumption

Trace date	JVCMMMD			VMLM			Improvement (%)			<i>t</i> -test
	Best	Worst	Avg	Best	Worst	Avg	Best	Worst	Avg	
03/03/2011	218.39	494.36	252.79	124.08	296.87	217.81	56.82	60.05	86.16	5.847
06/03/2011	491.31	560.01	340.99	113.05	307.28	209.59	23	54.87	61.47	9.458
09/03/2011	383.75	673.04	452.49	114.62	290.95	221.01	17.03	75.82	48.84	4.435
09/04/2011	434.03	584.35	386.8	122.24	293.02	194.76	20.92	67.51	50.35	6.626
12/04/2011	209.46	742.33	452.51	115.57	293.07	198.41	55.18	39.48	43.84	10.838
20/04/2011	357.11	583.02	404.84	114.17	293.82	179.57	31.97	50.4	44.35	8.291

**Table 4** Statistics results for VMs migration

Trace date	JVCMMMD			VMLM			Improvement (%)			<i>t</i> -test
	Best	Worst	Avg	Best	Worst	Avg	Best	Worst	Avg	
03/03/2011	327	790	465.73	204	470	342.87	62.39	59.49	73.62	6.982
06/03/2011	232	599	374.4	272	499	378.67	117.24	83.3	101.14	4.561
09/03/2011	414	868	598.67	232	452	322.53	56.04	52.07	53.88	6.702
09/04/2011	326	674	465	201	483	371.07	61.66	71.66	79.8	5.247
12/04/2011	484	794	587.47	227	480	333.8	46.9	60.45	56.82	-6.463
20/04/2011	224	800	642.47	236	493	355.2	105.36	61.63	55.29	-7.548

due to migration, and thus, VM migration processes are improved. Also, VMLM takes into account CCPU and ATHR parameters, which play an essential role in enhancing migration processes.

## 5.5 Statistical analysis

Here, a statistical analysis of the results obtained is presented. Tables 3 and 4 show the *t*-test result of our proposed algorithm and the JVCMMMD algorithm for the six benchmarks regarding energy consumption and virtual machine migration. The comparison is made only with JVCMMMD because the above experiments show that VMLM gives the best results compared to EVSP. For each trace date, we take 15 samples, after which we calculate the *t*-test using SPSS software. It can be concluded from the obtained results that there is a statistically significant difference between the proposed solution and the JVCMMMD algorithm. *T*-tests showed that our proposed solution gives a significantly lower value for JVCMMMD with a  $P < 0.05$  value. Therefore, we can deduce that

our proposed solution has the best performance concerning energy consumption and VMs migration number metrics.

## 5.6 Complexity analysis

Algorithm 1 is just for creating a machine learning model used during resource allocation. It does not participate in virtual machine migrations. Therefore, the complexity of the proposed solution relies on Algorithm 2. The complexity of the VMLM algorithm mainly depends on the size of the learning model and the number of virtual machines. Assume that  $N$  and  $M$  represent the total number of VMs and the size of the learning model, respectively, in the cloud. The time complexity of Algorithm 2 is  $O(N \times M)$  since Algorithm 2 traverses all instances of the learning model for each virtual machine. If the number of instances in the machine learning model equals the number of virtual machines, the complexity will become  $O(N \times N) = O(N^2)$ .

## 6 Conclusion

We have proposed a machine learning-based solution that manages the VMs migration in a coordinated fashion to enforce higher-level performance in energy consumption and minimize the number of VM migrations. By learning the state of the cloud environment and the behavior of VMs, our solution makes it possible to anticipate decisions about VM migration at run time. As a result, it becomes possible to choose the correct destination for each immigrant VM that provides minimal power consumption and takes a short time.

Our evaluation uses real workload traces at different dates to simulate a cloud environment. It shows significant gains for the proposed solution compared to other algorithms. We deduce that a machine learning-based solution for migrating virtual machines allows for high performance in workload distribution, but this is only possible if an effective learning model is included. This can be achieved by choosing efficient algorithms for classifying the learning model features. Moreover, the complexity of the VMLM algorithm is lower, which allows for getting the best results quickly. In addition, it takes into account the features of the cloud environment (AT, CCPU, etc.), allowing it to adapt to dynamic resource allocation automatically.

To our knowledge, this work is the first attempt to use a supervised machine learning model for a virtual machine migration problem in a cloud environment. The proposed learning model is designed to improve virtual machine migration selection and processes. This work can be further extended by using other mechanisms for creating the learning model with other attributes.

**Acknowledgements** Not applicable.

**Author contributions** AB conceived and designed the analysis, collected the data, contributed data or analysis tools, performed the analysis and wrote the paper. Other contribution: all authors reviewed the manuscript.

**Funding** The author declares that there is no funding.

**Data availability** Not applicable.

## Declarations

**Conflict of interest** The author declares that there is no conflict of interest.

**Ethics approval and consent to participate** Not applicable.

**Consent for publication** Not applicable.

## References

1. Erl T, Puttini R, Mahmood Z (2013) Cloud computing: concepts, technology, & architecture, Pearson Education
2. Hall J, Andrews J (2020) VMware certified professional data center virtualization on vSphere 6.7 study guide: exam 2V0-21.19, John Wiley & Sons
3. D-N Le et al. (2018) Cloud computing and virtualization, Scrivener
4. Le D-N, Kumar R, Nguyen GN, Chatterjee JM (2018) Cloud computing and virtualization, John Wiley & Sons
5. Wang Y, Wang X (2014) Performance-controlled server consolidation for virtualized data centers with multi-tier applications. *Sustain Comput Inf Syst* 4(1):52–65
6. Shirvani MH, Rahmani AM, Sahafi A (2020) A survey study on virtual machine migration and server consolidation techniques in dvfs-enabled cloud datacenter: taxonomy and challenges. *J King Saud Univ Comput Inf Sci* 32(3):267–286
7. El Naqa I, Murphy MJ (2015) What is machine learning? In: *Machine learning in radiation oncology*, Springer, pp 3–11
8. Kotsiantis SB, Zaharakis I, Pintelas P et al (2007) Supervised machine learning: a review of classification techniques. *Emerg Artif Intell Appl Comput Eng* 160(1):3–24
9. Noshay M, Ibrahim A, Ali HA (2018) Optimization of live virtual machine migration in cloud computing: a survey and future directions. *J Netw Comput Appl* 110:1–10
10. He T, Toosi AN, Buyya R (2019) Performance evaluation of live virtual machine migration in sdn-enabled cloud data centers. *J Parallel Distrib Comput* 131:55–68
11. Belgacem A (2022) Dynamic resource allocation in cloud computing: analysis and taxonomies, *Computing* 1–30
12. Belgacem A, Beghdad-Bey K, Nacer H (2018) A new task scheduling approach based on spacing multi-objective genetic algorithm in cloud, In: *International Conference on Computer Science and Information Systems*, pp 189–195
13. Pal S, Kumar R, Saravanan K, Abdel-Basset M, Manogaran G, Thong PH et al (2019) Novel probabilistic resource migration algorithm for cross-cloud live migration of virtual machines in public cloud. *J Supercomput* 75(9):5848–5865
14. Nashaat H, Ashry N, Rizk R (2019) Smart elastic scheduling algorithm for virtual machine migration in cloud computing. *J Supercomput* 75(7):3842–3865
15. Mandal R, Mondal MK, Banerjee S, Biswas U (2020) An approach toward design and development of an energy-aware vm selection policy with improved sla violation in the domain of green cloud computing, *J Supercomput* 1–20
16. Mao B, Yang Y, Wu S, Jiang H, Li K-C (2019) Iofollow: improving the performance of vm live storage migration with io following in the cloud. *Future Gener Comput Syst* 91:167–176
17. Yu Q, Wan H, Zhao X, Gao Y, Gu M (2019) Online scheduling for dynamic vm migration in multi-cast time-sensitive networks. *IEEE Trans Ind Inf* 16(6):3778–3788
18. Tyj NM, Vadivu G (2019) Adaptive deduplication of virtual machine images using akka stream to accelerate live migration process in cloud environment. *J Cloud Comput* 8(1):1–12
19. Xu H, Liu Y, Wei W, Xue Y (2019) Migration cost and energy-aware virtual machine consolidation under cloud environments considering remaining runtime. *Int J Parallel Prog* 47(3):481–501

20. Mekala MS, Viswanathan P (2019) Energy-efficient virtual machine selection based on resource ranking and utilization factor approach in cloud computing for iot. *Comput Electr Eng* 73:227–244
21. Bhattacharjee S, Das R, Khatua S, Roy S (2020) Energy-efficient migration techniques for cloud environment: a step toward green computing. *J Supercomput* 76(7):5192–5220
22. Ibrahim A, Noshay M, Ali HA, Badawy M (2020) Papsos: a power-aware vm placement technique based on particle swarm optimization. *IEEE Access* 8:81747–81764
23. Wang Z, Sun D, Xue G, Qian S, Li G, Li M (2019) Ada-things: an adaptive virtual machine monitoring and migration strategy for internet of things applications. *J Parallel Distrib Comput* 132:164–176
24. Moges FF, Abebe SL (2019) Energy-aware vm placement algorithms for the openstack neat consolidation framework. *J Cloud Comput* 8(1):1–14
25. Xiao X, Zheng W, Xia Y, Sun X, Peng Q, Guo Y (2019) A workload-aware vm consolidation method based on coalitional game for energy-saving in cloud. *IEEE Access* 7:80421–80430
26. Gholipour N, Ariyanan E, Buyya R (2020) A novel energy-aware resource management technique using joint vm and container consolidation approach for green computing in cloud data centers. *Simul Model Pract Theory* 104:102127
27. AlKadi O, Moustafa N, Turnbull B, Choo K-KR (2019) Mixture localization-based outliers models for securing data migration in cloud centers. *IEEE Access* 7:114607–114618
28. Xu X, Zhang X, Khan M, Dou W, Xue S, Yu S (2020) A balanced virtual machine scheduling method for energy-performance trade-offs in cyber-physical cloud systems. *Future Gener Comput Syst* 105:789–799
29. Zhang F, Liu G, Zhao B, Fu X, Yahyapour R (2019) Reducing the network overhead of user mobility-induced virtual machine migration in mobile edge computing. *Softw Pract Exp* 49(4):673–693
30. Xu X, Zhang Q, Maneas S, Sotiriadis S, Gavan C, Bessis N (2019) Vmsage: a virtual machine scheduling algorithm based on the gravitational effect for green cloud computing. *Simul Model Pract Theory* 93:87–103
31. Haghghi MA, Maeen M, Haghparast M (2019) An energy-efficient dynamic resource management approach based on clustering and meta-heuristic algorithms in cloud computing iaas platforms. *Wireless Pers Commun* 104(4):1367–1391
32. Li H, Li W, Zhang S, Wang H, Pan Y, Wang J (2019) Page-sharing-based virtual machine packing with multi-resource constraints to reduce network traffic in migration for clouds. *Future Gener Comput Syst* 96:462–471
33. Sayadnavard MH, Haghghat AT, Rahmani AM (2019) A reliable energy-aware approach for dynamic virtual machine consolidation in cloud data centers. *J Supercomput* 75(4):2126–2147
34. Khan MSA, Santhosh R (2022) Hybrid optimization algorithm for vm migration in cloud computing. *Comput Electr Eng* 102:108152
35. Gupta A, Namasudra S (2022) A novel technique for accelerating live migration in cloud computing. *Autom Softw Eng* 29(1):1–21
36. Verma G (2022) Secure vm migration in cloud: Multi-criteria perspective with improved optimization model. *Wireless Pers Commun* 124(1):75–102
37. Tran CH, Bui TK, Pham TV (2022) Virtual machine migration policy for multi-tier application in cloud computing based on q-learning algorithm. *Computing* 104(6):1285–1306
38. Hung L-H, Wu C-H, Tsai C-H, Huang H-C (2021) Migration-based load balance of virtual machine servers in cloud computing by load prediction using genetic-based methods. *IEEE Access* 9:49760–49773
39. Kratsch S (2016) On polynomial kernels for sparse integer linear programs. *J Comput Syst Sci* 82(5):758–766
40. Berrar D (2019) Cross-validation
41. Belgacem A, Beghdad-Bey K, Nacer H (2020) Dynamic resource allocation method based on symbiotic organism search algorithm in cloud computing. *IEEE Trans Cloud Comput*
42. line O (2022) planetlab-workload-traces, <https://github.com/beloglazov/planetlab-workload-trace>, accessed: 2022-05-23

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted

manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.