Computer Communications 209 (2023) 14–25

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Enhanced resource allocation in distributed cloud using fuzzy meta-heuristics optimization

Arun Kumar Sangaiah [a,b], Amir Javadpour [c,d,*], Pedro Pinto [d], Samira Rezaei [e], Weizhe Zhang [c]

[a] International Graduate Institute of AI, National Yunlin University of Science and Technology, Yunlin, Taiwan
[b] Department of Electrical and Computer Engineering, Lebanese American University, Byblos, Lebanon
[c] Department of Computer Science and Technology, Cyberspace Security, Harbin Institute of Technology, Shenzhen, China
[d] ADiT-Lab, Electrical and Telecommunications Department, Instituto Politécnico de Viana do Castelo, Portugal
[e] Leiden Institute of Advanced Computer Science (LIACS), Leiden University, P.O. Box 9512, 2300 RA Leiden, The Netherlands

ARTICLE INFO

Keywords:
Cloud computing
Ant colony optimization
Neural fuzzy system
Scheduling
Real-time resource allocation
Virtual machine migration

ABSTRACT

Cloud computing is a modern technology that has become popular today. A large number of requests has made it essential to propose a resources allocation framework for arriving requests. The network can be made more efficient and less costly this way. The cloud–edge paradigm has been considered a growing research area in the computing industry in recent years. The increase in the number of customers and requests for cloud data centers (CDCs) has created the need for robust servers and low power consumption mechanisms. Ways to reduce energy in the CDC having appropriate algorithms for resource allocation. The purpose of this study was to develop an intelligent method for dynamic resource allocation using Takagi–Sugeno–Kang (TSK) neural-fuzzy systems and ant colony optimization (ACO) techniques to reduce energy consumption by optimizing resource allocation in cloud networks. It predicts future loads using a drop-down window to track CPU usage. By optimizing virtual machine migration, ACO can reduce energy consumption. Simulations are provided by examining the implementation and a variety of parameters such as the number of requests made wasted resources, and requests rejected. In this paper, we propose the use of virtual machine migration to accomplish two main goals: evacuating additional and non-optimal virtual machines (scaling and shutting down additional active physical machines) and solving the resource granulation problem. We evaluated and compared our results with literature for rejection rates of virtual and physical machine applications. The performances of our algorithms are compared to different criteria such as performance in request rejection, dynamic CPU resource allocation with reinforcement learning, multi-objective resource allocation, NSGAIII, Whale optimization and Forecast Particle Swarm allocation. A comparison of some evaluation criteria showed that the proposed method is more efficient than other methods.

1. Introduction

Cloud, Fog, SDN (software-defined networking), 5G, VANET, and wireless networks are becoming more and more popular. Small and large businesses alike are using them. Providing efficient resource allocation mechanisms through the extensive use of the cloud network, BigData, and Internet of Things (IoT) computing is [1–5]. Cloud computing, as a bridge between distributed computing and computing resource markets, has reached a critical level in the market. Using the ability to rent a shared server for a large number of users, the cloud made it possible for customers to use computing resources more efficiently. Unlike conventional models, which provide a separate system for each user regardless of their needs, cloud computing can allocate computing resources based on users' specific needs [6]. It provides access to shared hardware and software resources for multiple users simultaneously [7–9]. However, the issue of energy consumption still needs to be investigated to reduce the number of active resources available on each server [9–11]. In the same way that sustainability considers all aspects of preserving the environment holistically, cloud computing must also take into consideration the front end as well as the back end. As compared to previous studies that focused only on data center energy use, resource allocation in distributed cloud examined all aspects of cloud computing. Cloud computing service providers should minimize their energy consumption, but they should also take into account the energy needed to transfer data to and from their end users, as well as the energy used for the user interface. Comparing and reducing energy consumption in a cloud environment can be accomplished by looking at several factors, including 1. Optimizing hardware resources, 2. Using energy efficiently, 3. Reducing costs and 4. Reducing data center footprints [12,13].

* Corresponding author at: ADiT-Lab, Electrical and Telecommunications Department, Instituto Politécnico de Viana do Castelo, Portugal.
E-mail address: a.javadpour87@gmail.com (A. Javadpour).

https://doi.org/10.1016/j.comcom.2023.06.018
Received 24 August 2021; Received in revised form 7 July 2022; Accepted 21 June 2023
Available online 24 June 2023
0140-3664/

Some studies have presented new algorithms to deal with online and dynamic resource management [3,14–16]. It can be considered an optimization problem with less server maintenance and proper distribution of requests [9,10]. With proper resource allocation, servers can serve a maximum number of requests. Cloud Manager is responsible for placing virtual machines on physical hosts. Because of the limited number of hosts, if the number of demands increases, resource allocation is a substantial challenge. There are three main aspects of resource allocation including virtual machines that host user requests, physical machines, and an efficient mapping technique between virtual machines and cloud data centers (cDC) [9,17]. In the following equations, PM is a list of physical machines. Eq. (2) shows that for the jth server, the total server capacity for each server must be greater than the resource requested by the virtual machines on the jth server.

$$PM(\text{Physical Machine}) : \{PM_{p1}, PM_{p2}, \dots, PM_{pj}, \dots, PM_{pm}\} \tag{1}$$

$$\begin{cases} ForAll\_CPU_j > \sum_{k=1}^{n} cpu_k \\ ForAll\_RAM_j > \sum_{k=1}^{n} ram_k \\ ForAll\_Storage_j > \sum_{k=1}^{n} storage_k \end{cases} \tag{2}$$

A combination of two fuzzy and Ant Colony Optimization (ACO) algorithms is used in this research to provide a resources management pipeline in cloud data centers. A fuzzy algorithm is used for the proper management of virtual machines to provide load balancing, reduced energy consumption, overhead, etc. ACO is used to allocate appropriate virtual machines.

Several parameters are taken into account when predicting cloud resource management, including an ant colony algorithm and a Neuro-Fuzzy system. The first step is to consider workload balance parameters when migrating virtual machines. Then, hybrid meta-exploration methods like Ant-Genetic and Ant-PSO were used in place of Ant Colony. It is also suggested that reverse control-based controllers be used to improve synchronization periods. Resource prediction using Neuro-Fuzzy technology is designed as the first level of load distribution, specifically for handling failures of availability zones and simulation of datasets.

In Section 2 we have reviewed the most recent articles in the literature. Section 3 provides details on the presented method on how to use fuzzy and ACO for our specific task. In Section 4, the evaluation criteria, as well as the test dataset, are explained. In Section 5, the results are presented and evaluated in comparison with the literature. Finally, we have provided our concluding notes as well as the future directions.

2. Literature review

Energy consumption in the cloud has different parameters and can be examined in many aspects [18,19]. Equipment efficiency and Virtual Machine Security (VMS) are two most critical parameters in power consumption and can affect the number of PMS in cDC. On the other hand, the workload has an important effect on energy. Although this parameter does not directly affect power consumption in the cloud, it can be used to reduce active servers by maximizing the efficiency of active servers during response time. To consider all the parameters, different policies are taken which define a threshold or a combination of thresholds in different performance metrics. It can be adjusted based on the aim of the network, such as long-term resources or minimizing energy consumption. It is important to prevent crashes due to server overloads. Using constraint programming and linear programming, energy optimization algorithms have been provided. Researchers have proposed multifunctional methods to investigate other factors such as latency and energy consumption as well. Optimization of energy consumption is reviewed in the literature by metaheuristic algorithms such as Genetic Algorithm [20] and bee optimization algorithm [21], Whale Energy-Resource [21], or NSGA [22]. The proposed scheduling algorithm in this endeavor is based on the cDC scale of dynamic processor voltage, known as meta-exploratory programming. It also uses a special formula similar to a crossover in a genetic algorithm.

A multi-purpose method to reduce energy consumption in the cloud environment is presented in [26] using a financial scale. Because finding patterns of resource consumption is more effective, it might be perceived that it is more important than resource planning. However, virtual machine scheduling on cloud servers is a fundamental issue. In resource-based models, the user requests resources for the duration of the virtual machine lease. Resource forecasting is a way to optimize energy consumption and resource allocation in a way to respond to customers in real-time. Cong et al. [26] proposed an enhanced learning approach to providing dynamic resources in virtual cloud data centers. An adjustment control method based on layered reinforcement training is provided for resource allocation while dealing with uncertainty in a dynamic cloud environment. The controller is provided to prevent refusal to perform the task as a threshold-based goal and to minimize energy as an auxiliary goal [27].

Although the presented method in this work is not focused on using cloud forecasting to minimize energy consumption, energy optimization is the first and most important item for cloud management. In cloud computing problems, resource consumption and response time are directly related to energy consumption. Therefore, any effort to optimize resource consumption leads to energy optimization. Using cloud forecasting methods [28], resource management is intended to minimize the operating cost of energy use. This work is based on three main aspects: forecasting the workload, identifying resource requirements for the next workload, and allocating resources according to the service cost. In [29], resource allocation is used by considering the required number of resources to reconfigure the system. This defines the optimal configuration to minimize energy consumption and manage the service level agreement by predicting the number of required resources for the next round. A comparison of the advantages and disadvantages of the studies in the literature are shown in Table 1.

3. Proposed method

In the data center deployment process, Cloud services have become a major challenge due to the rapidly increasing functional components demand. Traffic Engineering is traditionally based on central rules, but in this highly distributed environment, these are typically insufficient to allow for the efficient use of IT and network resources. We have conducted preliminary simulation experiments to assess the number of requests allocated and the average server resource utilization based on our proposal (SDAR-NFAT). ACO is a method for discovering optimal responses and can solve NP problems such as virtual machine placement (VMCP) problems [8,13].

ACO is inspired by the colony of ants and how they interact indirectly with each other. Pheromones are a source of information for ants to find their way in search of food, as other ants mark the path to food by placing a pheromone on it. The advantage of this system is that it is more likely to choose paths that have a better chance of creating a solution. Fig. 1 shows how ants learn to use the optimal path over time using the path on which more pheromones are. Initially, where there is no pheromone for the paths, the ants choose a random path to move. After a while, this technique helps them to find food and return easily because shorter routes are more crowded over time than longer routes [14,16]. Fig. 1 also shows the concepts and ideas of the proposed method. It is an inference system based on Takagi–Sugeno–Kang (TSK) neural-fuzzy system and ACO. After calculating the fuzzy relations between the input–output pairs, a non-fuzzy converter is applied to obtain a non-fuzzy result of the interference. Our comprehensive system for modeling workload time-series data waits for

Table 1
Review and compare previous works on the advantages and disadvantages.

References	Method	Disadvantages	Advantages
[23]	-Threshold defining based policies in cDC, RL-based approach	-Low efficiency in user maintenance and energy consumption	Simplicity and operational Easy implement capabilities
[20]	-Using the genetic algorithm to optimize energy consumption in cDC, MultiObjectivePSO-based approach	-Incapable of real-time execution -Incapable of user maintenance	-Ability to develop and improve because of the easy formulation of variables in Genetic algorithm
[24].	A reinforcement learning method for resources in cDC	-Provides the ability to control uncertainty specifically in cloud markets	-Reduce the whole network energy -Optimal policies to attract and retain customers
[22]	-Using NSGA for resource allocation and reducing energy consumption, the NSGAIII-based approach	-Incapable of real-time execution -Incapable of user maintenance	-High performance in simulation
[21]	-A new method for scheduling has been presented based on energy consumption for a virtual machine that limits energy consumption by performance metrics, the WhaleOptimization-based approach	-Incapable of user maintenance -Too much complicated for executing on enormous systems	-Calculating energy consumption metrics -Specializing in the rule of energy consumption metrics
[25]	Using reinforced Presented a new scheduler for reducing energy-based VMs in cDC	The significant problems in VM systems and cDC.	-Compatible Planning in a Cloud environment -Reduce the whole network energy
[22]	-Presented for requirement prediction that prediction process is used to resource consumption optimization and accordingly energy consumption optimization	-Not paying attention to optimizing energy consumption during virtual machine movement	-Using pre-preparation to avoid rejecting requests -High capability in the user maintenance
[26]	-Introducing a set of approaches that prediction capability can bring enormous optimization into a cloud environment	-Lack of adaptive analysis to balance between user maintenance and resource usage optimization objectives	-Investigating the rule of effective parameters in prediction -Addressing different aspects of applying prediction methods
[27]	Using MPC Model Parameter Tuning	The main problem is resource efficiency in the cloud	Maximize Efficiency in the long-run simulation
[28]	-Introducing workload unique prediction methods in the cloud due to resource allocation management, PSOARMASVR-based approach	-Not paying attention to optimizing energy consumption during virtual machine movement	-The ability to maintain customers with resource pre-preparation -Studying the rule of workload prediction in optimizing resources and energy
[29]	-Using a unique resource requirement parameter prediction approach to reconfigure the resource allocation system to optimize resource consumption and accordingly energy	-Using configuration rather than process setting cause more complexity and decrease the adaptation speed	-Using request prediction parameter rather than workload due to optimizing resource consumption

the next cDC workload. The real-time method of setting the number of running servers is known as dynamic resource allocation in the cloud. It directly affects the quality of real-time services in the cloud by providing real-time services to customers and allocating the appropriate virtual machine on physical servers. Another approach usually involves an over-allocation policy that tries to satisfy existing customers, while new requests may be rejected due to mismanagement of resources. In addition, over-allocation greatly increases energy consumption. In this work, we present Dynamic Services Resources Aware - Neuro-Fuzzy Ant Colony (DSRA-NFAC) as an optimal architecture using the Neuro-Fuzzy system and the ant colony algorithm for cloud resource management. A combination of the fuzzy nervous system for workload prediction and ACO for VM is presented in Fig. 2(A). The proposed architecture for optimizing energy consumption in the cloud with the ability to predict neural-fuzzy systems is shown in Fig. 2(B).

The cloud manager allocates existing resources to new customers. The main problem with these types of requests is re-assigning occupied resources by new requests as there are free spaces in different servers and the cloud is not able to provide suitable capacity for provision. The whole space is large because the servers are scattered, but the cloud cannot accept new requests. We consider the method for managing the free granular space that will make it unemployed, and we also consider optimizing the immigration of virtual machines with ACO. Implementing the design process of the method is shown in Figs. 3 and 4. In allocating resources to virtual machines, it is essential to find the location of physical hosts for virtual machines because each of these virtual machines corresponds to a request. In this process, each physical machine has its resources. These resources include processor, memory, and bandwidth. And with empty space, customer requests should be assigned to the physical machine t with virtual machines. Applications and capacity arrangements are shown in Eq. (3). Therefore, the allocation of VMs in PM can be defined as a limit. In this case, the capacity of physical machines can create a limit to the number of virtual machines assigned to them.

$$\begin{cases} VM_i : \{cpu_i, ram_i, bandwidth_i, storage_i\} \\ PM_j : \{CPU_j, RAM_j, BandWidth_j, Storage_j\} \end{cases} \tag{3}$$

3.1. Predict future workload in SDAR-NFAT

We use the workload forecast in each round to estimate the number of required servers. The monitored learning system can only solve classical classification problems. But in the forecast, we use the latest values of parameters to predict the size of their future work in a consecutive learning issue. To solve the learning under consecutive monitoring, we use a concept called the slider window. It converts a sequential learning algorithm to learning under classic supervision. In this technique, the bins are divided into consecutive units of N when we need to predict the resources. As a result, the first samples of N-1 are n input components and the N unit is the corresponding output. A new monitoring learning is similar to learning under classic supervision with N-1 input and output. Table 2 illuminates the definition of input and output collections in our proposed method.

3.2. The immigration of virtual machines is aimed to achieve two goals

Scaling and turning off additional active physical machines, and handling tasks with virtual machines. In the proposed architecture, cloud managers can reduce the number of active PMs based on the prediction. In the migration of virtual machines, several moves can be

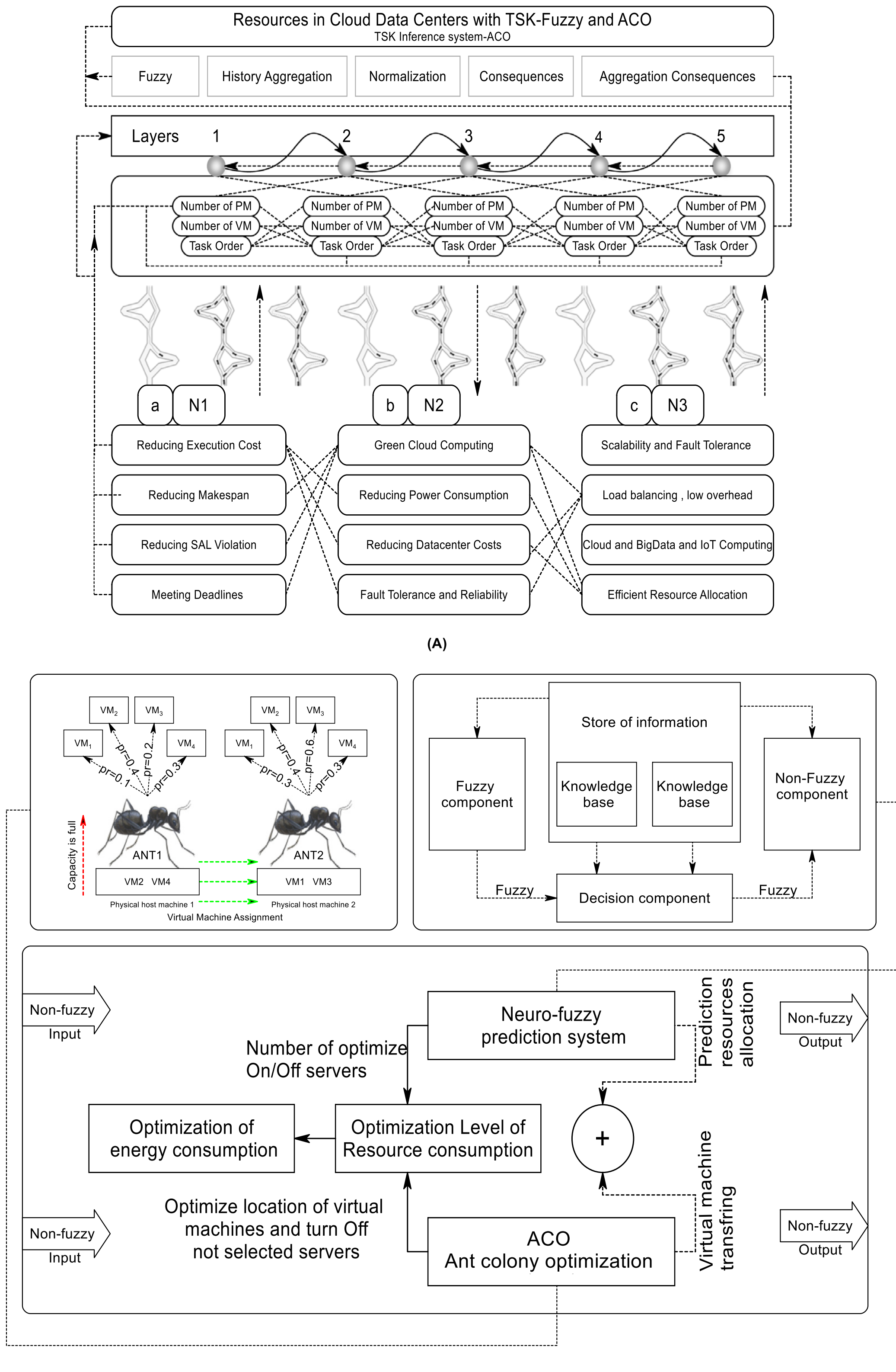


Fig. 1. A general architecture for TSK inference and ant colony algorithm in cDC (A), input and output details for fuzzy inference system (B).

considered in virtual machines to improve performance. Immigration can lead to heavy traffic in the network and reduce performance. For each physical machine, there is a C capacity vector that shows the physical resource capacity of the server. For each virtual machine I, the capacity of the time is shown, and the allocation variable is the case that the allotted bin is used for item *i*. Fig. 5 and Table 3 show the matrix that shows the spatial state of the X variable.

The cost function of the ant colony algorithm in resource allocation is to minimize physical servers that are active as some restrictions (see Eq. (4)). In optimizing the colonies of the ants, the probability of

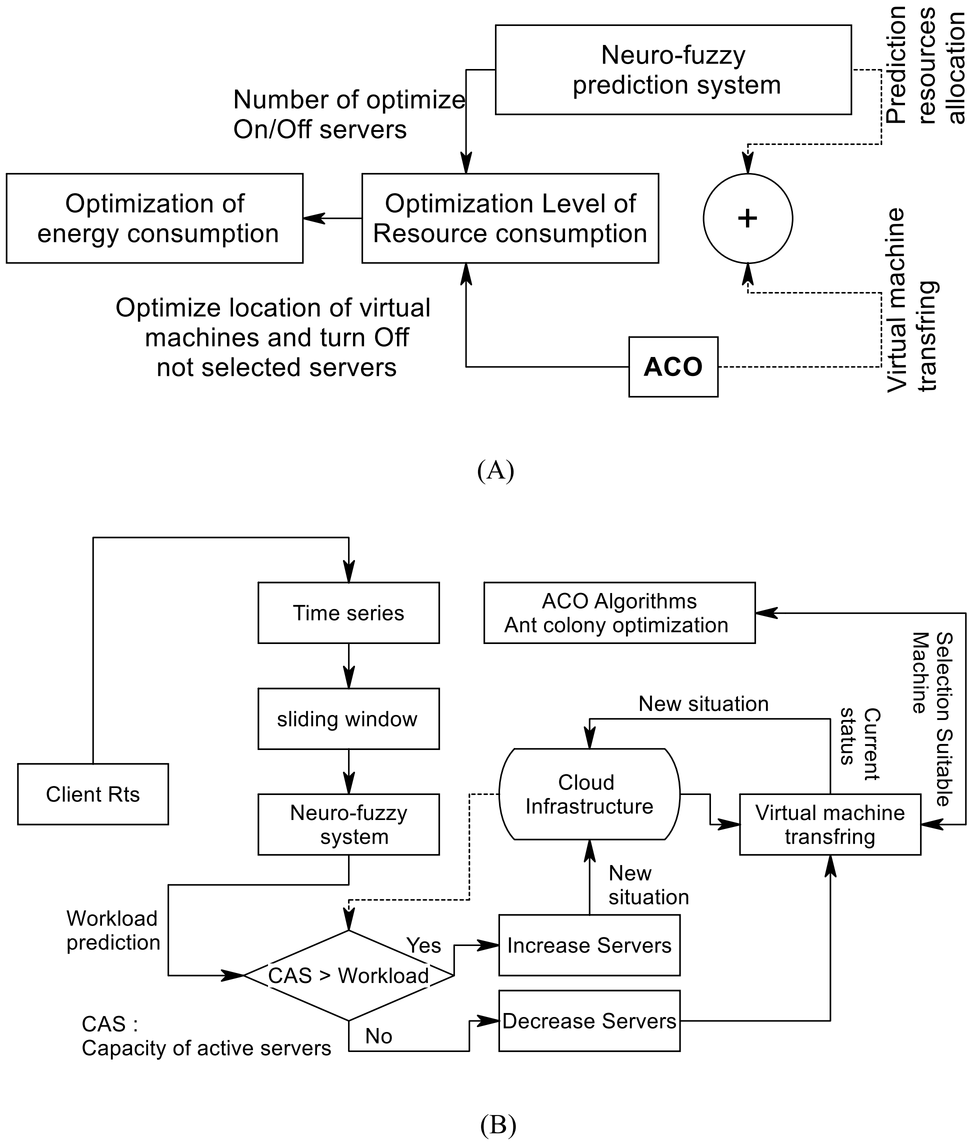


Fig. 2. The architecture of the SDAR-NFAT (A), Combining Neuro-Fuzzy and Ant colony algorithm (B).

Table 2
Inputs and outputs of Neuro-Fuzzy prediction system.

Parameter	Subject
$cDC\{\{cDC(CPU(a)_{t-2}), cDC(CPU(b)_{t-1}), cDC(CPU(t)_{t})\}\}$	Input
$CPU(cDC(m+t)_{t+1})$	Output

Table 3
Input parameters to resolve optimization issues with ACO.

B:{b1,b2,…,bn}	Bins (Physical Machines)
I:{i1,i2,…,im}	Items (Virtual Machine)
$R:\{CPU\ \ power, Ram, Bandwidth, \ldots\}$	Resources

choosing V Bin by the ant, or in other words, the probability of placing the virtual machine I * in the physical machine V is shown in Eq. (5). N_v is a VM (items) that do not fit in any physical machine (bucket) while meeting capacity constraints, N is the number of virtual machines that meet the conditions given in Eq. (6). B_v is the physical host capacity vector (bin) and is calculated by Eq. (7). In each iteration, S_{best} in Eq. (10) shows the best solution that corresponds to the best binary matrix and prevents premature evaporation leading to an unsuitable solution [30].

$$f(y) = \sum_{v=0}^{n-1} y_v \tag{4}$$

$$p_{i,v}(Cloud) = \frac{Cloud[\tau_{i,v}]^{\alpha} \times Cloud[v_{i,v}]^{\beta}}{Cloud \sum_{u \in N_v} [\tau_{u,v}]^{\alpha} \times [v_{u,v}]^{\beta}}, \quad \forall_i Cloud \in N_v \tag{5}$$

$$N_v(Cloud) := i; \begin{cases} Cloud \sum_{j=0}^{n-1} x_{i,j} = 0 \\ Cloud\, \vec{b}_v + \vec{r}_i \le \vec{C}_v \end{cases} \tag{6}$$

$$\vec{b}_v Cloud = Cloud \sum_{i \in B_v} \vec{r}_i \tag{7}$$

$$\tau_{i,v} := \frac{1}{|\overrightarrow{C_c} - (\overrightarrow{b_v} + \overrightarrow{r_i})|} \tag{8}$$

$$\tau_{\mathrm{i,v}} Cloud = (1-\mathrm{p}) \times Cloud\, \tau_{\mathrm{i,v}} + \Delta^{best}_{\tau_{\mathrm{i,v}}}, \quad \forall (i, B_v) \in ICloud \times BCloud \tag{9}$$

$$\Delta^{best}_{\tau_{\mathrm{i,j}}} = \begin{cases} \dfrac{1}{f(S_{best})} & if\ x_{i,v} = 1 \\ 0 & otherwise \end{cases} \tag{10}$$

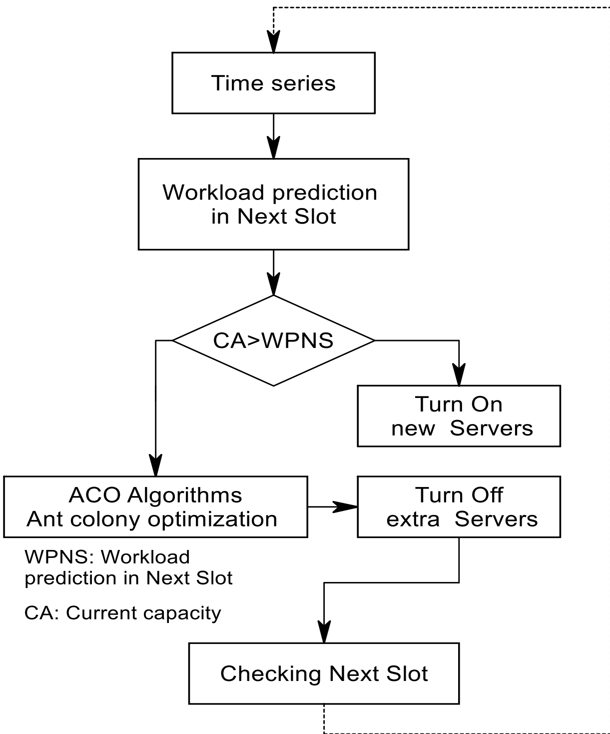


Fig. 3. The program of execution algorithm.

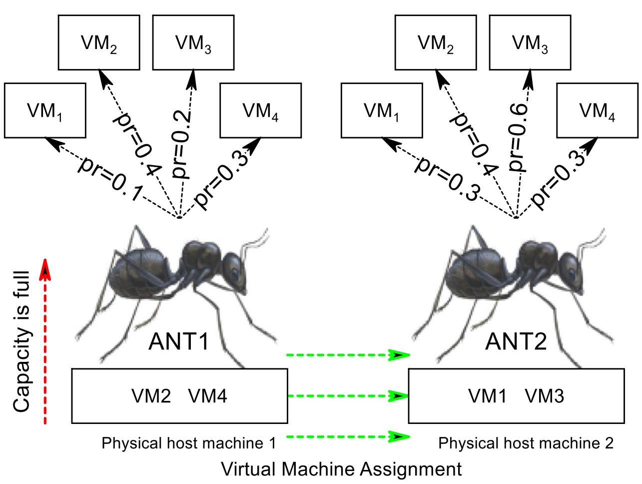


Fig. 4. Resolving the problem of assigning virtual machines with ACO.

	B1	B2	…	Bv	…	Bn
I1	1	0	…	0	…	0
I2	0	0	…	1	…	0
…	…	…	…	…	…	…
Ii	0	1	…	$x_{i,y}$	…	0
…	…	…	…	…	…	…
Im	0	0	…	0	…	0

Fig. 5. Value space of variable $x_{i,y}$.

4. Evaluation criteria

With CloudSim, developers can create seamless simulations by installing many packages "CloudletSchedulerTimeShared.java". Researchers and developers can modify these packages to create their own policies and algorithms. These packages are highly important in Cloud Sim since they include the necessary components for simulating a cloud environment. Cloudlet Schedulers can be introduced in this package to plan how to split CPU resources between Cloudlets based on their own policies. The default way of sharing resources is Space-Shared and Time-Shared. This package consists of mathematical distributions such as Lomax Distribution, to distribute probability. Other distributions include Random Number Generator, Lognormal, Exponential Distribution, Gamma Distribution, etc. This package is essential as it defines the provisioning policy for Virtual Machines running on a host (Fig. 6). Various cloud service providers use it to model their infrastructure. Depending on their hardware configuration, a cluster encapsulates either heterogeneous or homogeneous computing hosts. A data center characteristics model is used to model how data centers configure their resources such as Hosts, user base, cloudlet, broker, VMM allocation, and VM scheduler.

In this section, we present the results of the resource management and optimization of virtual machine migration. We use the window size of 4 in this research. In the following, we first explain the appropriate evaluation criteria for each part of the proposed method to define a specific criterion for comparing the performance of the proposed method. Next, we study the proposed NeuroFuzzy performance evaluation to estimate workload in the cloud. The cloud has many features of demand markets because it is the only technology that connects distributed computing resources with demand markets. One of the main features of the market is the frequent pattern of requests. These patterns provide a structure for product supply and demand that helps business managers make decisions (Fig. 7). According to the proposed architecture, cloud management planning relates to how new customers are allocated resources. We must consider those requests that require a large percentage of the available resources. These types of requests indeed consume most of the available resources, but they cannot be reassigned to new inquiries since the free spaces are hosted on different servers, and it is impossible to present new information in the cloud. This figure shows the implementation structure. It is shown on the left how ACO is applied to physical and virtual machines. The optimal resources are selected using ACO. A fuzzy-based TSK model is represented on the right side that provides a classification for virtual and physical machines [31]. As the solution in this research, ACO and TSK fuzzy combinations are presented in middle (Fig. 7). The simulated data set is based on the repetition of the demand trend every day. It takes two minutes to create and simulate this dataset based on the simulation criteria provided in section two. There are 24 * 30 or 720 run rounds in one day [21]. Another parameter that should be considered is how the demands are distributed in the cloud because it shows the distribution of demands at different times of the day. Although workload forecasting is not time-based, the relationship between consecutive time series values of workload determines the forecasting method (Figs. 8 and 9). By using the proposed method in resource optimization, appropriate allocation details can be obtained. In the proposed method, two approaches are used: the fuzzy method and a metaheuristic approach. Simulations are conducted using datasets. This kind of information is suitable for implementation using the DVFS method [11,19]. In addition, there is a discussion of migrations with heavy network traffic in SDAR-NFAT. Here are some traffic examples to evaluate such as Network traffic analysis involves examining packets passing along a network. Historically, this strategy was intended to investigate the sources of all traffic and volumes of throughput for the sake of capacity analysis. More recently, network traffic analysis has expanded to include deep packet inspection used by firewalls and traffic anomaly analysis used by intrusion detection systems. In our research,

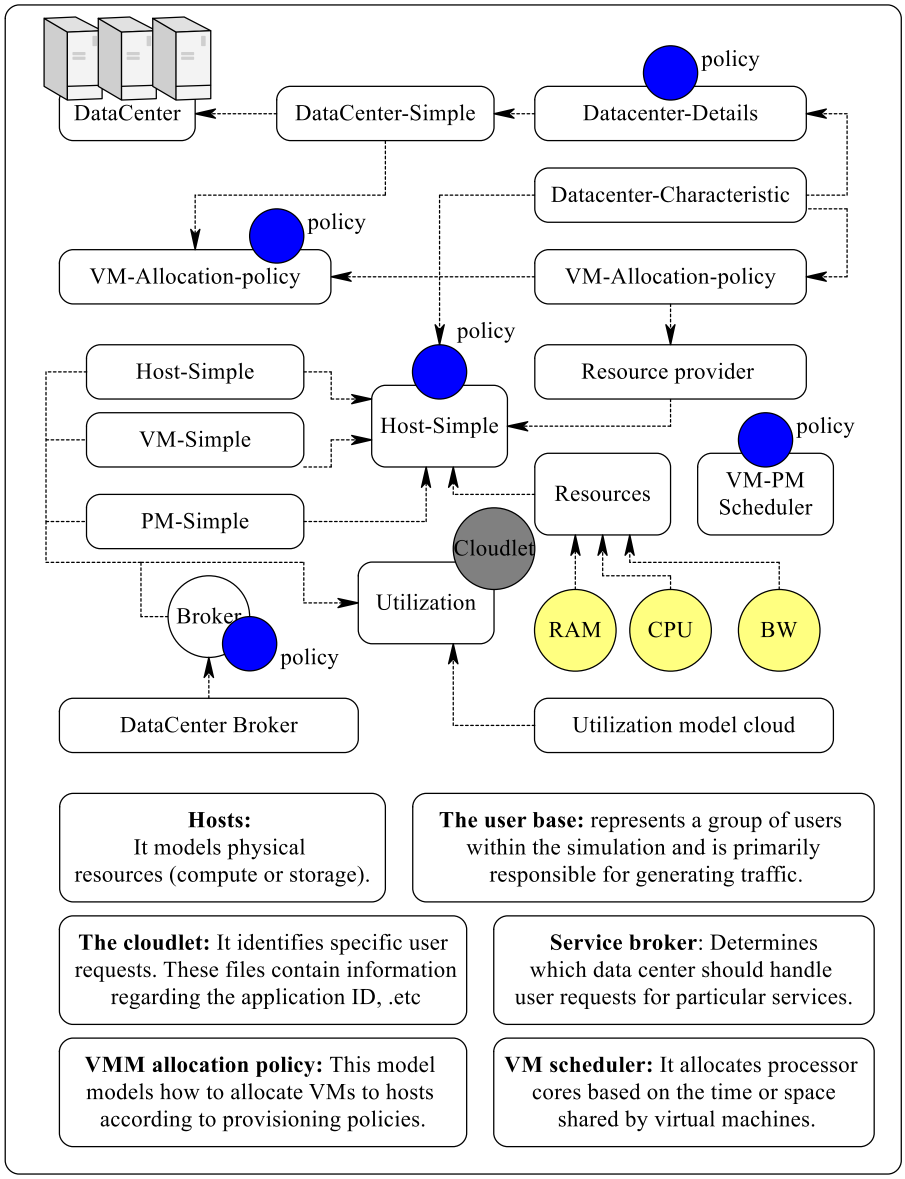


Fig. 6. Details on the installation of CloudSim packages.

we have considered the variety of traffic analysis tools such as leading network traffic analyzer, NetFlow and OpManager.

An important issue is the effect of noise on the data and the fact that the noise on different days should not affect the previous or future days. An acceptable amount of noise (10) is added to the daily distributed performance and independent of other days to prevent bias. This noise must have an average of zero to be acceptable to add to the distribution performance. Input requests for virtual machine allocation per day for the simulated data set and CPU demand are shown in Fig. 10 (sections A and B).

The Normalized Mean Error (nMSE) is a common evaluation criterion that measures the magnitude of the error and does not measure its direction. The calculation of the normal mean error is based on Eq. (11) where T_i is the concrete output and Y_i is the output generated by our model for members. Regression is another criterion that measures the reliability of the value of continuous estimation by correlation coefficient. It is between (−1, 1) and is an indicator for evaluating the linear correlation between the actual value and the estimation of a parameter. Zero regression shows no relationship between these two values. As regression approaches −1 or 1, it indicates a negative or positive linear correlation. Eq. (12) shows the reliability calculation criteria based on linear regression of correlation coefficients [23]. To calculate the dynamic resource allocation QoS, we consider the relationship between the number of banned requests and the total number of requests. In Eq. (13), calculates the ratio of rejected jobs arriving at time t. Finally, resource wastage is calculated based on the free capacity of the servers to the total capacity of the cloud servers (Eq. (14)).

$$\text{nMSE} \qquad nMSE(Cloud) = \frac{\sum_{i=1}^{N} Cloud(Y_i - T_i)^2}{\sum_{i=1}^{N} Cloud(Y_i - \overline{Y})^2}, \tag{11}$$

$$\text{Regression} \qquad R(Cloud) = \frac{In_{Cloud}\,\frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \overline{T}\right)\left(T_i - \overline{Y}\right)}{\sqrt{\frac{1}{N_i} Cloud \sum_{1}^{N}\left(Y_i - \overline{T}\right)^2} \times \sqrt{\frac{1}{N_i} Cloud \sum_{1}^{N}\left(T_i - \overline{Y}\right)^2}} \tag{12}$$

$$\text{The ratio of rejected jobs} \qquad \mathrm{RR}_t = \frac{\text{Rejected}_t}{\text{All jobs}_t} \tag{13}$$

$$\text{Resource wastage} \qquad Waste(Cloud) = \frac{\sum_{i=1}^{N} Cloud(Capacity_i - Load_i)}{\sum_{i=1}^{N} Cloud(Capacity_i)} \tag{14}$$

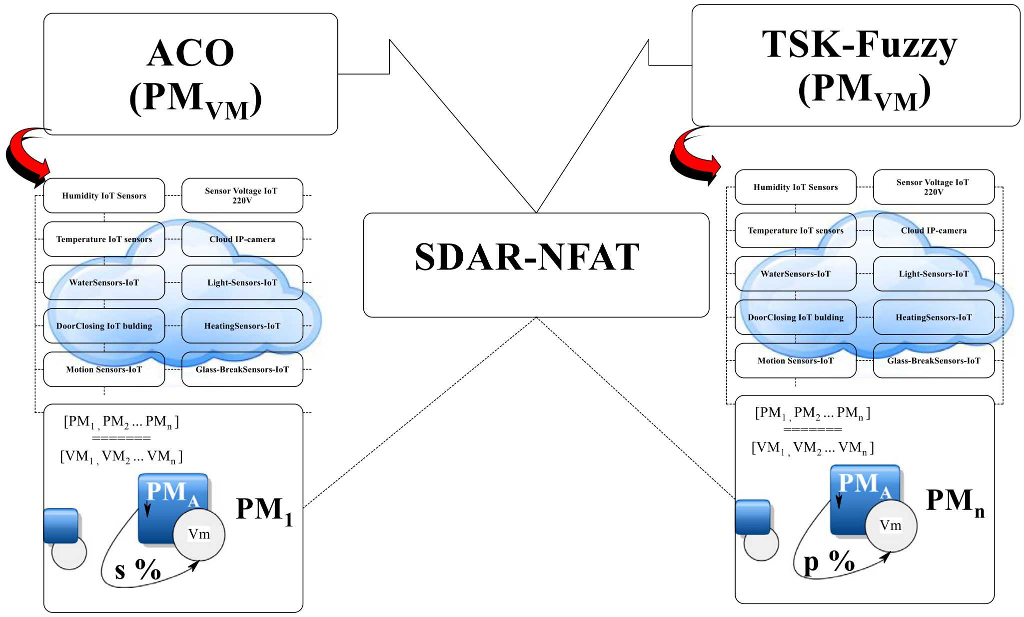


Fig. 7. Implementation structure: On the left side, the way ACO is applied on the physical and virtual machines is shown. ACO is applied to find the best resources. On the right side, the TSK fuzzy-based model is represented which classifies virtual and physical machines. In the center, a combination of ACO and TSK fuzzy is represented as the proposed method in this research.

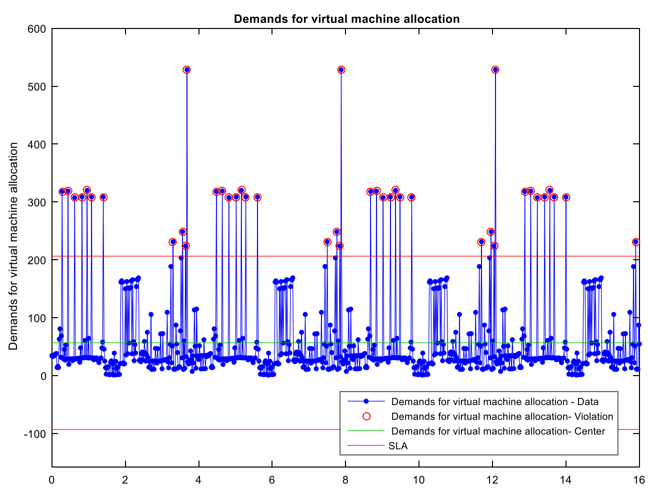


Fig. 8. An example of implementing the demand distribution function to simulate a Dataset.

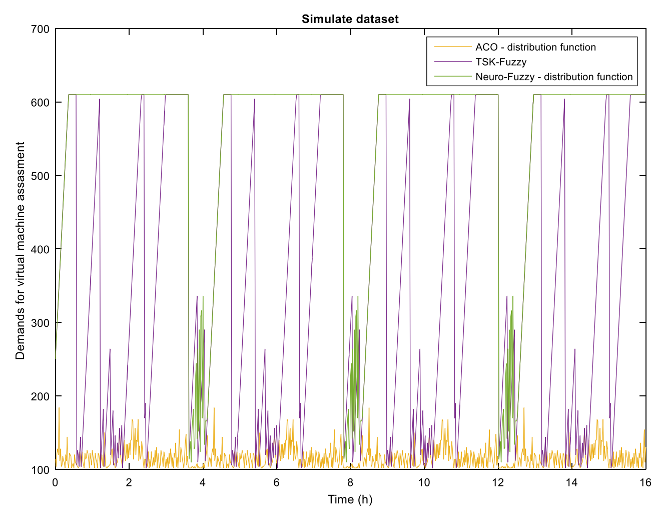


Fig. 9. Demands for virtual machine allocation with different scenarios, fuzzy and ACO.

5. Experimental results

Here we present the results of the presented method. We have compared SDAR-NFAT with the RL-based approach [21], MOPSO Approach [16], NSGAIII-based [20], WO [19], and PSOARMASVR [24] in several evaluation sources. In the performance, we study the fuzzy neuro-system to predict the workload and in the next section, we introduce the evaluation criteria of the proposed model.

5.1. Neurofuzzy system performance to evaluate the system

We calculated nMSE and regression for 15 experiments with and without noise (Table 4). The NeuroFuzzy system works well for noise prediction despite 15 noises and is well-suited as a research tool. We have calculated nMSE and regression for SDAR-NFAT, RL-based [23], MultiObjectivePSO-based [20], NSGAIII-based Approach [22], Whale-Optimization [21] and PSOARMASVR 3-Approach [28].

5.2. The performance of the dynamic resource allocation algorithm

Fig. 11 (sections A, B, and C) shows the quality of the dynamic resource algorithm in CPU workload, number of static virtual machines, and resource wastage. Control-based and static methods are good candidates for comparison. Although highly intelligent methods can model system dynamics, they require long-term learning to achieve proper performance in preventing rejection. Our model runs in real-time, so highly intelligent algorithms are not suitable for comparison.

It is important to note that the real-time cloud market usually uses static methods to prevent rejection in almost all static methods, resource consumption is sacrificed to prevent rejection. Due to the

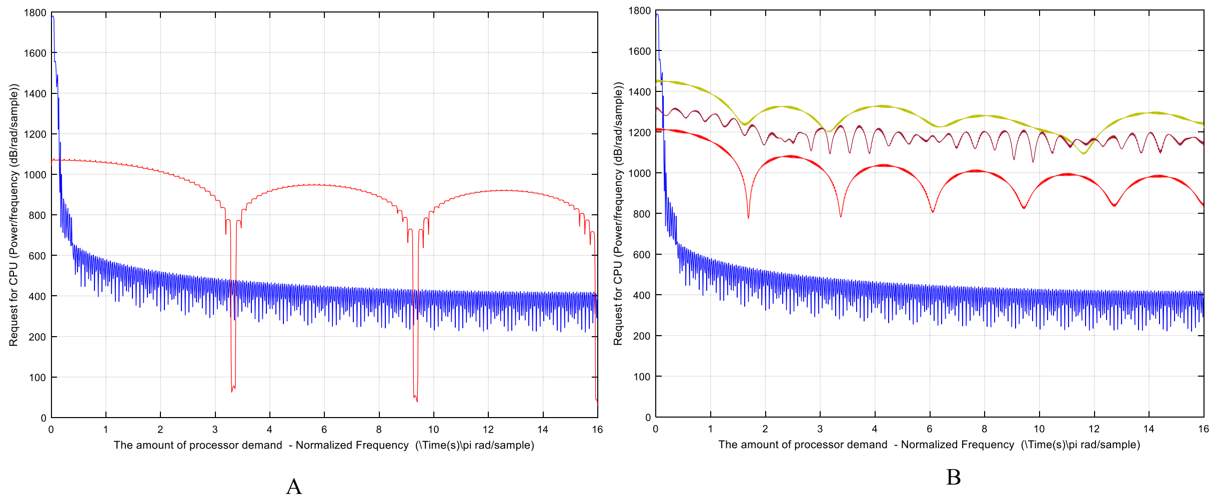


Fig. 10. The load applied to the CPU increases again, blue is performing tasks, red is loading in VMs, and green is loading in PMs.

Table 4
Inputs and outputs of Neuro-Fuzzy prediction system and comparing with other methods.

Methods	Best performance		Average performance	
Neuro-Fuzzy algorithm	R	nMSE	R	nMSE
Without noise	0.96123	0.00843	0.96013	0.00899
5% precent of noise	0.95478	0.00763	0.95205	0.00842
10% precent of noise	0.93145	0.01591	0.92442	0.01891
15% precent of noise	0.92073	0.01943	0.91440	0.01912
SDAR-NFAT	R	nMSE	R	nMSE
Without noise	0.97845	0.00898	0.96011	0.00863
5% precent of noise	0.95294	0.00775	0.95223	0.00456
10% precent of noise	0.93122	0.01555	0.92332	0.01468
15% precent of noise	0.92113	0.01942	0.91134	0.01984
RL-basedApproach [23]	R	nMSE	R	nMSE
Without noise	0.96113	0.00543	0.96243	0.00834
5% precent of noise	0.95488	0.00463	0.95265	0.00845
10% precent of noise	0.93148	0.01593	0.92482	0.01867
15% precent of noise	0.92057	0.01444	0.91493	0.01978
MultiObjectivePSO-basedApproach [20]	R	nMSE	R	nMSE
Without noise	0.93112	0.00839	0.94012	0.00800
5% precent of noise	0.93167	0.01461	0.94204	0.00855
10% precent of noise	0.91942	0.01797	0.93445	0.01791
15% precent of noise	0.91456	0.01945	0.91465	0.01823
NSGAIII-basedApproach [22]	R	nMSE	R	nMSE
Without noise	0.91203	0.00753	0.96013	0.02743
5% precent of noise	0.91234	0. 00843	0.85253	0.03890
10% precent of noise	0.91002	0. 01341	0.86466	0.04334
15% precent of noise	0.90273	0.01912	0.86446	0.04931
WhaleOptimization-basedApproach [21]	R	nMSE	R	nMSE
Without noise	0.91126	0.00341	0.84023	0.00891
5% precent of noise	0.91441	0.00743	0.98235	0.00142
10% precent of noise	0.93143	0.01734	0.97462	0.01391
15% precent of noise	0.93054	0.01943	0.97440	0.01911
PSOARMASVR-basedApproach [28]	R	nMSE	R	nMSE
Without noise	0.92127	0.00343	0.96017	0.01294
5% precent of noise	0.92448	0.00743	0.94288	0. 02242
10% precent of noise	0.93189	0.03541	0.95442	0. 03833
15% precent of noise	0.93095	0.03963	0.81445	0. 03912

complex mechanism of these methods, their implementation is impossible, therefore we consider static methods as unpredictable methods. Fig. 12 (sections A and B) shows the comparison function for wasting resources and request rejection. The results suggest that our models request resource wastage. It is also important to understand how our algorithm works on a large number of requests. Thus, the QoS method is presented in different application scales including small (thousand requests), medium (5000 requests), and large (10 thousand requests) (Fig. 13).

The average delay is examined in this section as it can be seen that the different methods described in the research background are compared. Delays are measured in milliseconds (Fig. 14). Virtual machines and physical machines are taken into consideration in this simulation. In the case of one physical machine and five physical virtual machines, this behavior is observed in all delay methods. It is convenient because the processing capabilities in different modes are similar. As the number of virtual machines increases in the proposed method, there is less delay than in other methods. The MultiObjectivePSO-based algorithm has similar latency as the PSOARMASVR method, the Wall algorithm, and the NSGAIII-based Approach algorithm due to the initial population. In all cases, the RL method has a higher latency. As we increased the number of physical machines as well as virtual machines, there was a delay for different modes. The trend is observed for 20 virtual machines and 4 physical machines, which have different gaps depending on the method. By increasing the processing load on the data center, all methods see an increase in performance. In comparison to other methods, the PSO method and the real method have a longer delay. The rate of improvement of the proposed method SDAR-NFAT compared with RL-based, MultiObjective PSO-based, NSGAIII-based, Whale Optimization and PSOARMASVR, respectively, 16.23%, 14,01%, 11,39%, 9.17%, 11.46%.

6. Conclusion

This study represents an approach to improving online resource allocation mechanisms (SDAR-NFAT). We increase the efficiency of these algorithms by adapting the performance of intelligent resource allocation methods to the number of requests. In this work, predictions are made with sliding windows and fuzzy neural systems to prepare computational resources so that real-time requests are not met with resources. The process of the proposed method is improved by an optimization process called the Ant Colony algorithm to improve energy consumption. To optimize performance and dynamic allocation algorithms in CPU workload, a number of virtual machines, and resource wastage at different scales of demand, we distribute virtual machines

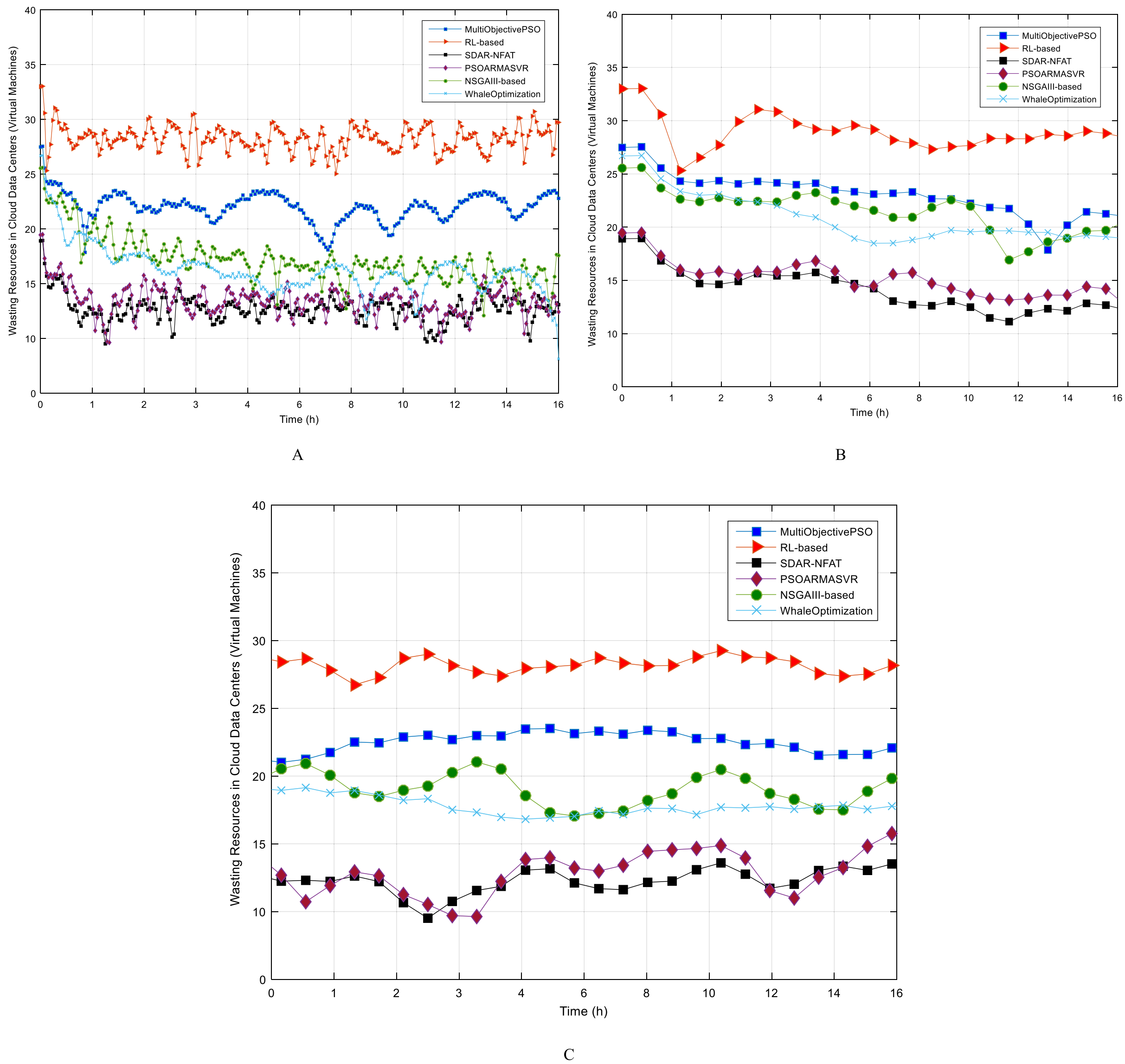


Fig. 11. (A) Dynamic allocation algorithm in CPU workload with the number of virtual machines and resource loss. (B) Comparing the number of virtual machine performances and wasted resources, (C) Dynamic allocation algorithm in CPU with real-time with several virtual machines and resource loss.

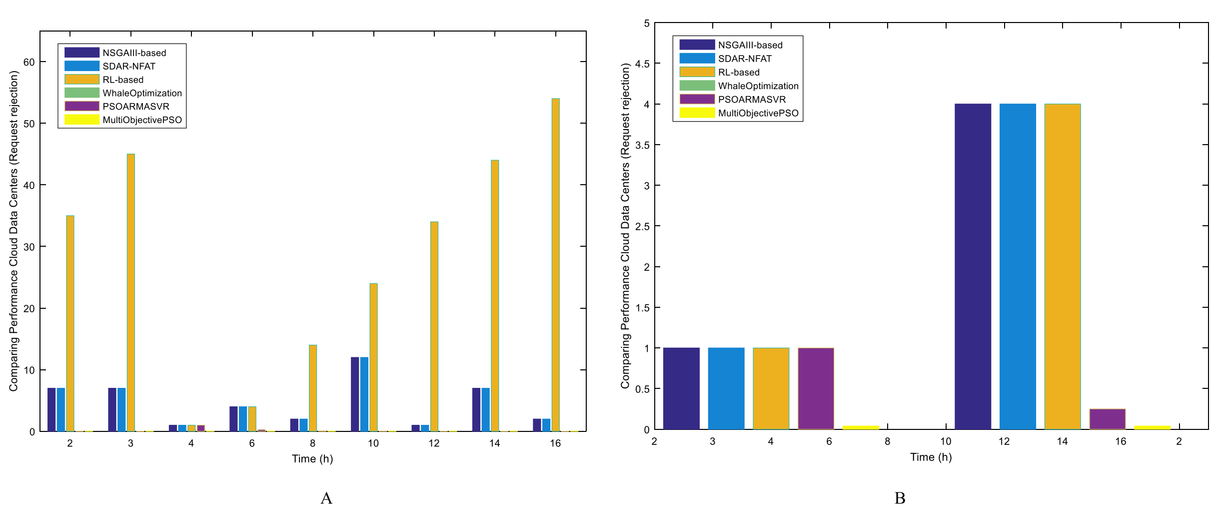
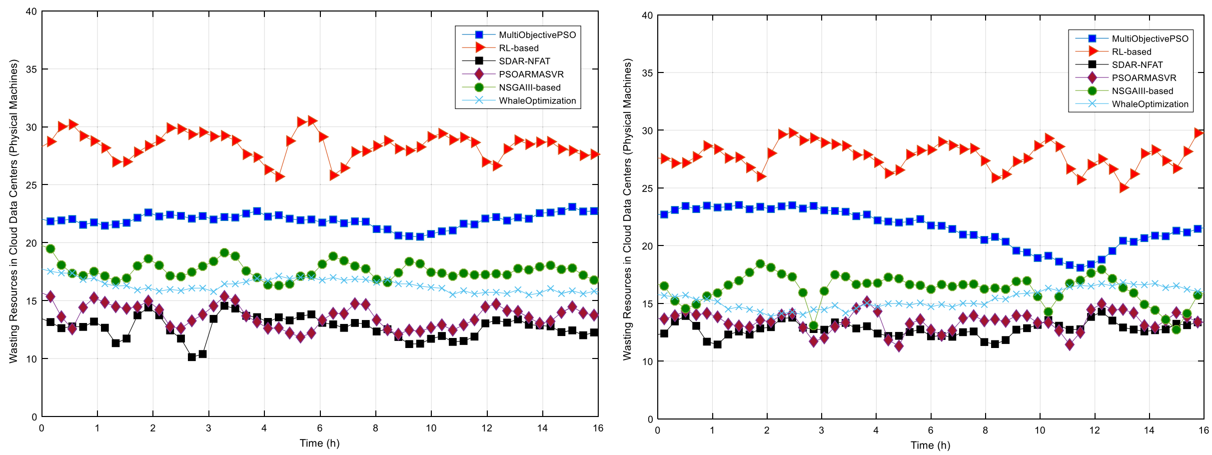


Fig. 12. Comparing performance in request rejection, A: Dynamic allocation algorithm in CPU with performance in request rejection a = 0.2, A: Dynamic allocation algorithm in CPU with performance in request rejection a = 0.4.



A B

Fig. 13. Performance of the algorithms in different request scales.

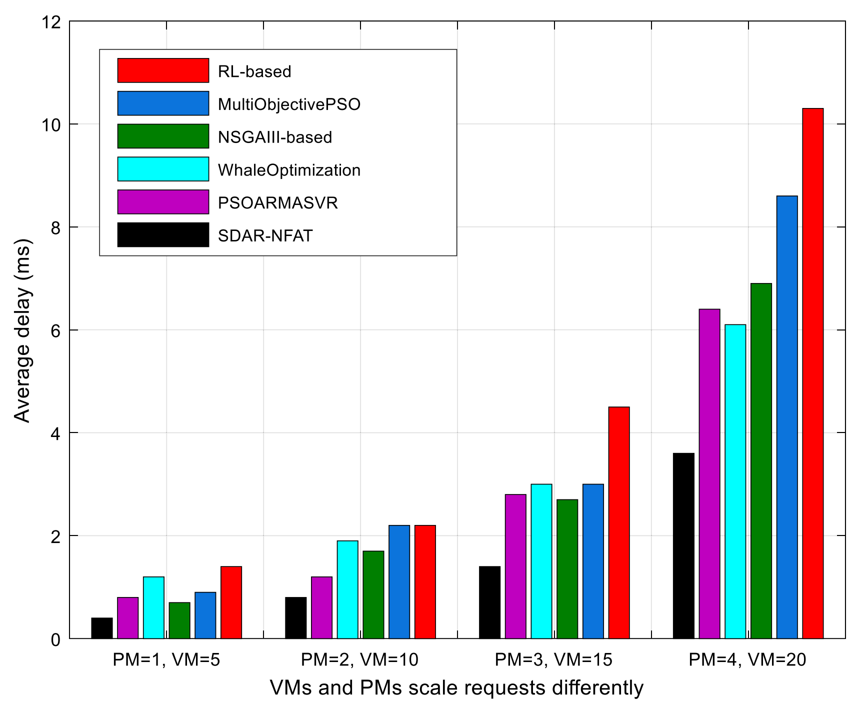


Fig. 14. The details of average delay in different request scales with different VM and PMs.

in a way that utilizes the full computing power of running servers. Finally, the performance of the proposed method and approaches in the literature are compared based on the reduction in the number of rejections and waste of resources. Experimental results show that energy consumption is reduced. We suggest the following ideas to improve the results of this study for future work. First, workload balance parameters can be considered for virtual machine migration based on load balancing [32], then hybrid methods in meta-exploration methods are used instead of Ant Colony, such as Ant-PSO, and Ant-Genetic. Using reverse control-based controllers to improve synchronization periods is also suggested. Virtual machine migration and forecasting and ultimately improve the proposed architecture by adding a failure analysis section to prevent a system crash when burst requests. Machine learning methods and prediction-based methods such as Markov can also be used to optimize resources.

CRediT authorship contribution statement

Arun Kumar Sangaiah: Collected the data, Contributed data or analysis tools, Performed the analysis, Wrote the paper. **Amir Javadpour:** Collected the data, Contributed data or analysis tools, Wrote the paper. **Pedro Pinto:** Collected the data, Wrote the paper. **Samira Rezaei:** Wrote the paper. **Weizhe Zhang:** Collected the data, Performed the analysis.

Declaration of competing interest

Authors declare that there is no conflict of interest.

Data availability

Data available on request from the authors The data that support the findings of this study are available from the corresponding author, upon reasonable request.

References

[1] A. Javadpour, Providing a way to create balance between reliability and delays in SDN networks by using the appropriate placement of controllers, Wirel. Pers. Commun. (2019).

[2] A. Javadpour, G. Wang, cTMvSDN: improving resource management using combination of Markov-process and TDMA in software-defined networking, J. Supercomput. (2021).

[3] A. Javadpour, Improving resources management in network virtualization by utilizing a software-based network, Wirel. Pers. Commun. 106 (2) (2019) 505–519.

[4] A. Javadpour, S. Rezaei, A.K. Sangaiah, A. Slowik, S. Mahmoodi Khaniabadi, Enhancement in quality of routing service using metaheuristic PSO algorithm in VANET networks, Soft Comput. (2021).

[5] A.K. Sangaiah, A. Javadpour, P. Pinto, F. Ja'fari, W. Zhang, Improving quality of service in 5G resilient communication with THE cellular structure of smartphones, ACM Trans. Sens. Netw. (2022).

[6] A. Javadpour, G. Wang, S. Rezaei, Resource management in a peer to peer cloud network for IoT, Wirel. Pers. Commun. (2020).

[7] S. Negi, M.M.S. Rauthan, K.S. Vaisla, N. Panwar, CMODLB: an efficient load balancing approach in cloud computing environment, J. Supercomput. (2021).

[8] A. Javadpour, S. Kazemi Abharian, G. Wang, Feature selection and intrusion detection in cloud environment based on machine learning algorithms, in: 2017 IEEE Int. Symp. Parallel Distrib. Process. with Appl. 2017 IEEE Int. Conf. Ubiquitous Comput. Commun., 2017, pp. 1417–1421.

[9] A. Javadpour, G. Wang, S. Rezaei, K.-C. Li, Detecting straggler MapReduce tasks in big data processing infrastructure by neural network, J. Supercomput. (2020).

[10] H. Zavieh, A. Javadpour, Y. Li, F. Ja'fari, S.H. Nasseri, A.S. Rostami, Task processing optimization using cuckoo particle swarm (CPS) algorithm in cloud computing infrastructure, Clust. Comput. 26 (1) (2023) 745–769.

[11] A. Javadpour, G. Wang, S. Rezaei, S. Chend, Power curtailment in cloud environment utilising load balancing machine allocation, in: 2018 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), 2018, pp. 1364–1370.

[12] A.K. Sangaiah, et al., Energy-aware geographic routing for real time workforce monitoring in industrial informatics, IEEE Internet Things J. (2021).

[13] A. Javadpour, G. Wang, X. Xing, Managing heterogeneous substrate resources by mapping and visualization based on software-defined network, in: 2018 IEEE Intl Conf on Parallel Distributed Processing with Applications, Ubiquitous Computing Communications, Big Data Cloud Computing, Social Computing Networking, Sustainable Computing Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom), 2018, pp. 316–321.

[14] A. Javadpour, A.K. Sangaiah, P. Pinto, F. Ja'fari, W. Zhang, A.M.H. Abadi, H. Ahmadi, An energy-optimized embedded load balancing using DVFS computing in cloud data centers, Comput. Commun. 197 (2023) 255–266.

[15] E.S. Alkayal, N.R. Jennings, M.F. Abulkhair, Survey of task scheduling in cloud computing based on particle swarm optimization, in: 2017 International Conference on Electrical and Computing Technologies and Applications, ICECTA, 2017, pp. 1–6.

[16] A. Javadpour, N. Adelpour, G. Wang, T. Peng, Combing fuzzy clustering and PSO algorithms to optimize energy consumption in WSN networks, in: 2018 IEEE SmartWorld, Ubiquitous Intell. Comput. Adv. Trust. Comput. Scalable Comput. Commun. Cloud Big Data Comput. Internet People Smart City Innov., 2018, pp. 1371–1377.

[17] D.P. Mahato, R.S. Singh, Maximizing availability for task scheduling in on-demand computing–based transaction processing system using ant colony optimization, Concurr. Comput. Pract. Exp. 30 (11) (2018) e4405.

[18] A. Thakkar, K. Chaudhari, A comprehensive survey on portfolio optimization, stock price and trend prediction using particle swarm optimization, Arch. Comput. Methods Eng. (2020).

[19] S.M. Mirmohseni, C. Tang, A. Javadpour, Using Markov learning utilization model for resource allocation in cloud of thing network, Wirel. Pers. Commun. (2020).

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

[21] S. Goyal, et al., An optimized framework for energy-resource allocation in a cloud environment based on the whale optimization algorithm, Sensors 21 (5) (2021) 1583.

[22] E. Parvizi, M.H. Rezvani, Utilization-aware energy-efficient virtual machine placement in cloud networks using NSGA-III meta-heuristic approach, Cluster Comput. (2020) 1–23.

[23] F. Bahrpeyma, H. Haghighi, A. Zakerolhosseini, An adaptive RL based approach for dynamic resource provisioning in cloud virtualized data centers, Computing 97 (12) (2015) 1209–1234.

[24] A. Ragmani, A. Elomri, N. Abghour, K. Moussaid, M. Rida, An improved hybrid fuzzy-ant colony algorithm applied to load balancing in cloud computing environment, Procedia Comput. Sci. 151 (2019) 519–526.

[25] I. Hussain, et al., Optimizing energy consumption in the home energy management system via a bio-inspired dragonfly algorithm and the genetic algorithm, Electronics 9 (3) (2020) 406.

[26] P. Cong, J. Zhou, L. Li, K. Cao, T. Wei, K. Li, A survey of hierarchical energy optimization for mobile edge computing: A perspective from end devices to the cloud, ACM Comput. Surv. 53 (2) (2020) 1–44.

[27] S.-Y. Hsieh, C.-S. Liu, R. Buyya, A.Y. Zomaya, Utilization-prediction-aware virtual machine consolidation approach for energy-efficient cloud data centers, J. Parallel Distrib. Comput. 139 (2020) 99–109.

[28] W. Zhu, H. Ma, G. Cai, J. Chen, X. Wang, A. Li, Research on PSO-ARMA-SVR short-term electricity consumption forecast based on the particle swarm algorithm, Wirel. Commun. Mob. Comput. 2021 (2021).

[29] A. Javadpour, F. Ja'fari, P. Pinto, W. Zhang, Mapping and embedding infrastructure resource management in software defined networks, Clust. Comput. 26 (1) (2023) 461–475.

[30] V. Maniezzo, L.M. Gambardella, F. de Luigi, Ant colony optimization, in: New Optimization Techniques in Engineering, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 101–121.

[31] A.-Y. Son, E.-N. Huh, Multi-objective service placement scheme based on fuzzy-AHP system for distributed cloud computing, Appl. Sci. 9 (17) (2019).

[32] A. Javadpour, A.M.H. Abadi, S. Rezaei, M. Zomorodian, A.S. Rostami, Improving load balancing for data-duplication in big data cloud computing networks, Cluster Comput. (2021).