ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal
Regular Issue, Vol. 12 N. 1 (2023), e31485
eISSN: 2255-2863
DOI: https://doi.org/10.14201/adcaij.31485

Host Detection and Classification using Support Vector Regression in Cloud Environment

Vidya Srivastava and Rakesh Kumar

Computer Science Department, MMMUT, Gorakhpur.
522vidya93av@gmail.com, rkiitr@gmail.com

KEYWORDS

cloud computing; support vector regression; energy consumption; execution time; resource utilization

ABSTRACT

Having the potential to provide global users with pay-per-use utility-oriented IT services across the Internet, cloud computing has become increasingly popular. These services are provided via the establishment of data centers (DCs) across the world. These data centers are growing increasingly with the growing demand for cloud, leading to massive energy consumption with energy requirement soaring by 63% and inefficient resource utilization. This paper contributes by utilizing a dynamic time series-based prediction support vector regression (SVR) model. This prediction model defines upper and lower limits, based on which the host is classified into four categories: overload, under pressure, normal, and underload. A series of migration strategies have been considered in the case of load imbalance. The proposed mechanism improves the load distribution and minimizes energy consumption and execution time by balancing the host in the data center. Also, it optimizes the execution cost and resource utilization. In the proposed framework, the energy consumption is 0.641kWh, and the execution time is 165.39sec. Experimental results show that the proposed approach outperforms other existing approaches.

1. Introduction

Cloud computing has become a well-known computing paradigm for delivering services to different organizations. The main benefits of the computing paradigm, including on-demand services, pay-as-per-use policy, and rapid elasticity, make cloud computing a more emerging technology to

lead with new methods (Arunarani *et al.,* 2019). Cloud computing has emerged as a popular area of study since it allows to efficiently expand numerous network services. It can be divided into three subcategories: public cloud, private cloud, and hybrid cloud. Demand services are organized by outside parties and distributed across numerous organizations using the public network in the public cloud. Without network capacity limitations, all data are handled within the organization using the private cloud (Adhikari & Koley, 2018). The hybrid cloud combines public and private clouds, unifying both computing environments (Zolfaghari *et al.,* 2021).

Services offered by cloud computing are SaaS, PaaS, and IaaS. In SaaS, users can access all programs instead of installing them. In PaaS, the user can run and manage the module without the complexity of maintaining the infrastructure. In IaaS, customers can use resources such as storage and processing devices. The scheduling of multimedia data sets on the World Wide Web (WWW) and the production of circuit boards are only a few examples of the many applications for which scheduling is employed today. The main application is to assign tasks to the appropriate processor (Garg *et al.,* 2022). Unfortunately, no polynomial time algorithms are available to optimize the proper allocation of computing resources. The scheduling problem is either NP-hard or NP-complete. In explaining the dilemma that we encounter, the difficulties that are 0.02% of candidate solutions lie between the make-span and optimum solution and 1.01 times the make-span of an optimum solution. These data show how difficult it is to find an optimum scheduling algorithm (Hosseinzadeh *et al.,* 2020). Data sizes in 2012 reached a record high, going from terabytes to a staggering number of petabytes for a single data set. High volume and high velocity are the main features of big data that require a new form of processing to enable process optimization and decision-making. High operating expenses and increased carbon emissions result from this energy demand. The result is that the data centers are no longer viable. The primary energy consumer in a data center is the information technology (IT) infrastructure which also comprises servers and other IT components. In recent years, the primary area of research has focused on reducing the amount of energy used by cloud infrastructure (Khan & Santhosh, 2022). Telecommunication, internet search, pharmaceuticals, etc. are real-world areas that generate large amounts of data. Alibaba set a record after 0:00 on November 11, 2013, when the overall transaction amount exceeded 35 billion (USD 5.71 billion), and the transaction amount reached about 100 million yuan (USD 16.3 million). One hundred eighty-eight million tx had been completed (Khaleel & Zhu, 2016). In the center of Aliyun, to process 40,000 tasks, they run on a cluster composed of 3000 nodes; 1.5 petabytes of tx records every day, result in 20 terabytes being generated.

The heterogeneity of computing resources, dynamic behavior, and complex reasoning make task scheduling a critical technology. The assumption that using cloud computing resources can be wise and effective implies that adequate allocation takes place when numerous jobs are requested for resources. Task scheduling is the process of task-resource mapping. By taking into account the attribute and status of cloud resources, the cloud system transfers these tasks to the related resources to run according to the attributes and necessities of these tasks (Magotra *et al.,* 2022). Contrary to this, if scheduling is not done properly, several parameters are affected, namely, resource consumption is increased, task execution time is extended, the whole system's throughput is reduced, and resource life is also reduced. Therefore, effective task scheduling is key to the successful completion of any computing system. In the case of distributed computing, such as cloud and grid computing, workload can be done by using several scheduling algorithms (program-based, heuristics, metaheuristics, greedy approach).

Load balancing is one of the essential factors in maintaining scheduling on a system. First, the nature of the load is checked, which can be either static or dynamic. Various mechanisms are proposed for static workloads, but nowadays, dynamic workload handling is an essential issue (Malik *et al.,* 2022). A popular strategy for managing a changing workload and achieving other goals is virtual machine (VM) consolidation. In this process, a running VM migrates from one physical machine to another without affecting the execution of an application. The decision to migrate depends on the detection of overload and underload host. Detecting these host systems demands a mechanism to predict the future use of resources to execute VM consolidation. Support vector regression (SVR) is a dynamic time series prediction model that accepts non-linear data, allows for the acceptance of error, and returns a productive prediction to achieve the necessary accuracy. Once the load is classified, a random selection policy is adopted for VM migration to balance the load among the hosts.

1.1 Motivation and Contributions

As the demand for scalable and convenient computing resources expands rapidly, cloud services are being widely adopted in many fields, such as research applications and business management. Cloud service providers have developed flexible features such as adaptive scale-out/scale-in and automatic deployment (Bal *et al.,* 2022). However, user- and developer-friendly makes it more challenging for service providers to maintain resources and efficiently provide high-level services. Utilizing resources to reduce energy consumption and execution time of the entire system becomes a challenging task for the cloud service provider. Early in 2011, the consumption of data accounted for 1.1-1.5% of total global energy consumption. The total requirement of power increased by 63% in a year (Pirozmand *et al.,* 2021). The continuous consumption of electricity makes cloud services unsustainable. Therefore, proper energy efficiency in the cloud becomes a great necessity. Workflow scheduling must consider useful dependency constraints and resource properties, aiming to enhance resource utilization (use of available resources) while reducing the makespan of application since scheduling is an NP-hard problem and must be optimized using approx. The solution is in a polynomial time. Apart from the makespan, energy consumption is another challenging parameter in the cloud environment.

Another knowledge gap is associated with previous research on task scheduling and virtual machine placement in the Cloud, as the operation is managed at two levels; the first one is task scheduling and the second one is VM placement. Scheduling and placement are coupled with each other. Researchers often address the two levels separately, with the first level working on task scheduling only for the profit of cloud users, and the second level, VM placement, for the profit of cloud providers. Thus, an optimal plan is crucial in enhancing resource utilization in the cloud environment (Mishra *et al.,* 2020).

The user aims to find suitable physical machines (PMs) for the available tasks, and the provider aims to use their infrastructure by accommodating tasks for the user demand. Quick expansion of computing could lead to an increase in the number of data centres, which would enhance resource utilization. Greenpeace estimation 2015 states that data centers increased from 12% in 2012 to 15% in 2017. From the previous study, it has been found that the infrastructure of the data centers can cause 70% of heat generation. In the paper of Kumar *et al.* (2019), Amazon EC2 encounters a variation of as high as approx. 30% -65% during unbalanced data transmission between the cloud host. Higher energy use results in more carbon dioxide (CO2) radiation, which harms the cloud environment. The energy is released into the atmosphere, along with "greenhouse gases", such as CO2, which cause global warming. To deliver services to Cloud consumers online, cloud companies require one or more data centers.

Most research communities are working to reduce their use of data centers. Resources for a task are shared among all the virtual computers on a cloud platform (Panwar *et al.,* 2022). Consequently, a single virtual machine cannot estimate the host's energy use. A migration strategy must be chosen that helps to balance the load on all the VMs.

The main contributions of this paper are defined as follows:

- The proposed cloud model is discussed considering several objectives, namely, energy consumption, execution time, resource utilization, and execution cost.
- Support vector regression prediction is used to achieve the upper and lower boundaries.
- On the basis of the upper and lower limit, the host is classified into four categories, namely: overload, under-pressure normal, and underload. In the case of load imbalance, a series of migration strategies have been highlighted.
- Experimental results show that ROA/SVR/RS improves scheduling and minimizes energy consumption and execution time by balancing the host in the data center. Also, it optimizes execution cost and resource utilization. Thus, the proposed schemes reveal an efficient solution for the cloud data center.

The rest of the paper is organized as follows: Related works are discussed in Section 2. Section 3 describes the proposed cloud model with the SVR method for boundary definition. A number of migration policies are outlined in the later part of this section. Section 4 analyses the simulation and the results. Finally, Section 5 concludes the conducted research and discusses future work.

2. Related Works

Sardaraz & Tahir (2019) provided a hybrid scheduling technique for scientific workflows in cloud environments. The authors created a list of tasks for the PSO algorithm in the first phase to reduce execution time bottlenecks and ensure that the highest priority activities are carried out. Tasks were scheduled in the second phase of the PSO algorithm to minimize time and expense while also keeping an eye on the load-balancing parameter to use cloud resources effectively. Scientific workflow was used to evaluate the proposed algorithm. The experiment's findings show that the proposed approach performs better than the standard methodology.

Marahatta *et al.* (2019) proposed a virtualized cloud data center (CDC); an energy-efficient dynamic scheduling method (EDS) using previously completed scheduling tasks and virtual machine classification. Similar tasks were combined to achieve optimal resource utilization, energy efficiency, and host operating frequency to conserve energy and create and delete virtual machines as needed. EDS algorithms enhanced the overall performance of scheduling by maximizing the jobs guarantee ratio, improving mean response time, and reducing energy consumption.

In the study of Abualigah & Alkhrabsheh (2022), the multi-verse optimizer genetic algorithm (MVO-GA) has been proposed for scheduling the task transfer queue by the available virtual machines. Depending on efficiency traits such as bandwidth speed and capacity, MVO works well to balance significant activities. Additionally, GA increases resource workload to reduce the time the planned tasks take to transfer. Crossover and mutation are optimizers between the jobs and available virtual machines. According to simulation results, the suggested approach assigns tasks more quickly

than utilizing MVO-GA separately. The efficiency of algorithms is demonstrated by the fact that MVO-GA successfully reduces task transfer times (1000–2000) by 15% compared to other existing algorithms.

Stavrinides & Karatza (2019) proposed a method for several multicore processors whose motive was to boost energy efficiency by trading off results, providing timeliness, and maintaining the quality of the job and cost demanded to complete the job at a reasonable level. The proposed algorithm provided promising results compared to the other two policies under various QoS requirements.

The study by Wang *et al. (*2017) introduced replication-based scheduling for maximising system reliability (RMSR), which incorporated task communication into system reliability. The suggested method finds the best path for the given tasks. During the task's replication phase, the user chooses the threshold value. An analysis of the performance of the RMSR method in comparison to other current algorithms shows that it outperforms both the randomly generated graph and the application graph.

Gupta *et al.* (2022) presented a modified version of the HEFT algorithm. The algorithm works in two phases: Rank generation and processor selection. In the first phase, different methodologies were used for rank calculation. In contrast, in the second phase, the modified mechanism was used for scheduling to reduce the makespan of workflows running on other virtual machines. Experimental analysis shows that the proposed HEFT algorithm performs better than the basic HEFT.

The study done by Abdullahi & Ngadi (2016) presented a new metaheuristic algorithm called discrete symbiotic organism search (DSOS) for solving numerical optimization problems and is suitable for large-scale problems DSOS mimics the symbiotic relationship exhibited by organisms in an ecosystem. Analysis done using a t-test reveals that DSOS performs significantly better than PSO individually in a large search space.

In Li *et al.* (2020), the shortest path algorithm that had been proposed aimed to reduce both execution time and energy usage. According to the sequence in which the tasks are executed, the directed acyclic graph of the tasks can be transformed into a hypergraph. The k-path hypergraph partition was then carried out to balance the hypergraph. The Dijkstra method was applied to each hypergraph partition to discover the best work scheduling algorithm. According to experimental results, the suggested workflow scheduling system successfully used cloud resources while minimizing energy usage.

Asghari *et al. (*2020) proposed a reinforced approach that was exploited in the multi-agent system for resource provisioning and task scheduling, aiming to minimize the makespan, optimize cost, reduce power, and maximize resource utilization simultaneously. The algorithm worked in two parts: in the first part, reinforced learning was used for task scheduling, then in the second part, including the information from the first part, resources were allocated in a multi-agent environment. Experimental results showed that this mechanism enhanced the efficacy of the used resources and minimized the cost. Simultaneously, execution time and execution cost were adequately reduced.

Safari & Khorsand (2018) provided a new time-constrained energy-aware scheduling approach that used the DVFS mechanism to have the host minimize the frequency at various voltage levels. The reduction of energy use, SLA violations, and improved resource utilization were the main objectives of this study. Results showed that when evaluating such metrics, the proposed strategy performed better. A comparative analysis of some existing approaches was discussed in Table 1.

Table 1. Comparative analysis of existing approaches

Author and Year of Publication	Applied Approaches	Advantages	Disadvantages	Scheduling Parameters
Sardaraz and Tahir (2019)	A hybrid algorithm for scheduling scientific workflow based on the PSO algorithm.	Reduction in execution time and monetary cost.	Restricted to some limited parameters.	Load balancing, cost, execution time.
Marahatta et al. (2019)	Presented an energy-efficient dynamic scheduling (EDS) for the virtualized cloud data center.	Improved overall scheduling performance.	Frequent tasks migration, high fault tolerance.	resource utilization, Response time, and task guarantee ratio.
Abualigah & Alkhrabsheh (2022)	Presented MVO-GA to schedule the task transfer according to the availability of the virtual machine.	Optimized the transfer time of the large set of tasks is about 15%	Not suitable for the more extensive data set.	Transfer time, throughput, speed.
Stavrinides & Karataza (2019)	Presented a DVFS on various heterogeneous multi-processor.	Increased efficiency, cost-effective scheduling.	Decrease the execution of the job by several degrees, SLA violation.	Qos, Energy-efficient, monetary cost.
Wang et al. (2017)	Presented RMSR (Replication based scheduling for enhancing the reliability of the system).	Achieve optimal system reliability.	Energy saving issue.	Reliability in the cloud computing environment.
Gupta et al. (2022)	Presented an improved version of the HEFT algorithm.	Optimal assignment of resources, reduce the makespan.	Not suitable for dynamic scheduling.	Makespan.
Abdullahi & Ngadi (2016)	Proposed the discrete symbiotic organism search (DSOS) algorithm to schedule workflows.	Enhanced performance through the degree of imbalance and makespan.	Only works for load balancing.	Overall execution time, degree of imbalance, makespan.
Li et al. (2020)	Presents a workflow scheduling depending on the shortest path algorithm.	Reduced completion time and energy consumption.	The shortest path algorithm does not always find the best path.	Completion time, load balancing, energy consumption.

Table 1. Comparative analysis of existing approaches (continued)

Author and Year of Publication	Applied Approaches	Advantages	Disadvantages	Scheduling Parameters
Asghar et al. (2020)	Proposed reinforcement learning mechanism for resource provisioning and task scheduling.	Achieved optimal solution, reduced cost, and reduced power utilization.	Resources are operated at high frequency, due to which transient errors occur.	Cost, resource utilization, power, makespan.
Safari & Khorsand (2018)	Proposed a new energy-aware scheduling algorithm using the DVFS mechanism at different frequency levels.	Optimal execution time, assured SLA, heuristic polynomial time.	Less efficient with a smaller number of processors.	SLA, resource utilization, energy consumption.

3. Proposed Cloud Model with Host Classification and VM Migration Policies

Cloud infrastructure is the large data center comprising m heterogeneous physical machines PM = {PM_1, PM_2, $PM_{3.....}$ PM_m}. Memory, processing capacity, and network bandwidth are the main features of the physical machine. According to requirements, PMs with varying capabilities in the data center can be activated or deactivated. The physical machine in the cloud data center uses the virtualization method to offer n types of the virtual machine represented as VM= {VM_1, VM_2, $VM_{3....}$ VM_n,} (Krue-Kaew & Kimpan, 2022). Each virtual machine (VM) includes processors with varying computational power, measured in millions of instructions per second (MIPS) and bandwidth. The capability of the virtual machine is being migrated between different physical machines (PMs). The proposed cloud model is represented in Figure 1.

Similarly, Cloud Sim produces physical and virtual machines according to the user's needs, and different instances are formed by varying parameters, such as memory, processing power, bandwidth, etc.:

a. Users/Brokers

A cloud broker is defined as a business or another individual that is the mediator between the seller of the service and the purchaser of that cloud computing service. Thus, a broker is a mediator between two or more parties while negotiating.

b. Task Scheduler

The task scheduler is mainly responsible for scheduling the task for the virtual machine. The main objectives of the scheduler are defined as follows:

- Ensure that each task waiting to be run gets its share of time in the virtual machine.
- Consider the priorities.
- Optimize and balance the use of available resources.
- Reduce response time.

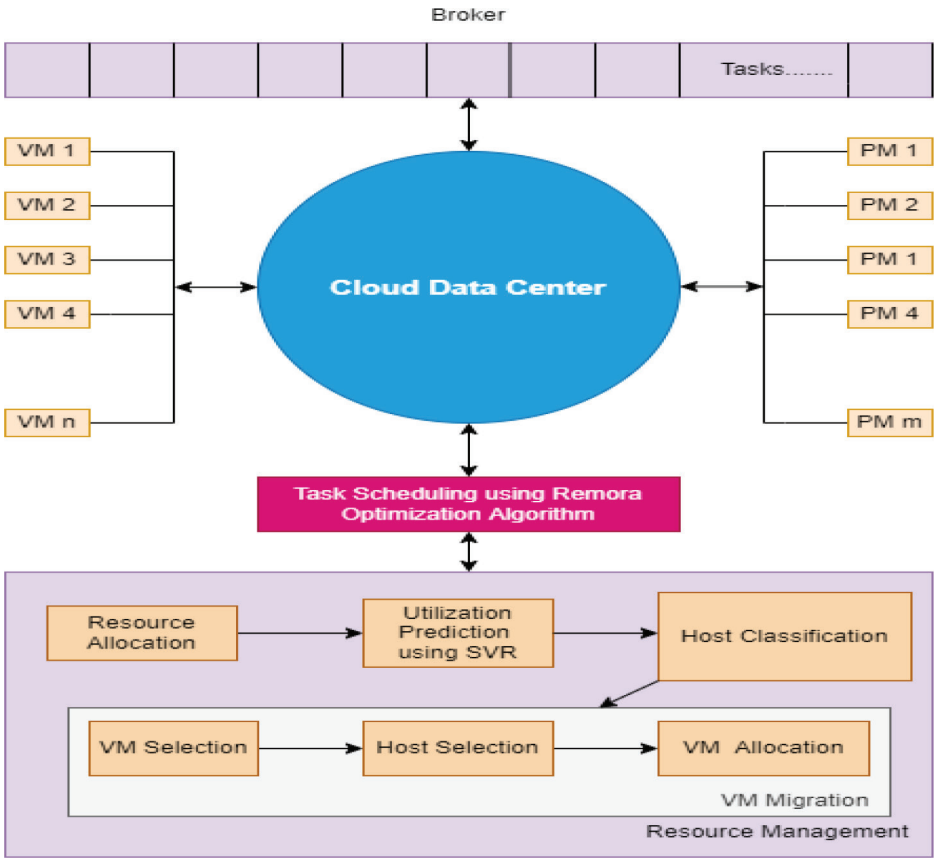

Figure 1. Proposed cloud model

c. Resource Allocator

The primary responsibility of the resource allocator is to assign available resources over the internet to the desired cloud application. It starves services when allocation is not controlled precisely; that problem is solved by permitting the service providers to manage the resources for each module.

d. Prediction Module

In the prediction module, we used support vector regression. It is a busy time series-based prediction model which forecasts future resource usage.

e. Host Detection Module

Support vector regression is used to detect the host; based on SVR host is classified into four parts: overloaded, under-pressure, normal, and underloaded. A list of notations used to indicate various parameters is given in Table 2.

Table 2. Notations used in the paper

Notation	Definition
t_k	Task
C_C	Computing Capacity
E_nC	Energy Consumption
E_xT	Execution Time
HU_i	Host Resource Utilization
BW_cost	Bandwidth Cost
RAM_cost	RAM Cost
CPU_cost	CPU Cost
Total_cost	Total Execution Cost
ED	Euclidian Distance
P_{max}	Maximum Power
P_{min}	Minimum Power
length(t_k)	Length of Tasks
Proc_cap	Processor Capacity
U_L	Upper Limit
L_L	Lower Limit

3.1. Multiobjective Formulation

3.1.1 Host Resource Utilization

CPU (Mips), RAM (Mbps), and bandwidth have different measures. We cannot calculate host utilization by finding the sum of these. To tackle this problem, the available objectives are divided by normalization constant (normConst), as shown in Eq. (1):

$$HU_i = \frac{CPU + RAM + BW}{normConstED} \tag{1}$$

Euclidean distance can be calculated as the difference between the current and previous host utilization as the normalization constant (A *et al*., 2019). Normalization constant is defined in Eq. (2):

$$normConstED = \sqrt{d\left(CPU\right)^2 + d\left(RAM\right)^2 + d\left(BW\right)^2} \tag{2}$$

The relative difference between the present and past CPU, RAM, and BW utilization is determined in this case by the values of d(CPU), d(RAM), and d(BW), respectively.

3.1.2 Energy Consumption

The energy model is discussed based on the linear relationship between processor utilization and energy consumption. In other words, for any task, processing time and processor utilization are the only parameters required for calculating the task's energy consumption (Lee & Zomaya, 2012). The definition of utilization HUi for a resource Ri at any time is shown in Eq. (3):

$$HU_i = \sum_{k=1}^{n} HU_{i,k} \tag{3}$$

Where n detect the number of tasks running at that time and $HU_{i,k}$ represents the resource usage of task tk. For resource Ri at any given time, energy consumption EnC is calculated in Eq. (4):

$$EnC = \left(P_{max} - P_{min}\right) * HU_i + P_{min} \tag{4}$$

Where Pmax detects the power consumption at peak time (100% utilization) and Pmin represents minimum power consumption in active time (as min as 1% utilization).

VM load is determined using Eq. (5), where Proc_cap is based on the number of processing elements and speed measured in MIPS (Singh *et al.,* 2022).

$$Load_{VM} = length\left(tk\right) * Proc_{cap} \tag{5}$$

3.1.3 Execution Time

The execution time ExT denotes the time interval required for executing task t_k on resources VM_{Cc} based on the computing capacity of the available resources (Mohanpriya *et al.,* 2018). This is defined in Eq. (6):

$$ExT\left(t_k, VM_{C_c}\left(t_k\right)\right) = \frac{\sum_{i=0}^{n} length\left(t_k\right)}{\partial\left(VM_{C_c}\right)} \tag{6}$$

Where length(t_k) denotes the length of task (t_k) that is about to be executed and $\partial(VM_{Cc})$ shows the computing capacity of the resource.

3.1.4 Execution Cost

The total execution cost obtained is the sum of BW_ cost, RAM_ cost, and CPU_ cost defined in the following Eq. (10). BW_ cost represents the cost of bandwidth usage. RAM_ cost represents the cost of used memory. CPU_ cost represents the cost of CPU usage.

$$BW_{cost} = ExT * rateofBWusage \tag{7}$$

$$RAM_{cost} = ExT * rateofRAMusage \tag{8}$$

$$CPU_{cost} = ExT * rateofCPUusage \tag{9}$$

$$Total_{cost} = BW_{cost} + RAM_{cost} + CPU_{cost} \tag{10}$$

3.2. Support Vector Regression

SVM is used for regression analysis, time series prediction, and pattern recognition. It has excellent implementation compared to other prediction models, such as LR and neural networks, which are based on the structure risk principle rather than empirical risk minimization. It was also beneficial in considering non-linear problems, merely using a support vector. The implementation of SVM is memory proficient. SVR is the regression scheme based on SVM (Zhu *et al.,* 2016). SVR is the mechanism used for estimating the function. In the first step linear scenario, training data (u_a, v_a) is taken, a = 1…p, where $u \in K^m$ and $v \in K$; additionally, v is an actual number, and u is an m-dimensional real vector. The forecasting function represented in Eq. (11)

$$f(u) = wt, u > +x \tag{11}$$

Where wt stands for weight vector, x for threshold, and <…… > denotes the inner product in k^m. The goal is to establish the ideal weight (wt) and the threshold (x). Later, the mistake caused by the value's estimation process must be decreased. This is known as the empirical risk and is illustrated in Eq. (12):

$$\left|v_a - f(u_a)\right| < \varepsilon \tag{12}$$

The separation between the point and the hyperplane (prediction function) is defined in Eq. (13):

$$dist_a = \frac{\left|wt, u_a > +x - v_a\right|}{\left\|wt\right\|} \tag{13}$$

From Eq. (12), we can deduce that:

$$dist_a = \frac{\left|wt, u_a > +x - v_a\right|}{\left\|wt\right\|} < \frac{\varepsilon}{\left\|wt\right\|} \tag{14}$$

Eq. (14) determines the upper bound of $dist_a$. Accordingly, the optimum hyperplane can be obtained by reducing $\|wt\|$. Thus, the linear regression can be changed to the given problem defined in Eq.(15).

$$minimize \frac{1}{2}\|wt\|^2$$

$$st.\begin{cases} wt,u_a > +x - v_a \le \varepsilon \\ v_a - wt,u_a > -x \le \varepsilon \end{cases} \tag{15}$$

Sometimes, however, not all training data properly find the given constraint, and thus slack variables γ^*, γ are presented to permit the predicted error of some point to be larger than ε. In such a scenario, the optimization problem can be explained as Eq. (16):

$$minimize \frac{1}{2}\|wt\|^2 + R\sum_{a=1}^{p}\left(\gamma^* + \gamma\right)$$

$$st.\begin{cases} wt,u_a > +x - v_a \le \varepsilon + \gamma^* \\ v_a - wt,u_a > -x \le \varepsilon + \gamma \\ \gamma^*,\gamma \ge 0 \end{cases} \tag{16}$$

In contrast, in Eq. (16), the first component creates a function to enhance generalization, and the regularisation constant R ensures that risks based on regularisation and those based on empirical data are aligned. As a result, the dual form is established by introducing the Lagrangian function:

$$minimize \frac{1}{2}\sum_{a,b=1}^{p}(\beta_{a-}\beta_b^*)(\beta_{b-}\beta_a^*) < u_a, u_b >$$

$$+\sum_{a=1}^{p}(\beta_a + \beta_b^*)\varepsilon - \sum_{a=1}^{p}(\beta_{a-}\beta_a^*)v_a$$

$$st.\begin{cases} \sum_{a=1}^{p}\left(\beta_a - \beta_a^*\right) = 0 \\ 0 \le \beta_a,\beta_a^* \le R \end{cases} \tag{17}$$

By evaluating Eq. (17) with optimum Lagrange multipliers β^* and β, $\hat{w}t$ and $\hat{x}$ are given as:

$$\hat{w}t = \sum_{a=1}^{p} \left(\beta_a^* - \beta_a\right) u_a = 0$$

$$\hat{x} = \frac{-1}{2} \hat{w}t, \tag{18}$$

In Eq. (18), u_c and u_d represent the support vectors. In a non-linear scenario, all training data are initially plotted in a high-dimensional feature space with the appropriate non-linear function. After that, linear regression is processed using a high-dimensional feature space. In a space with high-dimensional features, just the internal product is taken, kernel function K(u, v) overlap < φ(u),φ(v)> to be used in non-linear regression. In Eq. (19), the radial basis function (RBF) is used in the machine learning mechanism.

$$K\left(u, v\right) = exp\left(\frac{-\|u - v\|^2}{2\partial^2}\right) \tag{19}$$

Eq. (19) provides a scalar depending on the distance between two vectors. ∂ is the value set by available users to represent the speed of the scalar tending to zero. Overall, the principle of the SVR variable contributes to making a précised prediction.

3.3. Conceptualization of Problem

In cloud computing, scheduling is allocating virtual machines (VMs) to meet user needs or assigning tasks to a set of VMs, with specified constraints. Task scheduling cannot be carried out based on a single criterion. However, many factors that must be considered while mapping the task schedule are viewed as a problem that tends to find an optimal mapping of a set of sub-tasks of different tasks over the set of available resources (Ranjan *et al.,* 2020). An efficient scheduling method increases system throughput, maximizes resource utilization, reduces expenses, and shortens processing times. In the proposed research, task scheduling is based on remora optimization.

ROA is a new natural bio-inspired and meta-heuristic algorithm that aims to minimize energy consumption and execution time. Parasitic behavior is the main inspiration for the remora algorithm. According to it, host locations are updated (Jia *et al.,* 2021). In the case of a large host (giant whales), remora feed on the host's extermination and natural enemies. In the case of a small host, the remora chases the host to move fast(swordfish) to the bait-rich area to prey; from the above two cases, the remora makes a judgment based on experience. When it starts initiation to prey, the host is continuously updated and a global decision is made. If it eats by encircling the host remora, the local update is continued with, without changing the host. Because the host in ROA can be adequately substituted, such as ships, turtles, etc., this method is more likely to offer a fresh concept for a memetic algorithm. The suggested work must be merged with the tasks that need to be optimized and estimated using the data.

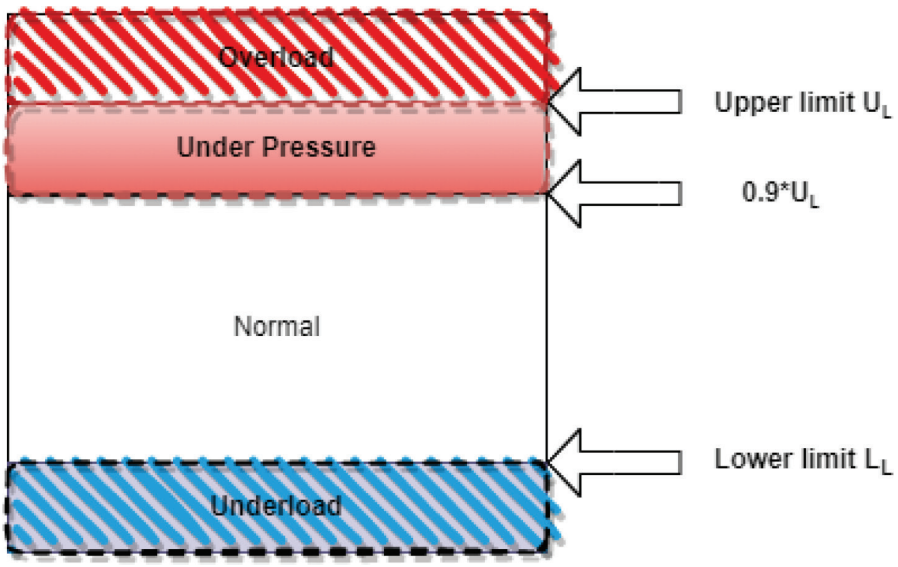

Figure 2. Category of host classification

After scheduling using remora optimization, tasks are distributed to different virtual machines according to their requirement. After scheduling, it is necessary to determine the load using Eq.(5) at each host. Hence, we have used the support vector regression prediction model. This SVR is used for host classification where the upper and lower boundary are identified. This helps to identify if the host gets overloaded or underloaded. Hence, with the help of these boundaries, we obtained four lists in our proposed approach, as represented in Figure 2. In the first case, if the load is greater than the upper boundary, the host is overloaded, and if the load is greater than 0.9*ul and less than the upper limit, then the host is treated as under pressure. The host is treated as normal if the load lies between the upper and lower limit. In the last case, if the load is less than the lower limit, it lies in under loaded situation (Khoshkholgi *et al.,* 2017). This is further illustrated in Figure 3. After host classification, an appropriate VM consolidation technique is applied for VM migration. Here, for VM migration, we have used a random selection policy.

Algorithm 1 describes multi-resource utilization (CPU, memory, bandwidth) taken as input and follows the above steps to train the SVR model to obtain a boundary line.

Algorithm 1. Host resource utilization (HU_i) prediction

Input: CPU utilization, memory utilization, bandwidth utilization.
Output: Decision boundary

1. Enter the input parameters to the model.
2. Use of radial based kernel to map non-linear data to higher-dimensional space.
3. Load training data set (input) and train model to decide boundary line that classifies data.
4. Boundary line (UL, LL).

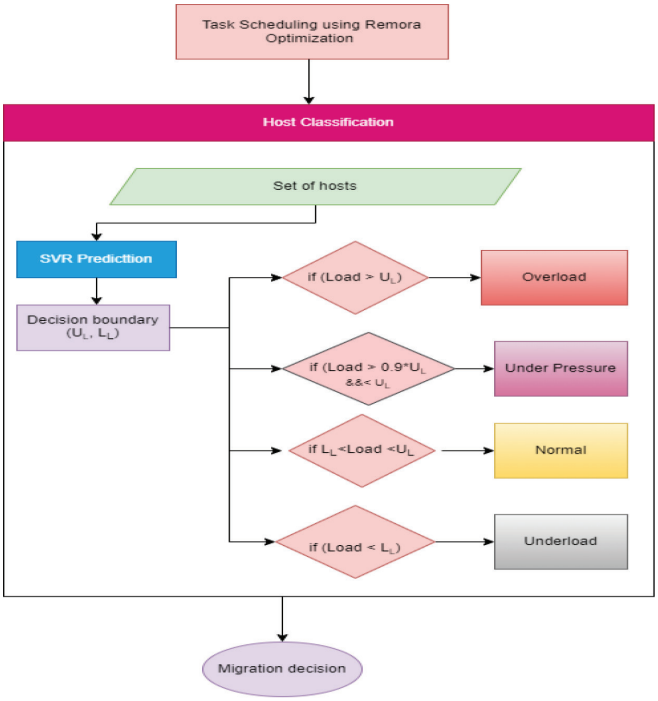

Figure 3. Working of the proposed model

3.4. Migration Policy

In the dynamic consolidation scheme of detecting overload/underload hosts, the selection of VM performs a crucial role in VM migration (Nehra & Nagaraju,2022). To reduce both energy usage and execution time, VM migration is carried out. Minimum utilization (MU), random selection (RS), minimum migration time (MMT), maximum correlation (MC), and constant fixed selection (CFS) are some of the ways for VM selection that are described in the literature.

a. Minimum Utilization
In this method, the VM with the lowest CPU utilization is chosen to reduce processing overhead.

b. Random Selection
According to the uniformly distributed random variable, this method chooses a VM at random for migration.

c. Minimum Migration Time
In this method, the VM with the shortest migration time is chosen. Calculating migration time is defined in Eq. (20).

$$Migrationtime = \frac{Vm}{B} \tag{20}$$

Where Vm is the RAM that the VM is using, and B is the bandwidth. Therefore, the VM with this ratio at a minimum migrates with the least delay.

d. Maximum Correlation
In this method, the application's resource utilization is correlated statistically, and the VM with the highest correlation value is chosen for migration.

e. Constant Fixed Selection
In this method, the VM is also chosen randomly, but the selection is constant, meaning that the VM in the first, middle, or last place is chosen.

Total complexity of the proposed scheme can be formulated in Eq. (21):

$$TotalTimecomplexity\left(ROA/SVR/RS\right)= timecomplexity\left(ROA\right)+timecomplexity\left(SVR\right)+timecomplexity\left(RS\right) \quad (21)$$

4 . Simulation and Results Analysis

CloudSim is a simulator program written in Java and used for designing and simulating the virtualized data center based on the Cloud. It also provides an interface for memory, bandwidth, storage, and VMs (Arroba *et al.,* 2017). A random selection policy is used here for VM migration.

4.1. Experiment and Parameter Setup

At the host level, CPU capacity is 50 MIPS, 800 heterogeneous hosts in total, with 512 MB of RAM each the corresponding bandwidth is 1000 Mbps, and storage is 100GB. At the VM level, the total no. of XEN virtual machines is 1175, the no. of tasks is set to be 3000, the memory is 1536MB, the CPU capacity is 1000 MIPS, the no. of CPU is 1, bandwidth is 1000 Mbps and storage is 1000 GB. Various performance varieties are evaluated to calculate the performance of energy consumption execution time and cost. The parameter setup of a VM in the cloud data center is described in Table 3. The parameter setup of the host is described in Table 4.

Table 3. The parameter setup of the VM in the data center

Parameters	**Value**
Number of Tasks	3000
Number VMs	1175
VMs Name	Xen
Number of CPU	1
CPU Capacity	1000 MIPS
Memory	1536 MB
Bandwidth	1000mbps
Storage	10000 GB

Table 4. The parameter setup of the host in the data center

Parameter	Value
Number of Hosts	800
Network Bandwidth	1000 Mbps
RAM	2048 MB
Storage	1000000GB
Capacity of CPU	1000MIPS

4.2 . Experiment Metrics

The experiments were conducted with the aim of evaluating performance at three levels, namely, energy consumption, total energy consumed by data center execution time, and execution cost. In this subsection, the experimental results of the proposed mechanism are discussed, considering energy consumption execution time and cost. To know the efficiency of the proposed mechanism, it is compared with prior existing approaches, such as the genetic algorithm (GA), the particle-swarm optimization (PSO) algorithm, minimum-migration time (MMT) policy, random selection (RS) policy, median absolute deviation (MAD) and interquartile range (IQR) algorithms (Li *et al.,* 2019).

Figure 4 provides a comparative analysis of the energy consumption of the proposed load detection mechanism. The proposed load detection mechanism consumed in total 0.641 kWh, which is less energy than other existing approaches. The result shows that energy consumption becomes reduced throughout the entire scheduling.

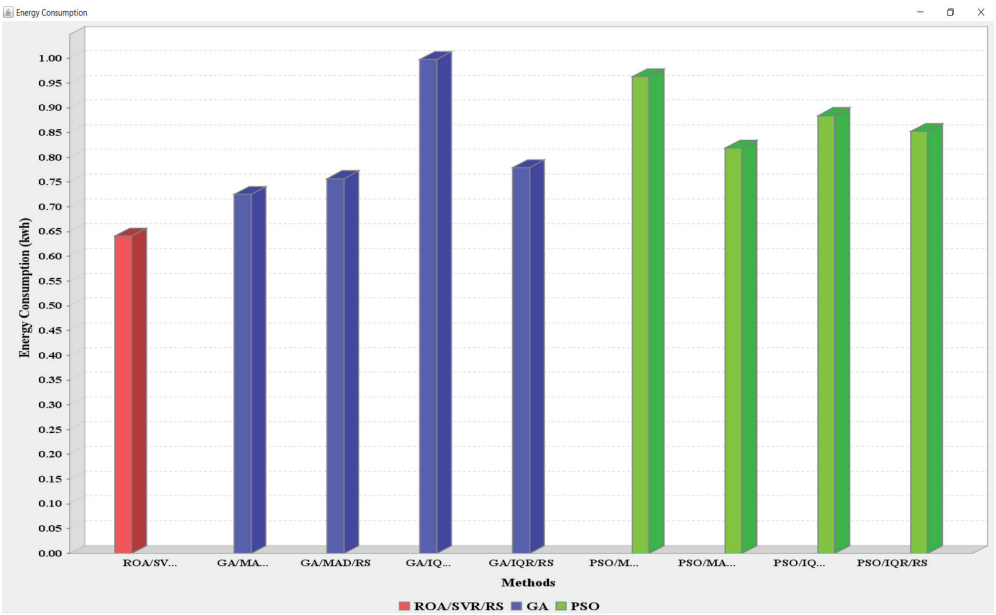

Figure 4. Comparison of the proposed approach in terms of EnC with eight existing approaches

Figure 5 provides a comparative analysis of the execution time of the proposed load detection mechanism. The time taken by the proposed mechanism is 165.39sec less than other existing approaches. Thus, the proposed mechanism consumed less time throughout the entire scheduling.

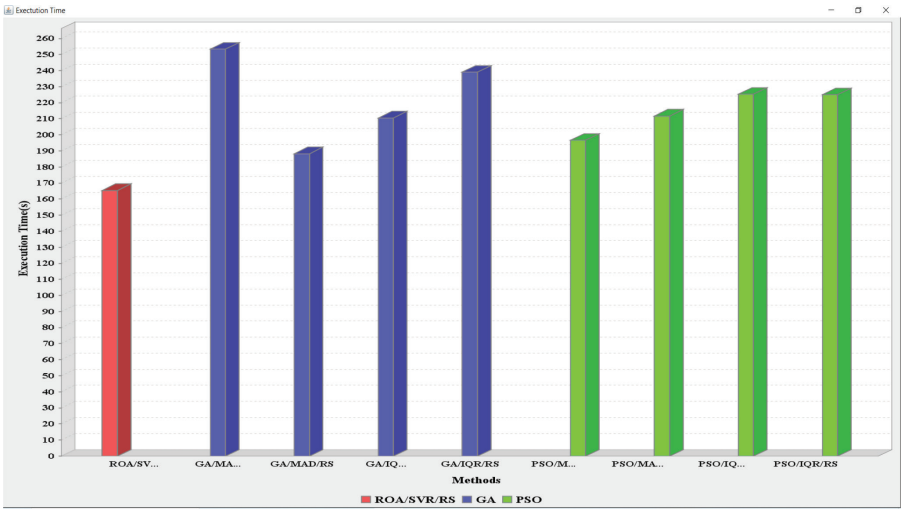

Figure 5. Comparison of ExT of the proposed scheme with eight existing approaches

Figure 6 depicts execution cost. When the no. of tasks was 3000, then the cost was 10,000$. When increased to 3500, the execution cost further rose to 20,000$. The no. of tasks was further increased to 4000, for which the computed execution cost was 30,000$, and finally for 4500 tasks the cost was 40,000$.

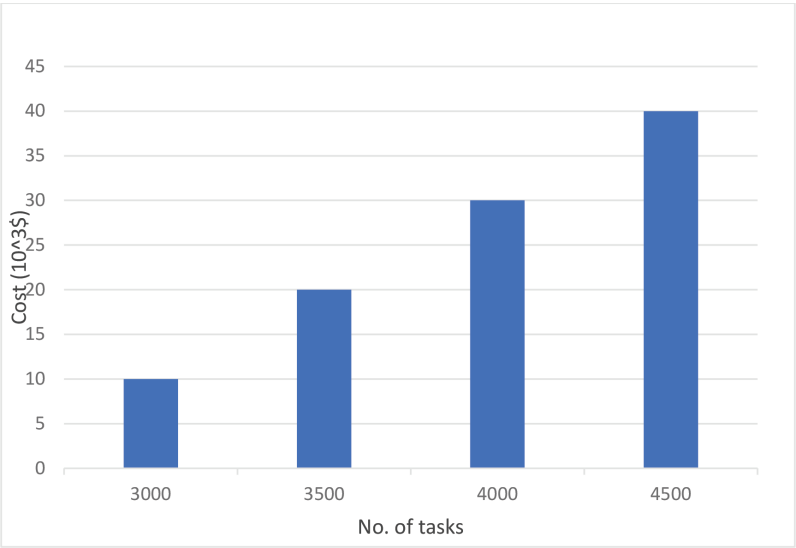

Figure 6. Execution cost

4.3. Result Discussion

Table 5 shows the performance analysis of the proposed algorithm with the eight existing meta-heuristic algorithms, including GA/MAD/MMT, GA/MAD/RS, GA/IQR/MMT, GA/IQR/RS, PSO/MAD/MMT, PSO/MAD/RS, PSO/IQR/MMT, PSO/IQR/RS. The performance analysis indicates that the proposed algorithms provide precise results when a comparison is performed with another existing mechanism for task scheduling.

Table 5. Performance analysis of proposed ROA/SVR/RS with the eight existing approaches

Approaches	**Execution time (sec)**	**Energy Consumption (kWh)**
Proposed ROA/SVR/RS	165.39	0.641
GA/MAD/MMT	253.475	0.725
GA/MAD/RS	188.265	0.757
GA/IQR/MMT	210.592	0.998
GA/IQR/RS	239.139	0.779
PSO/MAD/MMT	196.623	0.963
PSO/MAD/RS	211.769	0.819
PSO/IQR/MMT	225.287	0.884
PSO/IQR/RS	225.072	0.852

5. Conclusion and Future Scope

In cloud computing, scheduling is a significant process for assigning tasks to virtual machines. Energy consumption and execution time have recently become more important in research work. This research has developed a load detection and classification mechanism framework. After scheduling using remora optimization, the dynamic time series prediction model, SVR, has been used. SVR was done for host classification, where the upper and lower boundaries were identified. This helped to determine if the host is getting overloaded or underloaded. Hence, with the help of these boundaries, we have obtained four cases in our proposed approach: overloaded, under pressure, normal, and underload. After host classification, an appropriate VM consolidation technique was applied for VM migration. Here, for experimental analysis, random selection policy for VM migration was used to validate the result. Prediction defines the boundaries for better load distribution among the hosts, which minimizes energy consumption and execution time. In the experimental results, the proposed approach obtained an energy consumption of 0.641kWh, and an execution time of 165.39sec.

In the future, more traits can be added to improve accountability and integrated with different kinds of workflows, with different parameters to obtain more precise results. We plan to perform a comparison between available scheduling mechanisms concerning SLA violation. Work could be further expanded by a new task cum resource-aware scheduling methodology that will exploit the nature of the presented workload to provide the efficient mapping of tasks on the available cloud resources.

References

A. El-Moursy, A., Abdelsamea, A., Kamran, R., & Saad, M. (2019). Multi-dimensional regression host utilization algorithm (MDRHU) for host overload detection in cloud computing. *Journal of Cloud Computing*, *8*(1), 1-17. https://doi.org/10.1186/s13677-019-0130-2

Abdullahi, M., Ngadi, M. A., & Abdulhamid, S. M. (2016). Symbiotic organism search optimization based task scheduling in cloud computing environment. *Future Generation Computer Systems*, *56*, 640-650. https://doi.org/10.1016/j.future.2015.08.006

Abualigah, L., & Alkhrabsheh, M. (2022). Amended hybrid multi-verse optimizer with genetic algorithm for solving task scheduling problem in cloud computing. *The Journal of Supercomputing*, *78*(1), 740-765. https://doi.org/10.1007/s11227-021-03915-0

Adhikari, M., & Koley, S. (2018). Cloud Computing: a multi-workflow scheduling algorithm with dynamic reusability. *Arabian Journal for Science and Engineering*, *43*(2), 645-660. https://doi.org/10.1007/s13369-017-2739-0

Arroba, P., Moya, J. M., Ayala, J. L., & Buyya, R. (2017). Dynamic voltage and frequency scaling–aware dynamic consolidation of virtual machines for energy-efficient cloud data centers. *Concurrency and Computation: Practice and Experience*, *29*(10), e4067. https://doi.org/10.1002/cpe.4067

Arunarani, A. R., Manjula, D., & Sugumaran, V. (2019). Task scheduling techniques in cloud computing: A literature survey. *Future Generation Computer Systems*, *91*, 407-415. https://doi.org/10.1016/j.future.2018.09.014

Asghari, A., Sohrabi, M. K., & Yaghmaee, F. (2020). Online scheduling of dependent tasks of Cloud's workflows to enhance resource utilization and reduce the makespan using multiple reinforcement learning-based agents. *Soft Computing*, *24*(21), 16177-16199. https://doi.org/10.1007/s00500-020-04931-7

Bal, P. K., Mohapatra, S. K., Das, T. K., Srinivasan, K., & Hu, Y. C. (2022). A joint resource allocation, security with efficient task scheduling in cloud computing using hybrid machine learning techniques. *Sensors, 22*(3), 1242. https://doi.org/10.3390/s22031242

Garg, N., Neeraj, Raj, M., Gupta, I., Kumar, V., & Sinha, G. R. (2022). Energy-efficient scientific workflow scheduling algorithm in the cloud environment. *Wireless Communications and Mobile Computing, 2022*, 1-12. https://doi.org/10.1155/2022/1637614

Gupta, S. D., Iyer, S., Agarwal, G., Poongodi, M., Algarni, A. D., Aldehim, G., & Raahemifar, K. (2022). Efficient Prioritization and Processor Selection Schemes for HEFT Algorithm: A Makespan Optimizer for Task Scheduling in Cloud Environment. *Electronics*, *11*(16), 2557. https://doi.org/10.3390/electronics11162557

Hosseinzadeh, M., Ghafour, M. Y., Hama, H. K., Vo, B., & Khoshnevis, A. (2020). Multi-objective task and workflow scheduling approach in cloud computing: a comprehensive review. *Journal of Grid Computing, 18*(3), 327-356. https://doi.org/10.1007/s10723-020-09533-z

Jia, H., Peng, X., & Lang, C. (2021). Remora optimization algorithm. *Expert Systems with Applications*, *185*, 115665. https://doi.org/10.1016/j.eswa.2021.115665

Khaleel, M. I., & Zhu, M. (2016). Energy-efficient task scheduling and consolidation algorithm for workflow jobs in cloud. *International Journal of Computational Science and Engineering*, *13*(3), 268–284. https://doi.org/10.1504/IJCSE.2016.078933

Khan, M. S. A., & Santhosh, R. (2022). Task scheduling in cloud computing using hybrid optimization algorithm. *Soft Computing*, *26*(23), 13069-13079. https://doi.org/10.1007/s00500-021-06488-5

Khoshkholghi, M. A., Derahman, M. N., Abdullah, A., Subramaniam, S., & Othman, M. (2017). Energy-Efficient Algorithms for dynamic virtual machine consolidation in cloud data centers. *IEEE Access*, *5*, 10709-10722. https://doi.org/10.1109/access.2017.2711043

Kruekaew, B., & Kimpan, W. (2022). Multi-objective task scheduling optimization for load balancing in a cloud computing environment using a hybrid artificial bee colony algorithm with reinforcement learning. *IEEE Access*, *10*, 17803–17818. https://doi.org/10.1109/ACCESS.2022.3149955

Kumar, M., Sharma, S. C., Goel, A., & Singh, S. P. (2019). A comprehensive survey for scheduling techniques in cloud computing. *Journal of Network and Computer Applications*, *143*, 1–33. https://doi.org/10.1016/j.jnca.2019.06.006

Lee, Y. C., & Zomaya, A. Y. (2012). Energy efficient utilization of resources in cloud computing systems. *The Journal of Supercomputing*, *60*, 268-280. https://doi.org/10.1007/s11227-010-0421-3

Li, C., Tang, J., Ma, T., Yang, X., & Luo, Y. (2020). Load balance-based workflow https://doi.org/10.1016/j.jnca.2019.06.006 job scheduling algorithm in a distributed cloud. *Journal of Network and Computer Applications*, *152*, 102518. https://doi.org/10.1016/j.jnca.2019.102518

Li, L., Dong, J., Zuo, D., & Wu, J. (2019). SLA-aware and energy-efficient VM consolidation in Cloud data centers using the robust linear regression prediction model. *IEEE Access*, *7*, 9490–9500. https://doi.org/10.1109/ACCESS.2019.2891567

Magotra, B., Malhotra, D., & Dogra, A. K. (2022). Adaptive Computational Solutions to Energy Efficiency in Cloud Computing Environment Using VM Consolidation. *Archives of Computational Methods in Engineering*, *30*, 1-30. https://doi.org/10.1007/s11831-022-09852-2

Malik, S., Tahir, M., Sardaraz, M., & Alourani, A. (2022). A resource utilization prediction model for cloud data centers using evolutionary algorithms and machine learning techniques. *Applied Sciences,* 12(4), 2160. https://doi.org/10.3390/app12042160

Marahatta, A., Pirbhulal, S., Zhang, F., Parizi, R. M., Choo, K. K. R., & Liu, Z. (2019). Classification-based and energy-efficient dynamic task scheduling scheme for virtualized cloud data center. *IEEE Transactions on Cloud Computing*, *9*(4), 1376-1390. https://doi.org/10.1109/TCC.2019.2918226

Mishra, S. K., Sahoo, S., Sahoo, B., & Jena, S. K. (2020). Energy-efficient service allocation techniques in the Cloud: A survey. *IETE Technical Review*, *37*(4), 339–352. https://doi.org/10.1080/02564602.2019.1620648

Mohanapriya, N., Kousalya, G., Balakrishnan, P., & Pethuru Raj, C. (2018). Energy efficient workflow scheduling with virtual machine consolidation for green cloud computing. *Journal of Intelligent & Fuzzy Systems*, *34*(3), 1561–1572. https://doi.org/10.3233/JIFS-169451

Nehra, P., & Nagaraju, A. (2022). Host utilization prediction using hybrid kernel-based support vector regression in cloud data centers. *Journal of King Saud University – Computer and Information Sciences, 34*(8), 6481-6490. https://doi.org/10.1016/j.jksuci.2021.04.011

Panwar, S. S., Rauthan, M. M. S., & Barthwal, V. (2022). A systematic review on effective energy utilization management strategies in cloud data centers. *Journal of Cloud Computing*, *11*(1), 1–29. https://doi.org/10.1186/s13677-022-00368-5

Pirozmand, P., Hosseinabadi, A. A. R., Farrokhzad, M., Sadeghilalimi, M., Mirkamali, S. S., & Slowik, A. (2021). Multi-objective hybrid genetic algorithm for task scheduling problem in cloud computing. *Neural computing and applications*, *33*(19), 13075-13088. (SCIE). https://doi.org/10.1007/s00521-021-06002-w

Ranjan, R., Thakur, I. S., Aujla, G. S., Kumar, N., & Zomaya, A. Y. (2020). Energy-Efficient workflow scheduling using Container-Based virtualization in Software-Defined data centers. *IEEE Transactions on Industrial Informatics*, *16*(12), 7646-7657. https://doi.org/10.1109/tii.2020.2985030

Safari, M., & Khorsand, R. (2018). Energy-aware scheduling algorithm for time-constrained workflow tasks in DVFS-enabled cloud environment. *Simulation Modelling Practice and Theory*, *87*, 311-326. https://doi.org/10.1016/j.simpat.2018.07.006

Sardaraz, M., & Tahir, M. (2019). A hybrid algorithm for scheduling scientific workflows in cloud computing. *IEEE Access*, *7*, 186137-186146. https://doi.org/10.1109/access.2019.2961106

Singh, S., Kumar, R., & Rao, U. P. (2022). Multi-Objective adaptive Manta-Ray foraging optimization for workflow scheduling with selected virtual machines using Time-Series-Based prediction. *International Journal of Software Science and Computational Intelligence*, *14*(1), 1-25. https://doi.org/10.4018/ijssci.312559

Stavrinides, G. L., & Karatza, H. D. (2019). An energy-efficient, QOS-aware and cost-effective scheduling approach for real-time workflow applications in cloud computing systems utilizing DVFS and approximate computations. *Future Generation Computer Systems*, *96*, 216-226. https://doi.org/10.1016/j.future.2019.02.019

Wang, S., Li, K., Mei, J., Xiao, G., & Li, K. (2017). A reliability-aware task scheduling algorithm based on replication on heterogeneous computing systems. *Journal of Grid Computing*, *15*(1), 23-39. https://doi.org/10.1007/s10723-016-9386-7

Zhu, Z., Peng, J., Zhou, Z., Zhang, X., & Huang, Z. (2016). PSO-SVR-Based Resource Demand Prediction in Cloud Computing. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, *20*(2), 324-331. https://doi.org/10.20965/jaciii.2016.p0324

Zolfaghari, R., Sahafi, A., Rahmani, A. M., & Rezaei, R. (2021). Application of virtual machine consolidation in cloud computing systems. *Sustainable Computing: Informatics and Systems*, *30*, 100524. (19). https://doi.org/10.1016/j.suscom.2021.100524