

10-16-2021

## Hybrid Approach for Resource Allocation in Cloud Infrastructure Using Random Forest and Genetic Algorithm

Madhusudhan H S

Satish Kumar T

S.M.F D Syed Mustapha  
*Zayed University*

Punit Gupta  
*Manipal Academy of Higher Education*

Rajan Prasad Tripathi

Follow this and additional works at: <https://zuscholars.zu.ac.ae/works>



Part of the [Computer Sciences Commons](#)

---

### Recommended Citation

S, Madhusudhan H; Kumar T, Satish; Mustapha, S.M.F D Syed; Gupta, Punit; and Tripathi, Rajan Prasad, "Hybrid Approach for Resource Allocation in Cloud Infrastructure Using Random Forest and Genetic Algorithm" (2021). *All Works*. 4639.  
<https://zuscholars.zu.ac.ae/works/4639>

This Article is brought to you for free and open access by ZU Scholars. It has been accepted for inclusion in All Works by an authorized administrator of ZU Scholars. For more information, please contact [scholars@zu.ac.ae](mailto:scholars@zu.ac.ae).

## Research Article

# Hybrid Approach for Resource Allocation in Cloud Infrastructure Using Random Forest and Genetic Algorithm

Madhusudhan H S <sup>1</sup>, Satish Kumar T <sup>2</sup>, S.M.F D Syed Mustapha <sup>3</sup>, Punit Gupta <sup>4</sup>,  
and Rajan Prasad Tripathi <sup>5</sup>

<sup>1</sup>Department of Computer Science and Engineering, NIE Institute of Technology, Mysuru, Karnataka, India

<sup>2</sup>Department of Computer Science and Engineering, BMS Institute of Technology and Management, Bengaluru, Karnataka, India

<sup>3</sup>College of Technological Innovation, Zayed University, Dubai, UAE

<sup>4</sup>Department of Computer and Communication Engineering, Manipal University Jaipur, Jaipur, India

<sup>5</sup>Department of Electronics and Communication, Amity University Tashkent, Tashkent, Uzbekistan

Correspondence should be addressed to Punit Gupta; [punit.gupta@jaipur.manipal.edu](mailto:punit.gupta@jaipur.manipal.edu)

Received 18 April 2021; Revised 2 September 2021; Accepted 29 September 2021; Published 16 October 2021

Academic Editor: Marco Aldinucci

Copyright © 2021 Madhusudhan H S et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In cloud computing, the virtualization technique is a significant technology to optimize the power consumption of the cloud data center. In this generation, most of the services are moving to the cloud resulting in increased load on data centers. As a result, the size of the data center grows and hence there is more energy consumption. To resolve this issue, an efficient optimization algorithm is required for resource allocation. In this work, a hybrid approach for virtual machine allocation based on genetic algorithm (GA) and the random forest (RF) is proposed which belongs to a class of supervised machine learning techniques. The aim of the work is to minimize power consumption while maintaining better load balance among available resources and maximizing resource utilization. The proposed model used a genetic algorithm to generate a training dataset for the random forest model and further get a trained model. The real-time workload traces from PlanetLab are used to evaluate the approach. The results showed that the proposed GA-RF model improves energy consumption, execution time, and resource utilization of the data center and hosts as compared to the existing models. The work used power consumption, execution time, resource utilization, average start time, and average finish time as performance metrics.

## 1. Introduction

Cloud computing is a form of distributed computing that brings in utility models to deliver measurable and scalable resources remotely. Cloud is also a materialistic implementation of parallel computing, grid computing, and distributed computing [1]. The cloud environment offers a shared pool of resources to users as a service on an “on-demand” approach [2]. Effective computing capability and enormous storage capacity allow the users to access services of the cloud anytime and anywhere. A cloud data center comprises IT resources like databases, servers, communication devices, network, and software systems. More user’s demand for cloud resources makes the cloud providers scale

up the number of servers or required hardware. As a result, the creation of more physical nodes will lead to an increase in power consumption by the data center. Data centers consume 2% of today’s worldwide electricity. It is expected to reach 8% by 2030. There are three power consumers in a data center, namely cooling systems, data center networks, and servers. 10 to 25% of power is consumed by the network, cooling systems consume 15 to 30% power, and servers will consume power around 40 to 55% [3].

IaaS (Infrastructure as a Service) offers computing resources like RAM, CPU, Network, and Storage as a service and their use is likely abided by SLA (Service Level Agreement). Resource utilization is also having an impact on energy consumption. Low resource utilization is one of the

reasons for the energy inadequacy of the data center [4]. If the CPU utilization is as low as 10%, that means the workload is less and energy consumption is above 50% of the peak power. The virtualization techniques in IaaS play a vital role here, which efficiently improve resource or cloud utilization [5]. Virtualization offers resource sharing which allows virtual machines (VMs) to execute physical machines (PMs) to process user requests. Three possible operations performed using virtualization are VM isolation, VM migration, and VM consolidation. VM migration technique shifts the running virtual machines from a physical machine to another. In the VM consolidation process, virtual machines running on different hosts will leave that host and gather on fewer ones to reduce energy wastage by switching off the initial running host or moving it to hibernate mode [6]. Virtual Machine Placement (VMP) is a technique of executing virtual machines on suitable physical machines. To enhance power efficiency and maximize resource utilization, an efficient VMP technique is very much necessary [7]. The VMP problem is an NP-hard optimization problem [8].

In this work, efficient hybrid VMP technique is using the genetic algorithm (GA) and random forest (RF) algorithm. Our objective is to reduce the energy consumption of the data center while maintaining the load across several physical machines. Maximizing resource utilization of the physical machines is also taken as one of the key parameters for the evaluation of the proposed method. Cloud demands the least waiting time and least request completion time; therefore, minimizing execution time, average start time, and average finish time is the objective of this work. Another objective of the proposed model is to reduce the time to find the optimal solution which takes maximum time iterative metaheuristic algorithms like GA, ACO, PSO, and many more. The model is aimed to train the machine learning model with the best optimal solution and then, in the next iteration, the trained model can be used to predict the optimal solution in a constant time removing the time taken by evolutions in search of the global best solution.

One of the metaheuristic techniques used to find a globally optimal solution is the genetic algorithm. Firstly, the GA focuses on generating an optimized schedule for resource allocation which acts as training dataset that contains the mapping of virtual machines to the physical machines. Next, the dataset derived from GA is used to train the random forest algorithm and then it performs the classification, i.e., allocation of virtual machines to the physical machines. Classification accuracy of the RF is tested using the subset of the dataset obtained from the GA.

The random forest is a supervised machine learning technique and it can perform the placement of virtual machines on the best physical machines with good accuracy as it reduces overfitting in decision trees. Figure 1 represents the system model. It comprises several physical machines within a data center. Each physical machine can execute many virtual machines. The Virtual Machine Monitor (VMM), also known as Hypervisor, is a software program that facilitates the creation, managing, and monitoring of virtual machines. It also manages a virtualized environment

on top of physical machines. When a request for VM execution is received by the data center manager, firstly it gathers the status information from all available physical machines and delivers it to the VM scheduler. The VM scheduler is developed using GA-RF technique. In the next step, VM scheduler analyzes status information and then allocates virtual machines to apt physical machines.

The rest of the work is organized as follows: Section 2 brings in a literature survey with existing models and their comparison. The proposed method is discussed in Section 3, Section 4 gives an evaluation of the proposed method with experimental setup and results, and, lastly, conclusion and future work are given in Section 5.

## 2. Literature Review

This section brings in some of the work done by researchers. Table 1 depicts different approaches in the field of virtual machine placement and the parameters they have considered for the performance evaluation. Authors in [9] propose VMP technique using a combination of genetic algorithm and Tabu search algorithm. The authors have focused on achieving energy efficiency with an increase in load balance. They have also taken execution time for comparison of different algorithms in their work. Abohamama et al. [10] presented a VMP algorithm using an improved permutation-based genetic algorithm to improve energy consumption rate by reducing the number of active hosts that run VMs. The proposed method is compared with the Flow Shop Scheduling Problem and Traveling Salesman Problem. Yao et al. [11] introduced a VM placement procedure based on Weighted PageRank. They focused on minimizing the number of active physical machines and also increased resource utilization of all hosts in the data centers. To avoid the proposed method falling in local optimum solution, the impact of nonplaced virtual machines is considered. During the process of selecting a physical machine for VM, Weighted PageRank covers unplaced VMs of several types and the algorithm measures the possibility of a physical machine making complete use of resources under different conditions.

Authors in [12] proposed a stochastic VMP approach to increase energy efficiency and resource utilization of the data center. Here resource requirements are modeled as random variables instead of taking deterministic values to denote resource requirements. Due to variation in the resource requirements, the proposed optimization model is subject to a probabilistic restriction on resource overflow probability on every physical machine [13]. VMP in the heterogeneous data center using a Binary gravitational search algorithm (BGSA) was proposed. The aim of this work is to decrease energy consumption. The proposed method uses agents as objects and their mass is used to measure the performance. The object having a higher mass has a better solution [14]. Authors have proposed an evolutionary approach to increase energy efficiency. Here the proposed approach incorporates the reserved virtual machines also. Both simulation and real-time cloud environments were used to evaluate the performance of different techniques.

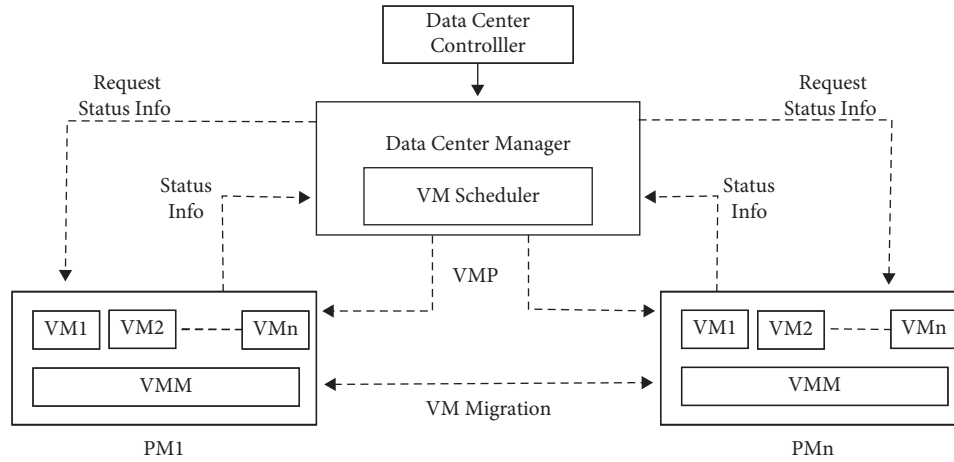


FIGURE 1: System model in cloud infrastructure for VM placement.

TABLE 1: Classification of different approaches in existing literature.

Reference	Technique/algorithm used	Energy consumption	Resource utilization	Load balance	SLA	No. of hosts shutdown/no. of active PM	Execution time/ elapsed/ simulation time
[9]	Metaheuristic (genetic and tabu search algorithm)	✓		✓			✓
[10]	Optimization (permutation-based genetic algorithm and multidimensional resource-aware best fit)	✓	✓				✓
[11]	Heuristic/Weighted PageRank		✓			✓	
[12]	Metaheuristic/stochastic VM placement algorithm	✓					
[13]	Heuristic/binary gravitational search algorithm	✓					
[14]	Optimization/evolutionary approach	✓					✓
[15]	Machine learning/reinforcement learning		✓	✓		✓	✓
[16]	Machine learning and metaheuristic/ (Naive Bayesian Classifier and Random Key Cuckoo Search)	✓			✓	✓	
[17]	Heuristic/HeporCloud	✓					✓
[18]	Heuristic	✓					✓
[19]	Optimization/game theory	✓					✓

Consolidation of VMs on lesser hosts resulted in a decrease in energy consumption.

Ghasemi et al. [15] have designed a reinforcement learning-based approach to address VM placement. The authors focused on load balancing while maximizing resource utilization and a number of hosts shutdown. The proposed method chooses an action from available acceptable actions and executes it on a cloud environment. It receives a reinforcement signal conforming to the suitability of the virtual machine placement solution by using that action [16]. The authors proposed a hybrid approach based on Naive Bayesian Classifier and Random Key Cuckoo Search for VM consolidation problem to minimize energy consumption. Here Naive Bayes is used for detecting the future state of the hosts which is necessary to perform virtual machine placement in an efficient manner. Khan et al. [17]

presented HeporCloud, a framework for a hybrid cloud platform that includes an integrated, workload-aware single resource scheduler and orchestrator. The proposed resource management may assign and forecast the placement and transfer of effective workloads. The empirical study shows that HeporCloud can efficiently plan and consolidate various types of workloads in terms of energy, performance, and cost. An extended version of the cloudsim simulator was proposed [20] to improve the accuracy and precision of the cloudsim. The evaluation of the extended version proved that it performed better in terms of energy, the performance of resource allocation, and even consolidation in heterogeneous data centers. Authors in [18] presented a consolidation approach that prioritizes the most efficient migration, which could be a VM, a container, or a specific application running within a container. Here authors modeled the

heterogeneity of cloud applications and resources and demonstrated how the consolidation of heterogeneous apps, containers, and virtual machines affects heterogeneous data center performance and energy efficiency. In [19], game-theoretic resource management techniques for multiaccess edge computing were developed. Google’s workload traces were used to evaluate the proposed work. The goal is to develop a resource management technique that is efficient in terms of energy, performance, and cost. [21] Ilias Mavridis and Helen Karatza proposed an approach to combine virtual machines and containers to enhance the isolation and extended functionality of the cloud. The authors highlighted the benefits of running containers on virtual machines, as well as an investigation of how different virtualization approaches and configurations influence the method’s performance. Docker containers were made to run on KVM and XEN virtual machines, and Linux containers were run on Windows Server to see how they performed. By running multiple benchmarks and installing real-world apps as use cases, authors were able to estimate the performance cost caused by the additional virtualization layer of virtual machines. Lastly, the authors investigated several operating systems designed to host containers, as well as techniques for storing persistent data, to see how isolation is implemented on virtual machines and containers.

### 3. Proposed Method

The proposed virtual machine placement technique in this work is a hybrid model using a genetic algorithm and random forest technique. A genetic algorithm is an optimization technique and here it will generate the dataset required for training the random forest algorithm. The training dataset comprises allocation or mapping of virtual machines to physical machines.

**3.1. Genetic Algorithm.** A genetic algorithm (GA) is a metaheuristic technique that generates a global optima solution. Here we use the genetic algorithm to generate a mapping of VM to available physical machines.

The following steps are incorporated in our work:

- (1) Initialize the population.

The population is initially generated in a random manner and all the virtual machines  $VM_i$  ( $i = 1$  to  $n$ ) are mapped to physical machines  $PM_j$  ( $j = 1$  to  $m$ ). Figure 2 shows the sample of the initial population.

- (2) Fitness function: fitness values of the host is derived as

$$\text{Fitness}(i) = \beta * \text{Power Efficiency}, \quad (1)$$

where  $\beta$  is the random constant. The individuals having higher fitness values are used for the reproduction process.

- (3) Selection: the fittest individuals are selected from the population using the tournament selection method. Tournament size is taken as  $N$  (individuals) and they

VM <sub>2</sub>	VM <sub>1</sub>	VM <sub>3</sub>	VM <sub>5</sub>	VM <sub>7</sub>	VM <sub>4</sub>	VM <sub>6</sub>
PM <sub>3</sub>	PM <sub>7</sub>	PM <sub>1</sub>	PM <sub>2</sub>	PM <sub>4</sub>	PM <sub>6</sub>	PM <sub>5</sub>

FIGURE 2: Initial population generation in a random manner.

are chosen randomly from the population. The winner of the tournament is taken for crossover operation.

- (4) Crossover: in this work, a 2-point crossover is used to generate new individuals/offspring from the selected best individuals. Crossovers like single-point crossover, multipoint, and uniform crossover are also available. To improve the diversity, a newly generated individual is added back to the population. An example of a 2-point crossover is given in Figure 3.
- (5) Mutation: mutation operation can be accomplished using swap, move, move and swap, and rebalancing; we have used swap operation to mutate individuals. Two points are identified and their values are swapped to create a new individual. Here swap operation will swap the physical machine allocated to the virtual machine as shown in Figure 4.

The pseudocode of the proposed genetic algorithm is shown in Figure 5. We have used 100 evolutions to get better results. The final result will have a mapping of virtual machines to physical machines.

**3.2. Random Forest Classifier.** Random forest is one of the supervised machine learning techniques used for both regression and classification. It is one of the flexible and easy-to-use algorithms. A random forest is made up of trees, and the more trees there are, the more resilient the random forest is. The random forest creates each decision tree by first selecting at random, at each node, a small set of features to split on and, secondly, by calculating the best split based on these features in the training set. Finally, it gets a prediction from each tree and chooses the best solution either by means of ‘majority voting’ or ‘performance voting’ as expressed in Figure 6.

Module 1 (dataset creation): the dataset is created using the genetic algorithm. The dataset consists of mapping of VM allocation to the best possible physical machine. The procedure to create the dataset using GA is discussed in the previous Section 3.1. Here, the dataset is divided into 2 sets. The first set consists of 80% of the dataset which is used as the training dataset to train the model and the remaining 20% of the dataset is used for testing.

Module 2 (training): consider a training dataset  $T = \{(x_1, y_1), \dots, (x_N, y_N)\}$  consist of  $N$  observations from the random vector  $(x, y)$ . Vector  $x = (x^1, \dots, x^P)$  contains predictors or independent variables and  $y \in C$  where  $C$  is the class label. Using this training set, the developed random forest will be an ensemble of  $B$  trees  $\{(t_1(x), \dots, t_b(x))\}$ . The ensemble results in  $B$  outputs  $\{y_1 = t_1(x), \dots, y_b = t_b(x)\}$ , where  $y_b, b = \{1, 2, \dots, B\}$  is





the prediction for the classified data object by the Bth tree. All the trees outputs are combined to produce the final class y which receives the maximum votes by all the trees.

**3.2.1. Tree Construction.** The training dataset is used in tree building and it follows a top-down approach. We use information gain to identify the attribute that best splits the given training set. “Best” is measured using information gain:

$$\Delta E = - \sum_i \frac{|D_i|}{|D|} E(D_i). \quad (2)$$

It is produced by partitioning the set D of examples into two subsets  $D_i$  according to the given attribute. Here  $E(d)$  is the entropy  $-\sum_{i=1}^N q_i \log_2(q_i)$  with  $q_j$  as the proportion of examples in d belonging to class i and  $|\cdot|$  is the size of the set. The process of selecting the attribute is repeated for each nonterminal node; the process is stopped when the node receives less examples or when it reaches the given depth.

Module 3 (testing): once the training of the model using the training set is over, prediction is performed on the test set. After training, the accuracy is checked using actual values and predicted values. If the obtained accuracy is less than the desired value, some tuning will be done and again the model is trained and tested. This repeats until the desired accuracy is achieved.

**3.2.2. Flow Diagram.** Figure 7 depicts the block diagram of the proposed hybrid technique. The genetic algorithm reads the PlanetLab dataset, which is a real-time dataset. The genetic algorithm goes through all the processes prescribed in Section 3.1 to generate the dataset having mapping of virtual machines to physical machines. This dataset is used as the dataset for a random forest classifier. Random forest classifier splits the dataset into training and testing datasets. Using training data, RF generates a specified number of random trees for each subsamples. The test dataset is used to test the classification process. Finally, majority voting decides the final class label (output value). Here the output value consists of a physical host number which is used to execute the required virtual machines on it.

Once the virtual machines are placed on the physical machines, it starts the execution. After some time, a physical machine may get more VMs to execute, in this situation, the physical machine which gets overloaded. We have used a VM migration technique to handle this kind of scenario. When a PM gets overloaded, some of the virtual machines will be selected and migrated to other PMs having less load, thereby maintaining load balance across all the physical machines in the data center. Interquartile range (IQR) is used to detect overloaded physical machines in the data center, which is one of the available overload detection methods [22]. Next, maximum correlation policy is used to select the VMs to be migrated from the overloaded PM. It selects the VMs having the maximum correlation of the CPU utilization with other VMs.

## 4. Simulation Results

Experimental setup, performance metrics, and experimental results are discussed in this section.

**4.1. Experimental Setup.** We have used the CloudSim 3.0 toolkit simulator to evaluate the proposed algorithm. Cloudsim provides different VM provisioning techniques and virtualized resources. To carry out the experiment, we have taken real workload traces from PlanetLab. PlanetLab is part of the CoMon project, which consists of CPU utilization from more than 1000 virtual machines running on various hosts in more than 500 locations around the world. In our experimental setup, we used 4 different types of virtual machines, Micro, Small, Medium, and Extra-Large instances. 800 heterogeneous hosts are deployed which belong to HP ProLiant G4 and HP ProLiant G5 category. Characteristics of these servers are shown in Table 2. For simulation, 500 hosts and VM vary from 500 to 650 with data center configuration as shown in Table 2. PlanetLab dataset is a log file of a real word data center with incoming traffic, task size, size of VM requested by task with VM configuration like RAM, number processor, and MIPS count.

**4.2. Performance Metrics and Results.** The following metrics are used to evaluate the proposed algorithm and other algorithms.

**4.2.1. Energy Consumption.** It denotes the total energy consumed by all the physical machines (PMs) in the data center. PMs energy consumption is calculated according to the linear cubic power consumption model. In this power model, the power consumption of the physical host grows linearly with an increase in the CPU utilization.

Let us consider the following parameters for the power model:

- (i)  $P_k^{\max}$ : maximum power consumed when the host k is completely utilized
- (ii)  $P_k^{\text{idle}}$ : idle power value of the host k
- (iii)  $U_k$ : current CPU utilization host k
- (iv) T: total number of hosts in the data center

The power consumption of host  $P_k$  can be expressed as

$$P_k = P_k^{\text{idle}} + (P_k^{\max} - P_k^{\text{idle}}) * U_k^3. \quad (3)$$

Our goal is to minimize power consumption of the data center; then, we aim to minimize

$$\sum_{k=1}^T P_k = \sum_{k=1}^T [P_k^{\text{idle}} + (P_k^{\max} - P_k^{\text{idle}}) * U_k^3]. \quad (4)$$

Figure 8 shows the energy consumption of different algorithms. The energy consumption of our proposed algorithm GA-Random Forest (RF) has declined on an average by 17%, 31%, and 39% compared to default GA (genetic algorithm), ACO (ant colony optimization), and

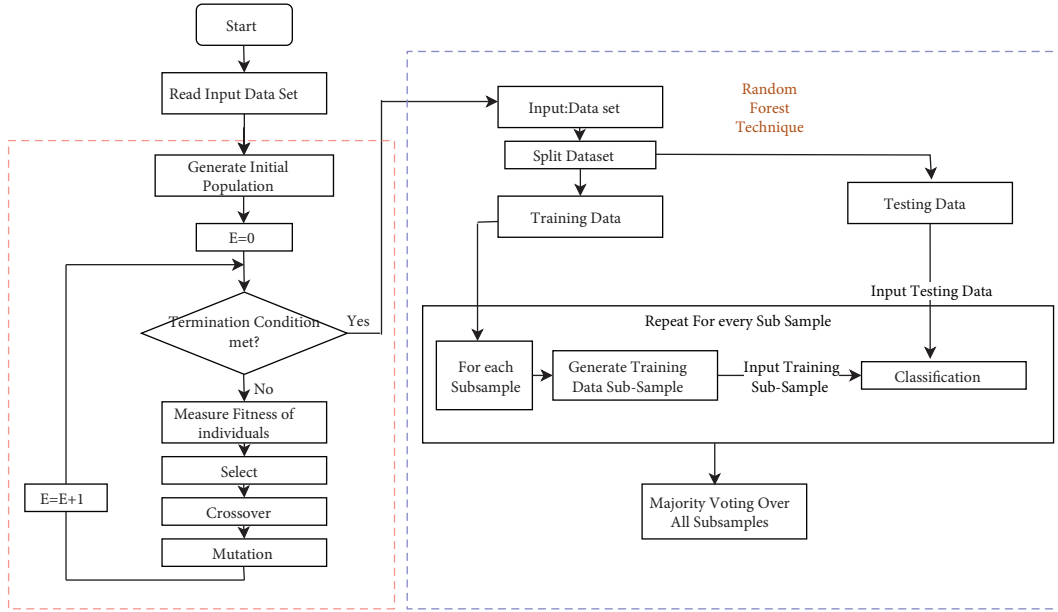


FIGURE 7: Block diagram of the hybrid proposed approach using GA and random forest.

TABLE 2: Characteristics of servers.

Machine type	Description
HP G4	1860 MIPS, 4 GB RAM, 2 GB network bandwidth 1.5 GB storage
HP G5	2660 MIPS, 4 GB RAM, 2 GB network bandwidth, 1.5 GB storage

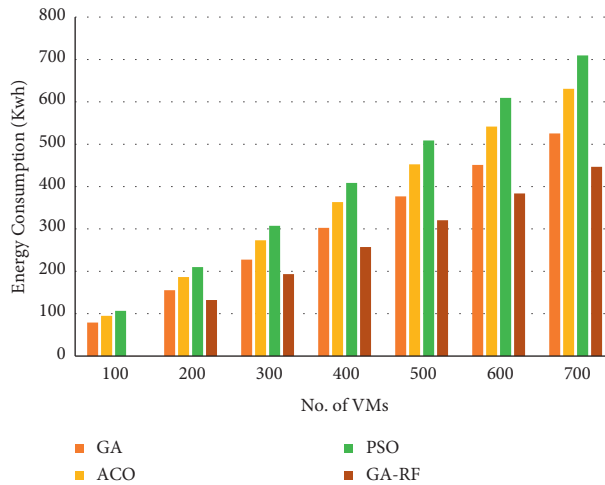


FIGURE 8: Comparative analysis of energy consumption of the proposed and existing approaches.

PSO (particle swarm optimization), respectively, for PlanetLab “20110409/planetlab-1\_amst\_nodes\_planet-lab\_org\_Arizona\_beta and 20110409/pl1\_rcc\_uottawa\_ca\_Google\_highground.

4.2.2. Execution Time. Executing all the user requests in lesser time is indeed an important factor from the cloud provider perspective. So, execution time is taken as one of the key performance factors to evaluate the algorithms.

Let  $T = \{T_1, T_2, \dots, T_N\}$  be the set of tasks.  $|T|$  represents the total number of tasks in the set.  $VM = \{VM_1, VM_2, \dots, VM_M\}$  denotes a set of virtual machines.  $T_E(i, j)$  represents the execution time of task  $T_i$  on virtual machine  $j$ . It is expressed as

$$T_E(i, j) = \frac{t_l(i, j)}{VM_c}, \quad (5)$$

where  $t_l(i, j)$  represents the time required to execute the instruction of length  $l$  on virtual machine  $j$  and  $VM_c$  denotes



the computational ability of virtual machine  $j$  which can be expressed as

$$VM_c = VM_{\text{pesNumb}} * VM_{\text{MIPS}}. \quad (6)$$

So, task  $i$  runtime on virtual machine  $j$  can be represented as

$$\text{Task Run Time}(i, j) = T_E(i, j). \quad (7)$$

The completion time of virtual machine  $j$  is the sum of the task run time of all tasks on that virtual machine.

$$CT\_VM_j = \sum_{i=1}^N \text{Task Run Time}(i, j). \quad (8)$$

Total execution time represents the time required to execute all the virtual machines thereby completing the execution of all the user requests. It can be expressed as

$$\text{execution time} = \sum_{j=1}^M CT\_VM_j. \quad (9)$$

As shown in Figure 9, the average execution time of GA-RF is less by 15%, 29%, and 37% compared to GA, ACO, and PSO, respectively; i.e., GA-RF has a faster execution time.

**4.2.3. Resource Utilization.** To process a user's request based on the resource requirement by the user, the Cloud data center creates various types of VMs. VMP technique aims to place a virtual machine on a suitable physical machine to improve resource utilization. The resource type we have considered is CPU. Suppose there are  $N$  PMs and  $M$  VMs. Let  $PM = \{PM_1, PM_2, \dots, PM_N\}$  denote the set of PMs, where  $PM_i \in PM$ .  $PMC_i$  represents the CPU capacity of  $PM_i$ . Let  $VM = \{VM_1, VM_2, \dots, VM_M\}$  denote a set of virtual machines, where  $VM_j \in VM$ .  $VMC_j$  represents the CPU requirement of  $VM_j$ .

Let  $P_{ij}$  denote whether  $VM_j$  is placed on  $PM_i$ . If  $VM_j$  is placed on  $PM_i$ , then  $P_{ij} = 1$  or else if  $VM_j$  is not placed on  $PM_i$ , then  $P_{ij} = 0$ . Equation (9) represents the requirements of all the virtual machines placed on a physical machine that cannot exceed the resource capacity of the physical machine.

$$\sum_{j=1}^M VMC_j \cdot P_{ij} \leq PMC_i \forall PM_i \in PM. \quad (10)$$

Once the virtual machine types are determined, the resource utilization of all the physical machines needs to be maximized. The CPU utilization of physical machine  $PM_i$  is expressed as

$$UCPU_i = \frac{\sum_{j=1}^M VMC_j \cdot P_{ij}}{PMC_i}. \quad (11)$$

Maximizing the CPU utilization of all the physical machines can be expressed as

$$\text{maximize} \sum_{i=1}^N UCPU_i. \quad (12)$$

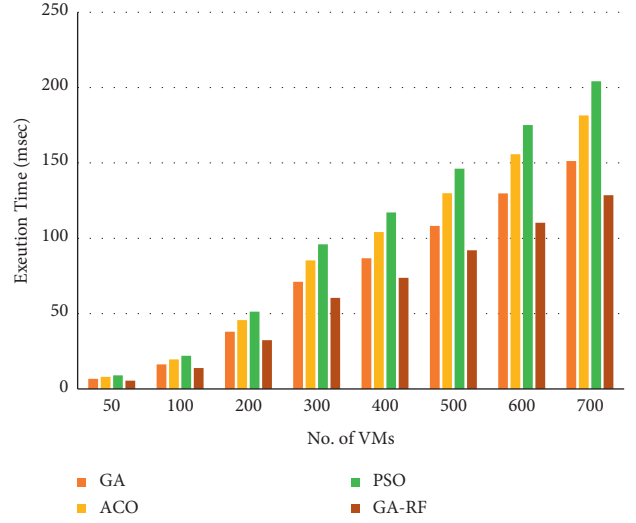


FIGURE 9: A comparative study of execution time for the proposed and existing approaches.

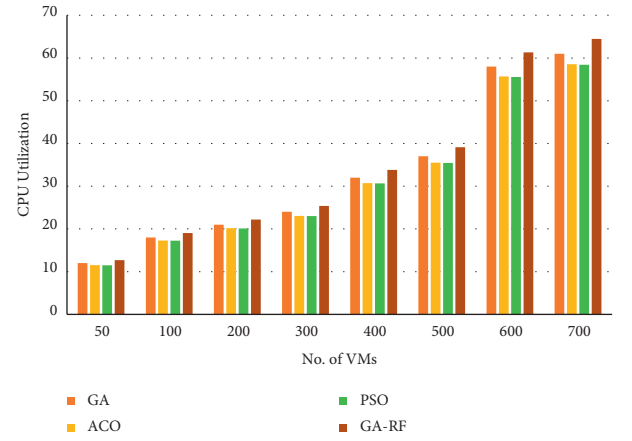


FIGURE 10: Comparative analysis of CPU Utilization using proposed and existing approaches.

Figure 10 shows the average CPU utilization of all the active PMs. Average CPU utilization of GA with the random forest is higher by 6%, 10%, and 11% compared to GA, ACO, and PSO, respectively.

**4.2.4. Average Start Time and Finish Time.** Delivering high performance to the cloud user is becoming an important criterion for the cloud provider. In this regard, parameters like start time and finish time of the user request/task can be considered as major factors. Figure 11 and 12 show the average start time and finish time of different algorithms. As a result, GA-RF can finish the user's requests/tasks in less time compared to other existing algorithms.

In cloud computing, time complexity plays an important role in studying the performance of the algorithm. There exist various studies which show the comparative analysis of the complexity of GA, ACO, and PSO [23,24]. The study shows that GA finds the best global optimal solution at the

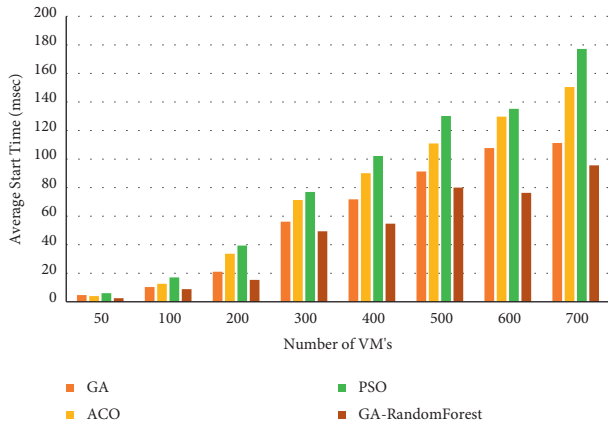


FIGURE 11: Comparison of average start time (msec) of GA-RF and existing approaches.

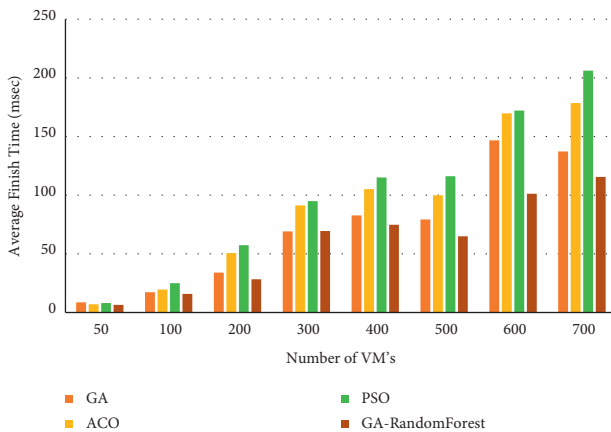


FIGURE 12: Comparison of average finish time (msec) of GA-RF and existing approaches.

cost of high searching time but finds a solution better than ACO and PSO. In order to reduce the searching time of GA, the optimal solution is trained to a random forest model for training and further prediction which gives an optimal solution in constant time. The cost overhead is only one time overhead, that is optimal solution generation using GA and training the model.

## 5. Conclusion

With the development of virtualization technology, designing a multiobjective virtual machine placement technique has become a hot research topic. Our work is two-fold. Firstly, a hybrid approach for VMP using a genetic algorithm and random forest algorithm is proposed to reduce the search time to find the best optimal solution. The genetic algorithm is used to find an optimal solution as a dataset that contains the mapping of the virtual machines to the physical machines. This dataset is used to train the random forest algorithm to place the virtual machines on apt physical machines and also the placement accuracy of RF is evaluated with the test data. Secondly, the trained model is used for load balancing through migrating the virtual machines from

overloaded physical machines to underloaded physical machines. This is accomplished using IQR technique to detect overloaded physical machines and VMs to be migrated are selected using the maximum correlation policy. The result shows that the proposed model provides an energy-efficient placement scheme by reducing power consumption, execution time, average start time, and finish time as compared to the existing approaches. In the future, the model can be tested with various machine learning and deep learning approaches for better solutions and performance study.

## Data Availability

The dataset for simulation is supported by parallel workload.com for real-time analysis.

## Conflicts of Interest

The authors have no conflicts of interest to declare

## Authors' Contributions

Madhusudhan H. S., Satish Kumar T., and Punit Gupta developed the theory and proposed model. Dr. S.M.F. D. Syed Mustapha and Rajan Prasad Tripathi worked on data preparation, ML model for the dataset, and verified the analytical methods. All authors discussed the results and contributed to the final manuscript. All authors confirm sole responsibility for the following: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation. All coauthors have seen and agreed with the contents of the manuscript.

## References

- [1] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud computing and grid computing 360-degree compared," in *Proceedings of the Grid Comput. Environ. Workshop*, pp. 1–10, Austin, TX, USA, November 2008.
- [2] R. Buyya, A. Beloglazov, and J. Abawajy, "Energy-efficient management of data center resources for cloud computing: a vision, architectural elements, and open challenges," *Eprint Arxiv*, vol. 12, no. 4, pp. 6–17, 2010.
- [3] K. Zheng, X. Wang, L. Li, and X. Wang, "Joint power optimization of data center network and servers with correlation analysis," in *Proceedings of the IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pp. 2598–2606, Toronto, ON, Canada, May 2014.
- [4] A. Beloglazov, R. Buyya, Y. C. Lee, and A. Zomaya, "A taxonomy and survey of energy-efficient data centers and cloud computing systems," *Advances in Computers*, vol. 82, pp. 47–111, 2011.
- [5] V. Petrucci, O. Loques, and D. Mossé, "March. A framework for dynamic adaptation of power-aware server clusters," in *Proceedings of the 2009 ACM symposium on Applied Computing*, pp. 1034–1039, Honolulu, Hawaii, USA, March 2009.
- [6] M. Masdari, S. S. Nabavi, and V. Ahmadi, "An overview of virtual machine placement schemes in cloud computing," *Journal of Network and Computer Applications*, vol. 66, pp. 106–127, 2016.

- [7] G. Hadi and M. Pedram, "Energy-efficient virtual machine replication and placement in a cloud computing system," in *Proceedings of the 2012 IEEE Fifth International Conference on Cloud Computing*, pp. 750–757, Honolulu, HI, USA, June 2012.
- [8] J. D. Ullman, "NP-complete scheduling problems," *Journal of Computer and System Sciences*, vol. 10, no. 3, pp. 384–393, 1975.
- [9] D.-M. Zhao, J.-T. Zhou, and K. Li, "An energy-aware algorithm for virtual machine placement in cloud computing," *IEEE Access*, vol. 7, pp. 55659–55668, 2019.
- [10] A. S. Abohamama and E. Hamouda, "A hybrid energy-Aware virtual machine placement algorithm for cloud environments," *Expert Systems with Applications*, vol. 150, Article ID 113306, 2020.
- [11] W. Yao, Y. Shen, and D. Wang, "A weighted PageRank-based algorithm for virtual machine placement in cloud computing," *IEEE Access*, vol. 7, Article ID 176369, 2019.
- [12] J. Zhou, Y. Zhang, L. Sun, S. Zhuang, C. Tang, and J. Sun, "Stochastic virtual machine placement for cloud data centers under resource requirement variations," *IEEE Access*, vol. 7, Article ID 174412, 2019.
- [13] F. Abdessamia, W.-Z. Zhang, and Y.-C. Tian, "Energy-efficiency virtual machine placement based on binary gravitational search algorithm," *Cluster Computing*, vol. 23, no. 3, pp. 1577–1588, 2020.
- [14] X. Zhang, T. Wu, M. Chen et al., "Energy-aware virtual machine allocation for cloud with resource reservation," *Journal of Systems and Software*, vol. 147, pp. 147–161, 2019.
- [15] A. Ghasemi and A. Toroghi Haghighat, "A multi-objective load balancing algorithm for virtual machine placement in cloud data centers based on machine learning," *Computing*, vol. 102, pp. 2049–2072, 2020.
- [16] Y. Moaly and B. A. Youssef, "Hybrid algorithm naive bayesian classifier and random key Cuckoo search for virtual machine consolidation problem," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 1, 2020.
- [17] A. A. Khan, M. Zakarya, I. U. Rahman, R. Khan, and R. Buyya, "HeporCloud: an energy and performance efficient resource orchestrator for hybrid heterogeneous cloud computing environments," *Journal of Network and Computer Applications*, vol. 173, Article ID 102869, 2021.
- [18] A. A. Khan, M. Zakarya, R. Khan, I. U. Rahman, and M. Khan, "An energy, performance efficient resource consolidation scheme for heterogeneous cloud datacenters," *Journal of Network and Computer Applications*, vol. 150, Article ID 102497, 2020.
- [19] M. Zakarya, G. Lee, H. Ali et al., "Epcaware: a game-based, energy, performance and cost efficient resource management technique for multi-access edge computing," *IEEE Transactions on Services Computing* 1 page, 2020.
- [20] M. Zakarya and L. Gillam, "Modelling resource heterogeneities in cloud simulations and quantifying their accuracy," *Simulation Modelling Practice and Theory*, vol. 94, pp. 43–65, 2019.
- [21] I. Mavridis and H. Karatza, "Combining containers and virtual machines to enhance isolation and extend functionality on cloud computing," *Future Generation Computer Systems*, vol. 94, pp. 674–696, 2019.
- [22] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," *Concurrency and Computation: Practice and Experience*, vol. 24, pp. 1397–1420, 2012.
- [23] X. Zhu, N. Li, and Y. Pan, "Optimization performance comparison of three different group intelligence algorithms on a SVM for hyperspectral imagery classification," *Remote Sensing*, vol. 11, no. 6, p. 734, 2019.
- [24] M. K. Patel, M. R. Kabat, and C. R. Tripathy, "A hybrid ACO/PSO based algorithm for QoS multicast routing problem," *Ain Shams Engineering Journal*, vol. 5, no. 1, pp. 113–120, 2014.