EAI Endorsed Transactions
on Cloud Systems
Research Article EAI.EU

An Intelligent Machine Learning and Self Adaptive Resource Allocation Framework for Cloud Computing Environment

Md. Shahidul Hasan[1,*], Balamurugan E[2], Md. Shawkat Akbar Almamun[1] and Sangeetha K[2]

[1]Research Scholar, Texila American University, Guyana
[2]University of Africa, Toru-Orua, Nigeria

Abstract

Resource allocation is one of the major concern in cloud computing model. When several problems exists in rendering a useful resource allocator. In this research , a self adaptive resource allocation frame work based on machine learning is proposed for modelling and analysing the problem of multi-dimensional cloud resource. This novel self-adaptive resource allocation architecture consists of three stages, QoS prediction model, Improved Bat Algorithm (IBA) and Energy Efficient Model (EEM). The first one, the QoS prediction model, which depends on the same scale of system's past events data, can attain a comparable accuracy with regard to QoS prediction. Secondly, an Energy Efficient Model, which is based on Modified Clonal Selection Algorithm (MCSA) is introduced for minimizing the energy depletion. Thirdly, a runtime decision-making algorithm that depends on improved bat algorithm can rapidly decide on a suitable function for resource allocation

Keywords: Cloud Computing, Resource Allocation, QoS prediction model, Improved Bat Algorithm (IBA), Energy Efficient Model (EEM), Modified Clonal Selection Algorithm (MCSA), Enhanced Recurrent Neural Network.

Handling Editor: Sathishkumar Karupusamy, University of Africa, Nigeria

Received on 18 May 2020, accepted on 18 June 2020, published on 25 June 2020

doi: 10.4108/eai.13-7-2018.165501

1. Introduction

Cloud computing platforms provides the consumers the sufficient ease of renting several kinds of computing resources that are Internet-accessible. In several scenarios, these platforms makes use of the recent progress made in virtualization field to show the consumers the resources to be unprocessed hardware devices, like machines, storage block devices, sensors, or links in the network [1]. Particularly, cloud computing has evolved to be a potential scheme for getting the services of IT infrastructures on a short-term on-demand plan. Using cloud computing, enterprises can upscale to marvellous capabilities instantaneously with no need for investment in new framework, training of fresh individuals, or the licensing process of new software. Cloud computing offers significant advantage to small and medium-sized companies who desire the complete outsourcing of their data-center architecture, or huge enterprises who want to achieve maximum load capability without taking the burden of the high expense of constructing massive internal data centre's [2]. In both scenarios, service consumers make use of whatever is required on the Internet and pay just for what they have used. Therefore, the operators of so-known Infrastructure-as-a-Service (IaaS) clouds, such as Amazon EC2, allow their customers assign, have access to, and manage a pool of virtual machines running within their data centre's and just ask to pay for the time duration for which the machines are in assigned state [3]. Hence, workflow control on cloud computing gains huge significance, when several jobs are submitted to cloud environment simultaneously.

Cloud resources could be considered to be any sort of resource, assuming physical or virtual, which are

[*]Corresponding author. Email: rethinbs@gmail.com

requested from the users from the Cloud. These comprise of network resources, memory, computational requirements like CPU time, or software applications also [4]. Generally, these resources are kept in multi-tenant data center having the capability of matching the resources and the amount of tasks that are being carried out at any moment of time so that an increase in the volume of business functionalities results in more amount of resources allocated and a relapse results in less amount of resources being allocated. Cloud is defined to be both the applications provisioned in the form of services on the Internet and the hardware and systems software in the data centre's providing those requested services. As per this, application fulfilment in the form of services (SaaS - Software as a Service) over the Internet and hardware services (IaaS - Infrastructure as a Service) form the two segments of cloud computing strategy [5]. As per the hardware service (utility computing) perspective, some novel concepts exists in cloud, and the foremost being the delusion of myriad computational resources and the capability to rent computational resources on a short-term when required. Since each of the resources used in cloud computing are provisioned in the form of a service, it is called as Service Oriented Architecture (SOA).

Quality of service and reliability has emerged a challenge since businesses and other companies shift from the classical inbuilt IT architecture to SAO, [6]. This is due to the fact that with inbuilt IT architecture, the in-house technical staff can easily monitor them in order to confirm the reliability of the resources and the quality of service. But, owing to the dynamic characteristic of users' requirements, it's not sure that the service providers might be capable of satisfying these demands on the entirety. The service-oriented architecture or service-centric systems bring in another new problem in the fields of quality, accessibility, usability, and robustness of services rendered[7]. Owing to the sophisticated characteristics of service demands from the cloud users and the service providers being unable to meet the requirements of the users completely and satisfy their demands, it is inevitable for both Cloud service provider and users to agree on what is desired by the consumer and what could be offered by the provider. This way, the SLA provisions to accept upon the QoS between an consumer and service Provider and specify the consumer resource needs and service Provider assurances, thereby guaranteeing a consumer that they are getting the services for which they have paid in turn for the Quality of Service in Distributed Systems called as the resource reservation control strategies are in place to ensure the performance and accessibility of a service to a specific extent[8]. QoS yields a degree of guarantee that the resource demands of differently web contents are provided stringent support.

Resource allocation refers to the procedure of allocating the current resources for optimal fulfilment of cloud services economically. It could also be considered to be any technique, which strives to ensure that the requirements of the applications as mentioned in the Service Level Agreement (SLA) are fulfilled perfectly by the service provider's framework [9]. Resource allocation could be defined to be the procedure of merging the cloud provider functions for using and assigning the meagre amount of resources, which may appear infinite to end-users, within the limits of the cloud environment such that the requirements of the cloud application are met in a flexible and straightforward fashion. Resource allocation mechanisms aid the two important parties(end-users and service providers) in cloud computing to accomplish their objectives [10]. Due to the Service-based characteristic of Cloud computing, end-users are worried about quality and dependability, therefore the end-users may desire to have a prediction about the resources needed for the task completion prior to the time estimated. However, this could result of the situation explained as over-provisioning.

Meanwhile, providers try maximizing their profit margins by making use of less amount of resources for each user such that more number of users can be accommodated and more profitability can be gained. This will result in under provisioning. But, the mutual optimum allocation of resources is hard owing to the absence of information sharing happening between them. Also, the ever-rising heterogeneity, changing environment and unreliability of resources present in the node that cannot be fulfilled with classical resource allocation impose much difficult problems for both players. The suggestions from users and providers are combined together for the optimal allocation of resources to avert the challenges of over/under provisioning of resources [11]. From the users' end, application needs and SLA are needed, whereas from the providers side, products, resources available, present condition of the resources and SLA are necessary. The demands from both the concerned parties are combined together for the optimal allocation of resources so that different user needs are met, and resource allocation mechanism must get through the following conditions as defined by,

1. Under provisioning of resources: a condition where an application is allocated less resources necessary to satisfy the QoS demands. 2. Over provisioning of resources: a condition where an application gets immense resources than required for meeting the QoS needs. 3. Contention for Resources: a condition where several applications attempt accessing the same resource simultaneously 4. Resource deficit/insufficient resources: a condition where the resources available are scarce when there are more demands made for these resources. 5. Resource division: this happen in the event of the resources being excessive but cannot be utilized by applications, which require them since they are not transmittable. This technical work fundamentally takes the important resource allocation mechanisms presently being utilized into consideration and then compares them in terms of five parameters: capability to prevent under provisioning and over provisioning, and prevent contention for resources, avert resource insufficiency and resource division.

In this technical work, a self adaptive resource allocation architecture that depends on machine learning for the modelling and analysis of the multi-dimensional cloud resource allocation problem is proposed and this technique makes use of the improved bat algorithm based decision making for allocation of resources. This novel self-adaptive resource allocation architecture consists of three phases called QoS prediction model, improved bat algorithm (IBA) - based runtime decision algorithm and Energy Efficient Model (EEM). Through the learning of a small-scale training set, the novel framework can ensure that the allocation accuracy, and resource usage in the solution proposed are very quiet equivalent to those achieved of the optimum allocation solution.

The following sections are organized as below, Section 2 explains about the different resource allocation schemes. Section 3 introduces the novel self-adaptive resource allocation architecture and Energy Efficient Model. In Section 4, the results and discussion are provided. Section 5concludes the research work and discusses the work intended for the future.

2. Literature Review

This section discusses few of the current methodologies of resource allocation and scheduling techniques. Also the benefits and drawbacks along with its techniques are discussed.

Wei et al [12] presented a Game theory technique, which is helpful in resolving the resource allocation problem. A feasible approximated solution having two steps given below is introduced. In the first step, all the participants solve their optimal problem on their own, without taking the multiplexing of resource allocations into consideration. A Binary Integer Programming approach is introduced for solving the independent optimization. Secondly, an evolutionary technique is developed, which modifies the multiplexed mechanisms of the initial optimum solutions of various users with the reduction in their loss of effectiveness. The algorithms used in the evolutionary approach consider both optimization and fair behavior. It is shown that Nash equilibrium is always present when the resource allocation process offers practical solutions.

An & Lesser et al [13] introduced a distributed negotiation technique, where the agents have a negotiation over both a contract cost and a decommitment penalty, permitting the agents to disengaged from contracts at an expense. This proposed technique is compared experimentally, employing representative conditions and workloads, to both combinatorial auctions and the fixed-price model that Elastic Compute Cloud of Amazon uses, and it is revealed that the negotiation model attains a much better societal goodwill.

Tsai et al [14] designed an improved differential evolution algorithm (IDEA) for the task scheduling optimization and resource allocation in accordance with the discussed expense and time models on cloud computing scenarios. The novel IDEA integrates the Taguchi technique and a differential evolution algorithm (DEA). The DEA exhibits a strong global investigation potential on macro-space and employs less number of control parameters. The flexible reasoning capability of the Taguchi technique helps in using the best participants on micro space to act as probable generations. Hence, the novel IDEA is quite improved and balanced in terms of exploration and exploitation. The novel cost model consists of the processing and receiving expense. Also, the time model merges receiving, processing, and waiting time. The multi-objective optimization technique, which in turn refers to the non-dominated sorting approach, not with normalized single objective technique, is used for finding the Pareto front of overall cost and make span.

Pawar et al [15] studied about an algorithm which took the Preemitable task execution and multiple Service Level Agreement (SLA) parameters like memory, network bandwidth, and necessary CPU time into consideration. The results attained from the experiments reveal that in a condition where contention for resource is high, this algorithm yields better resource usage.

Dr Balamurugan et al [16] suggested a market-based resource allocation approach for the allocation of services to participants with efficiency. The technique facilitate participants (1) to sort a fusion of services for workflows and co allocations and (2) to make reservations for future/present services in a forward/spot market. The analysis reveals that the technique demonstrates good performance in possible configurations.

Pillai et al [17] introduced a resource allocation technique for machines present on the cloud, depending on the concepts of coalition establishment and the uncertainty concept of game theory. Then, the results of using this technique with the available resource allocation techniques, which have been implemented on the cloud are compared. It is revealed that this resource allocation technique using coalition-creation of the machines on the cloud results in not just better usage of resources but also superior request fulfilment.

Abirami et al [18] proposed a Linear Scheduling for Tasks and Resources (LSTR) which carries out tasks and resources scheduling respectively. In this, the fusion of Nimbus and Cumulus services are transferred onto a server node to create the IaaS cloud environment and virtualization combined with LSTR scheduling for the allocation of resources when maximizing the system throughput and resource usage.

Tai et al [19] demonstrated novel burstiness-aware algorithms for balancing the spiked workloads across every computing site, and thereby to boost the system performance totally. In this, a smart load balancer is presented, which helps in leveraging the information of burstiness for forecasting the variations in the consumer requirements and on-the-go changes between the approaches, which are “greedy” (i.e., always choose the best region) and “random” (i.e., choose one randomly) as per the information expected. The results of both simulation and actual experiments depict that this novel load balancer can help in quick adaptiveness to the varying user requirements and the performance is improved by taking an intelligent site option for cloud users under both spikey and consistent workload scenarios.

Morshedlou et al [20] presented a novel proactive resource allocation technique targeting at reducing he SLA infringement effect. Novel technique depending on learning automaton for estimating these properties are also rendered. For the validation of this technique, few numerical simulations are carried out in critical scenarios. The results show that this technique is capable of increasing the customer satisfaction level, which can lead to business gain.

Wang et al [21] designed a Service Level Agreement (SLA) dependent resource provisioning and management among multiple CSPs. Every CSP has a group of strongly heterogeneous servers that support a generic type of application, and each one carries out resource allocation in these servers for processing requests. A central request dispatcher assigns the service requests to multiple servers (members of unique individual CSPs) in the cloud depending on the volume of resources allocated in those servers. Every CSP performs its profit optimization by itself, which specifies the overall revenue acquired through providing service to the clients minus the overall energy cost. The overall revenue is based on the average service request response time as the SLAs mentions. The resource allocation issue existing among different CSPs create a contentious normal-form game, as the revenue (profit) of every CSP relies not just on its individual resource allocation outcomes but also on the performances of the remaining CSPs. The proof for the presence and distinctness of Nash equilibrium in this scheme are given. Every CSP will seek its optimum technique at the Nash equilibrium point employing the convex optimization method. The results of experiments show the efficiency of the game theoretic resource provisioning architecture for the CSPs.

Teng et al [22] presented a new Bayesian Nash Equilibrium Allocation algorithm to resolve the problem of resource administration in cloud computing. This algorithm has the full consideration of multiple criteria like the heterogeneous dissemination of resources, sensible interchange of actions among the cloud users, missing generic information and dynamic sequential allocation. In comparison with earlier researches, the experimental results provided in this research work reveal that although there is ambiguity in the competitors' information, cloud users can get the Nash equilibrium allocation solutions through gambling in sequential stages. In addition, the resource price that the algorithm has assessed will get converged to the optimum cost finally at the gambling series.

It is quite evident from the above review that all the techniques have their own benefits and drawbacks. However, all those techniques focus just on the allocation of resources and do not consider energy usage and QoS demands. Therefore, this technical work highlights on increasing the energy efficiency dependent resource allocation in cloud computing scenarios

3. Proposed technique

In this technical work, a self adaptive resource allocation infrastructure that depends on machine learning for the modelling and analysis of the multi-dimensional cloud resource allocation problem is proposed and this technique makes use of the improved deep learning based decision making algorithm for resource allocation. This novel self-adaptive resource allocation architecture consists of three phases, which are QoS prediction model, improved bat algorithm (IBA) - based runtime decision algorithm and Energy Efficient Model (EEM).

- The first one, the QoS prediction model depending on the same scale of system's past event data, can attain a higher QoS prediction accuracy using the enhanced recurrent neural network (ERNN) and modular neural network (MNN).
- Secondly, an Energy Efficient Model depending on Modified Clonal Selection Algorithm (MCSA) is introduced for decreasing the energy usage.
- Thirdly, a runtime decision-making algorithm that depends on improved bat algorithm can rapidly decide on a suitable function for resource allocation.

3.1. System model

The quality offered by cloud-based software service differs with time due to the variations in its runtime environment. Generally, the environmental variations can be split into extrinsic or intrinsic variations, as per the factors that initiate [23]. The external factors primarily indicates the workloads (L), the changes of which are provided in this technical work, and the internal ones to the allocated resources (VM). Energy usage of a data center results from several sources like CPU usage, memory consumption, power supply devices, disk storage boxes and cooling systems. The energy consumption can be expressed in equation (1).

$$Energy\ Consumption = \int Power\big(u(t)\big)dt \qquad (1)$$

Where u(t) refers to the CPU utilization. The power usage is studied and can be defined using a linear association between the power usage and CPU usage.

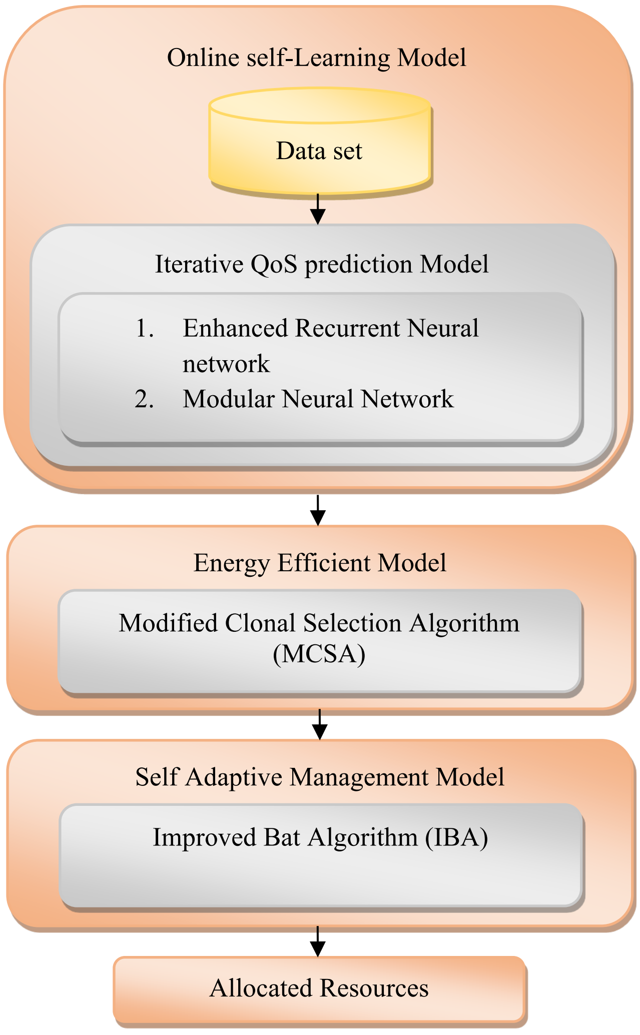

Online self-Learning Model
Data set
Iterative QoS prediction Model
1. Enhanced Recurrent Neural network
2. Modular Neural Network
Energy Efficient Model
Modified Clonal Selection Algorithm (MCSA)
Self Adaptive Management Model
Improved Bat Algorithm (IBA)
Allocated Resources

Figure 1. Depicts the overall process of the proposed self-adaptive resource allocation framework

Hence, power usage is directly proportional to CPU utilization as given in equation (2).

$$Power(u) = q.P_{max} + (1-q)P_{max}.u \qquad (2)$$

Where q indicates the fraction of energy depleted in the inactive server, P_{max}refers to the maximum power used by the completely used server and u refers to the CPU usage.

During the allocation of resources for cloud-based software services, self-adaptive systems have to create a balance between the quality of services (QoS) and the price of resources (Cost) as per the current targets. The fitness function is a generic technique used for evaluating the values (Fitness) of the current targets, which is as formally expressed in the Formula (3) given below:

$$Fitness = r_1 * \frac{1}{QoS} + r_2 * Cost + r_3 * Energy \qquad (3)$$

Where r1, r2and r3 stands for the parameters, the QoS weights, resource cost and Resource Energy are stated. In addition, QoS, Cost and Energy in the formula given above are the QoS value that has to be computed and the resource price to be defined in the subsequent paragraphs correspondingly. For real applications, a resource allocation plan that is devised better would provide a fitness function that has a smaller value. Hence, the assessment values of every resource allocation plans possible under the present workload can be predicted, and highly resourceful decisions can be made. As given in Formula (3), fitness relies on two aspects. The first one involves the resource price, which chiefly is from leased cost (CostL) and discontinued cost (CostD) of virtual machines, as expressed in Formula (4).

$$Cost = CostL + CostD \qquad (4)$$

As it can be seen in the formula, Cost L refers to the total cost of each of the allocated virtual machines, and CostD indicates the overall penalty value incurred in shutting down the virtual machines, called as discontinued price. Adjusting often will lead to unwanted expenses which consists of the resource expense taken for computation and extra system cost. The discontinued price in the formula targets at reducing the unwanted expenses and keep up the stability offered by the software services by preventing the unwanted shut down of the virtual machines allocated.

The QoS value is the second element, which can be computed by the Service Level Agreements (SLA) contract, is given in Formula (5) below:

$$Q_{actual} = SLA(RT, DH, \dots,), \qquad (5)$$

Where RT and DH indicate response time and data throughput correspondingly. Even though there are several parameters, which influence the QoS value, the response time and data throughput is focused on, and they are based on the kind of software services, whether it is either IO-intensive or CPU-intensive. It is to be noted that, the novel iterative QoS prediction model could also be regarded to be an sophisticated system having skills in resource allocation process. This model makes use of the workload information, the assigned resources, the present QoS value, and the present resource adjustment action in the form of inputs, and the QoS value prediction after the completion of the respective resource adjustment operation in the form of an output.

3.2. Adaptive resource allocation technique based on feedback loop

This section presents an adaptive resource allocation architecture that is dependent on feedback loop, and it can get a sensible resource allocation scheme by means of feedback and iterations. Fig. 2 shows the primary architecture of the novel self-adaptive resource allocation technique. The core concept here is to integrate a method known as feedback control loop used in control theory with the machine learning technique, and later facilitating this algorithm to employ feedbacks coordinating with the training data and hence mitigating the burden of learning from inadequate previous events data [23][29]. Consequently, this yields a sufficiently reasonable QoS and helps in the accuracy improvement of machine learning techniques. This architecture primarily comprises of three modules which include Online Self learning module, Energy Efficient module and Self-adaptive Management module. Cloud Resource module renders APIs for the process of monitoring and controlling the cloud resource' status in real time. Using these APIs, the monitoring of cloud resources can be done and then modified as required. Online Self-learning module makes use of previous event data for training the iterative QoS prediction model through two machine learning technique in the form of Enhanced Recurrent Neural Network (ERNN) and Modular Neural Network (MNN). Self-adaptive Management module yields the resource adjustment mechanism that depends on iterative QoS prediction model for automated decision-making process.

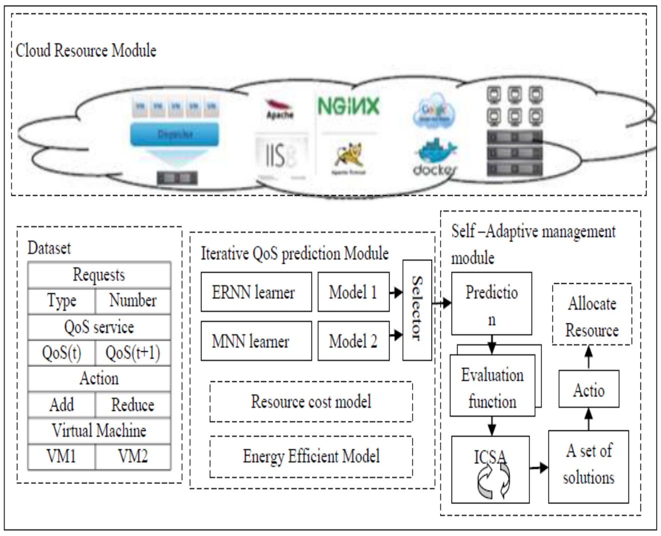


Figure 2. The comprehensive process of the novel self-adaptive resource allocation technique.

3.3. Iterative QoS prediction model

This subsection explains about the Online Self-learning module, especially on the means of usingpast event data for training an iterative QoS model that depends on machine learning. This model targets at the prediction of the QoS value getting the input information such as workload, assigned resources, present QoS value and resource adjustment functions. Generally, it can be expressed as in Formula (6):

$$QoSt + 1 \ = \ F \ (L, VMs, QoSt \ , Action), \qquad (6)$$

where the inputs L refers to the present on-going workload, VMs indicates the number of allocated virtual machines, QoSt stands for the quality of service where L and VMs represent the workload and virtual machine resources, Action refers to the present adjustment made to the assigned virtual machines, such as addition and deletion of virtual machines. The output QoSt+1 refers to the quality of service after the Action. In this, the dataset of past event data is used for training the iterative QoS model, as given in Table 1. The workload is denoted as (xi,0 xi,1 . . . xi,m), where xi,0 refers to the amount of workload and xi,j(1 ≤ j ≤ m) indicates the ratio of various kinds of tasks present in the workload. The allocated resources are indicated as (xi,m+1 xi,m+2 . . . xi,m+n), where xi,m+p stands for the number of virtual machine of pth kind. The adjustment action of assigned resources is indicated as (xi,m+n+1 xi,m+n+2 . . . xi,m+n+w), where xi,m+n+p specifies the number of virtual machine of pth kinds, which requires adjustment. The QoS value in the present state is denoted as (xi,m+n+w+1) and the QoS value after the adjustment is indicated as (yi).

Table 1. Past event data used for training the iterative QoS model

L	VMs	$Action$	QoS_t	QoS_{t+1}
$x_{0,0}\ x_{0,1} \ldots x_{0,m}$	$x_{0,m+1}\ x_{0,m+2} \ldots x_{0,m+n}$	$x_{0,m+n+1}\ x_{0,m+n+2} \ldots x_{0,m+n+w}$	$x_{0,m+n+w+1}$	y_0
$x_{1,0}\ x_{1,1} \ldots x_{1,m}$	$x_{1,m+1}\ x_{1,m+2} \ldots x_{1,m+n}$	$x_{1,m+n+1}\ x_{1,m+n+2} \ldots x_{1,m+n+w}$	$x_{1,m+n+w+1}$	y_1
...	...	...	...	...
$x_{u,0}\ x_{u,1} \ldots x_{u,m}$	$x_{u,m+1}\ x_{u,m+2} \ldots x_{u,m+n}$	$x_{u,m+n+1}\ x_{u,m+n+2} \ldots x_{u,m+n+w}$	$x_{u,m+n+w+1}$	y_u

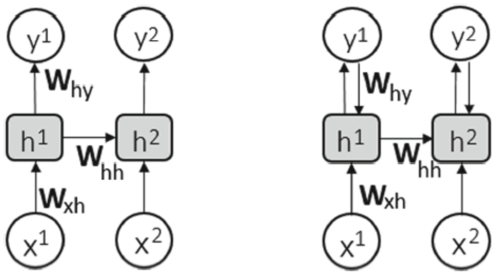
Here ERNN and MNN is used for training the iterative QoS model, which is in fact used for identifying the relationship between input and output.

a) Enhanced Recurrent Neural Network (ERNN)

Many times, Neural network models are limited by the less amount of labelled instances and move to modern frameworks and features. Recurrent Neural Networks (RNNs) is a popular scheme used in natural language processing. Even though, practically, RNNs are affected by the problem of vanishing/exploding gradient, and their small structure still provides efficacy and helps reduce the over fitting problem [24][30]. In this technical work, a recurrent neural network is suggested to be built with multiple semantically heterogeneous embedding's inside a self-training architecture. In this, it is shown that through the propagation of the weight matrix of the earlier labels, the performance of RNNs can be improved when maintaining the number of parameters in RNNs unmodified and adding just one step more for analysis purposes. Consequently, the models are still small and resourceful compared to other models that have sophisticated memory gates.

- Weight Matrix Propagation based Recurrent Neural Network (WMP-RNN)

An WMP-RNN is built by introducing one hidden layer in an artificial neural network over the passage of time. The hidden layer is found linked to itself using weights kept recurrently. One primary benefit of RNNs is the capability of learning temporal specifications from data with repetitive hidden layers.



(i) RNNs (ii) WMP-RNN

Figure 3. RNNs and WMP-RNN

a) Graphical Structure

Fig. 3a illustrates a recurrent neural network where W*xh* refers to the weight matrix of connections existing between input and hidden units; W*hy* stands for the weight matrix of connections between hidden and output units; W*hh* indicates the weight matrix of repetitive connections; b, c stand for the biases of output layers and hidden layers correspondingly. The predictive propagation recurrent neural network is identical to a recurrent neural network, with the exception being that the connections between hidden layer and output layer are bidirectional, as illustrated in Fig. 3b. In this, the direction upwards is utilized for prediction purposes whereas the direction downwards is used for transferring that prediction to the next subsequent time step. The sections that follow will explain the way in which inference and learning are performed in this model.

b) Inference

Inference in WMP-RNNs at every time step comprises of the computation of two states, which include the status of the output unit in the present time step for prediction, and the status of hidden unit, which would be utilized for propagating the present and earlier information, inclusive of the prediction, made for the next step. The steps of the Algorithm 1are given as below. Here, *f* refers to activation function and *s* indicates the softmax function.

$$s(X) = \frac{exp\ (x_i)}{\sum_{t'} exp\ (x_i)} \tag{7}$$

c) Learning

Here the RNNs are trained with every sample at a time instant. Especially, for every training pair $X^{1:T}, Y^{1:T}$ and $\tilde{Y}^{1:T}$ is inferred with the Algorithm 1 and the parameters are updated through the cross entropy minimization:

$$C = \frac{1}{T}\sum_{t=1}^{T}\sum_{l=1}^{L}[y_l^t log y_l^{-t} + (1 - y_l^t)\log(1 - y_l^{-t})] \tag{8}$$

where *L* refers to the number of classes.

Algorithm 1. Weight Matrix propagation Recurrent Neural Network

```
Input :$X^{1:T}$
Output: $o^{1:T}$
for t=1:T do
$\tilde{h}^t = f({x^t}^T W_{xh} + {h^{t-1}}^T W_{hh} + c^T)$
$\tilde{y}^t = s({\tilde{h}^t}^T W_{hy} + b^T)$
$o^t = argmax_k y_k^{-t}$
$h^t = f({x^t}^T W + {y^{-t}}^T W_{hy}^T + {h^{t-1}}^T W_{hh} + c^T)$
end
```

as WMP-RNN*p*. As an alternate, with the true labels, another technique is used for inferring ˜**y**1:*T* by substituting ˜**y***t* in the final expression in Algorithm 1 with **y***t*. Then the state of hidden unit at time *t* goes to be$h^t = g\left({x^t}^T W + {y^t}^T W_{hy}^T + {h^{t-1}}^T W_{hh} + c^T\right)$.A WMP-RNN is represented with this kind of inference for learning to be WMP-RNN*g*.

d) Modular Neural Network (MNN)

The popularly employed artificial neural networks exhibit a monolithic form. Many models operate on entirely connected networks or layers (Hopeld or multilayer Perceptron). The performance of these networks is quite good on a compact input space. But, there is an increase in complexity and the performance reduces quickly with an increase in the input dimension [25]. It is observed in supervised classification applications that the more accuracy the developed learning approach yields, the slower is its performance during the learning process. But, few applications exists where online learning and adaptation holds importance. There are multiple efforts made towards accelerating the learning in conventional NN paradigms (e.g., "smart" initialization, pruning during learning, and differences on gradient search algorithms depending on advanced learning rate, momentum adaptation, heuristic rules, or progressive optimization approaches. But, these methods always present extra parameters which are generally specified in accordance with some non-strict rules-of thumb. In case, those parameters are not selected right, they can, in fact, decrease the convergence speed.

One more technique for improving NN speed recommends the hardware realization of the learning procedure. But, still there are different technical problems for massive non modular NNs. The necessity for high-speed learning inspires MNN, where small modules (trained concurrently) would improve the convergence speed. A modular structure also helps increase the speed of learning by minimizing the impact of contradicting training information. Crosstalk deteriorates the network's ability to operate perfectly for a set of patterns, and therefore, leads to a delay in developing a superior solution.

A modular neural network can be considered to be a group of monolithic neural networks, which tackle with a section of a problem, and thereafter each of their outputs are integrated using an integration layer to make a globalized solution to the entire problem [26]. The primary concept is that a complicated problem can be partitioned into small sub problems, which can be resolved using ordinary neural networks and after this, the overall solution will become a fusion of the outputs obtained from the ordinary monolithic neural networks.

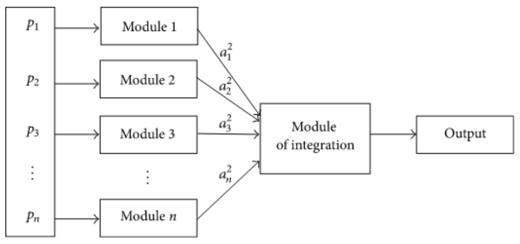


Figure 4. The structure of the MNN

In the novel scheme, each one of the MNN Modules considers the result of the resource cost model as its input. Each module is a two-unit Multilayer Perceptron and the output of the second unit of the ANN is received as a_i^2 $(for\ i \in \{1, \dots\dots, n\})$, where n refers o the maximal number of modules as illustrated in figure 4.

The output is computed as per the equation below, which, in fact, carries out a weighted combination of the module outputs,

$$o = \frac{\sum_{i=1}^{n} a_i^2 g_i k_i}{\sum_{i=1}^{n} g_i} \tag{9}$$

where a_i^2 refers to the module output; $i \in [1,2, \dots . n]$; $g$indicates the average deviation of the output values of the module i; k_i stands for the coefficient of presence of module i. This coefficient of presence of each module indicates the existence/miss of the corresponding module, with regard to the necessity of that module in that specific scenario.

For the sake of examples and to simplify the computations, it is chosen to minimize the MNN structure and 2 modules from the structure in Figure 5 are used. $k_1 = 1, k_2 = 1, k_3 = 0, \dots . k_{n=0}$.

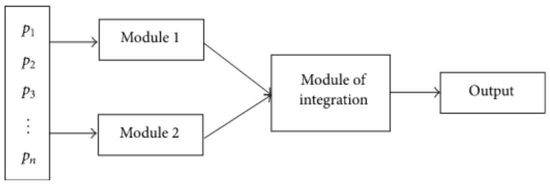


Figure 5. The MNN structure with 2 modules

The first module uses all the individual inputs (weak dissonance, dissonance, and strong dissonance). The second module considers the inputs, which exhibit high negative consonance, negative consonance, and weak negative consonance, and weak positive consonance, positive consonance, and high positive consonance. In the next module, few inputs can be reduced in case they exhibit highly strong positive consonance. On event of this, one of the inputs is removed. In case that they exhibit high negative consonance, that can also eliminate few inputs. There are other possible structures for the modular neural network, based on the way in which the inputs are chosen.

3.4. Energy Efficient Model using Modified Clonal Selection Algorithm (MCSA)

This section discusses the resource allocation along with energy-efficient approach employing Modified Clonal Selection Algorithm. The variables of the techniques which consists of the novel system framework, Energy- Efficient based modified clonal selection algorithm (EE-MCSA) along with their supposition and the respective models are described. The necessity for the management of different applications in Cloud data center, lead to the respective problem of on-demand allocation of resources and assignment of a variety of workloads that change. Figure 6 shows the energy-efficient model of resource allocation for Cloud data center. It depicts the way in which the user requests are propagated by the broker to the Cloud and at last to the datacenter. The request of the users is given to the Cloud broker at first, and after this, the broker will retrieve the result to the end-user depending on their degree of need and performance control of the available datacenters that the broker has subscribed. Once the broker's request get to the datacenter, the cloud manager will now go through the request and also takes a decision after making a comparison of the request on the VM manager module and resource scheduler module. These two modules will try to find the features of the resource consisting of reservation, on demand, availability and allocation. The Cloud manager approves any request after deciding the system availability. But, this technique has not resolved the problem regarding the ineffectiveness of the resource allocation policy, which led to in efficient resource usage and energy handling in cloud datacenters. Nonetheless, this technique has considered the abovementioned problem and presents a novel optimization approach that uses Modified Clonal Selection algorithm for resolving the issue.

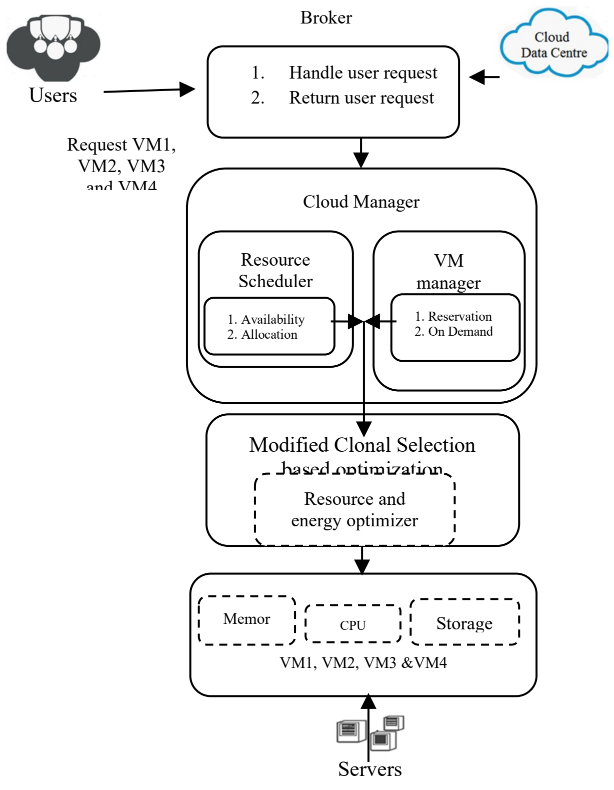


Figure 6. Energy-Efficient Model of Resource allocation

With the aim of generalizing the statement, it is assumed is that there exists a group of tasks and every task consists of multiple subtasks with precedence conditions. Every subtask is permitted to get processed on any free resource provided. A cloud resource yields a particular degree of capacity (e.g., CPU, memory, network, storage). A subtask gets processed on one resource at one time instant, and there is continuous availability of resources given. During the process of resource allocation in a cloud computing environment, the usage of MCSA for the generic process is as given:

Inputs: Let $R = (R1, R2, \ldots, Rj, \ldots, Rm)$ refer to the group of m free resources that must process n individual tasks represented by the set $T = (T1, T2, \ldots, Ti, \ldots, Tn)$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$. All of the resources have no correlation and are concurrent, and every task Ti can be executed on any subset $Rj \in R$ of available resources.

Outputs: The output got is an useful and effective resource allocation mechanism, which includes scheduling the tasks to suitable resources.

Objectives: The primary goal is to boost the energy efficiency of the data center so that an energy-efficient schedule is achieved.

As several practical design or decision making problems consists of parallel optimization of several goals, a resource allocation optimization model, which will completely merge the two aspects of energy-efficient optimization is designed.

1) Clonal Selection algorithm (CSA)

The Clonal Selection algorithm is a popular prototype for the immune system's responsive action to infection in human body. Clonal Selection Algorithms (CSA) belongs a specific group of Immune algorithms (IA), influenced by the Clonal Selection Principle. In order to increase the Algorithm's performance, this CSA has been refined through the implementation of concepts known as Adaptive mutation factor [27]. This novel function has a stable Factor maintained throughout the entire process, which is adaptively dependent on the attraction of antibodies.

A CLONALG is basically a population dependent Meta heuristic algorithm whose search strength depends on its mutation operator. This novel work focuses primarily on these mutation operators during the design of algorithms that perform better. The Clonal Selection Algorithm (CSA) generates individuals having strong affinities and chooses enhanced maturate progenies. This mechanism advises that the algorithm carries out a greedy search, where individual members will get optimized locally and the new off springs render a elaborate hunt of the search space. This feature increases the suitability of CSA for resolving optimization tasks.

2) Adaptive mutation factor based Clonal selection algorithm

Similar to CLONALG, the superior antibody is reproduced as per the cloning rate (β) and then the clones of best antibodies are produced. During this process, mutation is performed on the few best antibodies also in addition to the worst antibodies. Small numbers of best antibodies, which are cloned(copied)are considered and are then mutated in addition to the worst antibodies. Once the affinity difference between the worst antibody and the best antibody gets higher, the Worst antibodies attempt going and following the best like it happens in any other Evolutionary mechanisms. However, when their difference in affinity is quite less, all of the worst and best antibodies exist in the same enclosing region of the search space or to state otherwise, the worst antibody has arrived nearer to the region of the best antibody.

During this time, improving further may not happen quickly. This can be improved by taking some extra antibodies for mutation purposes. A possibility of a better convergence can be achieved by adaptively increasing the ratio of best cloned antibodies for mutation as described. The mutation factor can be increased adaptively depending on the affinity measure. This mutation factor is in proportion to the affinity of the antibodies i.e., in case, the ratio of the affinities between the best and worst antibody is less compared to the threshold value(μ), then the mutation factor must be increased. In this technical work, the adaptive mutation factor based CSA is introduced to reduce the premature convergence problem.

The pseudo code given below is added to the affinity function for preventing the premature convergence problem of the fundamental CLONALG:

$$If\ \left(\frac{aff[ab_b]}{aff[ab_w]}\right) < \mu \qquad (10)$$

$$\delta = \delta \times \rho \qquad (11)$$

Where ρrefers to a parameter, modified dynamically as given:

$$\rho = \left(\frac{iter}{maxiter}\right) * \alpha \qquad (12)$$

Where:

μ refers to the threshold value,

ab_b indicates the best antibody in the Antibody population.

ab_w stands for the the Worst antibody in the Antibody population.

$\alpha$is a constant multiplier that is based on the problem kind.

δ stands for the Adaptive Mutation Factor.

Iter refers to current Iteration.

Maxiter indicates for Maximum number of Iterations.

Aff(.) refers to a function employed for computing the affinity of the antibody.

Algorithm 2. Modified clonal Selection algorithm (MCSA)

```
Input: No. of. user requests and mutation probabilities $P_m$.
Output: the individual having minimal objective function value

    1. Randomly produce an antibody population A(0)
    2. Compute the affinity of the initial population A(k)
    3. Choose half of the antibodies having a higher affinity like the population $A_l$(k)
    4. Clone all individuals in $A_l$(k) to produce the population B(k), and the clonal number is in proportion to their affinity
    5. Carry out mutation from the population $A_l$(k) to produce the population C(k)
    6. Assess individual affinity after adaptive affinity function based mutation employing eq. (16,17). Re compute affinity values for new mutated antibodies.
    7. Perform selection operation from the population C(k) and obtain the next generation population.
    8. Repeat the steps 4-7 until the stopping criteria are met.
```

In the case of an energy efficient resource allocation, when there is a new request made for resource, the system will execute the MCSA to change the total allocation of the resources. Prior to getting the best solution with the MCSA, first the mapping correlations between resources and tasks is changed into a binary code in the form of a set of initial population $X(0)$. An individual is represented as $X_i^G = (X_{i1}^G, X_{i2}^G, \dots\dots.. X_{ip}^G,)$, where G represents the present generation, $i = 1, 2, \dots, s$, and sstands for the population size.

Every individual (antibody) indicates that a binary string of bits represents a candidate solution. The bit string length is appropriately chosen by the user to get a correct solution for the problem. Every gene in the chromosome takes a value of either 0 or 1. When the initial population is produced, the affinity value of every individual is assessed and then stored for operation further. The MCSA is used in resource allocation to tackle with the optimization problem, and the affinity function is developed according to the energy efficiency. The affinity function can be specified as below:

$$aff(x) = e^{minE_i + minMs} \qquad (13)$$

1) Minimizing Energy Consumption

In mathematical terms, the resource allocation model can be expressed as Eq. (14).

$$\sum_{x=1}^{Z,n} (PC_i + PD_i + PM_i) \times T_i \rightarrow U_i^j \qquad (14)$$

The cloud datacenter allocates m number of virtual resources $V = (V_1, V_2, V_3, \ldots. V_m)$ onto n free physical resources of the datacenter having the specified resource capacity, which is $P = (P_{c1}, P_{c2}, P_{c3}, \ldots. P_{cn}), P = (P_{d1}, P_{d2}, P_{d3}, \ldots. P_{dn})$ and $P = (P_{m1}, P_{m2}, P_{m3}, \ldots. P_{mn})$ to the users of the Cloud services $U = (U_1, U_2, U_3, \ldots. U_n)$so that there is a fitness maximization of j objective function $F = (F_1, F_2, F_3, \ldots. F_z)$.

The first objective of this technical work is to reduce the energy depleted by datacenter depending on the capacity needed for executing the cloudlets and also the user requests sent to Cloud datacenter environment. In simple terms, it is defined as the overall electricity power utilized for running the host or PMs in datacenters as defined in Eq. (15)

$$Energy\ Cons = \sum_{source}{i} \int_{str_{time}}^{fns_{time}} E_i(F, T) \qquad (15)$$

The energy consumed in a given resource i at a time T with placement F.Ei indicates the energy depleted by the resource i from its start time to finish time of usage.

2) Maximization of Resource usage

Resource Utilization Model: This is helpful in measuring the original amount of resources that the datacenters. Hence, the Cloud manager needs to seek means of effective allocation of the users request out of the available set of datacenter resources. The resource usage of the physical host can be defined by Eq. (16).

$$Resource\ Utiliz = \frac{\sum_{resource} i\ execution_{time}}{makespan\ or\ maxtask^i(i\ execution_{time})} \qquad (16)$$

Where make span refers to the highest time of completion after the resources are assigned to the end-users and execution time indicates the difference between the start time and end time of the requests.

3.5. On-line decision-making based on Improved Bat Algorithm (IBA)

This subsection explains about the Self-adaptive Management module, primarily on the ways of using the improved IBA to find a novel objective resource allocation plan depending on the QoS prediction model. In the form of an enhancement to the conventional BA, the modified IBA substitutes the movement of the original bat by the updating strategy inclusive of the new principle of distribution function, which will be described in the section that follows.

3.5.1. Fundamental concept of Bat Algorithm

The Bat Algorithm (BA) is a novel swarm intelligence optimization algorithm, which resemble the foraging characteristic of bats. Its is based on the principle of using the advanced echolocation skills of bats[28]. Echolocation is a type of sonar, whose operation is such that the bat (the important small bat) lets out a big and briefly pulsed sound. On the sound hitting an object, the echo will get back to their ears in a shorter duration of time; bats find and identify the location of the prey in this manner. With the aim of simulating the foraging process of bats, the biological strategy of the bat algorithm is explained as below. All the bats use echolocation for detecting the distances, and the technique utilized for identifying the hurdles and it is hard to understand the prey. Depending on the variable length waves λ, loudness A_0, and constant frequency f_{min}, the bat looks out for the prey with the velocity V_i at the position X_i. The bat changes the pulse wavelength as per the distance between itself and the prey. Meanwhile, when the prey is close, the frequency of the transmission $r \in (0, 1)$ will also be modified accordingly. The loudness varies between the maximum value $A_0$and the minimum value A_{min} during the search process.

The design of BA exploits the available algorithms and other exciting features, influenced by the remarkable characteristic of miniature bat echolocation. Depending on these presumptions, this algorithm produces a set of solutions randomly and thereafter makes use of the loop search for finding the optimum solution. The local search is used during this time period, which implies that near the optimal solution, the local solution is produced using random flight and a global optimal solution is produced. In the case of bats, in case their foraging space is present in the d-dimension at the $t - 1$ moment, theposition of bat i is X_i^{t-1}, the flight velocity is V_i^{t-1},, and the present global optimal position is X^*. Therefore, the position and flight velocity of bat i at time t can be defined by$f_i = f_{min} + (f_{max} - f_{min})\beta$, (17)

$$V_i^t = V_i^{t-1} + (X_i^{t-1} - X^*)f_i, \qquad (18)$$
$$X_i^t = X_i^{t-1} + V_i^t, \qquad (19)$$

where f_{min}refers to the minimum frequency of the sound waves that the bat generates and f_{max} is the maximum frequency. β is an evenly distributed random number positioned in the scope [0, 1]. During the initial setting process, the frequency of the sound waves emitted from the bat is evenly distributed in $[f_{min}, f_{max}]$ range. The respective frequency is computed as per equation (1), and then the local search is performed as per the equations (2) and (3). The bat performs a random walk as per the optimal solution, and the new solution is expressed by the equation as follows:

$$X_{new} = X_{old} + \varepsilon\bar{A}(t), \qquad (20)$$

where ε refers to a random number positioned in [− 1, 1], X_{old}indicates the solution chosen from the present optimal solutions randomly, and $\bar{A}(t)$indicates the average loudness produced by the bat if the number of iterations is t. By evaluating the loudness A_i and pulse emission rate r_i of the bat, it is noticed that the update rule can be given as below. In case, the bat knows that there is a prey, it will decrease the response of its pulsed emission and raise its pulse emission rate. The loudness A_i and rate r_i of the bat launch pulse are computed using the equations below,

$$r_i^{A_i^{t+1}=\alpha A_i^t} \tag{21}$$

$$r_i^{t+1} = r_i^t[1 + \exp yt], \tag{22}$$

where r_i^0refers to the initial rate and A_i^0indicates the initial loudness, which are all randomly selected. α and y refer to the constants (0 < α < 1, y > 0).

3.5.2. Distribution Factor based Bat Algorithm (DFBA)

In the case of the bat algorithm, as the individual is moving towards the optimal solution during the optimization period, generally there is an issue with premature convergence of the bats, therefore it is hard to get better optimization outcomes.

Algorithm 3. Pseudocode of improved bat algorithm

```
Input: No.VMs $x_i = (x_{i1,......}x_{iD})^T$ for $i = 1, ... ... . N_p$.
Output: Allocated Resources
init_bat(); eval=evaluate the new population;
$f_{min}$= find_best _solution ($x_{best}$); {initialization}
while termination_condition_not_meet do
    for i=1 to Np do
        y = generate_new_solution ($x_i$);
      if rand (0,1) > r then
        y = improve_the_best_solution ($x_i$);
      end if {local search}
      if $f_{new}$= evaluate _new_solution (y);
              $x_i = y; f_i = f_{new}$
      end if
$f_{min}$= find the best soluiton ($x_{best}$);
    end for
  end while
```

With the aim of preventing the early convergence problem of bat algorithm and force the individual get converged to the global optimal solution rapidly, a distribution factor that depends on Bat algorithm (DFBA) is introduced for the implementation, which not just helps maintain the versatility of the population but also increases the convergence efficacy. The novel DFBA can be achieved using the equations given below,

$$k = \frac{2}{|2-d-\sqrt{d-4d}|} \tag{23}$$

$$V_i^t = V_i^{t-1} + (X_i^{t-1} - X_*)f_i.*_{k,} \tag{24}$$

where d refers to a constant, and the value of the d depends on the experiment.

5. Results and Discussion

This technical work carried out evaluation on Cloud Stack, which is capable of accommodating three kinds of virtual machines in this experiment, which include the sizes of small, medium and large, correspondingly. The objectives of the experiment are to (1) evaluate if the iterative QoS model trained from past event data of the same scale has the same accuracy in comparison with the available machine learning techniques; (2) verify the performance of the resource allocation through its comparison in terms of the allocation output by classical energy efficient techniques. The following metrics were utilized for assessing the performance of the novel resource allocation techniques.

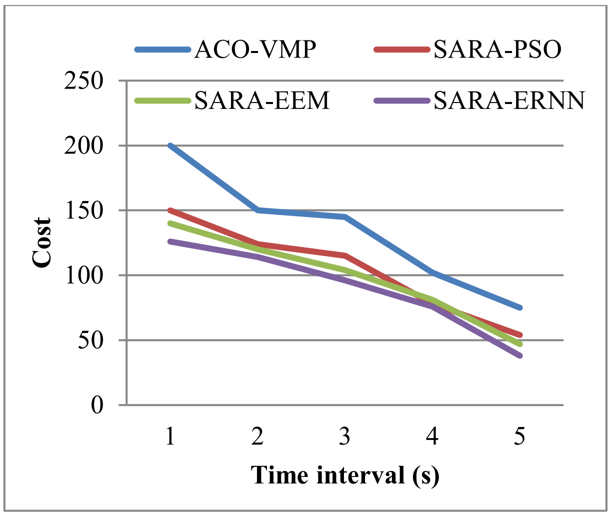


Figure 7. Cost efficiency of the novel and the available resource allocation methods

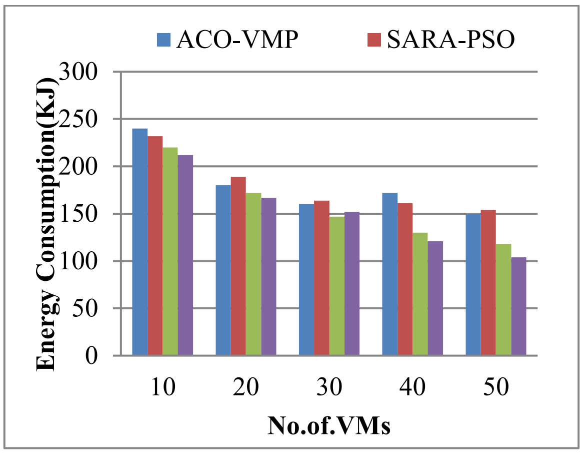


Figure 8. Evaluation performance of Energy Consumption range of the discussed and available resource allocation methods

In Fig.8. shows the performance of energy consumption range of the discussed and the available resource allocation methods. The performance of the novel SARA-ERNN approach is quite better than SARA-EEM, SARA-PSO and ACO-VMP. SARA-ERNN ensures better convergence employing Improved Bat algorithm (IBA). But, the convergence time is quite an issue owing to the distribution function based parameter changes. The result of experiment indicates that the novel SARA-ERNN approach consumes less than a minute for resolving an unpredictable allocation problem. This is because of the Qos prediction model readiness and scalability in finding an approximately optimal solution. Hence, this discussed technique is a suitable and more desirable approach for large-scale datacenters owing to rapid convergence.

The performance of the throughput in comparisons with the classical techniques SARA-EEM, SARA-PSO and ACO-VMP primarily is due to the fact that the accuracy of QoS prediction can have an effect on the resource allocation efficiency whereas the proposed technique offers better prediction accuracy as shown in Figure. 9. It is concluded from the graph that the discussed technique yields much better resource usage in comparison with the available approaches. The energy usage reduces owing to the efficiency of the novel approach, which has allocated several VMs. Hence, SARA-ERNN performs better than SARA-EEM, SARA-PSO and ACO-VMP with respect to resource usage and energy efficiency

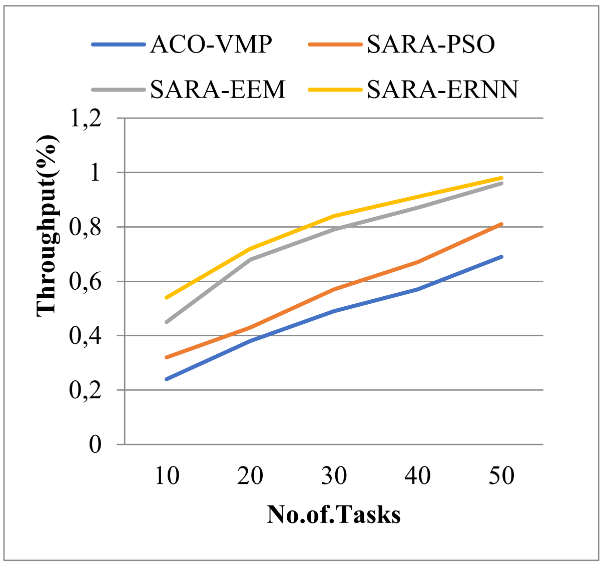


Figure 9. Evaluation performance of throughput of the novel and the available resource allocation methods

6. Conclusion

Cloud computing technology is on an exponential rise of being employed in companies and business global world. Resource allocation is the dire necessity of cloud providers for a rising number of users and with a lesser response time. In the case of cloud concept, an efficient resource allocation mechanism is necessary for attaining user satisfaction and this research work has shown the significance of resource allocation in cloud environment. This virtualization technology has helped in efficiently using the physical resources. This novel self-adaptive resource allocation architecture consists of three stages, which are QoS prediction model, improved bat algorithm (IBA) - based runtime decision algorithm and Energy Efficient Model (EEM). All the modules will improve the performance of the proposed approaches with the goal of satisfying the QoS needs and energy efficient model. The results of simulation reveal that the novel algorithm can help in effectively decreasing the energy usage of baseband and boost the resource allocation usage time of the system. As ap part of Future work, Incremental Relaying and other more complicated policies along with Decode and Forward and Amplify and Forward will be considered. Proportional Fairness will be regarded to be a goal rather than energy efficiency

References

[1] Patel, R., & Patel, S. Survey on resource allocation strategies in cloud computing. International Journal of Engineering Research & Technology (IJERT). 2013; 2(2) : 1-5.

[2] Parikh, S. M. A survey on cloud computing resource allocation techniques. In 2013 Nirma University International Conference on Engineering (NUiCONE). 2013;(pp. 1-5) : IEEE.

[3] Huang, L., Chen, H. S., & Hu, T. T. Survey on Resource Allocation Policy and Job Scheduling Algorithms of Cloud Computing1. Journal of Software. 2013; 8(2):480-487.

[4] Singh, S., & Chana, I. (2016). A survey on resource scheduling in cloud computing: Issues and challenges. Journal of Grid Computing. 2013; 14(2): 217-264.

[5] Anuradha, V. P., & Sumathi, D. (2014, February). A survey on resource allocation strategies in cloud computing. In International Conference on Information Communication and Embedded Systems (ICICES2014). February, 2014; (pp. 1-7): IEEE.

[6] Mastelic, T., Oleksiak, A., Claussen, H., Brandic, I., Pierson, J. M., & Vasilakos, A. V. Cloud computing: Survey on energy efficiency. ACM computing surveys (csur).2014; 47(2): 1-36.

[7] Krishnaveni, N., & Sivakumar, G.. Survey on dynamic resource allocation strategy in cloud computing environment. International Journal of Computer Applications Technology and Research. 2013 ; 2(6): 731-737.

[8] Nagpure, M. B., Dahiwale, P., & Marbate, P. An efficient dynamic resource allocation strategy for VM environment in cloud. In 2015 International Conference on Pervasive Computing (ICPC) . January 2015;(pp. 1-5): IEEE.

[9] Sareen, P., Kumar, P., & Singh, T. D. Resource Allocation Strategies in Cloud Computing. International Journal of

Computer Science & Communication Networks.2015; 5(6): 358-365.

[10] Dr Balamurugan E, Dr Sangeetha K, Dr Sathishkumar K , Dr Akpajaro J," Modified Support Vector Machine Based Efficient Virtual Machine Consolidation Procedure For Cloud Data Centers", Journal of Advanced Research in Dynamical and Control Systems, 2020, Vol 12, 04 Special Issue, Page No 501 -508

[11] Amjad Gawanmeh, A., Alomari, A., & April, A. Optimizing resource allocation scheduling in cloud computing services. Journal of Theoretical & Applied Information Technology, 2015; 95(1): 31-39.

[12] Wei, G., Vasilakos, A. V., Zheng, Y., & Xiong, N. A game-theoretic method of fair resource allocation for cloud computing services. The journal of supercomputing, 2010; 54(2): 252-269.

[13] An, B., Lesser, V. R., Irwin, D. E., & Zin. Automated negotiation with decommitment for dynamic resource allocation in cloud computing. In AAMAS. May, 2010; (Vol. 10: pp. 981-988).

[14] Tsai, J. T., Fang, J. C., & Chou, J. H. (2013). Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm. Computers & Operations Research,2013; 40(12): 3045-3055.

[15] Pawar, C. S., & Wagh, R. B. Priority based dynamic resource allocation in cloud computing. In 2012 International Symposium on Cloud and Services Computing, December 2012 ;(pp. 1-6): IEEE.

[16] Dr Balamurugan E, Md. Shahidul Hasan, Mohammad Shawkat Akbar Almamun, Sangeetha K," An Energy Efficient And Self Adaptive Resource Allocation Framework Using Modified Clonal Selection Algorithm For Cloud Based Software Services", Journal of Psychosocial Rehabilitation.2020; Volume 24, Issue 2, Pg.No. 5182-5203.

[17] Pillai, P. S., & Rao, S.. Resource allocation in cloud computing using the uncertainty principle of game theory. IEEE Systems Journal, 2014; 10(2): 637-648.

[18] Abirami, S. P., & Ramanathan, S.Linear scheduling strategy for resource allocation in cloud environment. International Journal on Cloud Computing: Services and Architecture (IJCCSA),2012; 2(1): 9-17.

[19] Tai, J., Zhang, J., Li, J., Meleis, W., & Mi, N. Ara: Adaptive resource allocation for cloud computing environments under bursty workloads. In 30th IEEE international performance computing and communications conference. November 2011, (pp. 1-8): IEEE.

[20] Morshedlou, H., & Meybodi, M. R. Decreasing impact of SLA violations: a proactive resource allocation approach for cloud computing environments. IEEE Transactions on Cloud Computing, 2014; 2(2):156-167.

[21] Wang, Y., Lin, X., & Pedram, M. (2014, April). A game theoretic framework of sla-based resource allocation for competitive cloud service providers. In 2014 Sixth Annual IEEE Green Technologies Conference. April 2014; (pp. 37-43): IEEE.

[22] Teng, F., & Magoulès, F. (2010, May). A new game theoretical resource allocation algorithm for cloud computing. In International Conference on Grid and Pervasive Computing. May 2010; (pp. 321-330): Springer, Berlin, Heidelberg.

[23] Chen, X., Wang, H., Ma, Y., Zheng, X., & Guo, L. Self-adaptive resource allocation for cloud-based software services based on iterative QoS prediction model. Future Generation Computer Systems,2020; 105: 287-296.

[24] Mikolov, T., Kombrink, S., Burget, L., Černocký, J., & Khudanpur, S. (2011, May). Extensions of recurrent neural network language model. In 2011 IEEE international conference on acoustics, speech and signal processing (ICASSP) May 2011, (pp. 5528-5531): IEEE.

[25] Happel, B. L., & Murre, J. M. (1994). Design and evolution of modular neural network architectures. Neural networks,1994;7(6-7):985-1004.

[26] Sotirov, S., Sotirova, E., Melin, P., Castilo, O., & Atanassov, K.. Modular neural network preprocessing procedure with intuitionistic fuzzy intercriteria analysis method. In Flexible Query Answering Systems 2016; (pp. 175-186): Springer, Cham.

[27] De Castro, L. N., & Von Zuben, F. J. The clonal selection algorithm with engineering applications. In Proceedings of GECCO , 2000, pp. 36-39 .

[28] Yang, X. S. (2011). Bat algorithm for multi-objective optimisation. International Journal of Bio-Inspired Computation, 2011; 3(5): 267-274.

[29] Laghari, A. A., He, H., Khan, A., Kumar, N., & Kharel, R. Quality of experience framework for cloud computing (QoC). 2018; 6, 64876-64890: IEEE Access.

[30] Laghari, A., He, H., Laghari, R., Khan, A., & Yadav, R.,Cache Performance Optimization of QoC Framework. EAI Endorsed Transactions on Scalable Information Systems, 2019; 6(20).