International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 4s
DOI: https://doi.org/10.17762/ijritcc.v11i4s.6303
***Article Received**: 19 December 2022 **Revised**: 22 January 2023 **Accepted**: 02 February 2023*

Hypervolume Sen Task Scheduilng and Multi Objective Deep Auto Encoder based Resource Allocation in Cloud

Ms.Neema George[1], Dr.Anoop B K[2]
[1]Research Scholar
Srinivas University
Mangalore, India
neemageo08@gmail.com
[2]Dept of AIML
Srinivas University
Mangalore, India
anoopbk2000@gmail.com

Abstract—Cloud Computing (CC) environment has restructured the Information Age by empowering on demand dispensing of resources on a pay-per-use base. Resource Scheduling and allocation is an approach of ascertaining schedule on which tasks should be carried out. Owing to the heterogeneity nature of resources, scheduling of resources in CC environment is considered as an intricate task. Allocating best resource for a cloud request remains a complicated task and the issue of identifying the best resource – task pair according to user requirements is considered as an optimization issue. Therefore the main objective of the Cloud Server remains in scheduling the tasks and allocating the resources in an optimal manner. In this work an optimized task scheduled resource allocation model is designed to effectively address large numbers of task request arriving from cloud users, while maintaining enhanced Quality of Service (QoS). The cloud user task requests are mapped in an optimal manner to cloud resources. The optimization process is carried out using the proposed Multi-objective Auto-encoder Deep Neural Network-based (MA-DNN) method which is a combination of Sen's Multi-objective functions and Auto-encoder Deep Neural Network model. First tasks scheduling is performed by applying Hypervolume-based Sen's Multi-objective programming model. With this, multi-objective optimization (i.e., optimization of cost and time during the scheduling of tasks) is performed by means of Hypervolume-based Sen's Multi-objective programming. Second, Auto-encoder Deep Neural Network-based Resource allocation is performed with the scheduled tasks that in turn allocate the resources by utilizing Jensen–Shannon divergence function. The Jensen–Shannon divergence function has the advantage of minimizing the energy consumption that only with higher divergence results, mapping is performed, therefore improving the energy consumption to a greater extent. Finally, mapping tasks with the corresponding resources using Kronecker Delta function improves the makespan significantly. To show the efficiency of Multi-objective Auto-encoder Deep Neural Network-based (MA-DNN) cloud time scheduling and optimization between tasks and resources in the CC environment, we also perform thorough experiments on the basis of realistic traces derived from Personal Cloud Datasets. The experimental results show that compared with RAA-PI-NSGAII and DRL, MA-DNN not only significantly accelerates the task scheduling efficiency, task scheduling time but also reduces the energy usage and makespan considerably.

Keywords- Cloud Computing, Cloud Server, Multi-objective, Auto-encoder, Deep Neural Network

I. INTRODUCTION

CC climate relate virtualization instrument to divide colossal actual assets into various virtual assets. A few clients use these virtual assets on a CC climate whenever and anyplace. Most existing cloud asset assignment techniques don't support the creating mode, explaining that they are deficient in endeavor the idealness and enhancement concerning asset distribution. However, clients' support more mindfulness of the practicality and advancement. In this manner, cloud specialist co-ops or cloud servers are very restless with how to control huge assets and improve asset use. A huge asset distribution is thus urgent to show up at these targets.

A Cloud Asset Distribution Calculation in light of an Equal and Further developed Non-ruled Arranging Hereditary Calculation II (RAA-PI-NSGAII) was proposed in [1]. To begin with, asset assignment need and it were figured out to match distances. Then, the proportion of proportion to reaction by means of multi-objective streamlining strategy for distribution the asset in CC climate was planned.Here, only minimum matching and resource proportion were employed.

Finally, a parallel allocation model was designed with the purpose of enhancing the quality that in turn not only improved average CPU utilization but also accelerated the speed significantly. Despite improvement found in minimizing the time involved, however several factors or objectives were not evolved, that in turn resulted in accuracy compromise. To address on this aspect, in this work, Hypervolume-based Sen's Multi-objective programming model is first designed for selecting on multiple quality assessment metrics and then mapping via Sen's function that in turn results in the improvement of both task scheduling time and task scheduling efficiency.

With the contemporary upsurge in the progression of CC data centers, and advancements in user service quality requirements, the framework of the entire cloud system has become more complicated. This in turn has also made the resource scheduling more demanding. Hence, the objective remained in sorting out the discord between cloud service providers (CSPs) who focus was to reduce energy costs and also optimize service quality. A deep reinforcement learning (DRL) was proposed in [2] on the basis of the Deep Q-network (DQN) algorithm. Here, the trade-off between energy consumption and task makespan was provided by fine tuning the reward of numerous optimization objectives. As a result, improvements were observed both in terms of energy consumption and task makespan.

Despite enhancements observed both in terms of energy consumption and makespan, the same performance metrics were found to be affected in case of distinct constraints between resources of the cloud. To address on this factor, Auto-encoder Deep Neural Network-based Resource allocation algorithm is presented that with the aid of Jensen–Shannon divergence function improves energy consumption and makespan even in case of distinct constraints.

A couple based task booking calculation for CC climate, in view of the Hungarian streamlining calculation was proposed in [3]. Here, inconsistent number of errands and mists were thought of and afterward undertakings were matched for settling on booking choice. With this the delay time was viewed as moved along. A survey on a few work planning components in cloud in light of need was figured out in [4].

A large portion of the common assets from CC climate were dispensed based on task necessities. Yet, being fundamental in dissecting the overarching asset accessibility and the probability of asset preceding the assignment of the mentioned tasks is said. With this, in [5], a mixture improved asset distribution strategy was proposed by utilizing bat enhancement calculation and molecule swarm streamlining calculation by thinking about asset status, distance, transfer speed, and undertaking necessities.

Distributed computing is overwhelmingly founded on the origination of on-request conveyance of asset, stockpiling, etc. As a substitute of getting, claiming, and keeping up with actual server farm and server clients access innovation administrations as need premise from a cloud supplier. However, the development of CC yet not found to be full proof and hence faces distinct number of challenges like security and scheduling.

In [6], a task scheduling technique concentrating on dynamic decision for micro service-based Cloud Computing Applications was proposed. The method executed both delay-sensitive applications and mobility with less cost. Moreover, the study concentrated on Task Scheduling issues like, matching of resources as requested by the user, sequencing of the corresponding user and scheduling them, therefore enabling improvement in resource utilization.

Yet another method employing Deep reinforcement learning was applied in [7] for optimal resource scheduling in cloud. However, only the response time were focused and less concentration was made on the aspect of revenue generation. A resource aware dynamic scheduling method was presented in [8]. Here, significant improvement in average response time along with the execution cost was found to be improved. Despite improvement in execution cost the makespan factor was less focused.

Motivated by the above papers, in this work, a method called, Multi-objective Auto-encoder Deep Neural Network-based (MA-DNN) is proposed. The main objective of this research is to propose a new optimized task scheduled and resource allocation model in cloud computing environment that performs task scheduling and resource allocation separately in an efficient manner. This optimization method addresses the limitations of the earlier resource allocation methods in cloud computing environment by its multi-objective model i.e., optimization and deep learning model. Further, the contributions of this paper include the following.

- To propose a method called, Multi-objective Auto-encoder Deep Neural Network-based (MA-DNN) method which is a combination of Sen's Multi-objective functions and Auto-encoder Deep Neural Network model
- To design a novel Multi-objective optimal scheduling algorithm. During algorithm optimization, multi-objective optimization is carried out with processing time, handover time, makespan time and workflow execution cost as evaluation indexes so as to obtain the relatively cloud task-scheduling. Our model can improve the task scheduling efficiency and time via Sen's Multi-objective functions.
- To present an Auto-encoder Deep Neural Network-based Resource allocation algorithm based on Kronecker Delta function and Deep learning model that ensures better

optimal resource allocation in the cloud computing environment.

- The proposed MA-DNN method in cloud computing environment has provided improved results for task scheduling efficiency, task scheduling time, energy consumption and makespan as performance evaluation measures.

The rest of the paper is organized as: The discussion about numerous scheduling techniques in cloud computing environment is presented in Section 2. In Section 3, the proposed Multi-objective Auto-encoder Deep Neural Network-based (MA-DNN) method has been elaborated. Experimental setup for validating the MA-DNN method is provided in Section 4. Comparative analysis with an elaborate discussion is described in Section 5 and finally, the conclusions are presented in Section 6.

II. RELATED WORKS

Cloud computing is considered as the modern computer technology that utilizes virtualized infrastructure for ensuring reliable services to users even in case of complicated scenarios. Over the past few years, the issue of scheduling dependent tasks has received great amount of attention for researchers in this domain.

A multi-objective Estimation of Distribution Algorithm (EDA) was presented in [9] with the objective of learning and optimizing fuzzy offloading technique from different types of applications. The algorithm split into distinct clusters with the objective of allocating to the respective tier for further processing. As a result, the system resources were found to be saved by ensuring scheduling in a minimal search space. However, makespan was not focused.

In [10], a novel Task Scheduling-Decision Tree (TS-DT) algorithm for assigning and implementing application's task was presented. Also an elaborate comparative study was provided and validation results found improvement not only in terms of makespan but also resulted in the improvement of resource utilization. A review on several resource management techniques in CC environment employing reinforcement learning was investigated in [11]. A survey of resource management in CC was presented in [12].

Since its evolution, CC environment has immensely changes our day to day living by associating the whole world via shared computational resources over the internet. In [13], a semi-dynamic real-time task scheduling method was presented in the cloud–fog environment. Here, initially, permutation-based optimization model was designed for scheduling. Second a modified genetic algorithm was designed with the purpose of ensuring distinct permutations for the tasks being arrived during each scheduling. Finally, the tasks were assigned according to the best permutation pattern, possessing sufficient resources, therefore minimizing the anticipated execution time. However, the cost involved was not focused. To address on this aspect, resource optimization algorithm employing particle swam optimization was designed in [14]. With this not only the cost factor was improved but also the task scheduling efficiency was found to be improved.

The work in [15] focused on the enhancement of the task distribution in virtual machine for CC utilizing load balancing technique. To achieve this objective modification was made in the Bat algorithm fitness value in the load balancer section. With this the improvement was found in both data center efficiency and accuracy. Yet another social group optimization algorithm was employed in [16] for concentrating on the throughput maximization.

In [17] heuristic approach was introduced that in turn not only reduced the turnaround time but also resulted in the improvement of response time. A task scheduling method based on the enhanced heterogeneous earliest finish time was proposed in [18] with the purpose of both optimizing task execution efficiency and concentrating on quality of service (QoS). Swarm-based ant colony optimization was introduced in [19] for optimal allocation of resources. Evaluation and validation in terms of load was focused in [20] by employing improved particle swarm optimization.

Based on the aforementioned techniques and methods, in this work, a method called, Multi-objective Auto-encoder Deep Neural Network-based (MA-DNN) for both task scheduling and resource allocation in cloud computing environment is proposed. The elaborate description of the MA-DNN method is provided in the following sections.

III. METHODOLOGY

First, Conventional competitive cloud task scheduling and optimization of resource is no longer acceptable for present day prerequisites. In this section a novel method called, Multi-objective Auto-encoder Deep Neural Network-based (MA-DNN) cloud task scheduling and optimization between tasks and resources in the CC environment is presented. Figure 1 shows the block diagram of MA-DNN method.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 4s
DOI: https://doi.org/10.17762/ijritcc.v11i4s.6303
***Article Received**: 19 December 2022 **Revised**: 22 January 2023 **Accepted**: 02 February 2023*

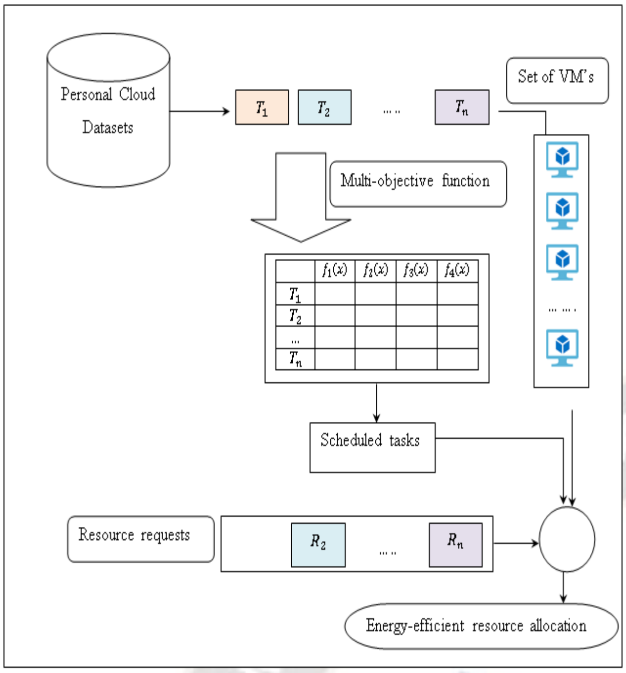

Figure 1. Block diagram of Multi-objective Auto-encoder Deep Neural Network-based (MA-DNN) task scheduling and resource allocation

As displayed in the above figure, most importantly, a multi-objective capabilities comprising of finishing time and cost is planned. With this resultant capability, the proposed technique depends on profound learning design by utilizing fitting cloud assets and considers the similitudes and profound show between cloud server modules. The development between cloud asset networks is coordinated with deduced data about the cloud asset and set up by the wellness requirements of cloud asset arranging. The nitty gritty depiction of Mama DNN technique is given beneath followed by the framework model.

A. System model

A multi-objective workflow application, '$G = (V, E)$', is designed by employing Directed Acyclic Graph (DAG), where 'V' represents the vertices forming the tasks '$T = (T_1, T_2, \dots, T_n)$' and '$E = (E_{ij} | T_i, T_j \in T)$' denotes the edges forming the association between tasks. The tasks weight is obtained by its reference execution time, depending on the number of information or data collected at a specific timestamp 't' and the edge weight represented by its output information or data size. Here, the task is said to be scheduled to single cloud resource, whereas, each resource execute multiple tasks by implemented one task at a time.

B. Hypervolume-based Sen's Multi-objective programming model

Scheduling of workflows in CC environment becomes more demanding when addressed in dynamic environment. A multi-objective optimization programming model is an optimization problem that possesses two or more objectives. In this work, a multi-objective optimization is carried out with time (i.e., processing time, handover time, makespan time) and cost (i.e., workflow execution cost) by employing hyper volume indicator via Sen's multi-objective function. The hyper volume indicator maps from 'n' dimension space (i.e., multi-objective function) to single dimensional space (i.e., optimized function) that evaluates convergence of obtaining solution set in an accuracy and timely manner. The hypervolume measure is of specific interest as it requires less time and resources in modeling multi-objective programming model. Figure 2 shows the structure of Hypervolume-based Sen's Multi-objective programming model.

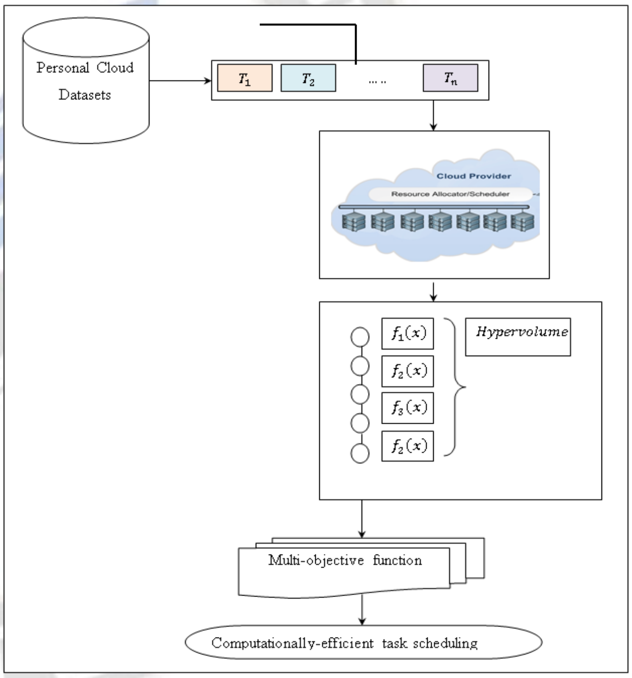

Figure 2. Structure of Hypervolume-based Sen's Multi-objective programming model

As shown in the above figure, with the tasks obtained from Personal Cloud Datasets at different time instances, resource allocation has to be made in an optimal manner. With this objective, a hypervolume-based mapping is designed that with the aid of four distinct objectives like, task processing time, handover time, makespan time and task execution cost, computationally efficient task scheduling is attained. Let us consider '$T = (T_1, T_2, \dots, T_m)$', a set of tasks, '$Res = (Res_1, Res_2, \dots, Res_m)$' a set of resources, providing distinct resource types and contain only one virtual machine '$VM =$

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 4s
DOI: https://doi.org/10.17762/ijritcc.v11i4s.6303
Article Received: 19 December 2022 Revised: 22 January 2023 Accepted: 02 February 2023

$(VM_1, VM_2, \dots, VM_o)$' at a time. Then, the processing time of task 'T_i' on resource 'Res_j' is mathematically formulated as given below.

$$f_1(x) = PT(T_i, Res_j) = \frac{RE_t(T_i)}{ResC_j} \quad (1)$$

From the above equation (1), the processing time 'PT' is evaluated based on the reference execution time '$RE_t(T_i)$', i.e., on the basis of the number of information obtained at each timestamp 't' and the resource cost '$ResC_j$' respectively. Second, the handover time between two tasks depends on bandwidth. Then, the handover time between two tasks 'T_i', 'T_j' with bandwidth '$Band$' is mathematically stated as given below.

$$f_2(x) = HOT(T_i, T_j) = \begin{cases} E_{ij}/Band, & if\ Res_j \neq Res'_j \\ 0, & if\ Res_j = Res'_j \end{cases} \quad (2)$$

From the above equation (2), the handover time 'HOT' is obtained based on the resultant value of the actual resource cost 'Res_j' and the estimated resource cost 'Res'_j' respectively. Finally, makespan time is measured, i.e., the problem of reducing makespan only assigns tasks to equivalent virtual machines. Let '$P_{ij} = 1$', if task 'T' is assigned to equivalent virtual machine and '$P_{ij} = 0$' otherwise. Then, with maximum completion time between all virtual machines being 'MCT', then, the makespan time is mathematically stated as given below.

$$f_3(x) = MST \geq \sum Equiv_j P_{ij} \quad (3)$$

Followed by the measurement of makespan time as given in equation (3), task execution cost is evaluated by employing the execution cost, transmission cost and storage cost respectively. This is mathematically formulated as given below.

$$f_4(x) = TEC = \sum_{E_{ij} \in E}(E_{ij} * EC_j) + \sum_{E_{ij} \in E}(E_{ij} * TC_j) + \sum_{E_{ij} \in E}(E_{ij} * SC_j) \quad (4)$$

From the above equation (4), the task execution cost 'TEC' for each edge 'E_{ij}' is evaluated by means of the execution cost 'EC', transmission cost 'TC' and the storage cost 'SC' of the resource 'r_j' at timestamp 't' respectively. As given above, the individual optima (i.e., time and cost) are obtained for each objective separately as given below.

$$Z_{optima} = [f_1(x), f_2(x), f_3(x), f_4(x)] \quad (5)$$

Then, the multi-objective function is mathematically stated as given below.

$$Maximize, Z = \frac{\sum_{j=1}^{m} z_j}{W_j} - \frac{\sum_{j=m+1}^{n} z_j}{W_{m+1}} \quad (6)$$

From the above equations (5) and (6), the maximization of the multi-objective function is derived so that resulting in the improvement of both the task scheduling efficiency and task scheduling time. The pseudo code representation of Hypervolume-based Sen's Multi-objective Optimized Task Scheduling is given below.

Input: Cloud Server 'CS', Task '$T = T_1, T_2, \dots, T_n$', Virtual Machine '$VM = VM_1, VM_2, \dots, VM_o$', resource '$Res = (Res_1, Res_2, \dots, Res_m)$',
Output: computationally efficient task scheduling
1: **Initialize** reference execution time '$RE_t(T_i)$', timestamp 't', resource cost '$ResC_j$', 'm', 'n', 'o' 2: **Initialize** execution cost 'EC', transmission cost 'TC' and the storage cost 'SC' 3: **Begin** 4: **For** each incoming tasks 'T' 5: Evaluate processing time of task 'T_i' on resource 'Res_j' as given in equation (1) 6: **For** each incoming tasks 'T_i' and 'T_j' 7: Evaluate handover time between two tasks 'T_i' and 'T_j' as given in equation (2) 8: Formulate makespan time as given in equation (3) 9: Measure task execution cost as given in equation (4) 10: Evaluate individual optima (i.e., time and cost) as given in equation (5) 11: Maximize multi-objective function as given in equation (6) 12: Return tasks scheduled 'TS' based on the results of multi-objective function 13: **End for** 14: **End for** 15: **End**

Algorithm 1. Hypervolume-based Sen's Multi-objective Optimized task Scheduling

As given in the above Hypervolume-based Sen's Multi-objective Optimized Task Scheduling algorithm, multi-objective (i.e., time and cost) simultaneously under common constraints. First, individual objectives like, task processing time, handover time, makespan time were formulated for each cloud user request task according to the number of information obtained at each specific timestamp of resource. Second, the task execution cost between the tasks is evaluated. Finally, multi-objective function is optimized by means of Sen's multi-objective programming model. With this computationally efficient task scheduling is ensured.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 4s
DOI: https://doi.org/10.17762/ijritcc.v11i4s.6303
***Article Received**: 19 December 2022 **Revised**: 22 January 2023 **Accepted**: 02 February 2023*

C. *Auto-encoder Deep Neural Network-based Resource* allocation

Based on the nonlinear features of cloud resource network in CC environment, a model is applied to deal with the issue of resource allocation. Figure 3 shows the structure of Auto-encoder Deep Neural Network-based Resource allocation model.

As shown in the figure 3, the Auto-encoder Deep Neural Network-based Resource allocations model is split into four sections. They are initialization, encoding, mapping and decoding. To start with, the equivalent matrix representing matrix vector of tasks scheduled is initialized and stored in the form of vector.

Second, in the encoding layer, the requests are obtained and analyzed whether same types of requests are being made by other cloud users. This is done by employing a task association matrix in the encoding layer that takes into account the similarities and deep presentation between cloud server modules. Third, the actual mapping process is performed by employing fitness constraints using Kronecker Delta function so that optimality is said to be maintained. Finally, in the decoding layer, appropriate allocation of resources is made

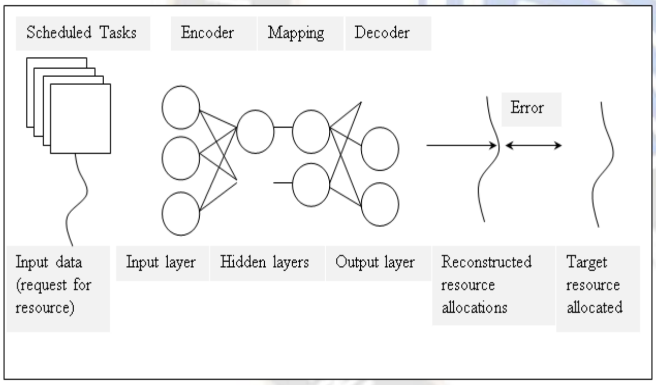

Figure 3. Structure of Auto-encoder Deep Neural Network-based Resource allocation model

Let us introduce an equivalent matrix '$EM = [EM_{ij}]$' to represent the parallelism of cloud resource in the CC environment as given below.

$$EM_{ij} = Exp\left[\sum_{k=1}^{N} \frac{(P_{ik} - P_{kj})^2}{\sigma_k^2}\right] \quad (7)$$

From the above equation (7), 'P_{ik}' represents the '$k-th$' feature of the equivalent matrix vector 'P_i' and 'σ' forms the scaling factor of the influence parameter for each dimension. In addition, we all consider the association between distinct tasks who are in requirement of the same resource by introducing a task association matrix '$TM = tm_{ij}$' given as below.

$$tm_{ij} = EM_{ij} - \frac{\sqrt{k_i^2 + k_j^2}}{\sum_{ij} EM_{ij}} \quad (8)$$

From the above equation (8), 'tm' is a further optimization of 'EM' that can be utilized to better acquire association between distinct tasks of cloud resource, with 'k_i'extent of resource 'i'. The data (i.e., task scheduled) of 'T' remains the input to the auto-encoder organization. In our work, the task scheduled are encoded by the encoder and then decoded by the decoder. Followed by which the decoded data (i.e., task scheduled) will be compared with the input data via reconstruction procedure. On the basis of the auto-encoder organization, the error between input data (i.e., resource requested) and output (i.e., resource assigned) data are reduced by employing Jensen–Shannon divergence in case of distinct tasks in requirement of same resource. With the input of auto-encoder by the task association matrix 'TM', the encoded task association matrix 'TM' is mapped into a low dimensional embedding matrix as given below.

$$EM_i = f(tm_{ij}) = KF(W_H tm_{ij} + B_H) \quad (9)$$

From the above equation (9), 'W_H' and 'B_H' denotes the weight and the bias of the endcoding layer and '$KF(T)$' denotes the Kronecker Delta function to model that is symmetrical in indexes 'i', 'k' respectively.

$$KF\ (T) = \frac{\partial}{\partial T_k}\sigma(T,i)[\delta_{ik} - \sigma(T,i)] \quad (10)$$

This kronecker delta function '$KF\ (T)$' given above acquires the task vector 'T' as input symmetrical in indexes 'i', 'k' and normalizes it into a low dimensional embedding matrix. Upon successful accomplishment of decoding, the hidden layer description 'H' is given back to the original data space for obtaining reconstruction of original data. This is mathematically stated as given below.

$$O_i = g(h_i) = KF(W_0 tm_{ij} + B_O) \quad (11)$$

From the above equation (11), the results of the original data 'O_i' or tasks scheduled for corresponding resource is obtained based on weight of decoding layer 'W_0', bias of decoding layer 'B_O' and task association matrix 'tm_{ij}' respectively. Moreover, to minimize the energy consumption of cloud data centers by reconstructing the original data 'TM' and deviation between data 'O' a low dimensional cost representation is modeled as given below.

$$Cost = argmin\ (TM, O) = argmin \sum_{i=1}^{N}(tm_i - O_i)^2 \quad (12)$$

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 4s
DOI: https://doi.org/10.17762/ijritcc.v11i4s.6303
***Article Received**: 19 December 2022 **Revised**: 22 January 2023 **Accepted**: 02 February 2023*

In addition, a sparsity constraint is added by utilizing a restriction condition on the neural network of auto-encoder in case of distinct tasks in requirement of same source is handled by employing the Jensen–Shannon divergence function.

$$JSD(T_i|T_j) = \frac{1}{2}(T_i - T_j)[logit\ (T_i) - logit\ (T_j)] \quad (13)$$

$$logit\ (T_i) = \ln\left[\frac{T_i}{1-T_i}\right]; logit\ (T_j) = \ln\left[\frac{T_j}{1-T_j}\right] \quad (14)$$

Hence, the total cost function of neural network for autoencoder with the objective of improving the makespan is mathematically stated as given below.

$$UCost = Cost + JSD(T_i|T_j) \quad (15)$$

From the above equation (15), '$Cost$' is the result obtained from equation (12), which is the reconstruction between 'TM' and 'O', whereas '$UCost$' represents the summation between cost and divergence factor 'JSD' respectively. The pseudo code representation of Auto-encoder Deep Neural Network-based Resource allocation is given below.

Input: Cloud Server 'CS', Task '$T = T_1, T_2, \ldots, T_n$', Virtual Machine '$VM = VM_1, VM_2, \ldots, VM_o$', resource '$Res = (Res_1, Res_2, \ldots, Res_m)$'
Output: Robust and optimized resource allocation

1: **Initialize** 'N', tasks scheduled 'TS', scaling factor '$\sigma = 0.1$'
2: **Begin**
3: **For** each incoming tasks scheduled 'TS' and matrix vector 'P_i'
//Initialization
4: Model equivalent matrix as given in equation (7)
5: **For** each tasks scheduled 'TS' for corresponding resource
//Encoding layer or input
6: Formulate task association matrix as given in equation (8)
//Mapping
7: Performing mapping as given in equations (9) and (10)
8: **End for**
9: **For** each tasks scheduled 'TS' for corresponding resource
//Decoding layer or output
10: Formulate reconstruction of original data as given in equation (11)
11: Formulate low dimensional cost representation as given in equation (12)
12: Formulate sparsity constraint is added by utilizing a restriction condition as given in equations (13) and (14)
13: Evaluate updated cost as given in equation (15)
14: Return allocation resources
15: **End for**
16: **End for**
17: **End**

Algorithm 2. Auto-encoder Deep Neural Network-based Resource allocation

As given in the above algorithm, the objective remains in designing Auto-encoder Deep Neural Network-based Resource allocation with improved energy utilization and makespan. The overall algorithm is split into four distinct processes. First, with the scheduled tasks provided as input, initialization process is formulated by means of equivalent matrix, comprising of scheduled tasks and requests being made with the cloud server. Second, an input layer is structured via formulation of task association matrix. Here, the tasks associated and also two or more requested tasks for similar requests of resources are taken into consideration and in case of deviation, Jensen–Shannon divergence function is introduced to minimize the error. With this the energy usage in allocating with the corresponding virtual machine is said to be reduced also only higher divergence results are utilized during the mapping process. Third, a mapping function, i.e., mapping tasks with the corresponding resources is made via Kronecker Delta function. Finally, decoding or corresponding resource allocation to the requested tasks is made with minimum error in the decoding layer. With this not only the energy consumption is reduced but also results in the improvement of makespan significantly.

IV. Experimental Setup

Trial assessment of the proposed Multi-objective Auto-encoder Profound Brain Organization based (Mama DNN) strategy is completed utilizing Cloud Test system utilizing Local Java Codes. The CloudSim network reenactment is a recreation and displaying device performing distributed computing administrations. The whole structure is performed on a conveyed server with various Virtual Machines associated with it. The dataset is taken from Individual Cloud Datasets: NEC Individual Cloud Follow (http://cloudspaces.eu/results/datasets). Trial assessment of proposed Mama DNN technique and two existing calculations in particular Cloud Asset Distribution Calculation in view of an Equal and Further developed Non-overwhelmed Arranging Hereditary Calculation II (RAA-PI-NSGAII) [1] and profound support learning (DRL) are reenacted utilizing the various measurements, for example, task booking effectiveness, task planning time, energy utilization and makespan.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 4s
DOI: https://doi.org/10.17762/ijritcc.v11i4s.6303
***Article Received**: 19 December 2022 **Revised**: 22 January 2023 **Accepted**: 02 February 2023*

V. DISCUSSION

A. Case scenario 1: Task scheduling efficiency

In this section the results of task scheduling efficiency is provided. The effectiveness of optimal resource allocation can be analyzed based on the number of tasks scheduled with respect to the user requested tasks. Higher the task scheduling efficiency better is the methods performance to be. The task scheduling efficiency is mathematically stated as given below.

$$TSE = \left[\frac{NTCS}{n}\right] * 100 \quad (16)$$

From the above condition (16), the errand planning effectiveness 'TSE' is estimated in view of the quantity of client mentioned undertakings put in the line 'n' and the quantity of assignments accurately booked 'NTCS'. It is estimated concerning rate (%). Different number of client mentioned errands running somewhere in the range of 100 and 1000 are thought about independently. The consequence of the recreation is recorded in table 1. Figure 4 shows the examination of assignment booking productivity upsides of three unique strategies, Mama DNN, RAA-PI-NSGAII [1] and DQN [2] individually. It is seen that the Mama DNN outflanks RAA-PI-NSGAII [1] and DQN [2].

Table 1. Tabulation of task scheduling efficiency using MA-DNN, RAA-PI-NSGAII [1] and DQN [2]

Number of user requested tasks	Task scheduling efficiency (%)		
	MA-DNN	RAA-PI-NSGAII	DQN
100	94	92	91
200	92.15	90.15	89.35
300	92	89.35	87.25
400	91.55	88.15	86.25
500	90	88	85.25
600	90.55	89.35	86
700	91.35	90	87
800	94.25	92.15	88.35
900	95	93	89
1000	93.15	92.45	88.15

Figure 4 shows the assignment booking proficiency accomplished by various kinds of client mentioned errands when served at the CC assets utilizing three distinct strategies. The x-hub portrays the quantity of client mentioned assignments that produce the cloud undertakings to be handled, though the y-pivot addresses their comparing task planning productivity accomplished with regards to rate (%).It shows how various kinds of client mentioned errands consume errands for designating the assets. It is addressed that the proposed technique Mama DNN displays higher better undertaking booking proficiency in the underlined climate upon correlation with [1] and [2]. The explanation for the improvement was because of the legitimate harmony between the use of hosts between cloud clients in the CC climate by utilizing Hypervolume-based Sen's Multi-objective Upgraded Assignment Planning calculation. By applying this calculation utilizing the Sen's multi-objective capability secures the genuine worldwide arrangement (i.e., ideal use of host guaranteeing task planning effectiveness) no matter what the cloud client demand errands in concern, quick assembly between under-use and over use.As a result, the task scheduling efficiency using MA-DNN method is said to be improved by 2% compared to [1] and 5% compared to [2] respectively.

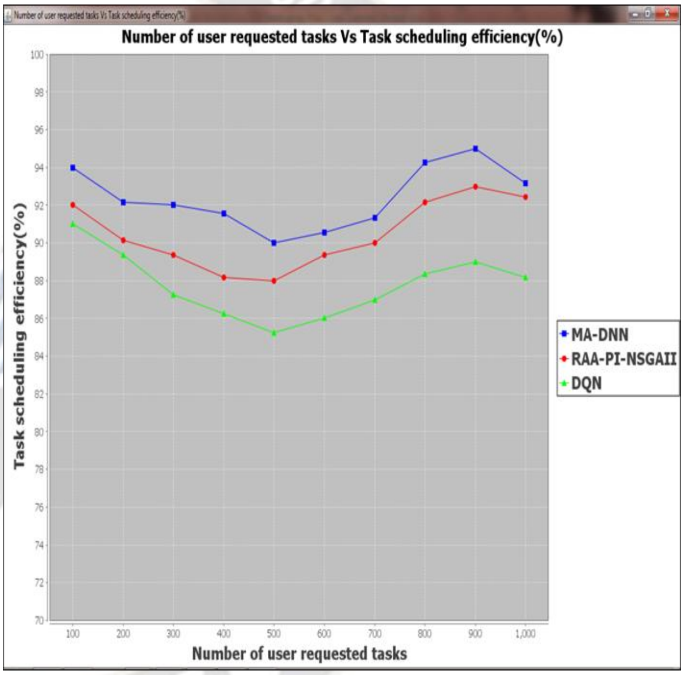

Figure 4.Comparison results of task scheduling efficiency

B. Case scenario 2: Task scheduling time

In this section, the result analysis of task scheduling time is discussed. A portion of time is said to be consumed during the processing of scheduling the tasks to the corresponding user. This is said to be the task scheduling time. It is mathematically stated as given below.

$$TST = \sum_{i=1}^{n} T_i * Time\ [ST] \quad (17)$$

From the above equation (17), task scheduling time 'TST' is measured based on the number of tasks involved in the simulation process 'T_i' and the time consumed in scheduling each and every task to the requested cloud users '$Time\ [ST]$'. It is measured in terms of milliseconds (ms). Different numbers of user requested tasks ranging between 100 and 1000 are considered separately. The result of the simulation is listed in

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 4s
DOI: https://doi.org/10.17762/ijritcc.v11i4s.6303
***Article Received**: 19 December 2022 **Revised**: 22 January 2023 **Accepted**: 02 February 2023*

table 2. Figure 5 shows the comparison of mean task scheduling time values of three distinct resource allocation methods. It is observed that MA-DNN method outperforms RAA-PI-NSGAII [1] and DQN [2].

Figure 5 delineates the errand booking time concerning 10000 particular client mentioned undertakings. From the above figure, x hub means the quantity of cloud client mentioned errands and y hub addresses the assignment booking time estimated concerning milliseconds (ms). Additionally from the above figure it is distinguished that by expanding the quantity of cloud client mentioned undertakings brings about the expansion in the quantity of cloud clients to be concentrated by the cloud server to defeat the point at issues in regards to, multi-objective capabilities, such as, handling season of errand, handover time, makespan time and assignment execution cost. In every one of the three strategies, expanding the quantity of cloud client mentioned errands brings about the increment of assignment planning time moreover. Notwithstanding, a critical improvement is noticed utilizing Mama DNN technique. This is a direct result of the utilization of Differential Development based Dark Wolf Streamlining model. By applying this model, individual optima (i.e., time and cost) are gotten for every goal independently through boost of the multi-objective capability by means of cloud server. With the usage of hyper volume pointer planning from 'n' aspect space (i.e., multi-objective capability) to single layered space (i.e., streamlined capability) brings about the assembly of acquiring arrangement set, in this way diminishing the likelihood of falling into the neighborhood ideal strategy. Accordingly, the undertaking planning time or the time consumed in booking of each assignment between the client demands are supposed to be diminished utilizing Mama DNN technique by 27% contrasted with [1] and 43% contrasted with [2] separately.

Table 2. Tabulation of task scheduling time using MA-DNN, RAA-PI-NSGAII [1] and DQN [2]

Number of user requested tasks	**Task scheduling time (ms)**		
	MA-DNN	RAA-PI-NSGAII	DQN
100	3.5	5.5	6
200	6.25	7.75	8.35
300	7.35	10.25	12.45
400	9.15	13.35	18.25
500	10.25	15.85	21.45
600	11.55	17	23.25
700	12.65	18.35	25.15
800	14.35	19.15	27.25
900	18	21.25	29
1000	19.85	24.35	31.32

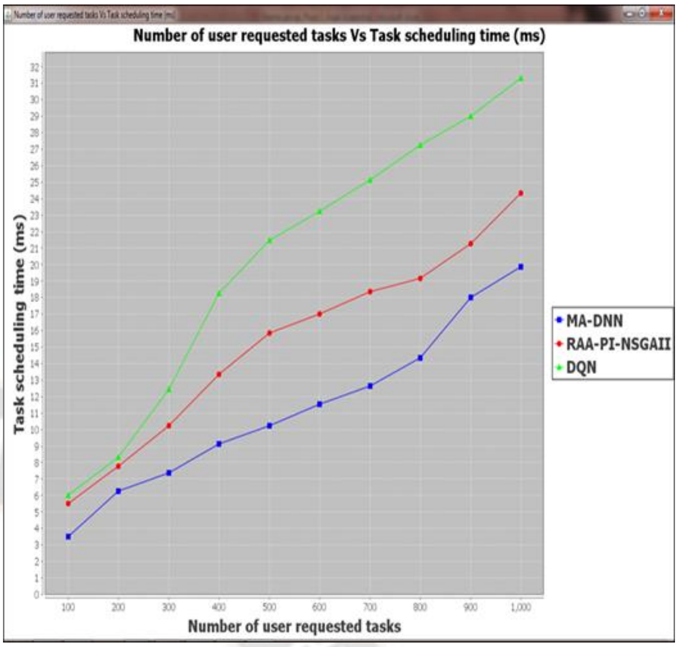

Figure 5 . Comparison results of task scheduling time

C. Case scenario 3: Energy consumption

In this section the energy consumption involved during the allocation of virtual machine for the corresponding cloud user requested task is provided. Energy is said to be consumed during the process of virtual machine allocation in the data center and hence is said to be unavoidable. However, using certain mechanisms, the energy consumption can be reduced. Energy consumption is mathematically stated as given below.

$$EC = \sum_{i=1}^{n} T_i * Energy(VM_{alloc}) \quad (18)$$

From the above equation (18), energy consumption 'EC' is evaluated based on the number of cloud user requested tasks involved in the simulation process 'T_i' and the actual energy consumed in allocating the virtual machine for each user '$Energy(VM_{alloc})$'. It is measured in terms of kilo watts per hour (Kwh). Number of user requested tasks ranging between 100 and 1000 are considered during the simulation of task scheduling and resource allocation.

Only after the scheduled tasks, the resource allocation is performed by the cloud server. The result of the simulation for energy consumption is listed in table 3. Figure 6 illustrates the comparison of mean values conducted for 10 different simulation runs of three different methods. It is observed that the MA-DNN method outperforms Standard RAA-PI-NSGAII [1] and DQN [2].

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 4s
DOI: https://doi.org/10.17762/ijritcc.v11i4s.6303
***Article Received**: 19 December 2022 **Revised**: 22 January 2023 **Accepted**: 02 February 2023*

Table 3 .Tabulation of energy consumption using MA-DNN, RAA-PI-NSGAII [1] and DQN [2]

Number of user requested tasks	**Energy consumption (kWh)**		
	MA-DNN	RAA-PI-NSGAII	DQN
100	53.2	61.5	68.5
200	58.15	69.55	78.35
300	63.25	75.85	90.25
400	68.55	83.25	105.35
500	75.35	95.25	135.35
600	82.35	115.35	140.15
700	90.15	125.25	155.35
800	95	140	170.25
900	98.35	155.35	195.35
1000	105	170	205.55

Figure 6 given above shows the energy utilization or the energy consumed while performing distribution of assets in an ideal way even if there should be an occurrence of unmistakable requirements between assets of the cloud. From the above allegorical portrayal, the energy utilization is viewed as in the rising pattern. More specifically, expanding the quantity of client mentioned errands brings about a proportionate expansion in utilization of energy for virtual machine portion. Reenactment performed with 100 client mentioned undertakings saw 53.2 kWh of energy being utilization utilizing Mama DNN, 61.5 kWh utilizing [1] and 68.5 kWh utilizing [2] individually. The energy consumed during the time spent asset designation utilizing Mama DNN technique was seen to be similarly lesser than [1] and [2].

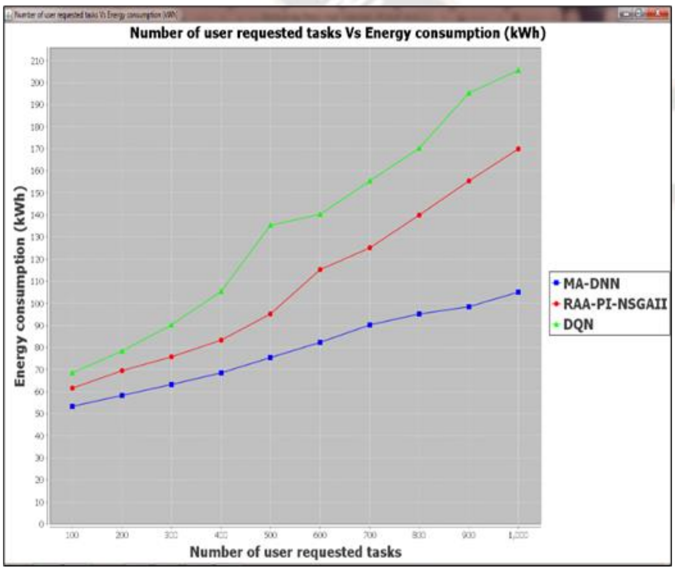

Figure 6. Comparison results of energy consumption

The purpose for the improvement was because of the use of Auto-encoder Profound Brain Organization based Asset assignment calculation. By applying this calculation, with errands being booked given as info, input layer was figured out through task affiliation network. In the event of comparative solicitations of assignments, Jensen-Shannon difference capability was used to diminish the mistake. Accordingly, the energy utilization was viewed as diminished likewise just higher difference results were used during the planning system.

Because of this, the energy utilization during asset distribution in the server farm utilizing Mama DNN strategy was viewed as worked on by 25% contrasted with [1] and 38% contrasted with [2].

A. *Case scenario 4: Makespan*

able 4 .Tabulation of makespan using MA-DNN, RAA-PI-NSGAII [1] and DQN [2]

Number of user requested tasks	**Makespan (ms)**		
	MA-DNN	RAA-PI-NSGAII	DQN
100	5	7.3	8.2
200	7.35	11.15	12.55
300	10.25	13.55	15
400	11.35	15.85	18.15
500	13	17.55	25
600	14.55	21.35	28.35
700	19	25	31
800	21.35	28.25	33.25
900	25	31	35

In this section, makespan referred to as the total time taken to process a set of jobs (i.e., task scheduling time and resource allocating time) for its complete execution is presented. The mathematical formulation for makespan is given below.

$$MS = \sum_{i=1}^{n} T_i * Time\ [TS + RA] \quad (19)$$

From the above equation (19), makespan 'MS' is measured based on the tasks involved in the simulation process 'T_i', the time consumed in both task scheduling and resource allocation '$Time\ [TS + RA]$' respectively. It is measured in terms of milliseconds (ms). The result of the simulation for makespan consumption is listed in table 4.

Figure 7 shows the assessment of execution measurements of underlined task planning and asset portion in CC climate concerning makespan. The exhibition metric makespan is considered as the most basic explicitly for asset allotment in CC climate. The previously mentioned measurement is assessed regarding milliseconds. It shows that

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 4s
DOI: https://doi.org/10.17762/ijritcc.v11i4s.6303
***Article Received**: 19 December 2022 **Revised**: 22 January 2023 **Accepted**: 02 February 2023*

the time consumed in planning client mentioned task and correspondingly dispensing the asset significantly contributes in the absolute season of serving demands. Be that as it may, time consumed in planning client solicitation and distribution of virtual machine is assuming a significant part in normal reaction time. This is attributable to the sending of client mentioned assignments towards the neighboring clients. In a presentation perspective, lower upsides of all previously mentioned makespan execution measurements are liked for better help provisioning and then again, higher makespan may corrupt and consequently disintegrate the whole exhibition. The makespan was viewed as relatively lesser utilizing Mama DNN strategy than [1] and [2]. The purpose for the improvement was attributable to the use of planning undertakings with the relating assets made by utilizing Kronecker Delta capability. By utilizing this capability settle time space data associated with planning, consequently joining these two portrayal together outcomes in the minimization of makespan utilizing Mama DNN strategy by 26% contrasted with [1] and 37% contrasted with [2] separately.

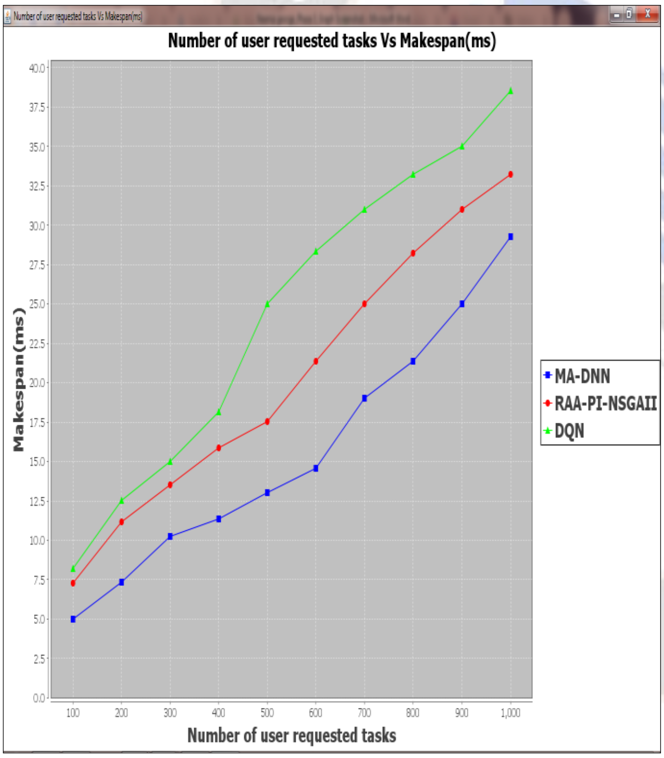

Figure 7 .Comparison results of makespan

VI. Conclusion

Due to several objectives of factors taking into consideration during task scheduling and involvement of distinct constraints during allocation of resources, scheduling efficiency and energy consumption in the data center remains to be focused. Also, owing to the reason that the same types of tasks and resources have to be allocated by the cloud server in an optimal manner, hence task scheduling between cloud user requests is required to reduce task scheduling time, energy consumption and ensure fast processing. In this paper, a method called, Multi-objective Auto-encoder Deep Neural Network-based (MA-DNN) in CC environment is proposed. The proposed task scheduling and resource optimization method provides information of the cloud user requested tasks to the cloud server by processing the generated data. After processing, a Hypervolume-based Sen's Multi-objective Optimized Task Scheduling is designed that schedules the tasks in a computationally efficient manner even in case of distinct factors (i.e., time and cost) by means of multi-objective function using Hypervolume-based Sen's Multi-objective programming model. Finally, employing Auto-encoder Deep Neural Network-based Resource allocation algorithm, optimal and energy efficient resource allocation is made between scheduled tasks. Simulations were performed to evaluate the performance of MA-DNN, RAA-PI-NSGAII, and DRL for cloud service provisioning. Simulation results revealed that the proposed MA-DNN method out performs RAA-PI-NSGAII, and DRL for cloud service provisioning implementations.

References

[1] Jing Chen, Tiantian Du, Gongyi Xiao, "A multi-objective optimization for resource allocation of emergent demands in cloud computing", Journal of Cloud Computing: Advances, Systems and Applications, Springer, Oct 2021[Cloud Resource Allocation Algorithm based on a Parallel and Improved Non-dominated Sorting Genetic Algorithm II (RAA-PI-NSGAII)]

[2] Zhiping Peng, Jianpeng Lin, Delong Cui, Qirui Li, Jieguang He, "A multi-objective trade-off framework for cloud resource scheduling based on the Deep Q-network algorithm", Cluster Computing, Springer, Jan 2020 [deep reinforcement learning (DRL)]

[3] Sanjaya Kumar Panda, Shradha Surachita Nand, Sourav Kumar Bhoi, "A pair-based task scheduling algorithm for cloud computing environment", Journal of King Saud University – Computer and Information Sciences, Elsevier, Oct 2018

[4] Vadlamaani , S. M., P. K. . Bharti, and M. R. . Kumar. "Generalized Statistical Indicators For Cloud Computing Fault Tolerance". International Journal of Intelligent Systems and Applications in Engineering, vol. 10, no. 1s, Oct. 2022, pp. 269 -281.

[5] D. Selvapandian, R. Santhosh, "A Hybrid Optimized Resource Allocation Model for Multi-Cloud Environment Using Bat and Particle Swarm Optimization Algorithms", Institute of Fundamental Technological Research, Oct 2022

[6] Abid Ali and Muhammad Munwar Iqbal, "A Cost and Energy Efficient Task Scheduling Technique to Offload Microservices Based Applications in Mobile Cloud Computing", IEEE Access, May 2022

[7] Priyadarsini Karthik, Karthik Sekhar, "Resource scheduling approach in cloud Testing as a Service using deep reinforcement learning algorithms", CAAI Transactions on Intelligence Technology, Wiley, Feb 2021

[8] Said Nabi, Muhammad Ibrahim, Jose M. Jimenez, "DRALBA: Dynamic and Resource Aware Load Balanced Scheduling Approach for Cloud Computing", IEEE Access, Apr 2021

[9] Chu-ge Wu, Wei Li, Ling Wanga, Albert Y. Zomaya, "An evolutionary fuzzy scheduler for multi-objective resource allocation in fog computing", Future Generation Computer Systems, Elsevier, Feb 2021

[10] Hadeer Mahmoud, Mostafa Thabet, Mohamed H. Khafagy, Fatma A. Omara, "Multiobjective Task Scheduling in Cloud Environment Using Decision Tree Algorithm", IEEE Access, Mar 2022

[11] Hoa Tran-Dang, Sanjay Bhardwaj, Tariq Rahim, Arslan Musaddiq, and Dong-Seong Kim, "Reinforcement Learning based Resource Management for Fog Computing Environment: Literature Review, Challenges, and Open Issues", Journal of Communications and Networks, Vol. 24, No. 1, Feb 2022

[12] M. O. Agbaje, O. B. Ohwo, T. G. Ayanwola, and Ogunyolu Olufunmilola, "A Survey of Game-Theoretic Approach for Resource Management in Cloud Computing", Journal of Computer Networks and Communications, Hindawi, Apr 2022

[13] A. S. Abohamama, Amir El-Ghamry, Eslam Hamouda, "Real-Time Task Scheduling Algorithm for IoT-Based Applications in the Cloud–Fog Environment", Journal of Network and Systems Management, Springer, May 2022

[14] Amira A. Amer, Ihab E. Talkhan, Reem Ahmed, Tawfik Ismail,, "An Optimized Collaborative Scheduling Algorithm for Prioritized Tasks with Shared Resources in Mobile-Edge and Cloud Computing Systems", Mobile Networks and Applications, Springer, Feb 2022

[15] Arif Ullah, Aziza Chakir, "Improvement for tasks allocation system in VM for cloud datacenter using modified bat algorithm", Multimedia Tools and Applications, Springer, Mar 2022

[16] S. Phani Praveen, K. Thirupathi Rao, B. Janakiramaiah, "Effective Allocation of Resources and Task Scheduling in Cloud Environment using Social Group Optimization", Arabian Journal for Science and Engineering, Springer, Oct 2017

[17] Mahendra Bhatu Gawali, Subhash K. Shinde, "Task scheduling and resource allocation in cloud computing using a heuristic approach", Journal of Cloud Computing: Advances, Systems and Applications, Springer, Oct 2018

[18] Varun Jasuja, Rajesh Kumar Singh "Enhanced MBFD Algorithm to Minimize Energy Consumption in Cloud", International Journal of Computer Engineering in Research Trends, , vol. 6, no. 2, Feb. 2019, pp. 266 -271.

[19] Harvinder Singh, Anshu Bhasin, Parag Ravikant Kaveri, "QRAS: efficient resource allocation for task scheduling in cloud computing", Applied Sciences, Springer, Jul 2021

[20] Huafeng Yu, "Evaluation of cloud computing resource scheduling based on improved optimization algorithm", Complex & Intelligent Systems, Springer, Sep 2020.