International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *26 August 2023* ***Revised:*** *20 October 2023* ***Accepted:*** *02 November 2023*

Bivariate Correlative Modest Adaptive Boost Resource Aware Task Scheduling in Cloud Environment

Mrs.J.Radha,

Assistant Professor, Department of Computer Applications, PSG College of Arts & Science, Coimbatore

Dr.V.Saravanan,

Professor & Head, Department of Information Technology, Hindusthan college of Arts and Science, Coimbatore

Abstract

Cloud computing is a rapidly evolving paradigm that provides accessible and virtualized resources through Internet technology. In this model, Cloud Service Providers (CSPs) offer online access to computing resources for users to execute their application tasks. Task scheduling in cloud computing involves the allocation of computational tasks to available resources within the cloud environment. The primary objectives of task scheduling are to optimize resource utilization, minimize task completion time, and enhance overall system performance. Task scheduling plays a vital role in cloud resource management as it directly impacts the efficiency of cloud data centers. With the increasing number of cloud computing users, task scheduling has become more challenging, requiring the use of suitable scheduling algorithms. To improve task scheduling efficiency, a novel method called Bivariate Correlative Modest Adaptive Boost Resource Optimized Task Scheduling (BCMABROTS) is developed for efficient service provisioning in the cloud environment. The BCMABROTS method begins by collecting the number of incoming user-requested tasks. After that, the resources of virtual machines, such as energy, bandwidth, memory, and CPU, are measured. The ensemble classifier constructs the set of weak learners as the nearest prototype centroid classifier. The weak classifier uses Bivariate Correlation to classify the resource-optimal virtual machine based on the available resources. The results of the weak learners are combined to provide strong classification results. Once the optimal virtual machine is categorized, the task is scheduled for that particular virtual machine by the task assigner. This approach ensures efficient cloud service provisioning with minimal time consumption. Experimental evaluation is carried out to assess factors such as task scheduling efficiency, makespan, throughput and average response time in relation to the number of cloud-requested tasks. The observed performance results confirm that the BCMABROTS method improves the task scheduling efficiency, throughput and minimizes the makespan as well as the average response time than the conventional machine learning methods.

Keywords**:** Cloud computing, Resource aware task scheduling, Modest Adaptive Boosting Classification, bivariate correlationanalysis, nearest prototype centroid classifier

1. Introduction

Cloud computing (CC) has emerged as a new paradigm for hosting and delivering services via the Internet. Resource-aware task scheduling plays a crucial role in effectively assigning and scheduling tasks across virtual machines within a cloud computing environment. The primary objective is to optimize resource utilization, minimize task completion time, and improve overall system performance. To achieve resource-aware task scheduling in the cloud, various techniques and algorithms, including heuristics, optimization algorithms, and machine learning approaches, have been utilized. The aim is to intelligently allocate tasks to resources by considering factors such as resource availability, task requirements, energy efficiency, and Quality of Service (QoS) constraints.

A Task Scheduling-Decision Tree (TS-DT) algorithm was designed in [1] to assign and schedule tasks within an application. The algorithm aimed to reduce the makespan and optimize resource utilization. However, it did not achieve the desired level of scheduling efficiency. The Meta Reinforcement Learning (MRL) on task scheduling in

cloud computing (MRLCC) method was developed in [2] to maintain a high utilization rate and minimize makespan. But the response time of the cloud user tasks was not reduced.

A resource-aware dynamic task scheduling approach was designed in [3] to enhance throughput and minimize response time. However, it failed to efficiently perform task and resource-aware scheduling on virtual machines (VMs). The task scheduling approach described in [4] was designed for deep neural network applications. However, it failed to effectively utilize learning-based methods to achieve a balance between performance and overheads of algorithms.

A failure-aware task scheduling approach was designed in [5] with the aim of predicting the status of given tasks during runtime. However, it was observed that the approach did not lead to a reduction in response time for task scheduling. A lightweight task-scheduling method was introduced in [6] to efficiently allocate cloud resources and minimize overhead. But, the method did not achieve optimal efficiency in task scheduling for resource allocation.

A Joint Neural Network and Heuristic Scheduling (JNNHSP) were introduced in [7] which integrate a neural network into the scheduling process. The aim of this approach is to improve the efficiency of the scheduling solution. But, the approach did not result in a reduction of the makespan.

A bi-objective algorithm for performance and energy optimization was designed in [8] specifically to enhance task scheduling. However, the algorithm did not incorporate CPU and memory-intensive task scheduling. The Efficient Task Scheduling Algorithm (EPETS) was designed in [9] to reduce execution time. However, energy-efficient task scheduling was not incorporated into the algorithm. The self-adapting task scheduling algorithm (ADATSA), designed in [10], utilized learning automata to achieve priority scheduling. However, the throughput was not improved.

1.1 Major contributions

To overcome the issues identified from the literature review, a BCMABROTS method technique is introduced with the following contributions as follows,

- To improve the resource aware task scheduling, a novel BCMABROTS method is introduced based on ensemble classification approach.
- The BCMABROTS method leverages the bivariate correlative modest adaptive boost technique to categorize virtual machines based on bivariate correlation analysis with multiple resources. The method employs the nearest prototype centroid classifier to initially classify virtual machines according to resource availability. By utilizing a strong classifier, the method achieves improved classification results with minimal generalization error.
- After selecting a resource-efficient virtual machine, the task assigner schedules the tasks to that virtual machine based on its higher throughput, aiming to provide a proper response with minimum time.
- An extensive experiment is conducted to estimate the performance of the BCMABROTS method and other related works with different evaluation metrics.

1.2 Paper organization

The paper is structured into different sections. The related works are discussed in brief in Section 2. The description of the proposed BCMABROTS algorithm with the block diagram is outlined in Section 3. The experimental evaluation is summarized in Section 4 followed by the results are discussed. Finally, the paper is concluded in Section 5.

2. Related works

An energy-saving task scheduling approach was designed in [11] using a greedy approach under a cloud environment. However, it failed to improve resource utilization in the task scheduling approach. A Directed Acyclic Graph (DAG)-based task scheduling approach was designed in [12], utilizing deep reinforcement learning and graph convolution network to minimize the makespan. But it failed to address large-scale virtual application management problems. A novel multiclass priority algorithm for task scheduling (MCPTS) was introduced in [13], aiming to minimize the makespan, response time, and resource utilization. However, it failed to consider the energy consumption of virtual machines (VMs).

An optimistic technique was introduced in [14] for VM placement to minimize the configuration overhead. But it failed to apply the machine learning-based approach to allocate the cloud data center resources more optimally. QoS-based resource allocation and scheduling approach was designed in [15] using a swarm-based ant colony optimization. However, resource management in cloud

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *26 August 2023* ***Revised:*** *20 October 2023* ***Accepted:*** *02 November 2023*

computing was not concentrated. The Clipped Double Deep Q-learning (CDDQL) approach was introduced in [16] for task scheduling and VM distribution. But it failed to ensure faster scheduling of multiple tasks.

A multi-objective task scheduling optimization was introduced in [17] with the help of a fuzzy self-defense algorithm to improve the resource utilization rate. However, the objective of achieving higher throughput in task scheduling was a challenging problem. A Deep Reinforcement Learning (DRL) approach was designed in [18] for resource provisioning and task scheduling. But the time consumption of designed task scheduling was not reduced. A Q-learning-based task scheduling approach was designed in [19] for an energy-efficient cloud computing model with the aim of minimizing task response time. However, the approach failed to apply in large-scale cloud environments with hundreds of virtual machines. Energy-efficient dynamic task scheduling method was introduced in [20] for virtualized cloud data centers with the aim of decreasing the mean response time, and energy consumption. However, the method did not include machine learning-based task scheduling.

3. Proposal methodology

Cloud computing has become a governing paradigm for large-scale information systems. Cloud data centers consist of physical and virtual infrastructure resources which include servers, network systems, and different virtual machines. Cloud computing becomes a smart technology as it offers a massive amount of storage to access these resources through appropriate task scheduling techniques. Task scheduling in cloud computing refers to the process of assigning computational tasks to available resources in a cloud computing environment. The main aim of task scheduling is to optimize resource utilization, minimize task completion time, and improve overall system performance. Based on optimization, a novel method called the BCMABROTS is introduced.

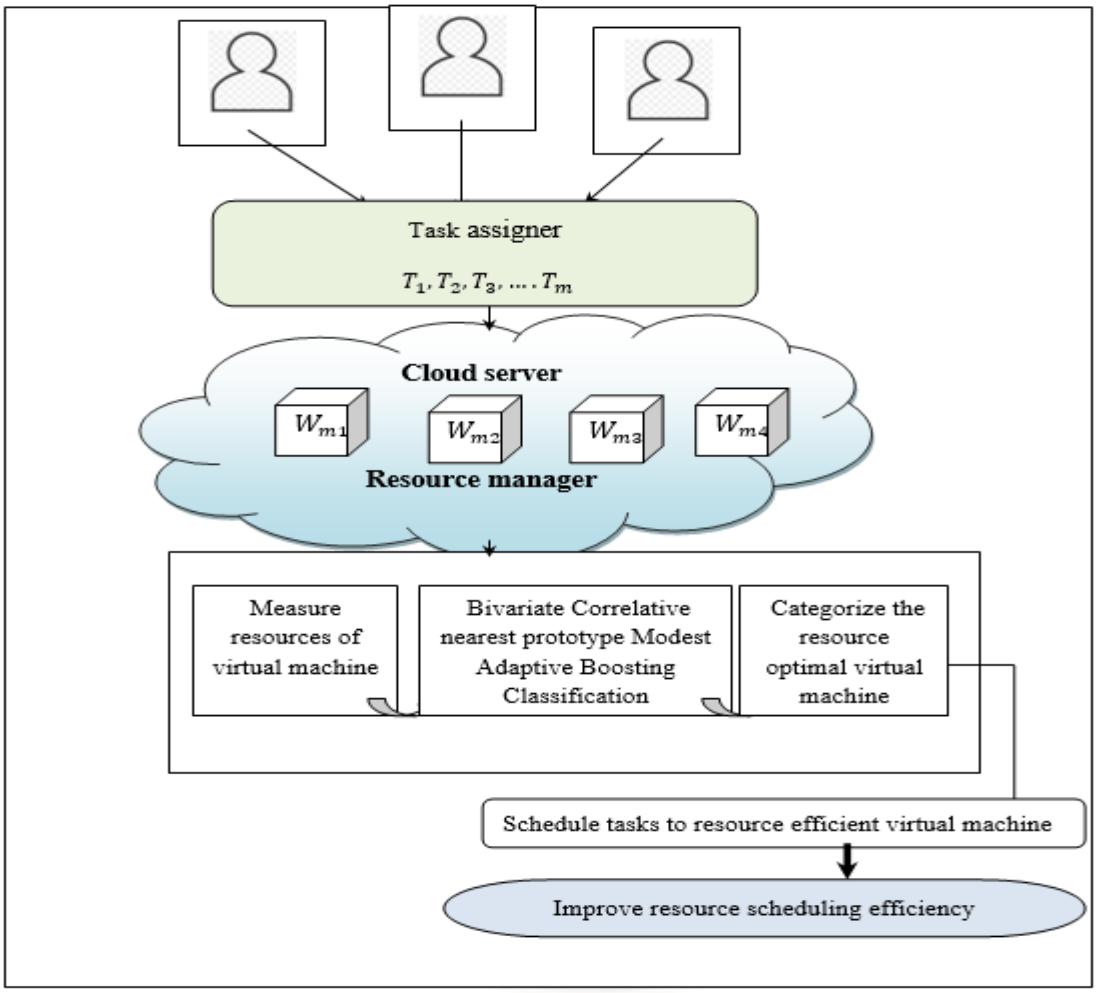

Figure 1 architecture diagram of proposed BCMABROTS method

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *26 August 2023* ***Revised:*** *20 October 2023* ***Accepted:*** *02 November 2023*

Figure 1 illustrates the architecture of the proposed BCMABROTS method for resource aware task scheduling in cloud computing. The cloud computing architecture incorporates the key components such as cloud users, cloud server, task assigner, and resource manager. The cloud task scheduling problem is defined as scheduling the various incoming tasks $T_i \in T_1, T_2, T_3, \ldots . T_m$ where, $i = 1,23,..m$ generated by the cloud users$Cu_k \in Cu_1, Cu_2, Cu_3, \ldots . Cu_k$. The cloud server ($CS$) consists of numerous virtual machines$W_{m1,}W_{m2,}W_{m3,} \ldots W_{mn,}$. In cloud server, the task manager is responsible for receiving requests from users and managing the scheduling process. It acts as an interface between users and the cloud server.

The task assigner implements the specific task scheduling algorithm. It takes input from the task assigner and makes decisions to allocate tasks to available resources such as virtual machine. The resource manager keeps track of available resources of virtual machines (VMs). It maintains information about resource capacities, current usage, and availability. The resource manager provides this information to the task manager for making informed decisions to schedule the task. The Bivariate Correlative nearest prototype Modest Adaptive Boosting Classification algorithm finds the resource aware virtual machine based on the factors such as Energy, Memory, bandwidth and CPU to allocate the incoming tasks 'T_i'. The brief explanation of the BCMABROTS method is given in the following subsections.

3.1 Bivariate Correlative nearest prototype Modest Adaptive Boosting Classification

A novel task scheduling algorithm using a multiobjective approach based on an ensemble technique is proposed, called Bivariate Correlative Modest Adaptive Boosting Classification. This machine learning ensemble technique aims to convert weak learners into strong ones, thereby improving classification performance. The weak learners in this context refer to base classifiers that struggle to accurately classify instances, while a strong learner is a classifier that exhibits a high correlation with the true classification. The main advantage of the Modest AdaBoost algorithm is used to minimize the generalization error as compared to other AdaBoost variants. The basic structure of the ensemble algorithm is given below.

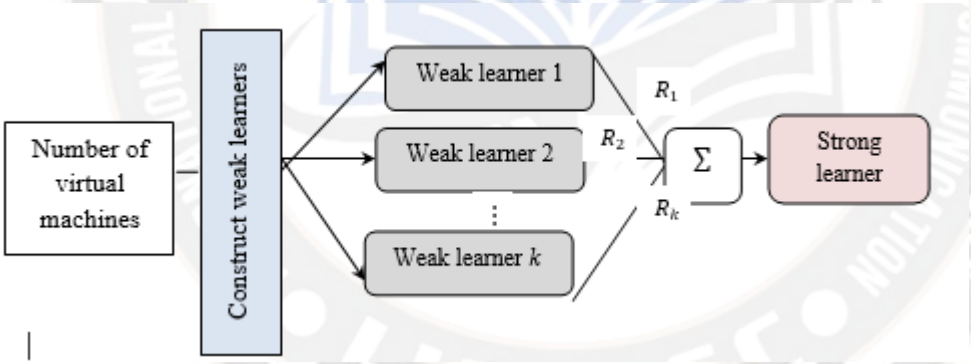

Figure 2 structure of the Bivariate Correlative Modest Adaptive Boosting Classification

Figure 2 portrays the structure of a Bivariate Correlative Modest Adaptive Boosting Classification to categorize the virtual machine $W_{m1,}W_{m2,}W_{m3,} \ldots W_{mn,}$ with efficient resource cosumption and minimum time. The ensemble classifier uses a training set $\{(W_{m,}, Y)\}$ where '$W_{m,}$' denotes the number of virtual machines, 'Y' denotes ensemble classification outcome. As shown in figure 2, the ensemble classifier initially constructs 'k' number of weak learners $Q_1, Q_2, Q_3, \ldots Q_k$ to categorize the virtual machine based on the resources. The proposed ensemble classifier uses the nearest prototype classifier model that assigns to observations the label of the class of training samples whose mean (centroid) is closest to the observation.

Let us consider the number of virtual machines $W_{m1,}W_{m2,}W_{m3,} \ldots W_{mn,}$ and the resource for each virtual machine$R_{1,}R_{2,}R_{3,} \ldots R_{m,}$. Therefore, the virtual machines with its features or resource is summarized in a matrix (M) by considering as given below,

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 26 August 2023 **Revised:** 20 October 2023 **Accepted:** 02 November 2023*

$$M = \begin{bmatrix} W_{m1}\,R_1 & W_{m2}\,R_2 & W_{m3}\,R_3 \\ W_{m4}\,R_4 & W_{m5}\,R_5 & W_{m6}\,R_6 \\ W_{m7}\,R_7 & W_{m8}\,R_8 & W_{m9}\,R_9 \end{bmatrix} (1)$$

From (1), the virtual machines with the resources are represented as a matrix 'M'. The resources for each virtual machine is considered as an Energy(φ_E), Memory (φ_M), bandwidth(φ_B), CPU (φ_{CPU}) to allocate the incoming tasks 'T_i'.

In cloud computing, energy resources are utilized to power the infrastructure and operations of the virtual machine. Therefore, the calculation for the energy availability of a virtual machine is performed as follows:

$$\varphi_{E(avl)} = [\varphi_{E_{tt}}] - [\varphi_{E_{cc}}] \quad (2)$$

Where, $\varphi_{E(avl)}$ signifies an energy availability of the virtual machine, $\varphi_{E_{tt}}$ indicates total energy, $\varphi_{E_{cc}}$ represents the consumed energy.

During the task scheduling processes, memory is another important resource for virtual machines. It represents the storage space required to handle the tasks. The availability of memory is measured as follows,

$$\varphi_{Mem(avl)} = [\varphi_{Mem(t)}] - [\varphi_{Mem(cc)}] \quad (3)$$

Where, $\varphi_{Mem(avl)}$ represents the memory availability of the virtual machine, $\varphi_{Mem(t)}$ indicates a total memory capacity of the virtual machine and $\varphi_{Mem(cc)}$ indicates a utilized memory space of a virtual machine.

The bandwidth refers to the maximum capacity of a virtual machine for processing and transferring tasks through a network connection.

$$\varphi_{B(avl)} = [\varphi_{B(t)}] - [\varphi_{B(cc)}] \quad (4)$$

From (4), $\varphi_{B(avl)}$ represents a bandwidth availability of the virtual machine, $\varphi_{B(t)}$ indicates the total bandwidth, $\varphi_{B(cc)}$ denotes a consumed bandwidth.

The CPU time of the virtual machine is determined based on the number of CPUs assigned to the task scheduling process with the shortest burst time. The CPU burst time refers to the duration taken by a CPU to complete a specific task.

$$\varphi_{cpu(time)} = T_{cc}\,(\,Exec\_task\,) \quad (5)$$

Where, $\varphi_{cpu(time)}$ represents a CPU time of the virtual machine, T_{cc} indicates the time consumed for executing the task ($Exec\_task$). It is measured in milliseconds (ms).

After measuring the resources of the virtual machine, the proposed ensemble classification algorithm constructs the weak learner with the virtual machine and their resource availability. A nearest prototype classifier model that assigns to observations the label of the class of training samples whose centroid is closest to the observation. The prototype classifier model uses the centroid value is a combination of all the resources availability values,

$$c_i = \sum\{\varphi_{E(avl)}, \varphi_{Mem(avl)}, \varphi_{B(avl)}, \varphi_{cpu(time)}\} \quad (6)$$

Where, c_i denotes a centroid value. Then the bivariate correlation function is applied for measuring the centroid 'c_i' with the estimated resource availability value 'ER_i'. The correlation (i.e. relationships) between the features and centroid are measured as given below,

$$CC = \frac{n*\sum ER_i*c_i - (\sum ER_i)(\sum c_i)}{\sqrt{[n*\sum {ER_i}^2 - (\sum c_i)^2]}\sqrt{[n*\sum\ {ER_i}^2 - (\sum c_i)^2]}} \quad (7)$$

Where, CC denotes a correlation coefficient, 'n' symbolizes a number of virtual machines. $(\sum ER_i)(\sum c_i)$ denotes a sum of the product of paired score of estimated resource and centroid, $\sum {ER_i}^2$ represents a squared score of ER_i and $\sum {c_i}^2$ represents a squared score of c_i. The correlation coefficient (CC) provides output results ranging from '-1' to '+1'. A coefficient (CC) of '+1' indicates a positive correlation, while '-1' represents a negative correlation. In the context of final classification results, positive correlated results are utilized. A positive correlation suggests a strong relationship between the estimated resource availability of a virtual machine and the centroid value. Consequently, virtual machines exhibiting positive correlation are selected for task scheduling.

With the aiming at further enhancing the accuracy of classification, the ensemble technique combines all weak classifier results.

$$Y = \sum_{i=1}^{k} R_i \quad (8)$$

Where, Y represents the output of the ensemble technique, R_i represents the output of the weak learner

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *26 August 2023* ***Revised:*** *20 October 2023* ***Accepted:*** *02 November 2023*

results. For each weak learner, initialize the weights similarly.

$$Y = \sum_{i=1}^{k} R_i \ \beta_j \quad (9)$$

Where β_j indicates the weight assigned to each weak learner results R_i. After assigning the weight, the generalized error is computed for finding the accurate classification results

$$GE = (Y_{act} - Y_{obs})^2 \quad (10)$$

Where, GE denotes a generalized error, Y_{act} denotes a actual outcome, Y_{obs} indicates a observed results. Based on the error value, weight gets updated. If the training data is correctly classified, then the input weight gets decreased. Otherwise, the weight gets increased. Subsequently, the proposed ensemble algorithm updates the weight instance using below,

$$\beta_{j+1} = \frac{\beta_j \ exp\ [-\mathrm{Y}\ R_i]}{\delta_t} \quad (11)$$

$$\delta_t = \sum_i \beta_j \ exp\ [-\mathrm{Y}\ R_i] \quad (12)$$

Where, β_{j+1} denotes a updated weight, β_j indicates a current weight, Y denotes a actual predicted results, R_i denotes a output of the weak learner results. Then final strong classifier result is acquired as follows,

$$Y = \arg min\ GE[R_i] \quad (13)$$

Where, $\arg min$ denotes an argument of minimum function, $GE[R_i]$ denotes a generalization error of classification. After that, the task scheduler assigns the incoming tasks to the resource-optimal virtual machine. This process enhances the scheduling efficiency of user incoming tasks resulting in minimized makespan. The algorithm of Bivariate Correlative nearest prototype Modest Adaptive Boosting Classification is described as given below,

Algorithm 1: Bivariate Correlative nearest prototype Modest Adaptive Boosting Classification

Input : cloud server (CS), virtual machines $W_{m1,}W_{m2,}W_{m3,}\ldots W_{mn,}$, number of tasks $T_i \in T_1, T_2, T_3, \ldots . T_m$, cloud users $Cu_k \in Cu_1, Cu_2, Cu_3, \ldots . Cu_k$, cloud service provider 'CSP' , Task assigner 'TA' , resource manager 'RM'
Output: Improve resource aware scheduling efficiency

Begin

1. **For each** virtual machines $W_{mi,}$
2. ***RM*** **computes the resource availability** Energy(φ_E), Memory (φ_M), bandwidth(φ_B), CPU (φ_{CPU})
3. Construct' empty set of weak learners $\{Q_1, Q_2, Q_3, \ldots Q_k\}$
4. Initialize centroid based on resource availability
5. **Measure bivariate correlation using (7)**
6. **If** ($CC = +1$) **then**
7. Classify the resource efficient virtual machines
8. **Else**
9. Classify the virtual machine as not resource efficient
10. **End if**
11. Combine a set of weak learners $Y = \sum_{i=1}^{k} R_i$
12. **For** each R_i
13. Assign the weight $Y = \sum_{i=1}^{k} R_i \ \beta_j$
14. Calculate generalization error 'GE'
15. Update the weight β_{j+1}
16. Find strong classification results with minimum error $Y = \arg min\ GE[R_i]$
17. TA schedules the tasks to resource efficient virtual machine
18. **end for**
19. **End for**

End

Algorithm 1 described above outlines the procedure for task scheduling in a cloud environment. Initially, the resource manager calculates the availability of resources on the virtual machine. Following that, the ensemble technique

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *26 August 2023* ***Revised:*** *20 October 2023* ***Accepted:*** *02 November 2023*

constructs sets of weak learners using the estimated resources of the virtual machines. The weak learner categorizes the virtual machines based on the bivariate correlation function. Subsequently, the ensemble technique combines the results from the weak learners, and weights are assigned to each weak learner's results. The generalization error is then computed for each observed result, and the weights are updated accordingly based on the generalization error value of each weak learner. Finally, the algorithm obtains strong classification results with minimal error. The task manager in the cloud schedules incoming tasks to the selected virtual machine with better resource availability. This process enhances task scheduling efficiency and minimizes the makespan.

4. Experimental Setup

In this section, experimental evaluations of the proposed BCMABROTS method and existing TS-DT [1] and MRLCC [2] are conducted using the Java language and the CloudSim network simulator. To perform the experiments, the Personal Cloud Datasets are used and it available at http://cloudspaces.eu/results/datasets for resource optimal task scheduling in cloud computing. The dataset consists of 17 attributes (or columns) and 66,245 instances. Out of the 17 attributes, two attributes, namely time zone and capped, were not utilized in our experiments. The remaining attributes were employed for task scheduling across multiple virtual machines in the cloud server. These attributes include row id, account id, file size (task size), operation_time_start, operation_time_end, operation_id, operation type, bandwidth trace, node_ip, node_name, quoto_start, quoto_end, quoto_total (storage capacity), capped, failed, and failure info.

5. Comparative performance analysis

This section presents a performance analysis of the proposed BCMABROTS method along with the existing methods, TS-DT [1] and MRLCC [2]. Various metrics, including task scheduling efficiency, makespan, memory consumption, and throughput, are used to evaluate and compare the performances of these three methods. The results are presented in a table format and supported by graphical representations.

5.1 Impact of task scheduling efficiency

The task scheduling efficiency metric is used to measures how effectively tasks are scheduled and allocated resources within the cloud server. Higher task scheduling efficiency indicates a more optimized and effective scheduling strategy. The task scheduling efficiency is determined by accurately assigning incoming user tasks to resource-efficient virtual machines. The calculation for task scheduling efficiency is as follows:

$$Task_{SE} = \left(\frac{n_{CS}}{n}\right) * 100 \quad (14)$$

Where, $Task_{SE}$ indicates a task scheduling efficiency, n_{CS} indicates the number of tasks correctly scheduled, n denotes the number of tasks. The task scheduling efficiency is measured in terms of percentage (%).

Table I Task Scheduling Efficiency

Number of user tasks	Task Scheduling Efficiency (%)		
	BCMABROTS	**TS-DT**	**MRLCC**
1000	97.5	91.2	93.7
2000	97	90	92.75
3000	96.66	89.16	92.66
4000	96.25	88.55	92.75
5000	95.8	88.3	92.64
6000	95.66	87.58	91.91
7000	95.42	87.5	91.6
8000	94.81	86.93	90.68
9000	94.22	86.66	90.22
10000	93.66	85.42	89.12

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *26 August 2023* ***Revised:*** *20 October 2023* ***Accepted:*** *02 November 2023*

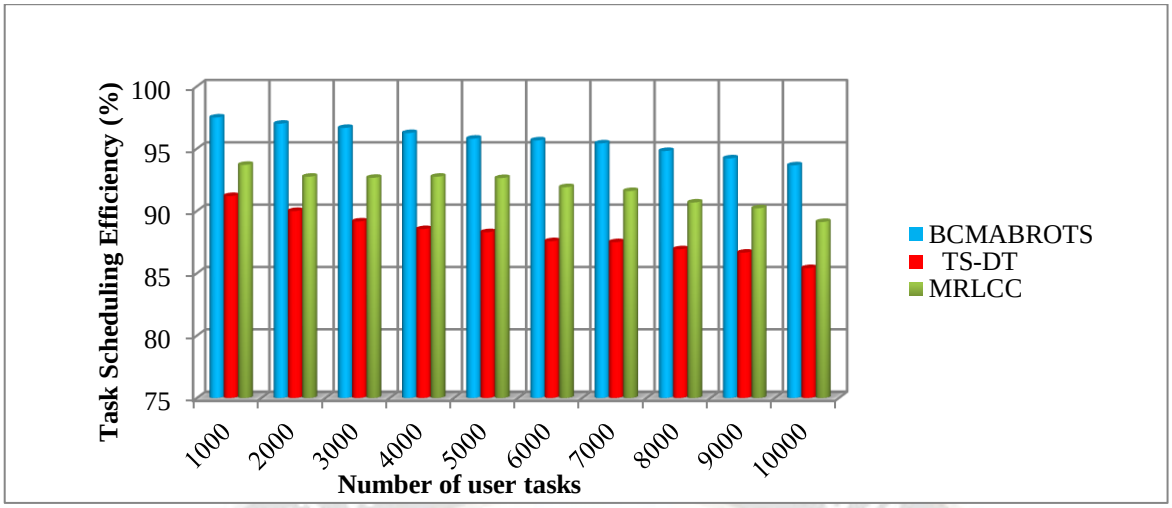

Figure 3 task scheduling efficiency versus number of user tasks

Figure 3 reveals the performance results of task scheduling efficiency using three different methods namely BCMABROTS method, along with the existing methods TS-DT [1] and MRLCC [2]. As shown in the graph, the number of user tasks ranging from 1000 to 10000 is represented on the horizontal axis, while the task scheduling efficiency is observed on the vertical axis. Among the three methods, the BCMABROTS method demonstrates higher task scheduling efficiency compared to the other existing classifiers. Let's consider 1000 tasks to calculate the task scheduling efficiency. The efficiency of the BCMABROTS method is observed to be 97.5%, while the efficiencies of the existing methods [1] and [2] are 91.2% and 93.7% respectively. Similar results are observed for different numbers of tasks in each method. Finally, the performance of the BCMABROTS method is compared to the existing methods. The average value of all the results indicates that the BCMABROTS method significantly increases the efficiency by 9% and 4% compared to existing methods [1] and [2] respectively. This is because the BCMABROTS method utilizes the adaptive boost resource ensemble classification technique to classify resource-efficient virtual machines based on energy, bandwidth, memory, and CPU. The task manager assigns incoming user-requested tasks to the selected resource-efficient virtual machine, thereby enhancing task scheduling efficiency.

5.2 Impact of Makespan:

The makespan refers to the total duration or completion time required to execute a set of tasks or processes. It refers to the duration required to schedule user-requested tasks onto virtual machines. Mathematically, it can be calculated as follows:

$$M = n * t\,(SST) \quad (15)$$

Where M indicates a makespan, n represents the number of tasks, t indicates a time, SST denotes the scheduling of one task. Makespan is measured in terms of milliseconds (ms).

Table 2 Makespan

Number of user tasks	Makespan (ms)		
	BCMABROTS	TS-DT	MRLCC
1000	26	33	28
2000	32	38	35
3000	36.6	45	39
4000	40	48	44

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *26 August 2023* ***Revised:*** *20 October 2023* ***Accepted:*** *02 November 2023*

5000	45	49.5	47.5
6000	51	60	54
7000	52.5	63	56
8000	54.4	64	60
9000	56.7	67.5	64.8
10000	60	78	75

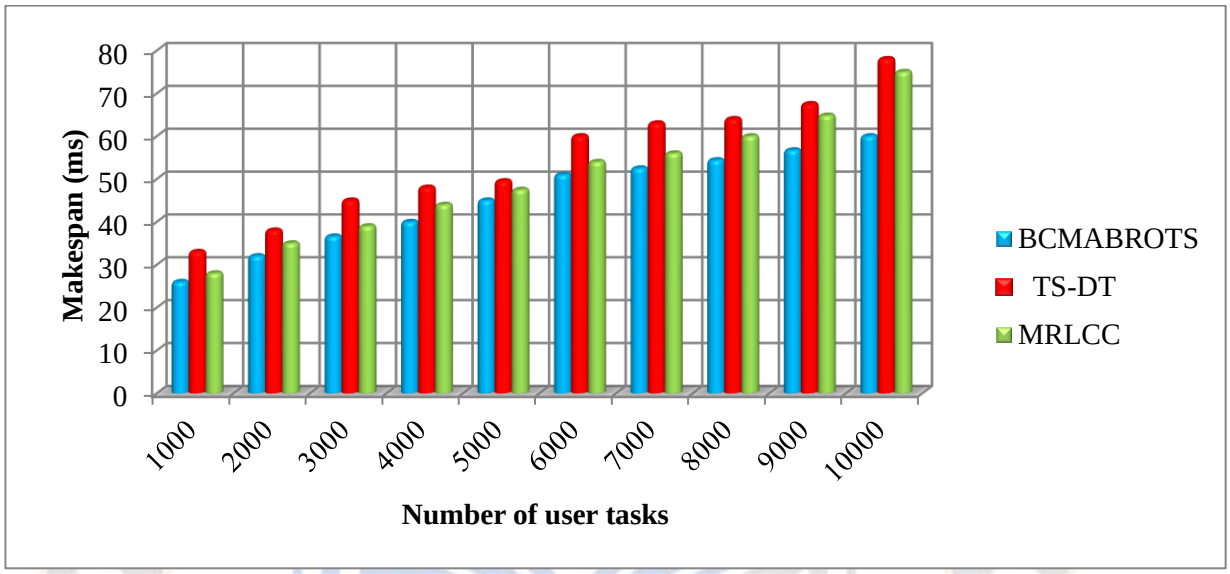

Figure 4 Makespan versus number of user tasks

Figure 4 illustrates the graphical representation of the Makespan for different methods: BCMABROTS method, existing TS-DT [1], and MRLCC [2], based on the number of tasks ranging from 1000 to 10000. As the number of tasks increases, the time required for task scheduling also increases. However, the BCMABROTS method shows a comparatively lower Makespan. This improvement is achieved by utilizing the modest adaptive boost ensemble technique to classify virtual machines. The ensemble classifier employs weak learners as nearest prototype centroid classifiers. The centroid classifier uses bivariate correlation to categorize the resource-optimal virtual machine and schedule tasks with minimal time consumption.

Let's consider the first iteration with 1000 tasks to calculate the Makespan. The time consumption of proposed BCMABROTS method was found to be $26ms$, while the existing techniques [1] and [2] was consumed $33ms$ and $28ms$, respectively. Similar comparisons were made for ten different scenarios. The comparative results demonstrate that the BCMABROTS method significantly reduces the Makespan by 17% and 9% compared to the existing techniques [1] and [2], respectively.

5.2 impact of Throughput

Throughput refers to the rate or speed at which tasks are executed within a system. "It is the measure of the rate at which user-requested tasks are performed in a given period of time within a cloud environment. Throughput is measured as follows.

$$TPT = \left[\frac{Number\ of\ tasks\ executed}{time\ (sec)}\right] \quad (16)$$

Where TPT denotes a throughput ad it measured in terms of tasks per second (tasks/sec).

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *26 August 2023* ***Revised:*** *20 October 2023* ***Accepted:*** *02 November 2023*

Table 3 Throughput

Number of user tasks	Throughput (tasks/sec)		
	BCMABROTS	**TS-DT**	**MRLCC**
1000	242	185	210
2000	535	355	415
3000	612	410	485
4000	745	455	512
5000	865	515	625
6000	985	624	745
7000	1120	745	974
8000	1165	945	1036
9000	1210	1026	1122
10000	1422	1085	1278

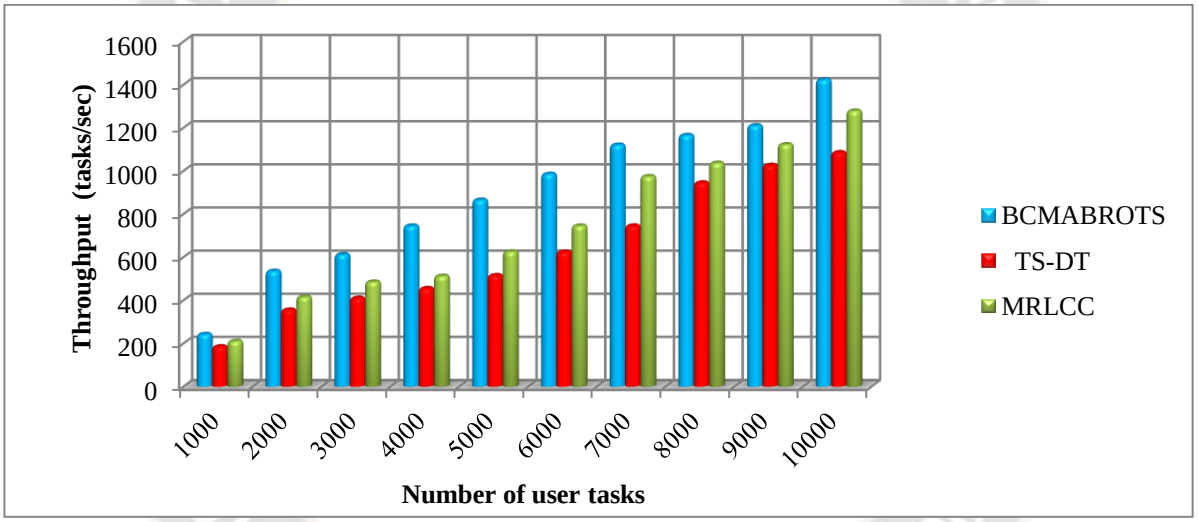

Figure 5 Throughput versus number of user tasks

The comparison of throughput during the task scheduling process, considering different numbers of tasks, is presented in Figure 5. The graphical results clearly demonstrate that the BCMABROTS method improves throughput compared to existing methods. This improvement is achieved due to the selection of virtual machines based on maximum bandwidth availability, which enhances the speed of task scheduling in the cloud environment. Let's take the example of 1000 tasks in the first iteration. The throughput performance was observed to be 242 $tasks/sec$, while the existing techniques [1] and [2] achieved the throughputs of 185 $tasks/sec$ and 210 $tasks/sec$, respectively. In total, ten different results were observed for varying numbers of tasks. These results consistently indicate that the proposed BCMABROTS method outperforms the existing techniques. The average of these ten results demonstrates that the throughput is improved by 44% compared to TS-DT [1] and by 23% compared to MRLCC [2].

5.3 **Impact of average Response time**:

Average Response time measures the time taken to response the user requested tasks in cloud. It is mathematically computed as a difference between the task finishing time and the arriving time of the tasks.

$$ART = ft_{task} - ar_{task} \quad (17)$$

Where ART indicates an average response time, 'ft_{task}' denotes the finishing time of the task and 'ar_{task}' indicates arriving time of user-requested tasks. Therefore, an average response time is measured in milliseconds (ms).

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *26 August 2023* ***Revised:*** *20 October 2023* ***Accepted:*** *02 November 2023*

Table 4 average response time

Number of user tasks	Average response time (ms)		
	BCMABROTS	TS-DT	MRLCC
1000	115	212	180
2000	132	262	220
3000	202	322	265
4000	313	432	385
5000	412	523	446
6000	475	580	500
7000	536	635	585
8000	632	765	712
9000	785	892	836
10000	823	966	924

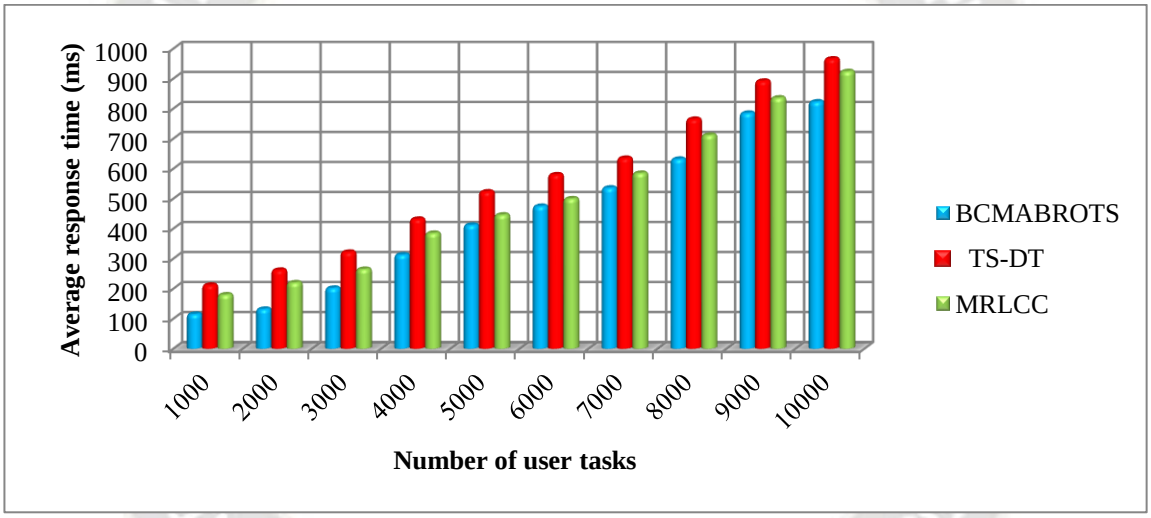

Figure 6 average response time versus number of user tasks

Figure 6 presents experimental results comparing the average response time of three different methods: BCMABROTS method, existing TS-DT [1], and MRLCC [2]. The graph shows that the average response time increases for all three methods when the number of tasks is increased. However, the BCMABROTS technique demonstrates a comparatively reduced average response time. This improvement is achieved by employing an ensemble technique that takes into account the number of virtual machines and their resource availability. The technique categorizes the virtual machines based on bivariate correlation and assigns incoming tasks to resource-efficient virtual machines. Consequently, the response time of tasks requested by cloud users is minimized. The average response time of the BCMABROTS method is minimized by 26% and 17% compared to [1] and [2], respectively.

6. Conclusion

The BCMABROTS (Bivariate Correlative Modest Adaptive Boost Resource Aware Task Scheduling) method is developed in a cloud environment to optimize task scheduling by considering resource awareness. In a cloud data center, a large number of VMs are deployed to serve end-user requests. However, configuring a VM takes a significant amount of time, leading to performance degradation. To address these issues, the BCMABROTS method is proposed. It employs a Modest Adaptive Boost ensemble strategy to schedule tasks across VMs and dynamically execute tasks in a dynamic environment. Moreover, BCMABROTS aims to effectively schedule tasks by considering different resources. The approach utilizes

bivariate correlation analysis and adaptive boosting techniques to dynamically allocate resources and improve scheduling efficiency. The proposed BCMABROTS method is implemented and evaluated through a comprehensive performance analysis using personal cloud datasets. The results demonstrate that the BCMABROTS method outperforms existing works by improving task scheduling efficiency, throughput, and minimizing makespan as well as average response time.

References

[1] Hadeer Mahmoud, Mostafa Thabet, Mohamed H. Khafagy, Fatma A. Omara, "Multiobjective Task Scheduling in Cloud Environment Using Decision Tree Algorithm", IEEE Access, Volume 10, 2022, Pages 36140 – 36151. DOI: 10.1109/ACCESS.2022.3163273

[2] Xi Xiu, Jialun Li, Yujie Long & Weigang Wu, "MRLCC: an adaptive cloud task scheduling method based on meta reinforcement learning", Journal of Cloud Computing, Springer, Volume 12, 2023, Pages 1-12. https://doi.org/10.1186/s13677-023-00440-8

[3] Said Nabi, Muhammad Ibrahim, Jose M. Jimenez, "DRALBA: Dynamic and Resource Aware Load Balanced Scheduling Approach for Cloud Computing", IEEE Access, Volume 9, 2021, Pages 61283 – 61297. DOI: 10.1109/ACCESS.2021.3074145

[4] Zheyi Chen, Junqin Hu, Xing Chen, Jia Hu, Xianghan Zheng, Geyong Min, "Computation Offloading and Task Scheduling for DNN-Based Applications in Cloud-Edge Computing", IEEE Access, Volume 8, 2020, Pages 115537 – 115547. DOI: 10.1109/ACCESS.2020.3004509

[5]Yanal Alahmad, Tariq Daradkeh, Anjali Agarwal, "Proactive Failure-Aware Task Scheduling Framework for Cloud Computing", IEEE Access, Volume 9, 2021, Pages 106152 – 106168. DOI: 10.1109/ACCESS.2021.3101147

[6] Thomas Dreibholz and , Somnath Mazumdar, "Towards a lightweight task scheduling framework for cloud and edge platform", Towards a lightweight task scheduling framework for cloud and edge platform", Internet of Things, Elsevier, Volume 21, 2023, Pages 1-16. https://doi.org/10.1016/j.iot.2022.100651

[7] Zhuo Chen, Peihong Wei, Yan Li, "Combining neural network-based method with heuristic policy for optimal task scheduling in hierarchical edge cloud", Digital Communications and Networks, Elsevier, Volume 9, Issue 3, 2023, Pages 688-697. https://doi.org/10.1016/j.dcan.2022.04.023

[8] Huned Materwala and Leila Ismail, "Performance and energy-aware bi-objective tasks scheduling for cloud data centers", Procedia Computer Science, Elsevier, Volume 197, 2022, Pages 238–246. https://doi.org/10.1016/j.procs.2021.12.137

[9] Mehboob Hussain, Lian-Fu Wei, Abdullah Lakhan, Samad Wali, Soragga Ali, Abid Hussain, "Energy and performance-efficient task scheduling in heterogeneous virtualized cloud computing", Sustainable Computing: Informatics and Systems, Elsevier, Volume 30, 2021, Pages 1-12. https://doi.org/10.1016/j.suscom.2021.100517

[10] Lilu Zhu, Kai Huang, Yanfeng Hu, Xianqing Tai, "A Self-Adapting Task Scheduling Algorithm for Container Cloud Using Learning Automata", IEEE Access ,Volume 9, 2021, Pages 81236 – 81252. DOI: 10.1109/ACCESS.2021.3078773

[11] Shuaishuai Liu ,Xinyu Ma, Yuanfei Jia, and Yue Liu, "An Energy-Saving Task Scheduling Model via Greedy Strategy under Cloud Environment", Wireless Communications and Mobile Computing, Hindawi, Volume 2022, April 2022, Pages 1-13. https://doi.org/10.1155/2022/8769674

[12] Hyunsung Lee, Sangwoo Cho, Yeongjae Jang, Jinkyu Lee, Honguk Woo, "A Global DAG Task Scheduler Using Deep Reinforcement Learning and Graph Convolution Network", IEEE Access , Volume 9, 2021, Pages 158548 – 158561. DOI: 10.1109/ACCESS.2021.3130407

[13] Hicham Ben Alla, Said Ben Alla , Abdellah Ezzati, Abdellah Touhaf, "A novel multiclass priority algorithm for task scheduling in cloud computing", The Journal of Supercomputing, Springer, Volume 77, 2021, Pages 11514–11555. https://doi.org/10.1007/s11227-021-03741-4

[14] Mehak Sharma, Mohit Kumar & Jitendra Kumar Samriya, "An optimistic approach for task scheduling in cloud computing", International Journal of Information Technology, Springer, Volume 14, 2022, Pages 2951-2961. https://doi.org/10.1007/s41870-022-01045-1

[15] Harvinder Singh, Anshu Bhasin & Parag Ravikant Kaveri, "QRAS: efficient resource allocation for task scheduling in cloud computing", SN applied science, Volume 3, 2021, Pages 1-7. https://doi.org/10.1007/s42452-021-04489-5

[16] Shashank Swarupa, Elhadi M. Shakshukia, Ansar Yasar, "Task Scheduling in Cloud Using Deep Reinforcement Learning", Procedia Computer Science,

Elsevier, Volume 184, 2021, Pages 42-51 . https://doi.org/10.1016/j.procs.2021.03.016

[17] Xueying Guo, "Multi-objective task scheduling optimization in cloud computing based on fuzzy self-defense algorithm", Alexandria Engineering Journal, Elsevier, Volume 60, Issue 6, 2021, Pages 5603-5609. https://doi.org/10.1016/j.aej.2021.04.051

[18] Toutou Oudaa, Hamza Gharsellaoui, Samir Ben Ahmed, "An Agent-based Model for Resource Provisioning and Task Scheduling in Cloud Computing Using DRL", Procedia Computer Science, Elsevier, Volume 192, 2021, Pages 3795-3804. https://doi.org/10.1016/j.procs.2021.09.154

[19] Ding Ding , Xiaocong Fan, Yihuan Zhao , Kaixuan Kang, Qian Yin, Jing Zeng, "Q-learning based dynamic task scheduling for energy-efficient cloud computing", Future Generation Computer Systems, Elsevier, Volume 108, 2020, Pages 361-371. https://doi.org/10.1016/j.future.2020.02.018

[20] Avinab Marahattaa, Sandeep Pirbhulal, Fa Zhang, Reza M. Parizid , Kim-Kwang Raymond Choo, Zhiyong Liu, "Classification-Based and Energy-Efficient Dynamic Task Scheduling Scheme for Virtualized Cloud Data Center", IEEE Transactions on Cloud Computing , Volume 9, Issue 4, 2021, Pages 1376 – 1390. DOI: 10.1109/TCC.2019.2918226