*IOSR Journal of Computer Engineering (IOSR-JCE)*
*e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 20, Issue 3, Ver. I (May. - June. 2018), PP 05-16*
*www.iosrjournals.org*

# Softmax Bootstrap Ensembled Multiclass Scheduling for Cloud Service Provisioning

*Sivagami KC[1], Dr. Sureshkumar C[2]
[1]*Ph. D Research Scholar Dept. of Computer Science, Periyar University, Salem*
[2]*Principal, Dr. Nagarathinam's College of Engineering, Namakkal*
*Corresponding Author: Sivagami KC*

***Abstract:*** *Job scheduling is an important process in cloud environment to schedule the tasks and to minimize resource utilization. Recently, a lot of research works have been designed for job scheduling using different data mining techniques. The efficiency of existing job scheduling techniques was lower due to misclassification results. In order to solve this limitation, Softmax Bootstrap Ensembled Multiclass Scheduling (SBEMS) Technique is proposed. The SBEMS Technique employs multiple learning algorithms to obtain better predictive performance for job scheduling in cloud environment. The SBEMS Technique is developed by combining Sofmax regression tree classification with boostrap aggregative ensemble method. The SBEMS Technique creates 'M' number of Softmax Regression Tree (SRT) for each user job request arrival to cloud server. The result of each SRT is considered as weak learner. After that, SBEMS Technique combines the results of all weak learners with helps of voting scheme in order to obtain strong learner. The strong learner efficiently finds resource and load optimized VM among multiple VMs on cloud server to schedule job and thereby providing user desired cloud services. Therefore, SBEMS Technique attains improved quality of cloud service provisioning and also balances the loads among multiple VMs as compared to state-of-the-art works. The SBEMS Technique conducts experimental results with help of metrics such as job scheduling efficiency, job scheduling time, energy utilization and resource utilization rate with respect to a number of jobs. The experimental result shows that SBEMS Technique is able to improve the job scheduling efficiency and also decreases the resource utilization rate for cloud service provisioning as compared to state-of-the-art works.*

***Keywords:*** *Boostrap Aggregative Ensemble Method, Job Scheduling, Resource Utilization, Softmax Regression Tree, Strong Learner, Weak Learners*

---

Date of Submission: 07-05-2018 Date of acceptance: 26-05-2018

---

## I. Introduction

Cloud computing is Distributed Computing paradigm which provides services to the customers. The cloud assists user to carry out their jobs with minimum costs. Job scheduling is a considerable problems to be solved in cloud for resource management. The aim of job scheduling in cloud computing is to distribute the task on processors. Different types of job scheduling algorithms have been introduced using many data mining techniques. The classification performance of existing job scheduling techniques was not sufficient. Therefore, this research work focus on designing a novel techniques for efficient job scheduling and resource allocation in cloud environment.

Hadoop scheduler was presented in [1] for scheduling task based on size. The scheduling performance was not effectual to reduce resource consumption. The max-min-cloud algorithm was intended in [2] to optimize the performance of the cloud service through workload allocation. The time complexity involved during workload scheduling was higher.

A Hybrid algorithm was designed in [3] by combining ACO and cuckoo search to lessening energy utilization of cloud service provisioning. The efficiency of job scheduling was poor. A Nash Bargaining Approach was presented in [4] to optimize resource utilization of server. The scheduling time was more.

Greedy task scheduler was developed in [5] to perform energy-efficient task scheduling. The task-scheduling efficiency was not adequate. A review of different techniques designed for scheduling task in cloud computing was analyzed in [6]. Dynamic task scheduling scheme was intended in [7] to enhance the total availability of the virtualized data center. Resource and the workload scheduling were not solved.

A novel approach was designed in [8] to distinguish various types of workloads and to balance the load among the nodes of virtual machines with higher energy efficiency. Multi-Objective Approach was presented in [9] for workflow scheduling in cloud environments. A QoS-based resource provisioning framework was designed in [10]. The scheduling problem was not solved efficiently.

A Dynamic Resource Scheduling Method was intended in [11] to enhance the resources scheduling efficiency and the quality of service (QoS) of cloud computing. The effect of improving the quality of service was not addressed. The problem involved in energy effectiveness of service provisioning on different cloud architectures was solved in [12].

In order to addresses above motioned limitations of existing techniques, SBEMS Technique is developed. The main contributions of SBEMS Technique are formulated as follows,

- To obtain improve classification performance for scheduling jobs in cloud environment, SBEMS Technique is designed. The SBEMS Technique unites results of multiple base learners to increase classification performance for effectively addressing job scheduling and resource optimization issues in cloud on the contrary to existing works.
- To reduce the resource utilization and to balance the loads of VMs on cloud server, SBEMS Technique allocate user requested job to VM which have higher exponential value. The exponential value based on priority and amount of resource utilized to execute user job on VM. This exponential value assists for SBEMS Technique to efficiently perform task scheduling through classification.

The remaining structure of the paper is formulated as follows. Section 2 portrays the different techniques developed for task scheduling and resource allocation. In Section 3, the proposed SBEMS Technique is explained with help of architecture diagram. The experimental settings and comparative results analysis of proposed SBEMS Technique is shown in Section 4 and Section 5. Section 6 presents the conclusion of the paper.

## II. Related Works

Adaptive cost-based task scheduling (ACTS) was designed in [11] that focuses on providing data access for executing each task with maintained costs. A load-balancing problem was needed to be addressed for providing efficient cloud computing services. An Efficient Scheduling was presented in [12] to schedule tasks with minimal execution cost.

A heuristic approach was introduced in [13] for performing task scheduling and resource allocation in cloud computing. A novel technique was developed in [14] in order to solve the challenge faced during resource management and task scheduling process in cloud environment. The optimal scheduling and load balancing was not attained in this technique.

A systematic review of diverse techniques designed using priority based job scheduling in cloud computing was presented in [15]. A Min-Min and Max-Min Algorithm was introduced in [16] for carried outing task scheduling in cloud environment. This algorithm does not manage huge numbers of cloud users during task scheduling process.

Ant Colony Optimization (ACO) algorithm was intended in [17] for allocating the incoming jobs to the virtual machines in cloud with higher efficiency. The computational time required for job scheduling was more. A Smarter MQS model was designed in [18] in order to get a high degree of job scheduling in cloud computing environment. This model does not consider decreasing the load on scheduler.

A novel task scheduling algorithm was introduced in [19] based on total processing power of the available resources and users' requests. The cost of task scheduling was not addressed in this algorithm. A survey of different existing scheduling algorithms in cloud computing was analyzed in [20] to improve their performance.

In order to overcome above mentioned existing issues, SBEMS technique is introduced which is explained below section.

## III. Softmax Bootstrap Ensembled Multiclass Scheduling Technique

Job scheduling is a significant problem to be solved in cloud computing. Allocation of workloads efficiently to VMs, increases cloud service performance. Therefore, Softmax Bootstrap Ensembled Multiclass Scheduling (SBEMS) Technique is designed. The main objective of SBEMS Technique is to improve the quality of cloud service provisioning through job scheduling with minimal resource utilization. The SBEMS is a machine learning technique where multiple weak learners are trained to solve the job scheduling and resource optimization problem in cloud environment. On the contrary to existing machine learning approaches, SBEMS Technique constructs a number of learners which are called as base learners to get higher classification performance. The SBEMS Technique is designed with application of Softmax regression tree (SRT) and bootstrap aggregative ensemble learning. The bootstrap aggregative ensemble learning helps SBEMS Technique to improve classification results of SRT for efficient job scheduling and resource optimization. The architecture diagram of Softmax Bootstrap Ensembled Multiclass Scheduling (SBEMS) Technique is shown in below.

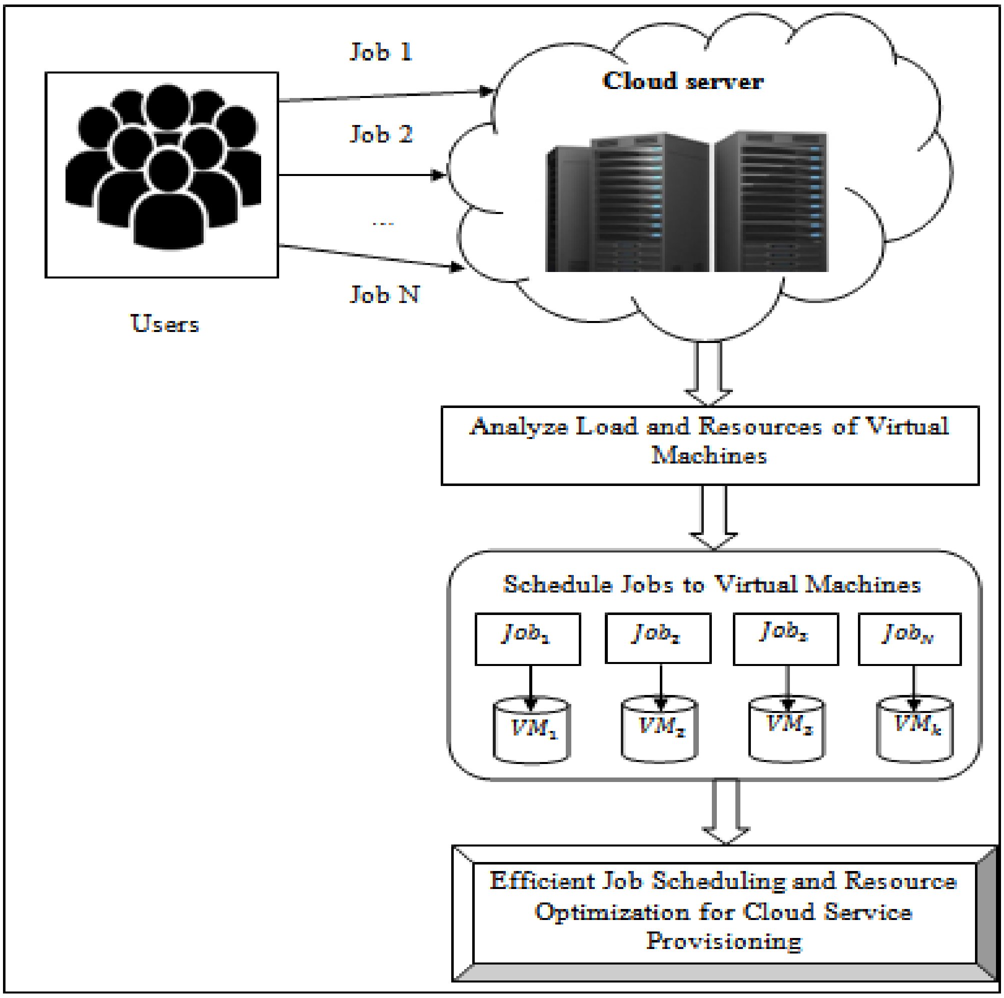


**Figure 1** Architecture Diagram of SBEMS Technique for Cloud Service Rendering

Figure 1 shows the overall process of SBEMS Technique to attain improved performance for job scheduling and resource optimization to render user requested services in cloud environment. As demonstrated in figure, the job requests are sent to cloud server. Then, load and resources utilization of each virtual machine is analyzed using SBEMS Technique and a job is scheduled to resource and load optimized VM on cloud server to complete user jobs. Hence, SBEMS Technique obtains efficient job scheduling and resource optimization for cloud service provisioning as compared to state-of-the-art works. The detailed process of SBEMS Technique is elaborated as follows.

SBEMS Technique is an ensemble method. SBEMS Technique combines the results from multiple machine learning algorithms collectively to formulate more accurate results than any individual model. SBEMS Technique reduces the variance of algorithm that has high variance (i.e. softmax regression tree) using bootstrap aggregative ensemble method. The process involved in SBEMS Technique is depicted in below Figure 2.

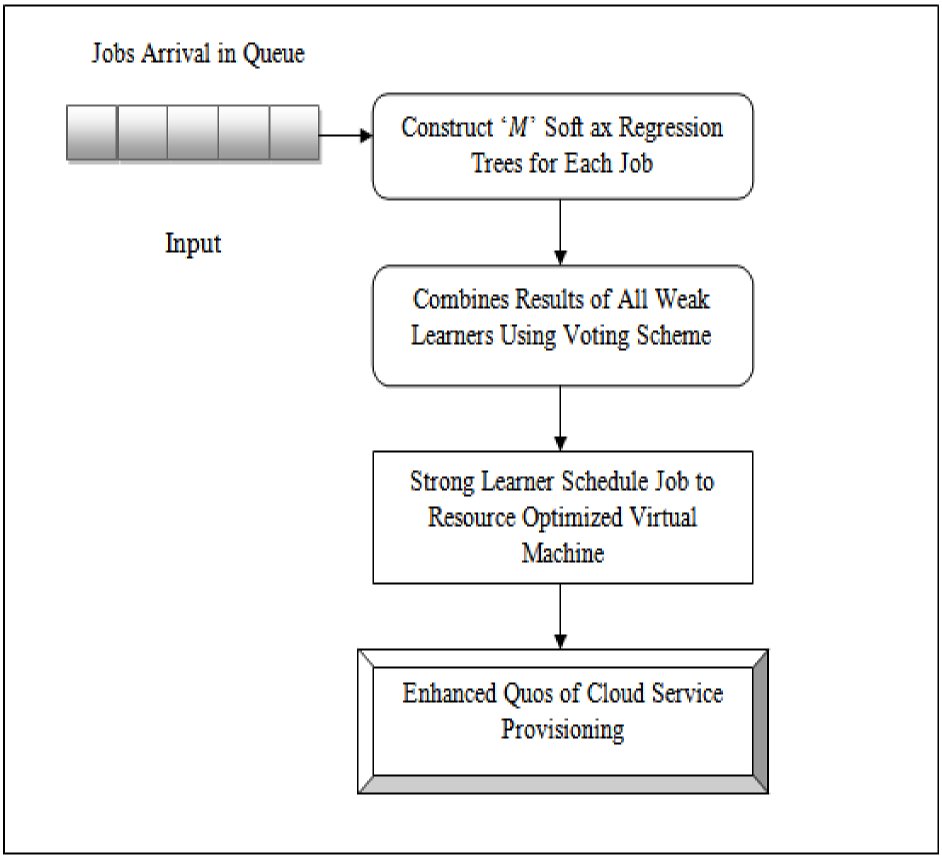


**Figure 2** Flow Processes of SBEMS Technique for Scheduling User Tasks to Multiple Virtual Machines

Figure 2 explains process of SBEMS Technique to get better Qos for rendering user requested services in cloud computing environment with lower resource utilization. As shown in figure, SBEMS Technique at first builds '$M$' number of Softmax Regression Tree (SRT) for each job arrival in queue at time '$t$'. In SBEMS Technique, the SRT is considered as weak learner or base learner in order to increases the classification performance of scheduling tasks to multiple virtual machines with minimal time complexity. Then, SBEMS Technique aggregates the results of all weak learners with application of voting scheme to find strong learner. At last, strong learner significantly schedule the task to resource optimized VM among multiple VMs on cloud server to provide user needed services. Thus, SBEMS Technique achieves enhanced Qos of cloud service provisioning as compared to state-of-the-art works.

Let us consider '$N$' virtual machines are available in cloud server. The jobs arrivals to cloud server is represented as '$J_i = (J_1, J_2, J_3, \ldots J_n)$' and it is necessary to schedule the jobs to available virtual machines. Each VM requires resources such as CPU, energy, memory, bandwidth to perform user job on cloud server. The SBEMS Technique used SRT to perform job scheduling and resource allocation process in cloud environment.

The job of users is executed on diverse virtual machines in cloud server. For an each virtual machine '$VM_i$' in cloud server, the priority is assigned based on residual workload capacity. Here, the workload refers to number of tasks executed on virtual machines. From that, priority '$P_{VM_i}$' is assigned to each '$VM_i$' in cloud server using below mathematical formulation as,

$$P_{VM_i} \rightarrow VM_{RW} = VM_{TC} - VM_{ET} \qquad (1)$$

From equation (1), '$VM_{RW}$' represent the residual workload capacity of '$VM_i$' on cloud server. Here, '$VM_{TC}$' refers the total capacity of virtual machine in which $VM_{ET}$ denotes the number of executing tasks (i.e. load) on VM of cloud server. The sum of loads on VM is formulated as,

$$VM_{ET} = \sum_{i=1}^{n} J_i \qquad (2)$$

From equation (2), '$n$' represent the number of task running on VM. By using equation (6) and (7), priority is allocated to each VM in cloud based on residual workload capacity. The VM which has the higher residual workload capacity has higher priority than other VMs in cloud server.

Followed by, SBEMS Technique measures resources utilization of virtual machine '$VM_i$' on cloud server to provide user requested services to corresponding users. The SBEMS Technique considers the energy, bandwidth, memory and CPU time of VM for rendering cloud services. Resource optimized cloud service provisioning significant to optimize the cost and thereby ensuring the higher Qos. The proposed SBEMS Technique finds the resource optimized VM to perform user task through classification.

In SBEMS Technique, energy determines an amount of power consumed by a VM to render the services to user at particular time instance. The energy utilization of VM '$EU_{VM_i}$' to carried out user job '$Job_i$' is mathematically formulated as,

$$EU_{VM_i} = P(Job_i) * t \quad (3)$$

From equation (3), '$EU_{VM_i}$' indicates energy consumed for service provisioning in cloud environment. The energy consumption of VM is estimated in terms of a joule (J). Here, $P(Job_i)$ denotes power utilized to perform user job on VM and '$T$' denotes a time of service provisioning. Besides, SBEMS Technique computes the bandwidth utilization of VM '$BU_{VM_i}$' for providing cloud service using below mathematical expression,

$$BU_{VM_i} = [B_I - BU_{J_i}] \quad (4)$$

From equation (4), '$B_I$' represents the initially available bandwidth of VM and '$BU_{Job_i}$' refers an amount of bandwidth utilized by VM to execute user job $J_i$ on cloud server.

Further, SBEMS Technique computes a total amount of storage space taken by VM to afford the user requested services. The memory consumption of VM '$MU_{VM_i}$' is measured in terms of mega bytes (MB). The amount of memory space utilized by a VM to complete user task is mathematically obtained as,

$$MU_{VM_i} = [M_I - M_U] \quad (5)$$

From equation (5), memory utilization is calculated as differences between the initial memory size '$M_I$' and unused memory '$M_U$'. Subsequently, SBEMS Technique measures the CPU time that estimates amount of time taken by a VM to accomplish the user needed job. The CPU time is mathematically determined as,

$$CT_{VM_i} = T(E(J_i)) \quad (6)$$

From equation (6), '$T(E(J_i))$' refers the time needed for VM to complete user job. By using the equation (3), (4), (5) and (6), SBEMS Technique determines the resource required for performing user jobs on VMs. The SBEMS Technique applies SRT based classification to schedule job to the VM which have highest priority and minimum resource consumption among multiple VMs in cloud server.

A SRT is used in SBEMS Technique in which it handles multiple VMs to schedule jobs arrivals to cloud server. The SRT analyzes the user jobs and identifies the VM to complete user requested service with minimal amount of resource utilization. On the contrary to state-of-the-art works, the SRT is employed in SBEMS Technique to handle multiple VMs for scheduling the jobs and efficient resource optimization for service provisioning in cloud computing environment.

SRT includes number of training set '$\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$'. Here '$x_1$' represents number of jobs arrival at cloud server and '$y_1$' indicates possible output class '$\{1,2,3, \ldots . k\}$' (i.e. resource optimized VM among multiple VMs to execute user job). The SRT based classification used Softmax Regression in order to determine the relationships among arrival user job and VMs to schedule the workload. The softmax regression function is obtained using below expression,

$$S(J_n) = \frac{\exp(VM_i)}{\sum_{i=1}^{n} \exp(J_i)} \quad \text{Where } i = 1, 2, 3 \ldots k \quad (7)$$

From equation (7), $S(F_n)$ denotes a softmax function of arrival user jobs $(J_n)$ at a particular time '$t$'. The output of softmax function measures probability distribution over $k$ different VMs to schedule to job to particular VM based on priority and resource utilization. The probability distribution (i.e. exponential value of each '$VM_i$' is measured based on priority and resource utilized to perform user job '$J_i$') is determined as,

$$P(y = i|J) = \frac{\exp(VM_i)}{\sum_{i=1}^{n} \exp(J_i)} \quad i = 1, 2, 3 \ldots k \quad (8)$$

$$Y = \begin{bmatrix} P(y = 1|J_i) \\ P(y = 2|J_i) \\ . \\ . \\ . \\ P(y = k|J_i) \end{bmatrix} = \frac{1}{\sum_{i=1}^{n} \exp(J_i)} \begin{bmatrix} \exp(VM_1) \\ \exp(VM_2) \\ . \\ . \\ . \\ \exp(VM_k) \end{bmatrix} \quad (9)$$

From equation (8) and (9), SRT identifies the particular VM among '$k$' number of VMs on cloud server using exponential value. Then the user job is allocated to a particular VM which has largest exponential value to render user needed services. The VM with higher priority and lower resource consumption has higher exponential value than VMs in cloud server. In the same way, all the incoming jobs of user are scheduled to VMs in cloud server to provide cloud services with minimum resource utilization. The SRT based classification technique distributes incoming workloads (i.e. jobs) to available VMs. However, misclassification errors may occur due to the probability distribution results of softmax regression. The misclassification error lacks the efficiency of job scheduling in cloud environment. In order to enhance the classification performance of cloud job scheduling, bootstrap aggregative ensemble technique is combined with SRT in proposed technique. This bootstrap aggregative ensemble technique helps for SBEMS to combines the result of all weak learners to make strong learner with application of voting scheme which shown in below Figure 3.

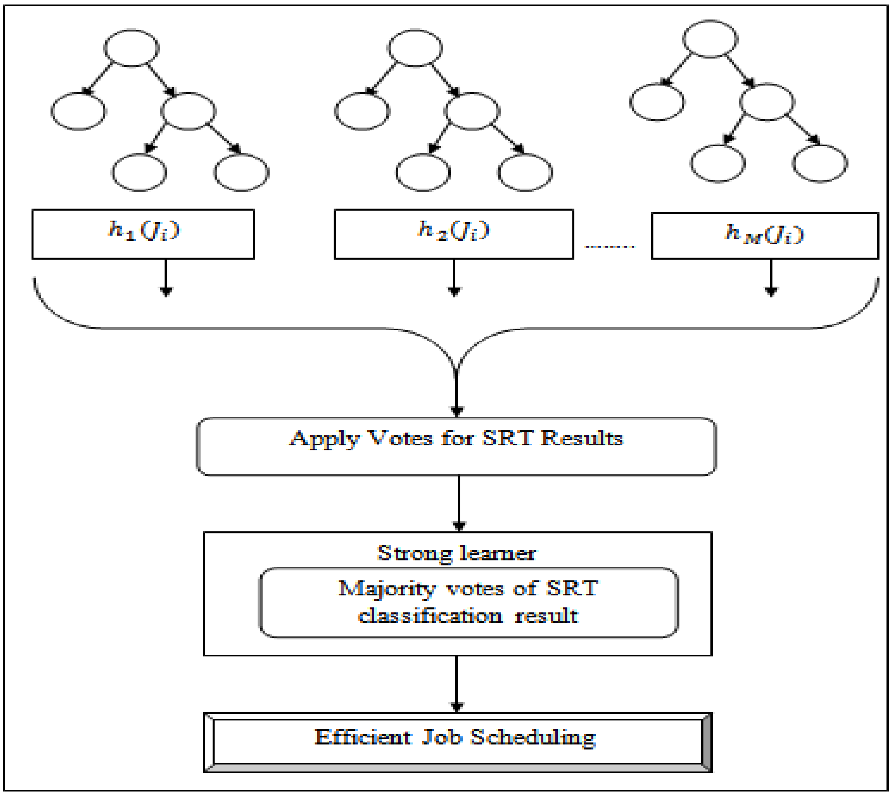


**Figure 3 Processes of Bootstrap Aggregative Ensembling for Job Scheduling**

As shown in Figure 3, SBEMS technique create '$M$' number of Softmax Regression Trees for each incoming job to cloud server. After construction, SBEMS technique aggregates output of the all the weak learners ($M$' number of Softmax Regression Trees) into a one strong classifier. The ensemble of all the weak learners is mathematically obtained as,

$$h(J_i) = h_1(J_i) + h_2(J_i) + \cdots + h_M(J_i) \qquad (10)$$

Followed by, SBEMS technique applies vote is applied for each weak learners '$h(J_i)$' which is determined as,

$$V_i \rightarrow \sum_{i=1}^{M} h(J_i) \qquad (11)$$

Then, majority vote of classification is employed in order to construct the strong classifier for efficiently scheduling jobs to VMs. From that, strong learner result is determined as,

$$f(x) = arg \max_{M} V(\, h(J_i)) \qquad (12)$$

From equation (12), $f(x)$ indicates the final decision of softmax regression tree ensemble with majority votes. Here '$arg \max_{n} M_n$' refers majority votes of softmax regression tree results whose decisions are known to the $n^{th}$ class. By using the strong learner, SBEMS technique allocate the jobs to VMs in cloud server with higher efficiency. The algorithmic process of Softmax Bootstrap Ensembled Multiclass Scheduling is demonstrated in below.

**// Softmax Bootstrap Ensembled Multiclass Scheduling Algorithm**
**Input:** Number of Jobs '$J_i = J_1, J_2, .. J_n$', Number of Virtual Machine '$VM_i = VM_1, VM_2, .. VM_n$'
**Output:** Enhanced Qos and optimized resource consumption for cloud service provisioning
**Step 1: Begin**
**Step 2:** **For** user job '$\boldsymbol{J_i}$'
**Step 3:** Measure Residual load for each '$\boldsymbol{VM_i}$'
**Step 4:** Assign priority '$\boldsymbol{P_{VM_i}}$' to each '$\boldsymbol{VM_i}$' based on load using (1)

**Step 5:** Compute resource need to complete jobs for each $\boldsymbol{VM_i}$ Using (3), (4), (5) (6)
**Step 6:** Construct '$\boldsymbol{M}$' Number Of Softmax Regression Trees using (7), (8), (9)
**Step 7:** **If** ('$\boldsymbol{M}$' number of trees are created) **then**
**Step 8:** All the weak learners are combined using (10)
**Step 9:** Apply vote '$\boldsymbol{V_i}$' for weak learners using (11)
**Step 10:** Obtain strong learner result using (12)
**Step 11:** Schedule the job to resource and load optimized VM on cloud server
**Step 12:** **End if**
**Step 13:** **End for**
**Step 14: End**

**Algorithm 1** Softmax Bootstrap Ensembled Multiclass Scheduling Technique for Rendering Cloud Services

Algorithm 1 shows the step by step processes of SBEMS Technique to significantly increase the classification performance of job scheduling and resource allocation in cloud computing environment. With help of the above algorithmic processes, SBEMS Technique allocate each job arriving at cloud server to the VM which have higher priority (i.e. residual workload capacity) and minimal resource consumption to perform user needed cloud services. This helps for SBEMS Technique to efficiently schedule incoming jobs to resource and load optimized VMs on cloud server with higher efficiency and lower amount of time utilization. Thus, SBEMS Technique balance workloads among multiple VMs and also decreases resource utilization rate of cloud service provisioning when compared to existing works.

## IV. Experimental Settings

To measure the performance of proposed, SBEMS Technique is implemented in Java Language with cloudsim simulator using personal cloud dataset. The SBEMS Technique used personal cloud dataset [23] for performing experimental process. The personal cloud dataset includes the following attributes such as row_id, account _id, file 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), failed and failure info. By using the above attribute information of personal cloud dataset, SBEMS Technique accomplishes job scheduling and resource allocation to provide user requested cloud services. The different size of files in personal cloud dataset is considered as jobs which are schedule to VMs in cloud server using proposed SBEMS Technique with higher efficiency and minimal amount of time consumption as compared to existing works for providing user needed service.

The efficiency of proposed SBEMS Technique is measured in terms of job scheduling efficiency, job scheduling time, energy utilization and resource utilization rate. The experimental work of SBEMS Technique is conducted for several instances with respect to various numbers of jobs in order to analyze the performances. The effectiveness of SBEMS Technique is compared with Hadoop scheduler [1] and max-min-cloud algorithm [2] respectively.

## V. Result and Discussions

In this section, the experimental result of proposed SBEMS Technique is discussed. The performance of SBEMS Technique is compared with existing two methods namely, Hadoop scheduler [1] and max-min-cloud algorithm [2] with help of parameters such as job scheduling efficiency, job scheduling time, energy utilization and resource utilization rate.

### 5.1 Performance of Job scheduling Efficiency

In SBEMS Technique, Job Scheduling Efficiency '$(JSE)$' is defined as the ratio of number of jobs that are correctly scheduled to resource optimized VMs to the total number of jobs arrival. The job scheduling efficiency is measured in terms of percentages (%) and mathematically determined as,

$$JSE = \frac{NJS_{RVMs}}{N} * 100 \tag{13}$$

From equation (13), job scheduling efficiency is determined to efficiently provide user requested cloud services with respect to different number of jobs. Here, '$N$' denotes a number of jobs arrival at cloud server whereas '$NJS_{RVMs}$' represents the number of jobs that are correctly scheduled to resource optimized VMs. When job scheduling efficiency is higher, the technique is said to be more effective.

The SBEMS Technique is implemented in Java Languages by considering a diverse number of jobs in the range of 10-100 to evaluate job scheduling efficiency in cloud environment. The experimental result of job scheduling efficiency using SBEMS Technique is compared against with existing Hadoop scheduler [1] and max-min-cloud algorithm [2]. When employing 50 number of cloud request (i.e. jobs) to conduct experimental process, SBEMS Technique gets 90 % job scheduling efficiency whereas existing Hadoop scheduler [1] and max-min-cloud algorithm [2] obtains 78 % and 80 % respectively. From that, it is expressive that the job scheduling efficiency using proposed SBEMS Technique is higher as compared to other existing methods [1], [2]. The comparative result analysis of job scheduling efficiency is demonstrated in below.

**Table 1 Experimental result of Job scheduling Efficiency**

| **Number of Jobs (N)** | **Job scheduling Efficiency (%)** | | |
|---|---|---|---|
| | **Hadoop Scheduler** | **Max-Min-Cloud Algorithm** | **SBEMS Technique** |
| 10 | 69 | 73 | 85 |
| 20 | 72 | 78 | 88 |
| 30 | 70 | 75 | 86 |
| 40 | 73 | 76 | 89 |
| 50 | 78 | 80 | 90 |
| 60 | 75 | 81 | 92 |
| 70 | 77 | 79 | 91 |
| 80 | 80 | 83 | 94 |
| 90 | 79 | 80 | 93 |
| 100 | 81 | 84 | 95 |

Table 1 presents the experimental result analysis of job scheduling efficiency along with various numbers of jobs in the range of 10-100 using three methods namely Hadoop scheduler [1] and max-min-cloud algorithm [2] and SBEMS Technique. As shown in the Table 1, proposed SBEMS Technique provides higher job scheduling efficiency for cloud service provisioning when compared to existing works namely Hadoop scheduler [1] and max-min-cloud algorithm [2]. Also while increasing the number of jobs for performing experimental work, the job scheduling efficiency is also enhanced using all the three techniques. But comparatively, job scheduling efficiency using SBEMS Technique is higher than other existing works. This is due to the application of SRT and Bootstrap aggregative ensemble mechanism in SBEMS technique.

By using the process of SRT, SBEMS Technique schedule jobs to VMs based on exponential value via classification. To further enhance the classification result of SRT, Bootstrap aggregative ensemble mechanism is applied in SBEMS Technique. With the processes of Bootstrap aggregative ensemble mechanism, SBEMS Technique combines outcomes of all weak learners using voting method. The majority votes of all weak learners are used to formulate strong learner. This strong learner gets higher classification performance for task scheduling in cloud environment. Hence, proposed SBEMS technique increases the job scheduling efficiency by 20 % and 15 % when compared to existing Hadoop scheduler [1] and max-min-cloud algorithm [2] respectively.

### 5.2 Performance of Job scheduling Time

In SBEMS Technique, Job Scheduling Time (JST) measures amount of time required for scheduling the job to a VM. The job scheduling time is estimated in terms of milliseconds (ms) and mathematically represented

$$JST = N * T(SJ) \tag{14}$$

From equation (14), scheduling time of job is computed with respect to various number of jobs arrival at cloud server. Here, '$N$' refers a number of jobs considered for experimental evaluation whereas '$T(SJ)$' indicates the time taken for scheduling one jobs to a VM. When job scheduling time is lower, the technique is said to be more effectual.

To compute amount of time needed for scheduling the jobs to a VMs, SBEMS Technique is implemented in Java Languages using various numbers of jobs in the range of 10-100. The experimental result of job scheduling time using SBEMS Technique is compared against with existing Hadoop scheduler [1] and max-min-cloud algorithm [2]. When considering 60 jobs to carry out the experimental evaluation, SBEMS Technique takes 48 ms time for scheduling the jobs to a VMs whereas existing Hadoop scheduler [1] and max-min-cloud algorithm [2] acquires 66 ms and 60 ms respectively. Thus, it is significant that the job scheduling time using proposed SBEMS Technique is lower when compared to other existing methods [1], [2]. The performance result analysis of job scheduling time is illustrated in below.

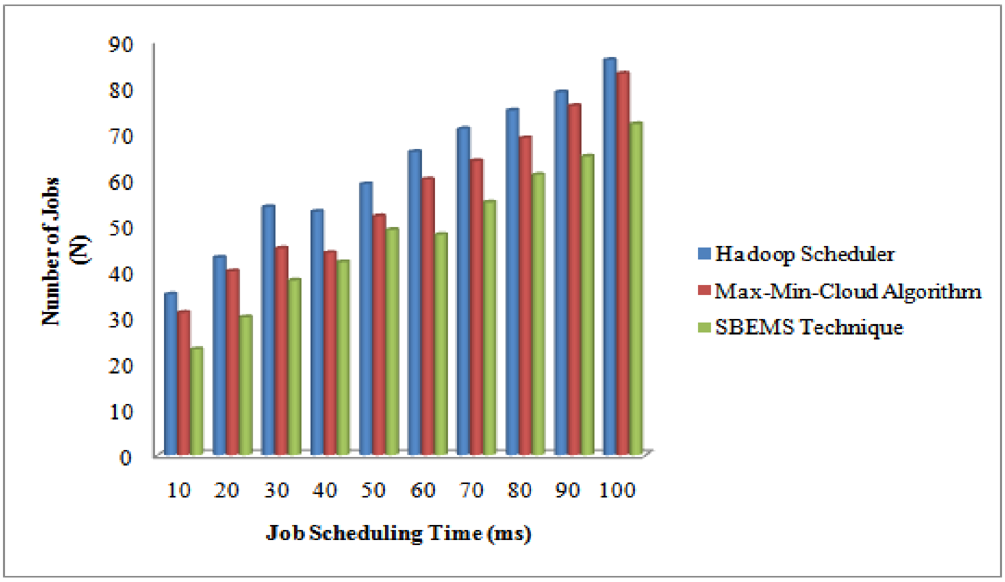


**Figure 4** Performance Result of Job scheduling Time Vs Number of Jobs

Figure 4 depicts the impact of job scheduling time with respect to diverse numbers of jobs in the range of 10-100 using three methods namely Hadoop scheduler [1] and max-min-cloud algorithm [2] and SBEMS Technique. As presented in Figure 4, proposed SBEMS Technique takes lower job scheduling time to render the requested cloud services as compared to existing works namely Hadoop scheduler [1] and max-min-cloud algorithm [2]. As well while increasing the number of jobs for conducting experimental process, the job scheduling time is also enhanced using all the three techniques. But comparatively, job scheduling time using SBEMS Technique is lower than other existing works. This is because of usage of SRT and boostrap aggregative ensemble method in SBEMS Technique.

With application of SRT classification process, SBEMS Technique assigns jobs to VMs using exponential value. To further improve the classification result of SRT, Bootstrap aggregative ensemble mechanism is employed in SBEMS Technique. With the concepts of Bootstrap aggregative ensemble mechanism, SBEMS Technique unites results of all weak learners and then applies voting method. The majority votes of weak learners are utilized to create strong learner. This strong learner obtains enhanced classification performance for job scheduling in cloud environment with minimal amount of time complexity. Therefore, proposed SBEMS technique reduces the job scheduling time by 23 % and 15 % when compared to existing Hadoop scheduler [1] and max-min-cloud algorithm [2] respectively.

**5.3 Performance of Energy Utilization**

In SBEMS Technique, Energy Utilization '($EU$)' determines the amount of energy consumed to render user requested cloud services. The energy utilization is measured in terms of joules (J) and mathematically obtained as,

$$EU = N * E(J_i) \qquad (15)$$

From equation (15), energy consumed for cloud service provisioning is evaluated with respect to diverse number of jobs. Here, '$N$' indicates a number of jobs assumed for experimental work in which '$E(J_i)$' refers amount of energy used for completing one user job. When energy utilization of cloud service provisioning is lower, the technique is said to be more efficient.

In order to measure energy used to provide user requested services in cloud environment, SBEMS Technique is implemented in Java Languages using dissimilar numbers of jobs in the range of 10-100. The energy utilization result obtained during cloud service provisioning using SBEMS Technique is compared against with existing Hadoop scheduler [1] and max-min-cloud algorithm [2]. When taking 70 numbers of jobs to accomplish experimental process, SBEMS Technique obtains 58 J amount of energy to render cloud services whereas existing Hadoop scheduler [1] and max-min-cloud algorithm [2] gets 73 J and 71 J respectively. Accordingly, it is clear that the energy utilization of cloud service provisioning using proposed SBEMS Technique is lower as compared to other state-of-the-art methods [1], [2]. The experimental result analysis of energy utilization is illustrated in below.

**Table 2 Experimental result of Energy Utilization**

| Number of Jobs (N) | Energy Utilization (J) | | |
|---|---|---|---|
| | Hadoop Scheduler | Max-Min-Cloud Algorithm | SBEMS Technique |
| 10 | 35 | 33 | 26 |
| 20 | 42 | 40 | 31 |
| 30 | 54 | 45 | 36 |
| 40 | 67 | 59 | 48 |
| 50 | 66 | 57 | 46 |
| 60 | 69 | 65 | 52 |
| 70 | 73 | 71 | 58 |
| 80 | 78 | 75 | 63 |
| 90 | 81 | 79 | 66 |
| 100 | 85 | 83 | 71 |

Table 2 explains the performance result of energy utilization along with different numbers of jobs in the range of 10-100 using three methods namely Hadoop scheduler [1] and max-min-cloud algorithm [2] and SBEMS Technique. As exposed in the Table 2, proposed SBEMS Technique used a lower amount of energy for perform user jobs on cloud server when compared to existing works namely Hadoop scheduler [1] and max-min-cloud algorithm [2]. In addition, while increasing the number of jobs for experimental evaluation, the energy utilization is also enhanced using all the three techniques. But comparatively, energy utilization using SBEMS Technique is lower than other existing works.

This is owing to application of SRT in proposed SBEMS Technique where it employs exponential value of VMs in cloud server for job scheduling. The exponential value is determined based on priority and resources such as energy, bandwidth, and memory and CPU time utilized to execute that particular job that VM. The VM which has a higher exponential value consumes minimal amount of energy for rendering user requested cloud services among other VMs in cloud server. As a result, proposed SBEMS technique reduces the energy utilization by 24 % and 19 % when compared to existing Hadoop scheduler [1] and max-min-cloud algorithm [2] respectively.

**5.4 Performance of Resource Utilization Rate**

In SBEMS Technique, Resource Utilization Rate (RUR) is evaluated as the ratio of a number of cloud resources utilized by a VM for providing cloud services to total cloud resources available. The resource utilization rate is measured in terms of percentages (%) and obtained using below mathematical representation,

$$RUR = \frac{resources\ \ utilized\ \ by\ VM}{Total\ available\ \ cloud\ \ resources} * 100 \quad (16)$$

From equation (16), resource utilization rate for cloud service provisioning is determined with respect to varied number of jobs. When resource utilization rate is lower, the technique is said to be more effective.

For determining resource utilization during process of cloud service rendering, SBEMS Technique is implemented in Java Languages by assuming varied numbers of jobs in the range of 10-100. The result of resource utilization rate using SBEMS Technique is compared against with existing Hadoop scheduler [1] and max-min-cloud algorithm [2]. While considering 100 jobs for experimental evaluation, SBEMS Technique averages obtains 72 % resource utilization rate whereas existing Hadoop scheduler [1] and max-min-cloud algorithm [2] attains 86 % and 78 % respectively. From these results, it is descriptive that the resource utilization rate of cloud service provisioning using proposed SBEMS Technique is lower when compared to other traditional works [1], [2]. The result analysis of resource utilization rate is presented in below.

**Table 3 Experimental result of Resource Utilization Rate**

| Methods | Resource Utilization Rate (%) |
|---|---|
| Hadoop Scheduler | 86 |
| Max-Min-Cloud Algorithm | 78 |
| SBEMS | 72 |

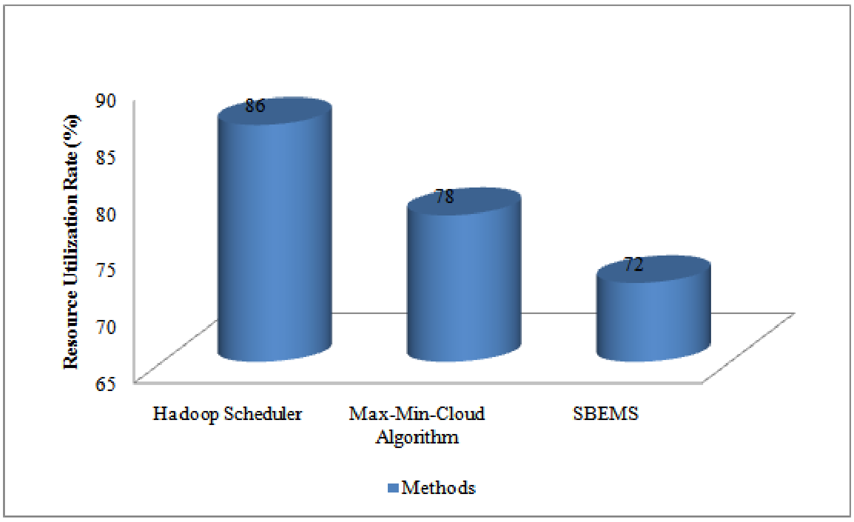


**Figure 5** Performance Result of Resource Utilization Rate for Rendering Cloud Services

Table 3 and Figure 5 illustrate the experimental result of resource utilization rate using three methods namely Hadoop scheduler [1] and max-min-cloud algorithm [2] and SBEMS Technique. As demonstrated in the Figure 5, proposed SBEMS Technique consumes minimal amount of resources to provide better quality of cloud service provisioning as compared existing Hadoop scheduler [1] and max-min-cloud algorithm [2]. This is because of the process of SRT applied in proposed technique. The SRT determines exponential value of each VM in cloud server by considering priority and resources such as energy, bandwidth, and memory and CPU time utilized to complete that particular job that VM. From that, VM with higher exponential value utilizes lower amount of resources for providing user desired cloud services as compared to other VMs in cloud server. Thus, proposed SBEMS technique minimizes the resource utilization rate by 16 % and 7 % when compared to existing Hadoop scheduler [1] and max-min-cloud algorithm [2] respectively.

## VI. Conclusion

An effective SBEMS technique is developed with goal of enhancing classification performance for solving job scheduling and resource utilization problem in cloud computing environment. The goal of SBEMS technique is achieved with assists of Sofmax regression tree classification with boostrap aggregative ensemble method. With help of boostrap aggregative ensemble method, SBEMS technique increase classification results of SRT for efficient job scheduling in cloud environment. The SBEMS technique distributes each job to resource and load optimized VM through performing classification. As a result, SBEMS technique effectively improves efficiency of job scheduling and also balances the loads on VMs for cloud service provisioning. The SBEMS technique helps for efficient utilization of VMs to render the user requested services with minimal resource utilization rate such as energy, memory, and bandwidth and CPU time when compared to state-of-the-art works. The performance of SBEMS technique is tested with the metrics such as job scheduling efficiency, job scheduling time, energy utilization and resource utilization rate and compared with state-of-the-art works. With the experimental process is conducted for SBEMS technique, it is significant that the job scheduling efficiency provides more accurate results for improving quality of services in cloud environment when compared to existing methods. The experimental result shows that the SBEMS technique provides better performance with enhancement of job scheduling efficiency and minimization of resource utilization rate for rendering user requested cloud services as compared to state-of-the-art works.

## References

[1]. Yi Yao, Jianzhe Tai, Bo Sheng, and Ningfang Mi, “LsPS: A Job Size-Based Scheduler for Efficient Task Assignments in Hadoop”, IEEE Transactions on Cloud Computing, Volume 3, Issue 4, Pages 411-424, 2015

[2]. Zhuoyao Wang, Majeed M. Hayat, Nasir Ghani, Khaled B. Shaban, “Optimizing Cloud-Service Performance: Efficient Resource Provisioning via Optimal Workload Allocation”, IEEE Transactions on Parallel and Distributed Systems, Volume 28, Issue 6, Pages 1689 – 1702, 2017

[3]. N.Moganarangan, R.G.Babukarthik, S.Bhuvaneswari, M.S.Saleem Basha, P.Dhavachelvan, “A novel algorithm for reducing energy-consumption in cloud computing environment: Web service computing approach”, Journal of King Saud University-Computer and Information Sciences, Elsevier, Volume 28, Issue 1, Pages 55-67, January 2016

[4]. Mohammad Mehedi, HassanAtifAlamri, “Virtual Machine Resource Allocation for Multimedia Cloud: A Nash Bargaining Approach”, Procedia Computer Science, Elsevier, Volume 34, Pages 571-576, 2014

[5]. Ziqian Dong, Ning Liu and Roberto Rojas-Cessa, “Greedy scheduling of tasks with time constraints for energy-efficient cloud-computing data centers”, Journal of Cloud Computing: Advances, Systems, Springer, Volume 4, Issue 5, 2015

[6]. Suman Sangwan, Sunita Sangwan, "An Effective Approach on Scheduling Algorithm in Cloud Computing", International Journal of Computer Science and Mobile Computing, Volume 3 Issue 6, Pages 19-23, 2014

[7]. Xiang zhen Kong, Chuang Lin, Yixin Jiang, Wei Yan, Xiaowen Chu, "Efficient dynamic task scheduling in virtualized data centers with fuzzy prediction", Journal of Network and Computer Applications, Elsevier, Volume 34, Pages 1068–1077, 2011

[8]. Prabhpreet Kaur, Monika Sachdeva, "Energy Efficient Task Scheduling in Cloud Computing", International Journal of Computer and Distributed Systems, Volume 4, Issue 4, Pages 132-137, 2016

[9]. Sonia Yassa, Rachid Chelouah, Hubert Kadima, and Bertrand Granado, "Multi-Objective Approach for Energy-Aware Workflow Scheduling in Cloud Computing Environments", Hindawi Publishing Corporation, The Scientific World Journal, Volume 2013, Article ID 350934, Pages 1-13, 2013

[10]. Sukhpal Singh Gill, "Resource Provisioning Based Scheduling Framework for Execution of Heterogeneous and Clustered Workloads in Clouds: from Fundamental to Autonomic Offering", Journal of Grid Computing, Springer, Pages 1–33, 2018

[11]. Zhijia Chen, Yuanchang Zhu, Yanqiang Di, and Shaochong Feng, "A Dynamic Resource Scheduling Method Based on Fuzzy Control Theory in Cloud Environment", Journal of Control Science and Engineering, Hindawi Publishing Corporation, Volume 2015, Article ID 383209, Pages 1-10, 2015

[12]. Leila Sharifi, Llorenç Cerdà-Alabern, Felix Freitag, Luís Veiga, "Energy Efficient Cloud Service Provisioning: Keeping Data Center Granularity in Perspective", Journal of Grid Computing, Springer, Volume 14, Issue 2, Pages 299–325, June 2016

[13]. Mohammed A. S. Mosleh, G. Radhamani, Mohamed A. G. Hazber, and Syed Hamid Hasan, "Adaptive Cost-Based Task Scheduling in Cloud Environment", Scientific Programming, Hindawi Publishing Corporation, Volume 2016, Article ID 8239239, Pages 1-9, 2016

[14]. Amandeep Verma, Sakshi Kaushal, "Cost-Time Efficient Scheduling Plan for Executing Workflows in the Cloud", Journal of Grid Computing, Springer, Volume 13, Issue 4, Pages 495–506, December 2015

[15]. Mahendra Bhatu Gawali, Subhash K. Shinde, "Task scheduling and resource allocation in cloud computing using a heuristic approach", Journal of Cloud Computing, Springer, Volume 7, Issue 4, December 2018

[16]. Er. Amanpreet Kaur, Bikrampal Kaur, Dheerendra Singh, "Challenges To Task and Workflow Scheduling In Cloud Environment", International Journal of Advanced Research in Computer Science, Volume 8, Issue 8, Pages 412-415, 2015

[17]. Swachil Patel, Upendra Bhoi, "Priority Based Job Scheduling Techniques In Cloud Computing: A Systematic Review", International Journal Of Scientific & Technology Research Volume 2, Issue 11, Pages 147-152, November 2013

[18]. Sanjay Kumar and Atul Mishra, "Application of Min-Min and Max-Min Algorithm for Task Scheduling in Cloud Environment Under Time Shared and Space Shared VM Models", International Journal of Computing Academic Research (IJCAR), Volume 4, Issue 6, Pages 182-190, December 2015

[19]. Medhat Tawfeek, Ashraf El-Sisi, Arabi Keshk and Fawzy Torkey, "Cloud Task Scheduling Based on Ant Colony Optimization", the International Arab Journal of Information Technology, Volume 12, Issue 2, Pages 129-137, March 2015

[20]. Jaspreet Singh and Deepali Gupta, "An Smarter Multi Queue Job Scheduling Policy for Cloud Computing", International Journal of Applied Engineering Research, Volume 12, Issue 9, Pages 1929-1934, 2017

[21]. Elhossiny Ibrahim, Nirmeen A El-Bahnasawy and Fatma A Omara, "Job Scheduling based on Harmonization between the Requested and Available Processing Power in the Cloud Computing Environment", Journal of Information Technology & Software Engineering, Volume 5, Issue 3, Pages 1-4, 2015

[22]. Fazel Mohammadi, Shahram Jamali and Masoud Bekravi, "Survey on Job Scheduling algorithms in Cloud Computing", International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 3, Issue 2, Pages 151-154, 2014

[23]. Personal Cloud Datasets: http://cloudspaces.eu/results/datasets


IOSR Journal of Computer Engineering (IOSR-JCE) is UGC approved Journal with Sl. No. 5019, Journal no. 49102.

Sivagami KC. "Softmax Bootstrap Ensembled Multiclass Scheduling for Cloud Service Provisioning." IOSR Journal of Computer Engineering (IOSR-JCE) 20.3 (2018): 05-16.