Machine Learning Approach to Select Optimal Task Scheduling Algorithm in Cloud

Chinmai Shetty[1], Dr. Sarojadevi H [2] and Suraj Prabhu [3]

[1] Assistant Professor, Dept of ISE, N.M.A.M. Institute of Technology, Nitte, Karnataka-574110, India.

[2] Professor, Dept of CSE, Nitte Meenakshi Institute of Technology, Yelahanka, Bangalore 560064, India.

[3] Programmer analyst, Oracle Managed Cloud Services, Bangalore, India.

Email: chinmai@nitte.edu.in[1], sarojadevi.n@nmit.ac.in [2] , surajrprabhu99@gmail.com [3]

ABSTRACT

The flexibility provided by the cloud service provider at reduced cost popularized the cloud tremendously. The cloud service provider must schedule the incoming requests dynamically. In a cloud environment tasks must be scheduled such that proper resource utilization is achieved. Hence task scheduling plays a significant role in the functionality and performance of cloud computing systems. While there exist many approaches for boosting the task scheduling in the cloud, it is still an unresolved issue. In this proposed framework we attempt to optimize the usage of cloud computing resources by applying machine learning techniques. The new proposed framework dynamically selects the scheduling algorithm for the incoming request rather than arbitrary assigning a task to the scheduling algorithm. The scheduling algorithm is predicted dynamically using a neural network which is the best for the incoming request. The proposed framework considers scheduling parameters namely cost, throughput, makespan and degree of imbalance. The algorithms chosen for scheduling are 1) MET 2) MCT 3) Sufferage 4)Min-min 5) Min-mean 6) Min-var. The framework includes 4 neural networks to predict the best algorithm for each scheduling parameters considered for optimization. PCA algorithm is used for extracting relevant features from the input data set. The proposed framework shows the scope for the overall system performance by dynamically selecting precise scheduling algorithms for each incoming request from the user.

Keywords: ***Cloud Computing, Machine learning, Task scheduling, Heuristic algorithms, Neural network.***

1.0 INTRODUCTION

With the rapid evolution of processing and storage technologies and the accomplishment of the internet, computing resources have become affordable, more robust and globally available than ever before. The different models of cloud computing architecture like a layered model of cloud computing, Business model which provides different services like 1)Infrastructure as a Service(IaaS) 2) Platform as a Service(PaaS) 3) Software as a Service(SaaS).The challenges also increase with the demand of emerging cloud services[1]. Efficient task scheduling to provide Quality Of Service (QOS) is a major issue.

The tasks in the cloud can be scheduled statically or dynamically. The traditional method of scheduling was time and space shared. First Come First Serve (FCFS) and Shortest Job First (SJF) scheduling methods developed later outperformed the traditional methods. Generalized priority algorithms developed in [3] performed better than SJF and FCFS .

Cloud computing is a popular service based technology in the business world nowadays. Cloud infrastructure provides users with a flexible network access for computing resources that are accessible on the internet and pay as you use. In a typical cloud environment, a scheduling algorithm is chosen beforehand and the same algorithm is used for every request made. The chosen algorithm might not be the optimal one for every request made. This leads to a game of chance, wherein for some user requests the chosen algorithm may be the best algorithm and other times there may be other algorithms which could have outperformed the default algorithm. Each algorithm considers a single objective or it may consider multiple scheduling criteria [2] depending on the scheduling algorithm. The main issue here is that the algorithm is chosen beforehand and never changed. All these algorithms are developed with different assumptions. Hence picking the suitable algorithm for solving a task assignment issue or certain nature becomes problematic. Here heuristic task scheduling algorithms based on six rules is considered. Machine learning technique for the incoming task requests to classify the best algorithm out of the six algorithms considered here for that request. The cloud environment is simulated where the scheduling best scheduling algorithm is dynamically chosen for every incoming request, which determines the best scheduling algorithm for each request on the go.

The machine learning techniques are applied in various areas including health care services and they increase the performance of health care services [4]. It maximized the cloud resource utilization by selecting appropriate VM. Machine learning has greatly performed in applications in the cloud. The new proposed model in this paper is an attempt to improve the system performance by dynamically selecting appropriate task scheduling algorithms by applying machine learning techniques. Here the Principal component analysis (PCA)is carried out to identify the suitable parameters for study. Multiple feed forward neural networks are designed for optimizing task scheduling parameters to predict the correct algorithm for scheduling the incoming request. Then the request is passed on to the corresponding Virtual Machine (VM) where the predicted scheduling algorithm is implemented. The proposed framework allows to achieve best accuracy in predicting optimization algorithms for the scheduling parameters, viz., degree of Imbalance, cost, makespan, throughput. The proposed framework contributes to the overall efficiency of the system performance through 1) dynamic selection of scheduling algorithms considering suitable workloads 2)Parameter optimization 3)PCA based feature selection.

2.0 RELATED WORK

The related work focuses on different algorithms and techniques used for task scheduling in cloud. It also includes different machine learning techniques used to address scheduling concerns in the cloud.

The performance contrast of task scheduling in an Infrastructure as a service (IaaS) cloud computing environment is done using six rule based heuristic algorithms [21]. The algorithms are implemented for homogeneous as well as heterogeneous environments by considering several parameters like throughput, degree of imbalance, cost and makespan. The results show that these six-rule based heuristic algorithms give the optimal solutions for the IaaS cloud environment. The framework designed helps the cloud users to choose the best algorithm for the performance metric they want to optimize.

To optimize the resource allocation by using priority-based task scheduling and allocating the tasks to the conflict free resources was proposed [5]. This has a better makespan and improves the throughput of the system. This improves the overall performance of the cloud environment.

The comparative study of most used task scheduling algorithms in cloud environments is carried out [6]. This study will help to select the parameters for scheduling. The heuristic techniques are the most efficient for scheduling tasks in a cloud environment as compared to other scheduling algorithms.

The heuristic approach proposed in [7] provides better Turnaround time and response time when contrasted with existing bandwidth aware divisible scheduling (BATS) and improved differential evolution algorithm (IDEA) frameworks. The tasks for the system are real Cybershake and Epigenomics scientific workflows. The cloud provider will be benefited if resources are efficiently used.

In [8] analysis of different heuristic and energy efficient scheduling algorithms for cloud is considered based on the different number of parameters taken for scheduling. The study suggests that the parameters considered for designing new scheduling algorithms can be increased in number so that it can be favourable to cloud providers also.

The survey of various heuristic scheduling algorithms for task scheduling is accomplished here [9]. The survey suggests the improvements for the emerging scheduling requirements, which include multidimensional, multi objective and dynamic scheduling.

An attempt to address scheduling problems in the cloud using a heuristic approach is accomplished in [10]. The goal was to impart uniform response time to requests(tasks). The approach used is to decrease the variance of response time by decreasing the variance of access time.

A venture to classify tasks in cloud environments using machine learning techniques is dealt in [11]. The idea behind the work proposed here is to optimize task scheduling by identifying the priority of each task and place them accordingly in different queues for further processing. To label tasks based on priority supervised machine learning techniques are used. The parameters used for comparison are accuracy, training time and speed of prediction.

The Deep Reinforcement Learning techniques are proposed for scheduling resources in an offline cloud environment [12]. DeepRM, DeepRM2 are modified to address scheduling problems of resources because basically they deal only with parameters relevant to CPU and Memory. The algorithms used here for scheduling are the shortest job first (SJF), Longest job first (LJF), Tetries and Random Algorithms. The study motivates that Deep Reinforcement Learning techniques can be used for similar optimization problems.

A framework to schedule tasks using machine learning techniques was proposed in [13]. The proposed system suggests that the best scheduling algorithm can be selected dynamically using supervised technique.

The review of task scheduling in cloud [14] lists the merits and demerits of each algorithm. The study helps the researcher to identify the parameters of scheduling in the cloud which needs more focus. The survey discloses that features such as optimization of cost and total execution time have already been covered. More focus is required towards methods for handling errors, providing reliable services and availability of services. Cloud services can be improved by implementing algorithms which consider more parameters.

Machine learning framework is applied in the field of aerospace applications [15]. The ML algorithms used are Mahout ML Library. The framework reduced the total life cycle cost by recognizing onset part failures, identifying irregularity and condition-based maintenance.

Traditional ML libraries do not support efficient large dataset [16]. To support parallel processing, ML is built upon as a SaaS. BigML, Bit Yota, Precog and Google prediction API are some of them.

In real time the QOS for resource allocation is difficult to achieve. The evaluated data for the current situation is compared with the historical data. Then, the optimal or near-optimal solution in the identical historical scenario is adopted to allocate the radio resources for the current scenario. Here a new design method is applied to allocate radio resources by using supervised machine learning techniques. [17].

A Survey of Machine Learning Applications to Cloud Computing [18] is discussed here. The article lists how ML applications can be used to improve dynamic resource management and prediction of VM size required. Artificial neural networks (ANN) and support vector machine (SVM) methods are used and it is situation dependent, which focuses on efficient resource usage and saves energy.

To optimize the performance of cloud scheduling inspired by Swarm Intelligence is proposed in [19]. Multiple criteria are considered here. The machine learning classifier chain algorithm identifies future algorithms to be used to schedule the tasks in every data center while having the best task execution time. The results improve significant performance of load balancing while scheduling.

The proposed technique in [20] for resource allocation with multidimensional requirements by using machine learning techniques. ML automatically understands both workload and system environments which help for intelligent resource allocation in cloud computing.

The proposed framework stands out in the literature from the following features: State of the artwork has focused on applying machine learning techniques for applications on cloud computing but the proposed framework is an attempt to improve the performance from the system end. We can apply machine learning to any existing scheduling algorithms to perform better. The scheduler acts intelligently by selecting appropriate scheduling algorithms. The scheduling can select any number of parameters including multi objective parameters. Even though in literature many algorithms are proposed to schedule tasks in cloud computing, machine learning is not used for scheduling prediction.

3.0 THE PROPOSED FRAMEWORK

The proposed framework is a new method of dynamically predicting the task scheduling algorithm for each incoming request. The architecture for the proposed framework is given in Fig.1. The requests are generated through the graphical user interface (GUI). The user requests IaaS from the Cloud service provider (CSP). The task is passed on to CSP, in turn to the task scheduler. The task scheduler selects the parameter to be optimized through the GUI provided. With the use of 4 neural networks the best scheduling algorithm is identified for the task. The task is allocated to the data centre where the corresponding VM is assigned. Each VM implements a different scheduling algorithm for the task and the task completes its execution.

4.0 IMPLEMENTATION

The implementation details include properties of simulating environment, data set generation and prediction method.

4.1 Simulation Parameters

The simulation was done using the CloudSim tool [22]. The dataset was generated in an environment having the following properties. This section explains the simulation setup.

No of Cloudlets in each set	: 500-750
Total No of machines	: 10-30
Length of cloudlets	: 500000- 800000
Machine capacity	: 1000 mips

4.2 Data set generation

The data set is generated using a CloudSim simulator. To simulate data a function was created which generates a set of tasks of random length between 50000-80000 mips and the number of tasks in the set is between 500-750. This set is passed to all the 6 scheduling algorithms and the details such as makespan, throughput, cost, degree of imbalance are obtained. Now all the features like number of tasks, Number of machines, Total length, Average load, Average arrival time difference are calculated, and the corresponding selected algorithm is appended. This makes up a single record. This function is iterated over 5000 times to produce a dataset for one performance metric and is stored in a .csv file. Then this entire process is repeated for the other three performance metrics which brings the total number of records to 20000.

The dataset consists of the following features:

- Number of tasks –the number of tasks to be scheduled.
- Number of machines –the number of machines on which the task has to be scheduled.
- Total length –sum of length of all the tasks.
- Average load - average load the selected machines.
- Average arrival time difference – sum of difference in arrival time between a task and the next task divided by the total number of tasks.

There are 4 datasets, one for each parameter considered (make span, throughput, cost, degree of imbalance). The data set in total contains 20000 records.

4.3 Prediction Methods

There are several methods available for the purpose of making predictions for the used data set. In this case it uses a neural network to predict the best scheduling algorithm for each metric i.e. cost, make span, throughput, degree of imbalance. Therefore, we have designed 4feed forward neural networks in which each of them predicts the best scheduling algorithm for each performance metric. The scheduling algorithms that are used here are: Minimum Execution Time (MET), Minimum Completion Time (MCT), Sufferage, Min-min, Min-mean, Min-var.

4.3.1 Performance Metrics Used

To compare the algorithms with each for a given task we have chosen 4 metrics [21].

1)Makespan: It estimates the maximum completion time, by computing the finishing time of the current task. It is characterized as the amount of time, from start to finish for completing a set of jobs. A good scheduling algorithm always tries to reduce the makespan. It is one of the most commonly used criteria for measuring scheduling efficiency in cloud computing.

$$Makespan = Max_{taski}(Fnsh_{time})$$

Where, Fnsh $_{time}$ represents the finish time of the i^{th} task.

2) Throughput: In a cloud environment, throughput means some tasks are accomplished in a convinced time period. Throughput is the amount of work completed in a unit of time. It can also be described as the total number of cloudlets or tasks that are executed successfully within a given time period in cloud computing.

$$Throughput = \sum_{Task\,i} (\mathrm{Exe}_{\mathrm{time}})$$

where Exe Time shows the execution time of i^{th} task

3)Degree of Imbalance: Degree of imbalance (DI) describes the amount of load distribution amongst the VMs regarding their execution competencies. Degree of imbalance is used for calculating the load of work in data centres. In simple terms, it is the ability to handle or process work in cloud computing.

$$DegreeOfImbalance = \frac{max_{task}(Exe_{time}) - min_{task}(Exe_{time})}{Avg_{taski}(Exe_{time})}$$

Where max_{task} indicates Maximum Execution time, Min_{task} indicates Minimum Execution time, Avg_{taski} indicates Average Execution time of task among all VMs.

4)Cost: The cost that each client pays for the time used by the VM. It is an amount that has to be paid against the usage of resources in cloud computing. Cost means the total payment generated against the utilization or usage of resources, which is paid to the cloud providers by the cloud users .It is profit and revenue for the cloud providers and expense for the cloud users besides the utilization of resources in cloud computing. Assume the cost of a VM varies from one another based-on time substantiality and the VM's specification as specified by the cloud providers, then the following equation holds for the cost of Task Execution of a VM.

$$Cost = \sum_{i=1}^{n} task^{\,i}\,(Ci * Ti)$$

where C_i represents the cost of resource i per unit time, and T_i represents the time of utilization of resource.

4.3.2 Algorithms Used

It is observed that a number of task ordering methods have been developed and improved over a course of time. These methods can be categorized to either belong to exact or heuristic or meta- heuristic methods. In cloud computing, heuristic algorithms are designed to resolve the problematic issues faster than meta-heuristic algorithms, when their performance is too slow. The solutions obtained by the heuristic methods are optimum or near optimum in nature by using a smaller number of computer resources and computational time. In the proposed framework we have implemented the heuristic algorithms and pseudocode as referred from [21].

The 6 Scheduling Algorithms used are listed below:

- MCT
- MET
- Min-min
- Min-mean
- Min-var
- Sufferage

4.4 Feature Selection

In the process of selecting a subset of relevant features for use in model construction we chose 5 features: number of tasks, number of machines, total length of all the tasks in a set, average load on the machines, average arrival time difference of tasks. Hence the dimension of input is 5. In order to select the most relevant set of features for our particular model we have to perform a principal component analysis (PCA) of our dataset. PCA is a very important intermediate step in data analysis. In order to interpret large datasets, methods are required to drastically reduce their dimensionality in an interpretable way, such that most of the information in the data is preserved. PCA is one of the most used methods to decrease dimensionality. Also, PCA is used to discover important features of a large data set. It reveals relationships between variables which help us decide whether the variable actually has any effect on the prediction process or not. The Principle component analysis (PCA) of the proposed framework is given in Fig.2, where the above mentioned five dimensions are mapped to a 2D grid.

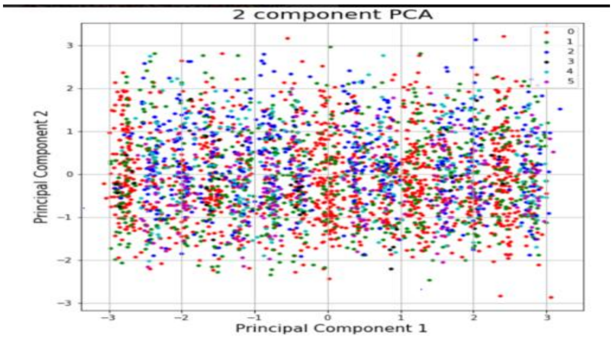
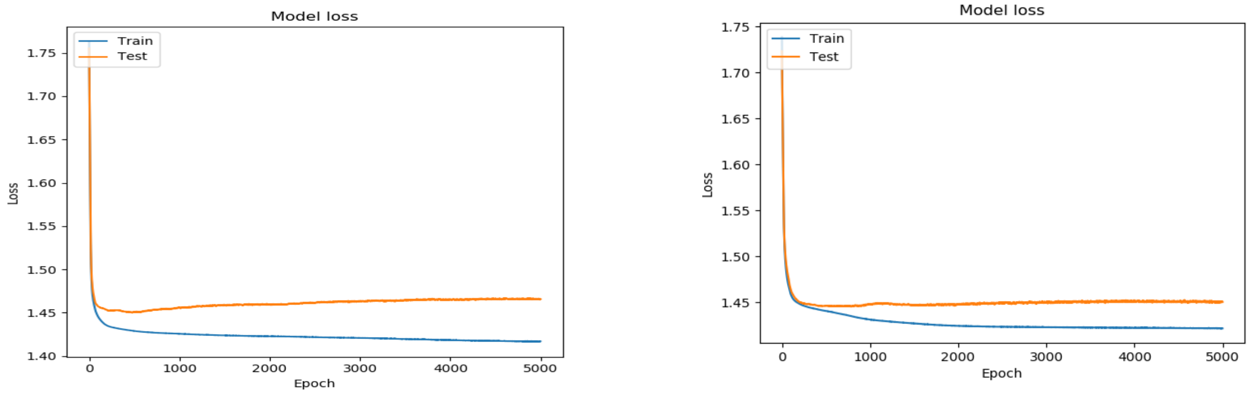


Figure 2. PCA of the proposed system

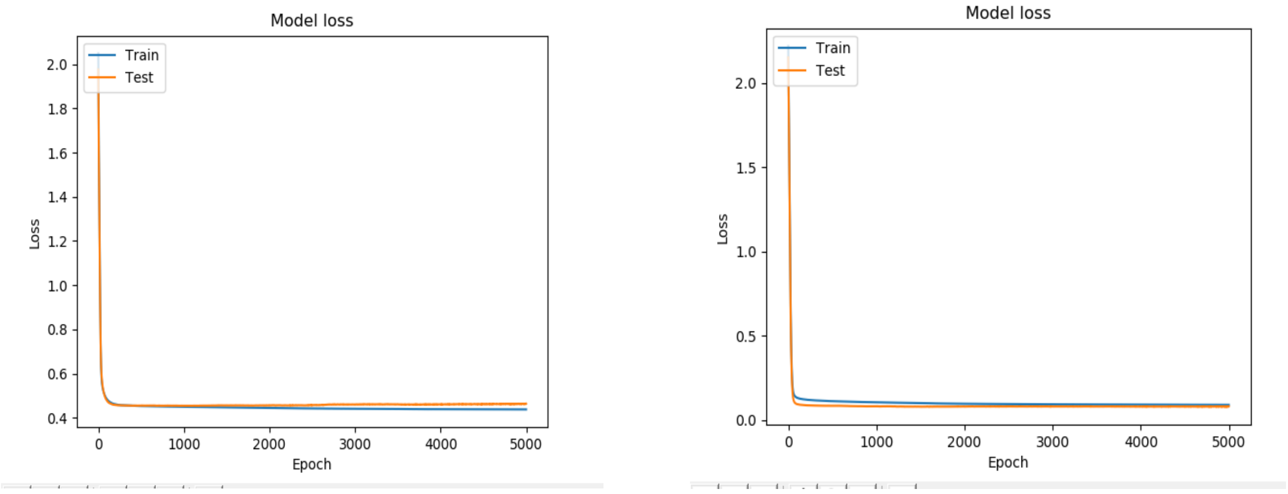
4.5 Neural network architecture

For each neural network the sample dataset was split into train and test sets where the split ratio was set to 0.1. Since there are five input features (Number of tasks, Number of machines, Total length, Average load, Average arrival time difference) the input layer has 5 neurons where each neuron is for each input feature, followed by 3 hidden layers where each hidden layer has 6 neurons and an output layer with 6 outcomes. The Activation functions used in the hidden layers are 'ReLU' and the activation function used for the output layer is 'softmax'. The loss function used is 'categorical cross-entropy' and the optimizer function is 'adam'. The training process was performed for 5000 epochs and a batch size of 1000.The graphs shown in Fig.3 are the loss vs epoch graph for all the neural networks. The graphs clearly show how the loss of the model has dropped with every epoch. It also gives an idea as to when to take a decision to stop the training process. Looking at the final loss value it also gives a picture on how well the model has been trained.



(a)

(b)

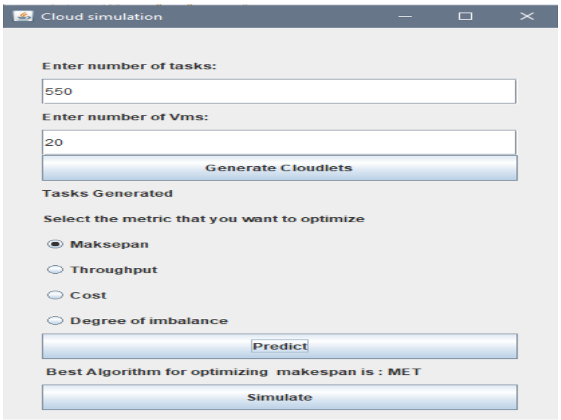


(c) **(d)**

Figure 3.loss vs epoch graph for model which predicts the best algorithm based on a) Cost b) Throughput c) Makespan d) Degree of imbalance

4.6 User Interface for the proposed framework

The requests from the user for IaaS must be generated. The User Interface is designed to mimic the activities of the client. The snapshot of the same is given in Fig.4.



Enter number of tasks: 550

Enter number of Vms: 20

Tasks Generated

Select the metric that you want to optimize

● Maksepan

○ Throughput

○ Cost

○ Degree of imbalance

Best Algorithm for optimizing makespan is : MET

Figure 4. UI design for the implemented system

Fig.4 depicts the instance of UI design where the user wants to optimize the makespan. The client enters the number of tasks that they want to submit and enters the number of VM. Now the application generates a specified number of tasks. The UI is created to mimic the activities of the clients in the real-world scenario. But here the application generates the tasks for simplicity. Then the client has the option of selecting the parameter that they want to optimize. Once the details are given and the Predict button is pressed the application calculates the features required i.e. number of tasks, Number of machines, Total length, Average load, Average arrival time difference.

4.7 Prediction process

As mentioned earlier a neural network is built using Python the Java application must send the features calculated to a Python file which performs the prediction. In order to achieve this the application creates a Python child process in which the prediction module in python is executed. All the calculated features are sent as arguments to the child process. The child process now has all the features required and can predict the outcome based on the received features. This prediction result is sent back to the parent Java application and shown to the user. As shown in the UI the user has the choice to select the parameter, they want to optimize so based on the user's choice these features are given to that particular neural network(NN) model as input and the NN model returns the best scheduling algorithm for that particular input set and then the submitted set of tasks can be scheduled using the algorithm returned by the neural network.

5. 0 RESULTS

Results about data set generation, overall evaluation of neural network, prediction outcomes, comparative analysis of scheduling algorithms are presented.

5.1 Data set generation sample

The data set was generated using different algorithms on the same set of tasks. Here we have considered 3 arbitrary sets of tasks and collected the results from all the six scheduling algorithms so that one can clearly see the changes in the values of performance metrics depending upon the algorithm used for scheduling.

Table1. Sample Data Set 1

Number of tasks: 550, Total length: 27896943, Number of machines: 20, Avg Arrival Diff:10310, Avg Load :2657				
	Makespan	Cost	Throughput	DI
MET	10470	81.5	163	2.4
MCT	10514	110.0	220	1.54
Sufferage	10499	85.5	171	1.0
Min-min	10514	84.0	168	0.739
Min-mean	10499	86.0	172	0.586
Min-var	10514	86.0	172	0.472

Table2. Sample Data Set 2

Number of tasks :600, Total length: 31259116, Number of machines: 25, Avg Arrival Diff: 11370, Avg Load: 2305				
	Makespan	Cost	Throughput	DI
MET	11476	65.5	130	2.4
MCT	11478	65.5	135	1.7

Sufferage	11476	65.5	127	1.0
Min-min	11478	63.5	129	0.81
Min-mean	11476	64.5	129	0.65
Min-var	11478	64.5	127	0.55

Table3. Sample Data Set 3

<table>
<tr><td colspan="5">Number of tasks:650, Total length: 31819340, Number of machines: 30 ,
AvgArrival Diff:12293 Avg Load :2653</td></tr>
<tr><td></td><td>Makespan</td><td>Cost</td><td>Throughput</td><td>DI</td></tr>
<tr><td>MET</td><td>12379</td><td>41.5</td><td>83</td><td>2.66</td></tr>
<tr><td>MCT</td><td>12383</td><td>41.5</td><td>83</td><td>1.33</td></tr>
<tr><td>Sufferage</td><td>12381</td><td>34.5</td><td>69</td><td>0.94</td></tr>
<tr><td>Min-min</td><td>12379</td><td>34</td><td>68</td><td>069</td></tr>
<tr><td>Min-mean</td><td>12380</td><td>34</td><td>69</td><td>0.57</td></tr>
<tr><td>Min-var</td><td>12379</td><td>34.5</td><td>83</td><td>0.48</td></tr>
</table>

Table 1,2,3 are the sample data set generated using CloudSim using randomly generated properties of the tasks. They are given as input to different scheduling algorithms. Here we have considered six scheduling algorithms. The output is the different set of values for makespan, cost, throughput and DI.

5.2 Results related to Confusion matrix of Neural networks

The different feed forward neural networks are evaluated based on the confusion matrix method. The metrics are indicated as below.

Precision: Precision talks about how precise/accurate your model is out of those predicted positive

$$\text{Precision} = \text{True positive} / (\text{True positive} + \text{False positive}) \quad (5)$$

Recall: Recall shows how many of the Actual Positives our model capture through labelling it as Positive

Recall = True positive/ (True Positive + False negative) (6)

F1 score: F1 Score shows the balance between Precision and Recall.

F1 score = 2*((Precision* Recall)/ (Precision +Recall)) (7)

Based on the values confusion matrix generated, the performance measures are calculated and a summarized table for each optimization metric is given in table 4. Here only the performance measure of the best scheduling algorithm considered is given.

Table 4. Performance Measure

		Performance Measure		
Parameter	Best scheduling Algorithm	precision	recall	f1-score
Throughput	Min-min	0.50	0.01	0.0
Makespan	MET	0. 91	1.00	0.95
Cost	MET	0. 43	0.98	0.59
Degree of imbalance	Min-var	0.91	1.00	0.95

From table 4, we have considered precision as an accuracy measure. Looking at the above table Min-min seems to be the best scheduling algorithm for optimizing throughput and the accuracy is 50%. MET seems to be the best scheduling algorithm for optimizing makespan and the accuracy is 91.04%. MET seems to be the best scheduling algorithm for optimizing cost and accuracy is 41.6%. Min-var is clearly the best scheduling algorithm for optimizing Degree of imbalance and the accuracy is: 90.60%. From the calculated values we can see that there is no clear winner when it comes to optimizing throughput and cost. In the case of throughput, we can see that both Min-min and MET precision values are very close and in the case of cost, MET and MCT have very close precision values. So, it becomes difficult to choose the best scheduling algorithm in these cases. But in the case of makespan we have MET as a clear winner and for Degree of imbalance we have Min-var.

5.3 Prediction vs actual value:

In this section we see the performance of the actual application. Here we assign the same set of tasks to all the six scheduling algorithms and obtain their results. The same set of tasks is sent to the application which returns its prediction and then these values are compared for verifying if the algorithm that was predicted by the model is in fact the best scheduling algorithm.

Table 5. Evaluation Task Set.

	Number of tasks	Total length	Number of machines	Avg. Arrival Diff	Avg. Load
Set 1	550	28049693	10	10474	2677
Set 2	550	26823302	15	10377	2317
Set 3	580	29186306	20	10957	2950

Set 4	600	29576432	10	11217	2183
Set 5	600	29732234	25	11288	2037
Set 6	700	34525047	15	13211	2456
Set 7	700	34411823	25	13278	2350
Set 8	660	33694510	19	12503	2448
Set 9	680	34345303	22	12696	2607
Set 10	742	36627615	29	13901	2760

Table 5 gives the details of 10 sets of tasks used for the evaluation.

Evaluation of actual performance in makespan optimization is shown in table 6. The first six columns contain the makespan values when scheduled using the corresponding algorithms. This gives a comparison of the actual algorithm to the predicted algorithm.

Table 6. Performance Evaluation in Makespan Optimization.

	MET	MCT	Sufferage	Min-min	Min-mean	Min-var	Actual Least Makespan	Predicted algorithm
Set 1	10627	10629	10627	10628	10628	10629	MET	MET
Set 2	10495	10496	10496	10495	10496	10498	MET	MET
Set 3	11205	11205	11206	11207	11206	11207	MET	MET
Set 4	11342	11342	11344	11342	11343	11344	MET	MET
Set 5	11395	11398	11397	11398	11398	11398	MET	MET
Set 6	13340	13340	13340	13340	13340	13340	MET	MET
Set 7	13337	13339	13339	13338	13337	13338	MET	MET
Set 8	12656	12659	12658	12658	12658	12657	MET	MET
Set 9	12776	12777	12777	12777	12776	12778	MET	MET
Set 10	14006	14009	14008	14008	14008	14009	MET	MET

Evaluation of actual performance in throughput optimization is shown in table 7. The first six columns contain the throughput values when scheduled using the corresponding algorithms. This gives a comparison of the actual algorithm to the predicted algorithm.

Table 7. Performance Evaluation in Throughput Optimization.

	MET	MCT	Sufferage	Min-min	Min-mean	Min-var	Actual Max Throughput	Predicted algorithm
Set 1	148	148	156	132	150	153	Min-min	Min-min
Set 2	164	164	171	164	171	170	Min-min	Sufferage
Set 3	232	232	197	195	198	198	Min-min	Min-min
Set 4	105	119	130	129	130	131	MET	Min-min
Set 5	100	103	96	98	98	98	Sufferage	Sufferage
Set 6	133	133	24	24	24	24	Sufferage	Sufferage
Set 7	67	67	23	25	25	25	Sufferage	Sufferage
Set 8	151	151	129	119	124	124	Min-min	Min-min
Set 9	62	71	68	67	67	68	Min-min	Sufferage
Set 10	99	99	79	81	83	83	Min-min	Min-min

Evaluation of actual performance in cost optimization is shown in table 8. The first six columns contain the cost values
when scheduled using the corresponding algorithms. This gives a comparison of the actual algorithm to the predicted algorithm.

Table 8. Performance Evaluation in Cost Optimization.

	MET	MCT	Sufferage	Min-min	Min-mean	Min-var	Actual Least Cost	Predicted algorithm
Set 1	66.0	66.0	78.0	74.0	75.0	76.5	MET	MET
Set 2	82.2	82.0	85.5	82.3	85.5	85.0	MCT	MET
Set 3	116.0	116.0	98.5	97.5	99.0	99.0	Min mean	MCT
Set 4	52.5	59.5	65.0	64.5	65.0	65.5	MET	MET
Set 5	51.0	48.0	51.5	51.5	51.5	51.5	MCT	MCT

Set 6	12.0	12.3	12.3	12.1	12.0	12.1	MET	MET
Set 7	33.5	33.5	11.5	11.5	11.5	11.5	Sufferage	MET
Set 8	75.5	75.5	64.5	59.5	62.0	62.0	MET	MET
Set 9	31.0	35.5	34.0	33.5	33.5	34.0	MET	MET
Set 10	49.5	49.5	39.5	40.5	40.5	40.5	Sufferage	MET

Evaluation of actual performance in Degree of optimization is shown in table 9. The first six columns contain the degree of imbalance value when scheduled using the corresponding algorithms.

Table 9. Performance Evaluation in Degree of Imbalance.

	MET	MCT	Sufferage	Min-min	Min-mean	Min-var	Actual Least DI	Predicted algorithm
Set 1	2.2	1.5	1.05	0.78	0.62	0.51	Min-var	Min-var
Set 2	2.4	1.2	0.93	0.85	0.66	0.54	Min-var	Min-var
Set 3	2.66	1.23	0.85	0.69	0.56	0.46	Min-var	Min-var
Set 4	2.5	1.1	1.6	0.53	0.80	0.62	Min-var	Min-var
Set 5	2.2	1.7	1.06	0.89	0.65	0.54	Min-var	Min-var
Set 6	3.0	1.25	0.88	0.78	0.68	0.52	Min-var	Min-var
Set 7	2.2	1.5	0.91	0.70	0.50	0.55	Min-var	Min-var
Set 8	3.0	1.38	0.94	0.72	0.58	0.58	Min-var	Min-var
Set 9	2.5	1.65	1.05	0.81	0.64	0.52	Min-var	Min-var
Set 10	2.8	1.27	0.82	0.68	0.55	0.44	Min-var	Min-var

Table 6,7,8,9 present an analysis of actual to predicted algorithms for optimizing makespan, throughput, cost, degree of imbalance in the view of task scheduling dynamically in the simulated cloud environment. Comparing the results of actual to predicted value in each optimization metric the accuracy is fairly matching with the results in the reference [21]. The dynamic selection of task scheduling algorithm using neural network based prediction approach is a novel contribution of this work. It is evident that machine learning techniques to dynamic selection of algorithm improves the overall system performance and adaptability.

6.0 Conclusion

The aim of the proposed framework is to design a new approach to dynamically schedule tasks in an IaaS cloud environment. In contrast to the previous works, this approach uses machine learning techniques to predict dynamically the scheduling algorithm for the incoming request. For a given set of tasks, the best algorithm is predicted for each of the following four parameters viz., cost, degree of imbalance, makespan and throughput. Task scheduler unit outputs the best scheduling algorithm to be used based on the parameter to be optimized. For the algorithms considered here by looking at the dataset, the neural network predicts that in terms of Degree of Independence Min-Var seems to be the best algorithm in most of the cases; for makespan, MET seems to be the best algorithm in most of the cases. while for throughput and cost there are no such visible patterns. Since task scheduling in a distributed computing environment is considered as a np-hard problem, we hereby propose a framework for solving this problem. PCA is used to capture the relevant features and make a machine learning model that can predict the best task scheduling algorithm for each set of tasks. In contrast to the previous works, integrating machine learning to determine the scheduling scheme helps improve the QOS and also save energy and resources that are valuable and most importantly computation costs, which in turn improves cloud accessibility to a wider demographic.

REFERENCES

[1] Qi Zhang, Lu Cheng and Raouf Boutaba , "*Cloud computing: state-of-the-art and research challenges*", Journal of Internet Services and Applications, volume 1, Issue 1, pp 7–18,May 2010.

[2] Vijindra and Sudhir shenai, "Survey *on scheduling issues in cloud computing", International conference on modeling optimization and computing* (ICMOC-2012), Procidia Engineering 38, pp 2881-2888, 2012.

[3] Dr.Amit Agarwal and Saloni Jain, "*Efficient Optimal Algorithm of Task Scheduling in Cloud Computing Environment*", International Journal of Computer Trends and Technology (IJCTT), volume 9, number 7, Mar 2014 .

[4] Ahmed Abdelaziz, Mohamed Elhoseny, Ahmed S. Salama and A.M. Riad, "*A Machine Learning Model for Improving Healthcare services on Cloud Computing Environment*", Measurement ,2018.

[5] Anagha Yada and S.B.Rathod, "*Study of Scheduling Techniques in Cloud Computing Environment."* International Journal of Computer Trends and Technology (IJCTT), volume 29 -Number 2, November 2015.

[6] Mayank Sohani, Saniya Jain, Ishita Narang, Kaushal Agarwal and Saahil Arora, "*A Survey of Different Task Scheduling Algorithm in Cloud Computing",* International Journal of Advanced Studies in Computer Science and Engineering, Volume 6 Issue 4, 2017.

[7] Mahendra Bhatu Gawali and Subhash K. Shinde, "*Task scheduling and resource allocation in cloud computing using a heuristic approach",* Journal of Cloud Computing, Article number 4, Springer Open, 2018.

[8] Teena Mathew, K. Chandra Sekaran and John Jose, "*Study and Analysis of Various Task Scheduling Algorithms in the Cloud Computing Environment",* ICACCI, 2014.

[9] Nasim Soltani, Behzad Soleimani and Behrang Barekatain," *Heuristic Algorithms for Task Scheduling in Cloud Computing: A Survey", I.* J. Computer Network and Information Security,volume 8, pp 16-22 ,2017.

[10] Satyasundara Mahapatra, Rati Ranjan Dash and Sateesh K. Pradhan,*"Heuristics Techniques for Scheduling Problems with Reducing Waiting Time Variance", Heuristics and Hyper-Heuristics - Principles and Applications,* April 13th 2017.

[11] Naoufal Er-raji, Faouzia Benabbou, Mirela Danubianu and Amal Zaouch, "*Supervised Machine Learning Algorithms for Priority Task Classification in the Cloud Computing Environment"*, International Journal of

Computer Science and Network Security (IJCSNS), volume 18, No.11, November 2018.

[12] Hatem M. El-Boghdadi and Rabie A. Ramadan, *"Resource Scheduling for Offline Cloud Computing Using Deep Reinforcement Learning"*, IJCSNS International Journal of Computer Science and Network Security, volume .19, No.4, April 2019.

[13] Chinmai Shetty and SarojaDevi H., "*Framework for Task scheduling in Cloud using Machine Learning Techniques*", IEEE International Conference on Inventive Systems and Control (ICISC), pp 713-717, pp 729 – 731, January 2020.

[14] Arghavan Keivani and Jules-Raymond Tapamo, "*Task Scheduling in Cloud Computing: A Review",* IEEE International Conference on Advances in Big Data, Computing and Data Communication Systems , 2019.

[15] Brian Xu, D. Mylaraswamy and P. Dietrich, "*A Cloud Computing Framework with Machine Learning Algorithms for Industrial Applications*", worldcomp-proceedings, ACM,2013.

[16] Daniel Pop, "Machine *Learning and Cloud Computing: Survey of Distributed and SaaS Solutions*". IEEE Network Magazine, 2016.

[17] Jun-Bo Wang, Junyuan Wang, Yongpeng Wu, Jin-Yuan Wang, Huiling Zhu and Min Lin, Jiangzhou Wang, "*A Machine Learning Framework for Resource Allocation Assisted by Cloud Computing*", IEEE Network Magazine, April 2018.

[18] Joe Fiala, "*A Survey of Machine Learning Applications to Cloud Computing*", http://www.cse.wustl.edu/jain/cse570-15/ftp/cld_ml/index.html.

[19] Gaith Rjoub and Jamal Bentahar, "*Cloud Task Scheduling based on Swarm Intelligence and Machine Learning",* IEEE 5th International Conference on Future Internet of Things and Cloud, 2017.

[20] Renyu Yang, Xue Ouyang, Yaofeng Chen, Paul Townend and Jie Xu, "*Intelligent Resource Scheduling at Scale: a Machine Learning Perspective"*, IEEE Symposium on Service-Oriented System Engineering, 2018.

[21] Syed Hamid Hussain Madni, Muhammad Shafie Abd Latiff, Mohammed Abdullahi, Shafii Muhammad Abdulhamid and Mohammed Joda Usman, "*Performance comparison of heuristic algorithms for task scheduling in IaaS cloud computing environment", Journal*-PLOS One, May 3, 2017.

[22] Rajkumar Buyya, Rajiv Ranjan and Rodrigo N. Calheiros, "*Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities"*, Proceedings of the 7th High Performance Computing and Simulation (HPCS 2009) Conference Leipzig, Germany, June 21 - 24, 2009.