International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

# Average True Range Approach for Resource Scheduling and Allocation in Cloud Computing Networks

[1]Uma Maheswara Rao I, [2]Dr. JKR Sastry

**Submitted:** 25/10/2022 **Revised:** 23/12/2022 **Accepted:** 21/01/2023

**Abstract**: **Purpose**: Resource allocation in cloud computing has significant importance, in adhering to optimal utilization of resources, and maintaining service quality among the service networks. Scores of machine learning models and other contemporary solutions were proposed for handling the resource allocation process. However, one of the challenges with many of the existing solutions is complexity in the process and occupying the space for internal application executions.

**Approach**: Thus, in this manuscript, the attempt is to propose a simplistic approach of statistical model in the combination of average true range (ATR) and exponential moving average (EMA) models for multi-layer decision-making process. The proposed system is about identifying the average true range of service requests received and processed over time and applying the decision-making matrix at multiple scenarios.

**Findings**: An experimental study of the model executed on a data generated from Cloudsim simulations, refers to the effectiveness of the system, wherein the accuracy is high in terms of indicating the potential need for alternate resources or optimal use of the existing resources.

**Originality**: Considering the linearity of the model, it is effective for small-scale IaaS environments. In future research, a potential mode of applying the solution over a machine learning approach to improve the overall efficacy of the system can be considered more pragmatic for enhancing the outcome.

**Keywords***: Cloud Resource, Exponential Moving Average (EMA), Average True Range (ATR), longest expected processing time (LEPT), Machine learning.*

## 1 Introduction

Cloud computing solutions are fundamentally about effectively deploying information systems and applications so that organizations can reap the benefits of quick and anytime access, reduced downtime impacts, a reduction in the cost of managing the IT infrastructure, and a slew of other advantages. However, there are also challenges associated with cloud computing solutions, such as data privacy and security concerns and the difficulty in fully integrating cloud solutions with existing IT systems. However, considering the gaps in the system developments, there is a distinct set of critical success factors to be managed effectively for improving the overall experience of handling cloud computing services [1].

One of the critical elements of paramount importance in cloud computing networks is handling the resource scheduling process effectively. While there are scores of cloud resource scheduling patterns adapted in a real-time environment, the crux of such solutions is the time and resources required for the execution of the process. Allocating an optimal amount of time for scheduling tasks and resources should be a priority for cloud administrators in order to ensure efficient utilization of resources and minimize idle time. In simple instances, for an IaaS (Infrastructure as a Service) to accommodate the service requirements of various client applications in its service network, there is a need for more optimal utilization of the virtual machines integral to the network environment [2].

Unlike the service giants, for the small-scale service providers, adapting high-end machine learning solutions, and deploying AI-based applications are herculean tasks. Also, in terms of economies of scale, the question of affordability in deploying such machine-learning models stands as a challenge. Therefore, cloud administrators should focus on the efficient utilization of existing resources to provide a better quality of service. There is a need for minimalistic solutions that can be deployed at surface levels without occupying too much system

*1Research Scholar 2Professor*
*1,2Dept., of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Andhra Pradesh, India*
*1inkolluchanti@gmail.com 2drsastry@kluniversity.in*

bandwidth and are able to deliver the required outcome from the process [3].

The objective of this manuscript is to develop a linear and effective tracking model that depicts the resource overload conditions for the VMs. The proposed model should be capable of accurately recognizing these overload situations in order to assist the cloud administrator in taking appropriate remedial measures. While many of the solutions prominent in the machine learning domain are resourceful for handling the cloud resource scheduling process, the model explored in this manuscript is a statistical approach highly resourceful for small-scale IaaS set-up or even to work as a monitoring solution in high-end cloud computing applications.

Two key objectives of the proposed application are: (1) developing a statistical performance solution that can be significant for addressing cloud resource scheduling across the VMs in a cloud network; and (2) creating a linear and effective tracking model that can help the cloud administrator in recognizing and dealing with resource overload situations.

Secondarily, developing a solution that is simpler in the case of monitoring for the server admins, and in terms of offering a visual representation that can indicate better decision-making insights.

The model discussed in this study shall be compared to the other relative models in the domain, to understand the efficacy of the proposed model. In the further sections of this report, section 2 refers to the related work in the domain, and section 3 provides insights into the actual model, the process flow, material, and methods. Section 4 provides insights into the experimental study outcome, followed by the conclusion discussed in the section 5.

## 2 Related Work

Cloud Systems and Infrastructure resource scheduling are the key areas of the information systems domain, wherein there is more advanced research studies are taking place. Considering the growing demand for IaaS kind of systems, there is distinct kind of solutions being explored to optimize the service quality, and utilization of resources effectively in the cloud systems environment. In this related work section, the outline of various segments in which the resource scheduling models are being explored in the systems is discussed [4].

Firstly, in the case of the simple and linear models, the focus is on understanding the models that are integral to the network and hardware systems integral to the IaaS environment. A few of the key models discussed in the segment are about bandwidth and processer load levels tracking, using the network monitoring tools, which can help in tracking the load and resource allocation priorities [5], [6].

Some of the advanced models that were discussed is the heuristic models, and the efficiency monitoring set of customized algorithms that can help in improving the resource scheduling models. One such model in [6] discusses modified analytic hierarchy process (MAHP), bandwidth aware divisible scheduling (BATS) + BAR optimization, longest expected processing time preemption (LEPT), and divide-and-conquer methods to perform task scheduling and resource allocation [6]. Multiple sets of complex models were combined into one attempt, wherein the solutions could be more impacting and load significance [6], [7], [8].

There are scores of machine learning models and evolutionary computing algorithms proposed in the cloud resource scheduling models that can help in more effective ways of allocation of cloud resources. Table 1 below depicts some of the distinct kinds of resource scheduling algorithms considered significant in the process.

**Table 1:** The distinct kind of resource scheduling algorithms considered significant in the process

| Frameworks | Purpose | Algorithm | Environment | Parameters |
|---|---|---|---|---|
| Optimal Algorithm | To observe the difference of time between grid and cloud | Task Scheduling algorithm | Cloud & Grid Environment | execution time |
| Deadline based Resource Provisioning algorithm | To minimize execution cost | Resource Scheduling Algorithm, PSO, and MHOA | Cloud Environment | PSO, MHOA, Execution Cost |
| Dynamic Resource | To suspend low priority job | Resource Scheduling | Cloud | Priority based |

| | | | | |
|---|---|---|---|---|
| Allocation Scheme | and allow high priority job | Algorithm | Environment | |
| Hybrid job scheduling Algorithm | It reduce the population | Job scheduling Algorithm | Cloud & Fuzzy set | Execution time cost |
| Firefly Algorithm | To Reduce balancing time | Load Balancing | Cloud Environment | Index calculation Schedule list, Execution time |
| Star | To reduce SLA-Violation | Based on the genetic property of self-management | Cloud Environment | Execution time, cost, latency, reliability |
| (IDEA) Algorithm | To allocated resource & task efficiently | Task & Resource Scheduling Algorithm | Cloud Environment | Make span, cost, Performance |
| Cuckoo | Scheduling | Task Scheduling Algorithm | Cloud Environment | Execution time |
| Greedy based Job Scheduling Algorithm | To decrease the job completion time | Job scheduling Algorithm | Cloud Environment | Execution Time |
| Resource aware Hybrid scheduling Algorithm | To minimize execution cost | Task & Resource Scheduling algorithm | Cloudsim | Dynamic Clustering & Abstract Modeling |
| Autonomous agent | Calculation of load on virtual machine | Load Balancing | Cloud Environment | Load |
| Multiple Objective | To Reduce the Execution time | Task Scheduling Algorithm | Cloud Environment | Execution time |
| Honeybee Algorithm | To balance load as well as assign the task according to the priority | Load balancing | Cloud Environment | Priority Response Time |
| Credit based Scheduling | To combine cloudlet priority and length mechanism | Task Scheduling Algorithm | Cloud Environment | Cloudlet Priority and length, Make span |

While many of such algorithms focus on different metrics, one of the common aspects to consider is the quick turnaround time in which the solutions can be processed by the application system. Thus, focusing on the dynamics of the model, the emphasis is on improving the quality of the solution, whilst ensuring that the heavy load factor over the existing application [9-15].

Zheng et al., [16] explored a multi objective bat algorithm (MOBA) that intended to optimize the cloud resource scheduling. In order to avoid the impact of local minima and increase the loudness in the search direction and the pulse emission rate, the MOBA is using the back-propagation technique based on the mean square error as well as the conjugate gradient method. However, the process complexity is proportionate to the load, hence, the MOBA is suboptimal in real time scenarios with high rescheduling frequency and low job completion frequency. Kumar et al., [17] proposed a particle swarm optimization approach called PSO-BOOST to optimize the cloud resource scheduling under multiple quality of service parameters. However, the resource scheduling and allocation performance of the PSO-BOOST is inversely proportionate to the load of the tasks buffered at an event of time.

In summary of the reviewed work, it is evident about the application of numerous machine learning kind of algorithms can function in improving the resource scheduling modality. However, one of the key aspects to consider is to keep the application linear and lighter in approach, in terms of enabling the servers to occupy

minimal space for its internal applications, and improve the overall process quality while handling requirements for external applications.

## 3 Methods and Materials

The following subsequent sections explores the materials used and methods contributed in "Average True Range Approach for Cloud Resource Scheduling" with multiple Quality parameters listed as "minimal rescheduling frequency", "Maximal job completion frequency" and "maximal resource utilization frequency".

### 3.1 Gap Analysis

In line with the literature reviewed for the model, it is evident how the current models of resource scheduling are effective. However, highlighting the implications of some of the robust resource scheduling solutions, it is evident how the organizations are facing complexities in terms of deploying new solutions, and handling the additional burden on the servers in terms of deploying the machine learning solutions [12].

In simple terms, when an entity has "X" as the bandwidth from the systems to cater to clients, if 10% of the X is used for deploying the resource scheduling systems, it might lead to more load on the virtual machines in the network. There is a need for a specific kind of scanning model that assesses the current bandwidth utilization conditions, and accordingly, develops a systematic signal for admins to make informed decisions.

The primary reason or implications observed in some of the current models are about issues pertaining to how the algorithms are switching the servers frequently, thus leading to increased energy consumption and efforts towards altering the VMs, and dynamic load balancing efforts. Fluctuation in the resource utilization capacities signifies the chances of small ups and downs in the system, and it is not essential to switch the application solutions in too fast range. Thus, there is a need for the application to focus on the holistic scenario of whether there is a significant drop or spike in the load management, and accordingly make necessary decisions to cater to the conditions.

### 3.2 Average True Range Approach for Resource Scheduling and Allocation

The Resource scheduling model proposed in this study is about the application of the unit / metric value analysis using a statistical approach of the boxing technique. The boxing or grid technique discussed in this manuscript is called ATR (Average true range) charts which are predominantly used in securities analysis, wherein the objective is to understand the optimal value for underlying process execution and to ignore any small variances occurring in the process as a notional loss. In simple terms, when the overall network path is clear for handling the requirements, little load factor during a given session need not be considered as an alarm. Thus, the proposed model is based on developing Grids in handling the distinct kind of traffic reaching the VMs, and if there is any instance wherein the solutions are able to act as signal alerts indicating a significant shift in the move, such solutions can be more resourceful for resource scheduling conditions [18].

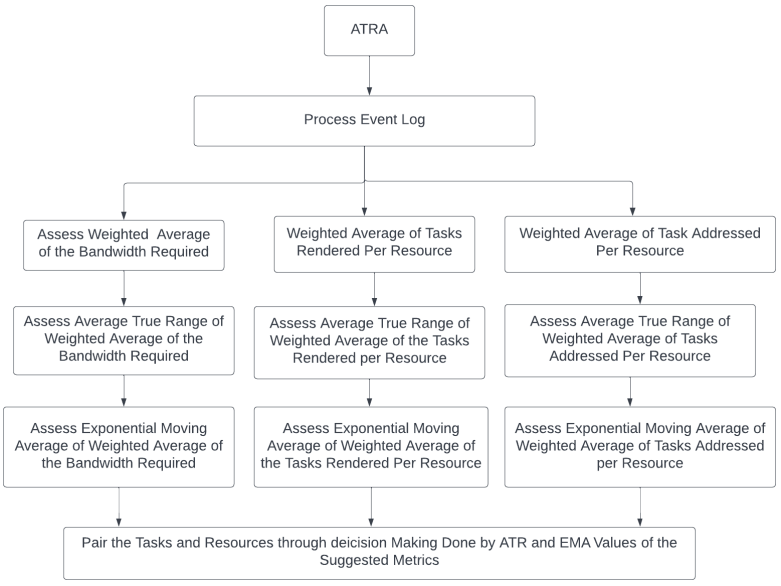


**Fig. 1:** Block diagram of the suggested ATRA based resource scheduling strategy

Fundamentally, the solution relies on developing grids that are based on the average true range of the transaction bandwidth, and it stands irrespective of the volume spike in the scenario. The quantum of session requests being processed within a period of time shall be calculated for average true range. When the system pars the number of session requests above one particular range, then it is considered as significant for action [18].

The system works in a pattern, wherein when the ATR range value is created, a block is formed, and subsequent block is formed above only when a complete ATR value of session requests is processed. The uptrend is marked by a range and the lower value of range in a matrix and marked by different alert signals. The block formation has no insights into bandwidth consumed for processing the requests [15], [19], [20], [21].

To eliminate the issues of any abrupt spike or issues in the bandwidth consumption-related issues, the scope of using the Exponential Moving Average (EMA) value of the bandwidth for the chosen time frame is reviewed. If there is any abrupt drop in the EMA values and a sudden spike in the transaction requests, it could be considered a high-alert mode (as it could be DoS attacks or interruptions from the system) impacting the system efficacy. The overall phases involved in the illustrated model can be found in figure-1.

### 3.2.1 Illustrative Scenario.

- Per Say, the average true range estimated stands at 532 transactions.
- The ATR is formed for the session requests received during the time frame, and the block count stands as X holds the range.
- When the server handles the subsequent 532 transactions, then a new block is formed as Y based on its ATR Range.
- The absolute difference between service requests processed and received is estimated to understand the overall load impact on the system.
- If the server has failed to handle 532 transaction requests stand pending, then the grid allocation shall be into a specific decision-making section.
- For the time period of alert, the exponential moving average of the absolute load is plotted as an average to understand the divergence or convergence conditions. If the confluence is high in terms of the number of requests being pending and bandwidth drop is imperative, the alternative systems can be activated, and the requests routed to the alternate servers [18].

### 3.2.2 Justification of the Model

The critical advantage of the model is about reducing alerts on minor interim disturbances in the system allocation and resource scheduling conditions. Also, in the instances of the sudden spike in the requests, similar to the other resource scheduling models, the alert mechanism in the current model is possible, as there shall be multiple blocks of rejections that form in the grid, and the trigger alert is sounded. Thus, the proposed model can be used for the manual switch of the server load capacities, or as a surface monitoring system on top some of the existing machine learning models. Adapting such solutions can improve the overall process outcome, and the measures integral to the process [18].

## 3.3 Metrics

### 3.3.1 Bandwidth

The bandwidth of the server in terms of server network capacity allocation, and the requisite metrics to handle the server allocation stands critical in offering quality services to the clients in the IaaS environment. There is a need for focusing on the bandwidth conditions wherein the solutions for bandwidth management require timely estimation of the load balancing requirements or other implications encountered in the bandwidth management. One of the key metrics for assessing how far a specific bandwidth category is yielding results can be based on the quantum of transactions completed within a given period of time [18], [21].

Required Bandwidth of each Make-span (RBEM): This metric represents the volume-weighted averages of the bandwidth used for each make span for each time frame. The following estimate is to be made for the specified metric:

Find the volume-weighted averages of the bandwidth used for each make-span $m$ and each resource $cR = \left\{ r_j \exists j = 1, 2, 3..., |cR| \right\}$ as follows:

Determine the absolute product of bandwidth utilised per time unit and the total time units that the related work-flow lasted for each work-flow $wF_m = \left\{ w_k \exists k = 1, 2, 3, ..., w_{|wF_m|} \right\}$ in the provided make-span. The results of all work-flows should be aggregated further, and the final aggregate value should be divided by the total time units $tU_m$ that made it through the relevant make-span Eq 1:

$$\forall_{j=1}^{|cR|} \left\{ rwbt_{(r_j \rightarrow m)} \leftarrow \left| \left( \left( tU_m \right)^{-1} * \sum_{k=1}^{|wF_m|} \left\{ bcu * tU_k \right\} \right) \right| \exists r_j \in cR \right\} \quad \text{... (Eq 1)}$$

The objective of the model is about using the different metrics that can have a direct impact how the requests are processed in the cloud computing environment, for

handling the resource requests. Depending on the context of the application systems the equations for the metrics mentioned in the box above can be adapted, and necessary computation can be processed using the steps detailed in the following sub-sections.

### 3.3.2 Tasks / Session Requests

In the other dimension, the task (transaction) and session requests are other key aspects important in tracking the performance of the servers. For instance, if the systems are able to handle the transaction requests in a quick turnaround time, it helps the application process the customer requests much more effectively. In the absence of such effective processing, the application response to user requests shall be low, and could affect the service quality assurances. The cumulative set of transactions assessed in the process stands critical for the overall process handled by the specific servers. The aggregation of the requests could be from multiple applications deployed from the server [14]. The suggested metrics of the tasks and sessions are explored in the following.

- Rendered Tasks per Resource ( $rtr$ ): The weighted-average of all tasks completed for each make-span to the specific resource is used to calculate this measure. The following estimate is to be made for the specified metric:

Determine the volume-weighted average of jobs completed for each resource $cR = \{r_j \exists j = 1, 2, 3..., |cR|\}$ throughout each make-span $m$ , which is as follows.

Find the total number of jobs completed under all work-flows for each of the provided make-spans and resources. The total work-flows must be presented for each make-span in the following notation.

$$\forall_{j=1}^{|cR|} \left\{ rtr_{(r_j \rightarrow m)} \leftarrow \left( |wF_m| \right)^{-1} * \sum_{k=1}^{|wF_m|} \left\{ |aT_k^j| \right\} \exists r_j \in cR \right\}$$

Begin // represents the weighted-average of make-span to resource's total jobs scheduled.

- Completed Tasks per Resource $(ctr)$ : The volume-weighted averages of all tasks completed by the target resource for each make span is used to calculate this indicator. This is how the aforementioned measure will be estimated:

Determine the volume-weighted averages of the total tasks serviced for each make-span $m$ and each resource $cR = \{r_j \exists j = 1, 2, 3..., |cR|\}$, and is as follows.

Determine the total tasks completed in each work flow $wF_m$, together with the percentage of the overall amount that corresponds to the number of work flows completed in the corresponding make-span.

$$\forall_{j=1}^{|cR|} \left\{ ctr_{(r_j \rightarrow m)} \leftarrow \left( \left( |wF_m| \right)^{-1} * \sum_{k=1}^{|wF_m|} \left\{ |aT_k^j| \right\} \right) | \exists r_j \in cR \right\}$$

//The weighted=average of the jobs that are planned throughout the whole make-span to resource is represented by this equation.

$wF_m$ : Total workflows of scheduled make-span $m$

$aT_k^j$ : completed tasks at $j^{th}$ resource scheduled to $k^{th}$ workflow

The objective of using these metrics is to target the direct outcome of the cloud resource scheduling process. Eying at the significant elements of the system that impact service quality, various parameters of session requests management are accounted for analysis and estimated for true range conditions.

Based on the prioritization of the bandwidth or session requests or any other kind of services that could affect the service quality, the computation process is chosen, and accordingly, the Average True Range model for the chosen metrics is executed. Also, in the case of a complex resource scheduling environment, using multiple sets of cloud resource scheduling, tasks to assess the complexities and make informed decisions for scheduling.

### 3.3.3 ATR (Average True Range)

ATR Charts refers to the optimal value for an underlying metric action or by focusing on the ATR (Average True Range) estimated for the period, the existing variance from the current trend to the optimal trend can be assessed. More often, the visual representation of the ATRs can help in understanding the variance between the optimal value proposition and the current performance for a metric or unit [21].

In the context of the proposed application model, the emphasis is on developing a comprehensive solution, wherein the blocks shall work as a trigger alert to denote the changes to the current movement in consideration to the earlier movements.

Estimation of ATR (Average True Range) using the formulae Eq 2

$$TR = Max[(H - L), Abs(H - C_p), Abs(L - C_p)]$$

$$ATR = \left(\frac{1}{n}\right)\sum_{(i=1)}^{(n)} TR_i$$

*Where*;

$TR_i =$ *A particular truerange*

$n =$ *Thetime period employed*

... (Eq 2)

Followed By, $X$ is the estimation of the value for the server requests received for the period, wherein the true range is estimated initially and subsequently the moving average for the value is carried out for the periods. Similarly, for the quantum of requests processed in the system $Y$ is estimated for the true range values. Considering the significant variance in the true range conditions, the emphasis in the model is to improve the optimal value-based decision-making [21].

#### 3.3.4 Exponential Moving Average (EMA)

The EMA is a statistical model wherein the recent data is given more weightage than the complete historic data and the conditions are ascertained based on any kind of sudden spurt or drop in the volumes indicating potential ways in which the resource server overload or reduction in the process can be ascertained [22], [23],[ 24].

Profoundly used in the analysis of price movements in the securities market, the adjusted formulae of the volume weighted average used in the current model is

The fundamental purpose of using the EMA is about understanding the generic volume of bandwidth consumption taking place in the model, and towards understanding how the proposed solution is effective for the real-time environment [25].

The objective of using the EMA analysis in the proposed model is about understanding any sudden variation in the transaction processing, that might impact the overall scheduling process effectively.

### 3.4 Process Flow

Step 1. The Volume of Transactions requests received for a Periodic Time Frame (per se a day) is collected in terms of low, high, and close values denoted as SRR

Step 2. The volume of Transaction requests processed for a periodic time frame (per se day) is collated in terms of low, high, and close values denoted as SRP

Step 3. The ATR Requests received and processed are estimated and the value differences are estimated as (SRR-ATR) $"X"$ and (SRR-ATR)$"Y"$. (Average True Range is estimated using 10-period Simple Moving Average)

Step 4. Upon Completion of Step-3, If $X > (Y + 10\%)$ , then the first level decision is given as "Monitor", Else, the "Continue Execution" is depicted.

Step 5. If there is a "Monitor" alert for consecutively two periods, then the second decision system shall be reviewed (the Second Decision Review system is detailed in steps 6-7)

Step 6. As a referral view, the absolute difference between $X$ and $Y$ is calculated and denoted as $Z$

Step 7. followed by the 10-period exponential average for the movement is estimated and denoted as $K$

Step-7 If $Z > K$ , "Monitor" is displayed as Caution, Else, "Switch" is displayed as the actionable request.

Step 8. The Final Decision shall be based on the two-level decision inputs as per the following Grid.

Step 9. In the condition of divergence between the First Level Decision and Second Level Decision condition, the trigger shall be monitored for any further necessary action over the task.

Step 10.

**Table 2:** Decision Grid

| First Level Decision | Second Level Decision | Response System |
|---|---|---|
| Alarm | Switch | Switch Load to Alternative VMs |
| Alarm | Monitor | Stand-by Scenario, wherein the alternates can be activated if the situation persists |
| Continue | Switch | Watch Zone, and measures for load-balancing by adjusting bandwidth |

| | | |
|---|---|---|
| Continue | Monitor | Ignore Until subsequent periods with same indication. |

### 3.5 Algorithm Flow

Let $"B"$ be the bandwidth speed at which the transaction request received and processed are assessed for a given time period,$"t"$, and $R$ be the cumulative set of records

Begin

{

**Estimation of ATR Range for $X$**

$\{$ATR Range of Session Requests Received$\}$

Session Requests Received (SRP)

- $SRP\_0\ (Open\ Value) = a_1$
- $SRP-H\ (High\ Value) = b_1$
- $SRP\_L\ (Low\ Value) = c_1$
- $SRP\_C\ (Close\ Value) = d_1$
- $SRP\_PL\ (Previous\ Low) = e_1$
- $SRP\_PC\ (previous\ close) = f_1$

True Range Estimation

$$Let\ x_1 = b_1 - c_1,\ x_2 = b_1 - e_1,\ x_3 = f_1 - c_1$$

$$X_{tr} = Max\ (x_1,\ x_2,\ x_3)$$

$$X = 10\ period\ Moving\ Average\ (X_{tr})$$

**Estimation of ATR Range for $Y$**

$\{$ATR Range of Session Requests Processed$\}$

Session Requests Processed (SRR)

- $SRR\_0\ (Open\ Value) = a_2$
- $SRR-H\ (High\ Value) = b_2$
- $SRR\_L\ (Low\ Value) = c_2$
- $SRR\_C\ (Close\ Value) = d_2$
- $SRR\_PL\ (Previous\ Low) = e_2$
- $SRR\_PC\ (previous\ close) = f_2$

True Range Estimation

Let

$$Y_1 = b_2 - c_2,\ Y_2 = b_1 - e_1,\ Y_3 = f_1 - c_1$$

$$Y_{tr} = Max\ (Y_1,\ Y_2,\ Y_3)$$

$$Y = 10\ period\ Moving\ Average\ (Y_{tr})$$

**First Level Decision – $DT_1$**

IF $X > (Y + 10\%)$,

{then

"$DT_1 = Monitor"and\ Verify\ DT_2$

Else

"Continue"

}

Second Level Decision – $DT_2$

$$Z = Absolute\ Difference\ (X,\ Y)$$

$$K = 10\ Day\ Exponential\ Moving\ Average\ of\ "Z"$$

If $Z > K$,

{

$(True)$ – $DT-2$ "*Monitor*"

Else

$$DT-2 = "Switch\ Load"$$

}

Based on the Decision Grid matrix (Table-2), the necessary load balancing action shall be initiated.

## 4 Experimental Study

The input data used in experimental study has been generated with Cloudsim [26], [27], [28], [29]

simulations, which enables the modelling of a high-dimensional Cloud Computing network reconstituting input jobs without taking into account the jobs' priority sequence. Without using preemption, the constraints are put into action to simulate from one CPU to another.

### 4.1 Empirical analysis

While the cumulative of 100 records were used, wherein the period time sequence (t), and the session requests processed by one particular server VM during the time period chosen is reviewed, the insight into the model refers to the potential way in which the Average True Range solutions can be adapted for improving the monitoring and controls for better levels of resource scheduling.

Among the cumulative set of 100 records $R$ , the first nine records were chosen to compile the moving averages and exponential moving average at the respective levels. Accordingly, the further 91 records were chosen for the detailed analysis in terms of how the solutions are resulting from the process, at two levels of decision-making.

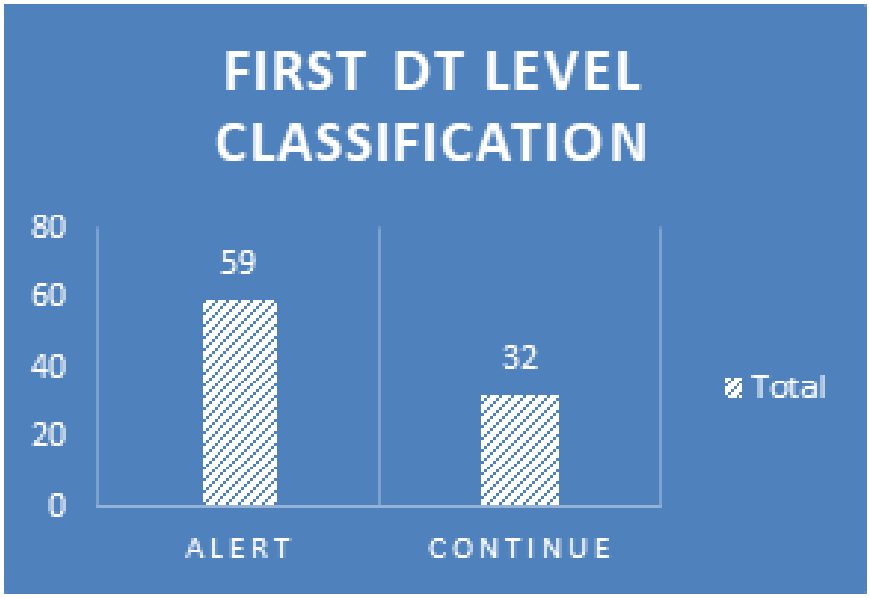


**Fig. 2:** The first DT level classification

The figure 2 Among the existing set of records chosen for the assessment, the cumulative number of records depicted for the “Alert” mode are around 59 records, and thus, when the system refers to such factors for more accurate watch, it helps in reducing the complexity of regularly switching the solutions and creating impact factors. Thus, in the context of second level decision monitoring, the further assessment is carried out in terms of the second layer assessment, in the case of the 59 dataset records considered for secondary confirmation.

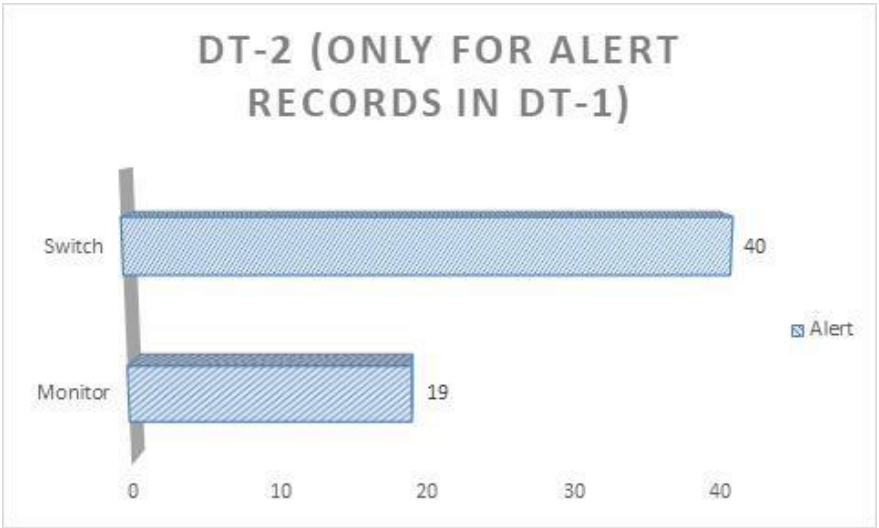


**Fig. 3:** The DT-2 only for alert records in DT-1

From the records Figure 3 chosen for the secondary review impact, 67% of the records are directed in the secondary decision process for the "Switch" position which indicates further action necessary by the administration teams. In lines with the functional processes considered for the process, the necessary "Switch" action can be performed by the team for the load balancing purposes. And in the case of the 23% of the records wherein the "Monitor" status is evident, the teams can track the conditions for any periodical switch essential in the model.

Thus, in a holistic time frame of 91 record period assessed for the cloud server, the proposed system has directed only 60% of the periods for load balance requirements, and in the other conditions the system has provided subtle indications. Also, if the variance is increased from the current 10% variation to the 20% variation in the server load balancing for X and Y variance (depending on the commitment of server response and minimum service quality) to the end-clients, the noise reduction for monitor conditions could be reduced.

In a comprehensive view of the solution, the other critical aspect performed in the model is to understand how the parallel execution of DT-1 and DT-2 reflects on the overall system. The matrix formation in the model is evident in the following table 3.

**Table 3:** The matrix formation in the model

| Row Labels | Monitor | Switch | Grand Total |
|---|---|---|---|
| Alert | 19 | 40 | 59 |
| Continue | 12 | 20 | 32 |
| **Grand Total** | **31** | **60** | **91** |

The above matrix table 3 refers to the combined view of the how the records are distributed among the four quadrants, ALERT, CONTINUE, MONITOR, and SWITCH. From the inputs while the red coded matrix point stands significant for the switch decision requirement, the yellow highlighted quadrant is a possible indication of imbalance in the load balancing, despite the load being addressed during the timeframe effectively. Considering application of other alternate VMs in the process can be more resourceful in the possible section.

In a distinct dimension, a comparative analysis of the model among various set of VMs that can help in improving the overall conditions of the IaaS environment and enable in taking better decisions among how the load balancing and resource scheduling can handled for betterment.

### 4.2 Performance Analysis

The approach of performance analysis contrasts the suggested ATRA model with existing models PSO-BOOST [17] as well as MOBA [16]. Based on the QoS criteria taken into account by the suggested and other current techniques for performance evaluation, they are distributing the resources. Additionally, we took several alternative looks at the performance indicators we'll discuss in the next section. The suggested ATRA has been assessed by contrast with modern methods like PSO-BOOST [17] as well as MOBA [16]. Various QoS indicators, such as task accomplishment frequency, resource usage frequency, as well as rescheduling frequency, would be used to gauge performance in this case. The observed frequency of resource usage for ATRA would be the maximum when compare to MOBA as well as PSO-BOOST. In this situation, the ATRA model's perceived rescheduling frequency would be linear and negligible in comparison to other methods. The scheduling frequency is thought to be minimal in ATRA, which leads to an ideal frequency of project completion. The complexity of the ATRA process would be low because of the updated scalable approach for determining the level of resource optimality.

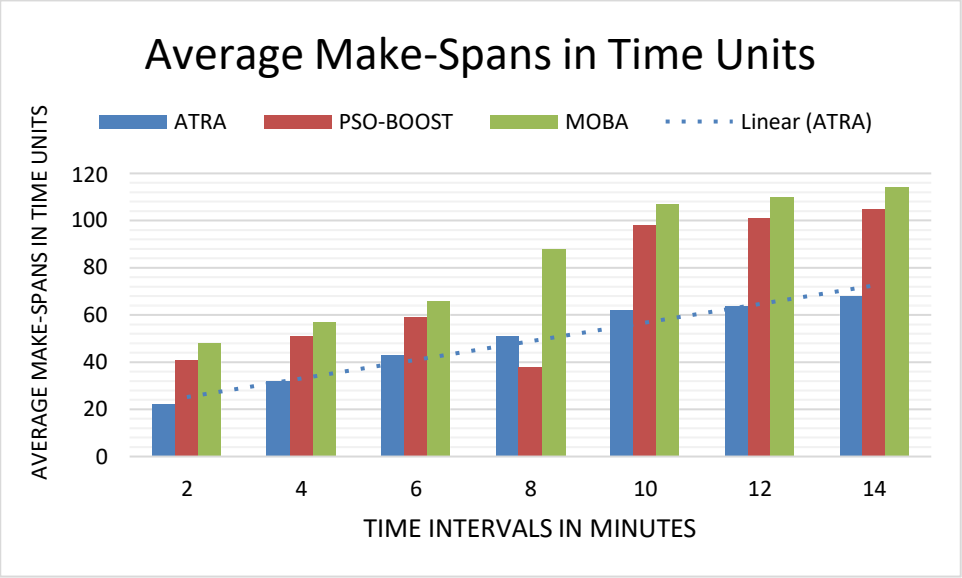


**Fig. 4:** Average Make-spans at fixed cloud resources and variable workflows

Let's start by examining how much time each of the alternatives that we are comparing requires. The "make-span" is a metric for calculating the interval between two continuous transactional intervals. In the initial batch of research (Figure 4).

The suggested ATRA approach does have a shorter make-span than the PSO-BOOST [17] as well as MOBA [16] methods, which is the first item that can be inferred from Figure 4. Because they simply consider the amount of time it necessitates to complete a job, the PSO-BOOST as well as MOBA tactics don't perform as well as ATRA. They ignore other crucial variables like latency or resource utilisation, which have a significant impact on the make-span. The second lesson you can take away with Figure 4 seems to be that when there are more Workflows, the make-span gets longer. This is due to the fact that additional Workflows are dependent on a particular number of cloud resources, thus increases the workload on each resource. The overall make-span will always be greater as a result of the higher wait periods between jobs.

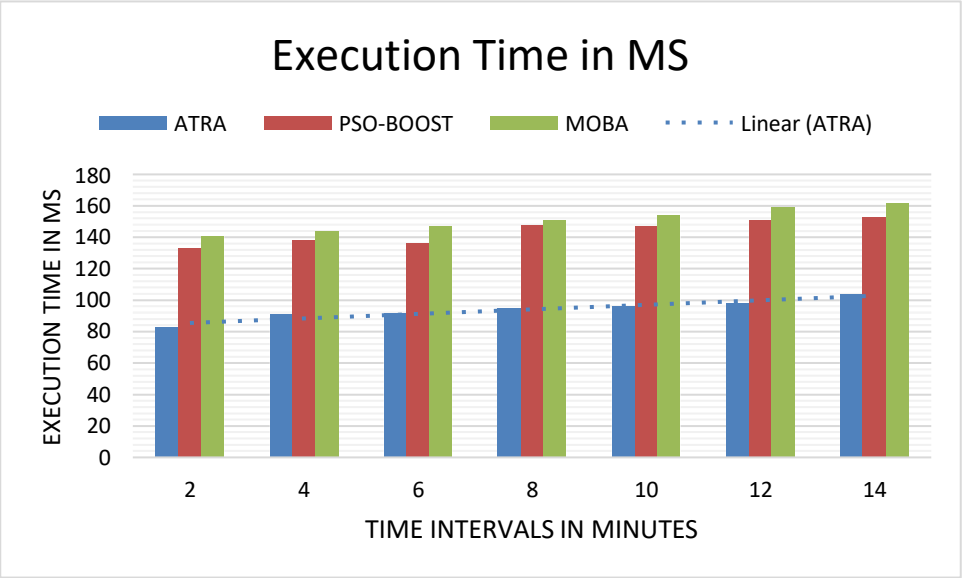


**Fig. 5:** Process time in milliseconds

The processing time of each approach was examined as a second metric. To do this, numerous trails of the experiments were conducted, each using a distinct set of resources as well as operations. Figure 5 demonstrates that process time is inversely related to the amount of workflows, which is reasonable given that processes take longer the more workflows, which depend on cloud resources. Additionally, since there are more cloud resources available, different algorithms run faster. This is due to the fact that additional cloud resources need sources to consider more inputs when creating their preference lists. Figure 5 demonstrates that compared to

the PSO-BOOST as well as MOBA techniques, our ATRA strategy is quicker to use.

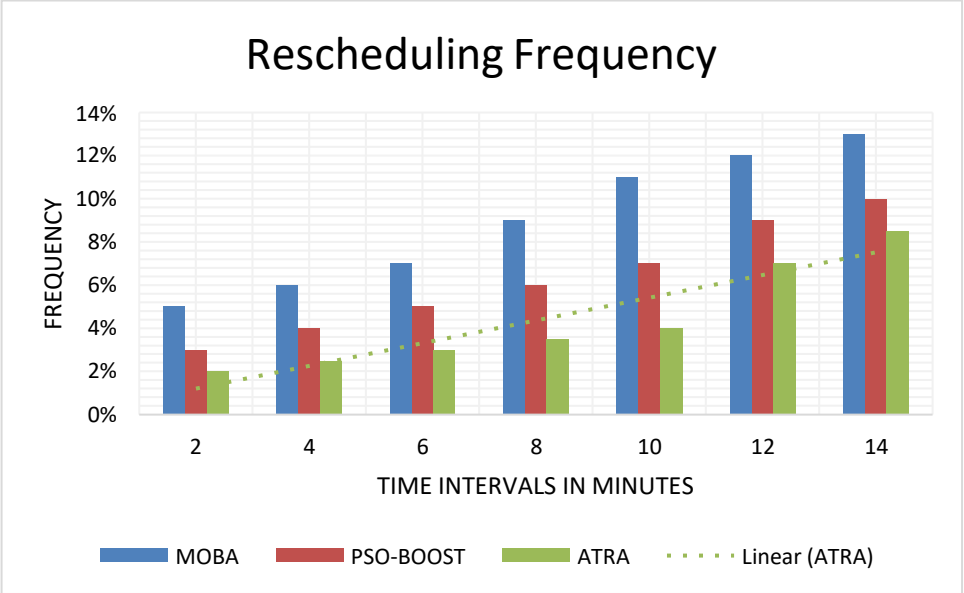


**Fig. 6:** Resource rescheduling frequency perceived

The rescheduling frequency seen at various time periods is shown in Figure 6. As seen in Figure 6, the rescheduling frequency at different time periods has been varied. Task load and perceived rescheduling frequency are negatively related. The results of the study showed that as the amount of tasks to be completed increases, there is a decrease in perceived rescheduling frequency. This figure shows that the projected ATRA model has significantly outperformed other approaches in lowering the frequency of rescheduling.

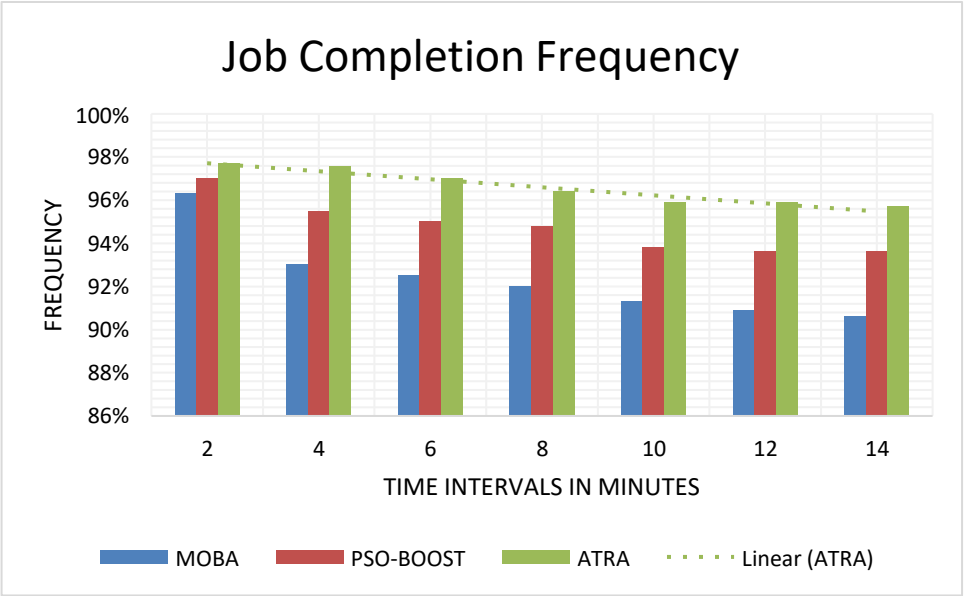


**Fig. 7:** Job completion frequency perceived

Additionally, Figure 7 demonstrates that, compared to the existing methodologies MOBA and PSO-BOOST, the suggested model ATRA would result in a greater perceived frequency of task completion. This suggests that ATRA can complete tasks more frequently and quickly than the existing methods, leading to better results with less effort. This could be particularly beneficial for those completing time-sensitive tasks, as ATRA could make them more efficient and improve the quality of their output.

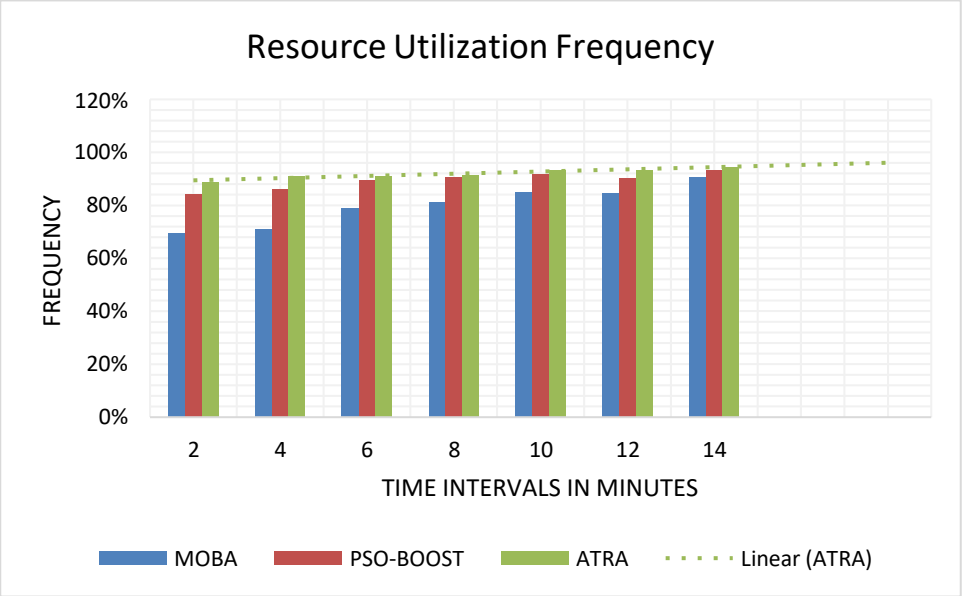


**Fig. 8:** Resource utilization frequency observed

Figure 8 shows that, in regards to the frequency of resource usage, which is a primary goal of resource scheduling strategies, the ATRA has a significant advantage against the other two current models MOBA and PSO-BOOST. This is due to the fact that the ATRA model takes into account the number of resources being used, rather than just focusing on throughput. By incorporating the number of resources into its decision-making process, the ATRA model is able to more effectively determine how much of a resource to allocate for each job.

## 5 Conclusion

Resource scheduling in the cloud environment is one of the critical challenges facing the IaaS service providers. More specifically in the case of the small-scale service providers, deployment of high-end machine learning applications could impact the economies of scale. Thus, in this manuscript, a contemporary approach of relying on the multi-setup kind of average true range analysis for the transaction requests received and processed is explored. Fundamentally, the system relies on the estimation of true range values, and the set of moving average estimations for developing the matrix of decision making. Two-stage set-up of decision-making on the resource scheduling tracking proposed in this study is assessed over the Kaggle datasets, and the outcome from the model signifies the efficacy of the system. However, considering the implementation of the models over the large-scale IaaS environment, it is important to reduce the trigger of the "Continue-Switch" segment of the matrix, which is indicating a probable gap in the server load balancing. As a part of the future research scope, the model can be applied to some machine learning models to understand the distinct kind of signals prompted by the system, and accordingly make informed decisions. Such a model can improve the overall efficacy of the proposed ATR range-based resource scheduling analysis.

## References

[1] Dewangan, B. K., Agarwal, A., Venkatadri, M., & Pasricha, A. (2018). Resource scheduling in cloud: A comparative study. International Journal of Computer Sciences and Engineering, 6(8), 168-173.

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

[3] Yu, H. (2021). Evaluation of cloud computing resource scheduling based on improved optimization algorithm. Complex & Intelligent Systems, 7(4), 1817-1822.

[4] Liu, Y., Wang, L., Wang, X. V., Xu, X., & Zhang, L. (2019). Scheduling in cloud manufacturing: state-of-the-art and research challenges. International Journal of Production Research, 57(15-16), 4854-4879.

[5] Lavanya, B. M., & Bindu, C. S. (2016). Systematic literature review on resource allocation and resource scheduling in cloud computing. international Journal of Advanced Information Technology (IJAIT), 6(4), 1-15.

[6] Gawali, M. B., & Shinde, S. K. (2018). Task scheduling and resource allocation in cloud computing using a heuristic approach. Journal of Cloud Computing, 7(1), 1-16.

[7] Strumberger, I., Bacanin, N., Tuba, M., & Tuba, E. (2019). Resource scheduling in cloud computing

based on a hybridized whale optimization algorithm. Applied Sciences, 9(22), 4893.

[8] https://www.researchgate.net/post/How-can-I-get-cloud-computing-data-sets

[9] Zhan, Z. H., Liu, X. F., Gong, Y. J., Zhang, J., Chung, H. S. H., & Li, Y. (2015). Cloud computing resource scheduling and a survey of its evolutionary approaches. ACM Computing Surveys (CSUR), 47(4), 1-33.

[10] Varshney, S., & Singh, S. (2018). A survey on resource scheduling algorithms in cloud computing. International Journal of Applied Engineering Research, 13(9), 6839-6845.

[11] Singh, A., & Malhotra, M. (2013). A comparative analysis of resource scheduling algorithms in cloud computing. Am. J. Comp. Sci. Eng. Surv, 1(1), 1-19.

[12] Maheshwari, S., Shiwani, S., & Choudhary, S. S. (2021, March). The Efficient Resource Scheduling Strategy in Cloud: A Metaheuristic Approach. In IOP Conference Series: Materials Science and Engineering (Vol. 1099, No. 1, p. 012027). IOP Publishing.

[13] https://www.kaggle.com/pitasr/scheduling-in-cloud-computing

[14] Priya, V., Kumar, C. S., & Kannan, R. (2019). Resource scheduling algorithm with load balancing for cloud service provisioning. Applied Soft Computing, 76, 416-424.

[15] Rohit Shere,Sonika Shrivastava, R.K. Pateriya(2017). CloudSim Framework for Federation of identity management in Cloud Computing. International Journal of Computer Engineering In Research Trends, 4(6), 269-276.

[16] Zheng, Jianguo, and Yilin Wang. "A hybrid multi-objective bat algorithm for solving cloud computing resource scheduling problems." Sustainability 13.14 (2021): 7933.

[17] Kumar, Mohit, and Subhash C. Sharma. "PSO-based novel resource scheduling technique to improve QoS parameters in cloud computing." Neural Computing and Applications 32.16 (2020): 12103-12126.

[18] Cao, J., & Wan, Y. (2017). U.S. Patent No. 9,635,134. Washington, DC: U.S. Patent and Trademark Office.

[19] Chen, J. (2016). Research on resource scheduling in cloud computing based on firefly genetic algorithm. Int. J. of Grid and Distributed Computing, 9(7), 141-148.

[20] Rudra Kumar, M., Rashmi Pathak, and Vinit Kumar Gunjan. "Diagnosis and Medicine Prediction for COVID-19 Using Machine Learning Approach." Computational Intelligence in Machine Learning. Springer, Singapore, 2022. 123-133

[21] Joga Singh, Supreet Kaur , Jasjit Kaur (2015). Security Issues' in Cloud Computing and its Solutions. International Journal of Computer Engineering In Research Trends, 2(8), 457-462.

[22] Rudra Kumar, M., Rashmi Pathak, and Vinit Kumar Gunjan. "Machine Learning-Based Project Resource Allocation Fitment Analysis System (ML-PRAFS)." Computational Intelligence in Machine Learning. Springer, Singapore, 2022. 1-14.

[23] Chan Phooi M'ng, J., & Zainudin, R. (2016). Assessing the efficacy of adjustable moving averages using ASEAN-5 currencies. Plos one, 11(8), e0160931.

[24] Suneel, Chenna Venkata, K. Prasanna, and M. Rudra Kumar. "Frequent data partitioning using parallel mining item sets and MapReduce." International Journal of Scientific Research in Computer Science, Engineering and Information Technology 2.4 (2017).

[25] Hyndman, R. J. (2011). Moving Averages.

[26] M. N. Prasad* et al., "Reciprocal Repository for Decisive Data Access in Disruption Tolerant Networks," International Journal of Innovative Technology and Exploring Engineering, 2019, 9(1), pp. 4430–4434.

[27] Kumar, V., et al. "Dynamic Wavelength Scheduling by Multiobjectives in OBS Networks." Journal of Mathematics 2022 (2022).

[28] van Rossum, H. H. (2019). Moving average quality control: principles, practical application and future perspectives. Clinical Chemistry and Laboratory Medicine (CCLM), 57(6), 773-782.

[29] Hicham, Gibet Tani, and El Amrani Chaker. (2016). "Cloud Computing CPU Allocation and Scheduling Algorithms Using CloudSim Simulator." International Journal of Electrical & Computer Engineering (2088-8708) 6.4.