CLR: Cloud Linear Regression Environment as a More Effective Resource-Task Scheduling Environment (State-of-the-Art)

https://doi.org/10.3991/ijim.v16i22.35791

Mohammed E. Seno[1(✉)], Omer K. Jasim Mohammad[2], Ban N. Dhannoon[3]
[1] Computer Science Department, Al-Ma'arif University College, Ramadi, Iraq
[2] Quality Assurance and Accridition, University of Fallujah, Fallujah, Iraq
[3] College of Sciences, Al-Nahrain University, Baghdad, Iraq
mohammed.e.seno@uoa.edu.iq

Abstract—The cloud paradigm has swiftly developed, and it is now well known as one of the emerging technologies that will have a significant influence on technology and society in the next few years. Cloud computing also has several benefits, including lower operating costs, server consolidation, flexible system setup, and elastic resource supply. However, there are still technological hurdles to overcome, particularly with real-time applications by providing resources. Resources allocation management most charming part of cloud computing; therefore, several authors have worked in the area of resource usage. This study introduces an innovative cloud machine learning framework-based linear regression approach called cloud linear regression (CLR), which entails both cloud technology and machine learning concept. CLR using machine learning yielded good prediction results for resource allocation management, as appeared with many researching, and still seek, research to raise optimal solutions to the resources' allocation problem as the aim of this study. This study discusses the relation between cloud resource allocation management and machine learning techniques by illustrating the role of linear regression methods, resource distribution, and task scheduling. The analytical analysis shows that the CLR promises to present an effective solution for resources (scheduling, provisioning, allocation, and availability).

Keywords—task scheduling, resource allocation, machin learning algorithms, linear regression, cloud-host, VM-placement, VM-migration

1 Introduction

Now just place the cursor in Cloud computing has grown in popularity throughout the internet computer family, and it acts as a hot emerging topic in different sectors including medicine, education, and libraries. It also demonstrates a new trend of deploying apps and services via the internet using virtualization technologies [1]. It is made up of a variety of shared resources that vary in terms of the cost of completing activities and the kind of scheduling resources available. Naturally, cloud computing

deploys anything as a service style on software or hardware level, and all such services are entirely dependent on task scheduling and resource availability [1, 2]. Cloud computing uses application service resource pools to aggregate all apps supplied by Internet service providers (ISPs). So, a physical machine resource pool is also utilized to give resources to hosts such as CPU or Memory [3]. While the remaining resource table and the usage rate table are used to figure out how to increase or decrease the number of virtual machines (VMs) needed by each application service [4]. This study surveys many kinds of research related to the enhancement of resource allocation and task scheduling in cloud computing-based machine learning. Resource allocation is the process of assigning and managing cloud resources to the needed cloud applications over the internet. Several key resource allocation challenges such as maximum computer performance and green computing, have lately attracted the attention of academics. While task scheduling is critical for improving cloud service flexibility and dependability. In a cloud system, the task scheduling mechanism aims to spread the load evenly across VMs based on resource capacity, such that no resource is overloaded or underutilized. Commonly, there are two forms of scheduling in a cloud environment: dynamic and static scheduling. Dynamic scheduling is used to rebalance the system by running, while the static scheduling is used configure the system balance from the schdulign start [4, 5]. The main goal of this study is to suggest a new machine-learning framework that includes an algorithm used for optimizing priority task scheduling and resource allocation. Such framework-based linear regression algorithm focuses on classifying each cloud-user with different levels to prioritize their tasks while arranging them in the task queue. However, the quality of service (QoS) in the private cloud-based mainly on investigated optimal scheduling for the tasks that minimizes the completion time, cost, and system load. In addition, it doesn't ignore the level of the cloud-users in the institution that uses the server. Hence, CLR should have the ability to change its behavior in ordering tasks in the queue dynamically. Consequently, CLR is based to create a sub-optimal resource allocation system for cloud computing and measure the reaction time in the next measurement period. However, regarding resources, it’s reallocated depending on the present condition of all virtual machines deployed in physical computers. Finally, the remainder of the study is organized as follows: Section 2 defines the concept of linear regression. The study novelty is shown in Section 3. Section 4 gives a short concept about the literature survey and the state of the art. Section 5 illustrates the steps of cloud linear regression in detail. The details analytical analysis of CLR is shown in Section 6. Finally, Section 7 details the conclusions of the study and future exploration areas.

1.1 Linear regression concept

Machine learning is widely utilized in a variety of areas to handle complex issues that are difficult to solve with traditional computer methods. The linear regression algorithm is one of the most basic and widely used machine learning techniques. It's a mathematical method for performing predictive analysis. Francis Galton [6] and Cleveland [7] initially proposed the notion of linear regression (LR) in 1894 and described it as a mathematical test that is used to assess and quantify the relationship between the

variables during consideration [8]. LR is a statistical process that predicts the estimations of the numeric or continuous individual and is known as regression. Generally, LR helps to compile and perceive the connotation between two continuous variables. A variable usually symbolized by x is called the predictor and another denoted by y is called the response variable see, $Eq.1$.

$$\hat{y}_i = \beta_0 + \beta_1 x_i \quad (1)\ [8, 18]$$

Where x_i is the predictor values, $\hat{y}_i$ are the predicted values, β_0 is the intercept and β_1 is the slope.

LR is widely employed in mathematical methods because it allows for the measurement and modeling of expected effects using numerous input variables. It is a data analysis and modeling technique that develops linear connections between dependent and independent variables via Evaluating Regression Model (ERM) [9]. ERM is to predict a continuous numeric value and realize two processes: (i) creating a scatter plot- predicted values and (ii) computing a statistical metric, to evaluate the predictive accuracy of a regression model. Two declared processes depend on the visual results obtained from a 2D. However, scatter plot-predicted models are simple to understand: the x-axis contains the predicted values, and the y-axis contains the response values [10,11]. Belong to Eq. 1, it predicts the actual values of y_i , but, the difference between predicted values of $\hat{y}_i$ with actual values of y_i is denoted as the remainder of the error and it can be calculated by the $Eq.2$.

$$Re_i = y_i - \hat{y}_i \quad (2)\ [18]$$

In order to reduce and minimize the error rate and calculate the estimation fulfill the value of $y_i\ and\ \hat{y}_i$, kindly read carefully in [18, 19].

2 CLR novelty

This section shows many novelties related to the proposed cloud machine learning solution that is listed as follows:

1. Proposing a prediction framework for anticipating current and future CPU utilization.
2. Utilizing a host's underload detection algorithm to reduce energy consumption (EC) by specifying a threshold value that will be set in the experimental work.
3. Utilizing the application-service prediction module (ASPM) that was deployed to forecast response time in line with the measurement period.
4. Classifying the priority levels (user and VM).

3 Related work

Newly, several studies have been published regarding to resource allocation and task scheduling based machine learning techniques. However, no universal solution for this

topic has been upcoming yet. Inspite of enhancements are achieved in several parts, most of such enhancements focus on the cloud task scheduling area, task resource requirements, resource allocation, and execution time. For example, Josep Ll. Berral et al. [11] propose a framework for dealing with uncertain data while maximizing performance by utilizing multiple strategies like turning on/off machines, power-aware consolidation algorithms, and machine learning techniques. In addition, such dealing utilizing by provides an intelligent consolidation methodology. Regrading to the machine-learning topic, they have applied models learned from prior system behaviors in the machine learning technique to estimate power consumption levels, CPU loads, and SLA timings, and enhance scheduling decisions. Moreover, this study evaluates and measures the use of such techniques with a proposed framework depending on simulation with representative heterogeneous workloads. However, in this study, the quality of the result indicates that the suggested is close to the optimal placement and behaves better when the level of uncertainty increases.

Akindele A. Bankole and Samuel A Ajila [12] present a new model for cloud client prediction models based Transaction Processing Performance Council web application (TPCW), In a bid to secure Service Level Agreement (SLA) requirements, Virtual Machine (VM) resources must be provisioned few minutes ahead due to the VM boot-up time. One way to do this is by predicting future resource demands. Also, to increase the response time and throughput, this study applied three techniques of machine learning Support Vector Machine (SVM), Neural Networks (NN), and Linear Regression (LR). The result of the study authors showed that the Support Vector Machine provides the best prediction model.

Chenn-Jung Huang et al. [13] illustrated a forecast model by employment Support Vector Regressions (SVRs). This proposed model is used to determine response time in all virtual machines in the next measurement interval it using genetic algorithms (GAs) to locate the resources in a virtual machine. The obtained result shows such a model is the best way to provide available hardware resources and is eligible in an effective cloud environment.

Jun-Bo Wang et al. [14] provided a novel machine learning framework for resource allocation with the use of cloud computing. The optimal or near-optimal solution of the most comparable historical situation is used to distribute radio resources for the current scenario, based on the extracted similarities. After that, the authors showed an example of beam allocation in multi-user massive MIMO systems to show that suggested machine-learning-based resource allocation outperforms traditional techniques. However, this study does not complete a holistic solution for resource allocation inside a cloud data center.

Jixian Zhang1 et al [15] suggested a model to evaluate the multi-dimensional cloud resource allocation problem using machine learning classifications and present two resource allocation forecasting approaches based on linear and logistic regressions. The suggested approach has an efficient influence on resource allocation in cloud computing, according to the findings of the experiments.

Jing Chen et al. [16] offered a proactive resource allocation strategy in cloud computing based on adaptive resource request prediction. It creates a paradigm for multi-objective resource allocation optimization that reduces resource allocation delay and

balances the consumption of different types of physical machine resources. The results of the experiments reveal that this strategy achieves a balanced consumption of CPU and memory resources while also shortening resource allocation time.

Thang Le D. et al. [17] illustrated a study of effective resource provisioning in combined edge-cloud systems and investigates technologies, techniques, and strategies for improving the dependability of distributed applications in varied and heterogeneous network environments. The survey is organized around a three-part breakdown of the dependable resource-provisioning problem: workload characterization and prediction, component placement, system consolidation, and application elasticity and remediation.

Nirmal Kr. Biswas et al [18] present a novel New Linear Regression (NLR) prediction model. The NLR model's primary intention is to take that the model goes through a straight line and a mean point. Future CPU utilization is predicted based on the proposed NLR model. The experiment shows that proposed algorithms reduced EC and SLAV in cloud data centers and can be used to construct a smart and sustainable environment for Smart Cities. This study proposed three basic algorithms, Hosts Overload Detection (isHOD), Hosts Underload Detection (isHUD), and Modified Power-Aware Best Fit Decreasing (MPABFD) algorithms. However, NLR depends on a straight line and a mean point in the prediction process.

As shown in Table 1, many studies attempted to solve the resource scheduling and provisioning problem in line with green services, however, no holistic research for applied novel learning prediction model work in a cloud environment. Herein, the proposed framework strives to present a perfect solution for scheduling problems.

Table 1. Summary of existing studies

Authors	Cloud Platform Engin used	Algorithm Contribution	Factor consideration	Features	Future Enhancement
Josep Ll. Berral et al[11]	Xen Server	Reinforcement Learning algorithms	CPU, power consumption	(CPU Usage, Timing SLAs	Needing to be extended to work rest of resources
Akindele A. Bankole Samuel A Ajila.[12]	Java Environment	SVM), Neural Networks (NN) and Linear Regression (LR).	CPU, cost	Response time and throughput	Needing to be extended to work with all resources that client needed or requested.
Chenn-Jung Huang et al[13]	Hyper-V	Support Vector Regressions (SVRs) and genetic algorithms (GAs)	Service Level Agreements (SLA)	utilization of resources	Applied model only in some resources (CPU, Memory) , need to approve the model can work all of resources.
Jun-Bo Wang et al. [14]	Cloud Sim	k-nearest neighbor, beam allocation algorithm.	Radio resources	multi-user massive MIMO systems, task scheduling	None
Jixian Zhang et al.[15]	Amazon EC2	linear regression, logistic regression		utilization of resources	Needing to support scheduling

Jing Chen et al.[16]	Java Environment	short-term prediction	CPU and memory	utilization of resources	None
Thang Le D. et al.[17]	Google, Amazon EC2).	multiple linear regression model	CPU, response time	utilization of resources	Needing to be extended to work with all resources that client needed or requested.
Nirmal Kr. Biswas [18]	Cloud- Sim	New Linear Regression(NLR) prediction model	CPU-Utilization predication and Host load computing	Resource scheduling enhancement	NLR assumes the model goes through a straight line and a mean point

4 CLR: Proposed framework

This section shows the main building of the proposed solution for optimizing resources allocation, resources scheduling, provisioning, and task scheduling.

4.1 Main building

Several daily services are offered by cloud technology like resource provision, resource allocation, task scheduling, etc, which in turn revealed an imbalance in the loads on cloud-host. Due to the heavy request for green services, it's essential to continuously enhance the performance and reliability of the cloud environment. As shown in Figure 1, the proposed cloud-machine learning framework includes a physical constitution besides software logically constitute.

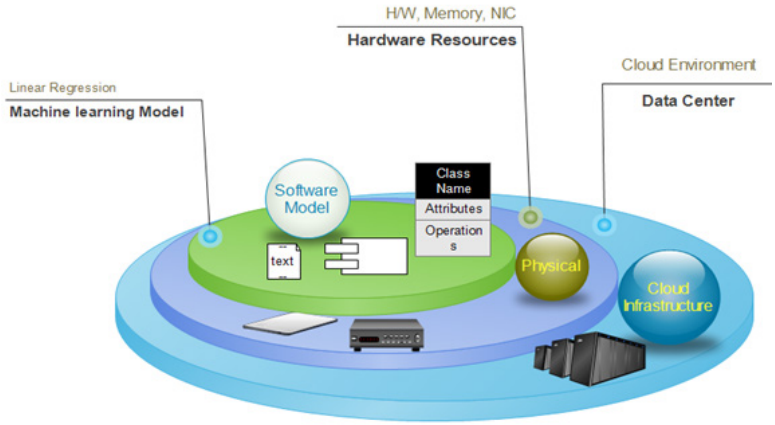

Fig. 1. Cloud-machine learning platform

Regarding physical constitute, it includes cloud-VMs, cloud infrastructure, and cloud storage. Cloud storage (internal or external) is one of the main contents of the framework; it's used to store a big amount of historical data. Historical data contains

information about the No. of users, No. of requests, No. of tasks, No. of VMs. etc. While the logical constitute includes No. of cloud-users, software models, and machine learning engines. Figure 2 depicts that the CLR framework completely depends on the linear regression approach.

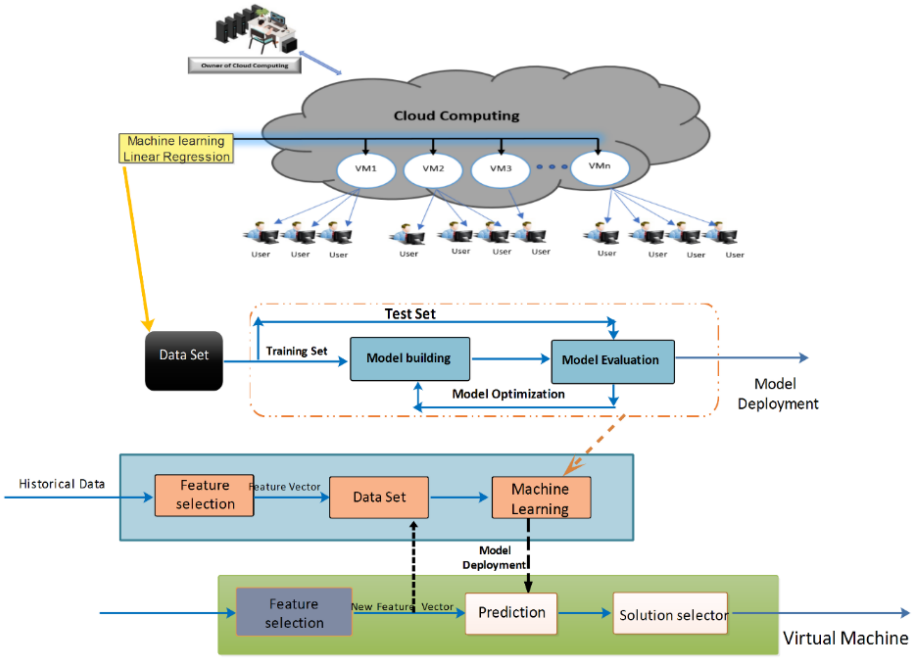

Fig. 2. CLR- main building model

CLR model going to be implemented on all cloud-VMs and taking into account its logical dependence on building an integrated machine learning model that complies with the requirements of cloud computing. Hence, CLR logically includes the following [18, 19, 20, 21]:

Dataset. Data Set (DS) is a collection of variable-data, usually presented in the standard form like a table template (row and column) and it represents a starting point to build machine-learning model. It gives values for each of the variables such as the height and weight of an object [19]. In the cloud environment, datasets are becoming progressively more pertinent when executing the performance assessment of task scheduling, resource-allocation, and resource provisioning. It is used for the examination of efficiency and performance in a real-world cloud. Logically, any update or change in the dataset-behavior or dataset-nature is reflected in the performance of scheduling and resource-allocation policies [20]. Due to the type of users' data confidentiality and policies in the open cloud environment, the real cloud workload is hard to measure for performance analysis. Therefore, using real tests obstructed the experiments to the scale of the test. Consequently, the performance accuracy with real-world datasets is most important in the field of research. The formal dataset and evaluation

process (training test) in a cloud environment revolves around the following parameters:

- Number of VMs: number of nodes used in the cloud.
- Number of cloud-users = number of users used in the cloud.
- Number of requests= number of requests used in the cloud.
- CPU usage data provided by cloud providers (CSP) and CPU utilization.
- Users level classification: type of user's priority.

Learning building model. Although various types of machine learning will have different tactics for training the model, there are basic steps that are utilized by most models. Machine learning models are authoritative tools used to efficiently and excellently perform energetic tasks and solve complex problems. These models have a wide range of uses in finance, medical, research, computing, banking, security, and diagnostic tools. The Heartbeat of all machine learning models are algorithms, they usually need huge amounts of high-quality data to be professionally trained. Such data need many steps to the preparations and formalization, which in turn builds an excellent effective model. Accordingly, each model must be well planned and managed from the beginning. The following points act as the basic steps to building a CLR-building model:

1. **Contextualizing:** this step is responsible for determining the requirements of the CLR -building model needs such as objectives, source of training data, and basic parameters that are needed to start. This point defines the problem that the CLR model needs to solve it and its success.
2. **Data Exploration and selection algorithm:** this step identifies the type of model that is required. CLR model depends on the type of task and the features of the dataset. Initially, the data should be explored by a cloud-administrator through the process of exploratory data analysis. This gives the cloud-administrator an initial understanding of the dataset, including its features and components, as well as a basic grouping. Regarding the selection algorithm, it completely depends on the understanding of the core data and the problem that needs to be solved. Generally, three major types of machine learning algorithms models such as supervised (labeled dataset), Unsupervised (unlabeled dataset), and reinforcement models (trail error feedback). In this study, CLR -building model is classified as a supervised algorithm.
3. **Dataset Preparation and Cleaning:** to ensure that the CLR -building model is an accurate model, ML needs arrays of high-quality training data. Typically, the model will learn the relationships between I/O data from the training dataset. The type of ML-training model specifies the form of the training dataset (labeled or unlabeled). The data quality and data standardization are mainly influenced by supervised and unsupervised algorithms. In addition, the data should be checked and cleaned. All mentioned characteristics lead to building an effective and accurate learning model.
4. **Dataset Splitting and Cross Validation:** the ability of ML to generalize and apply the learned from training data to new unseen data is a more important characteristic to build a real-world effective learning model. Systemically, if the selected algorithm is too closely aligned to the original training data, the obtained result will drop in

accuracy when encountering new data in a training environment. Usually, the preparation process split data into training and testing data. The training data is summarized in "*train and build*", while the testing data acts as new and unseen data, allowing it to be assessed for accuracy and levels of generalization [18, 19]. Hence, the cross-validation process validates the effectiveness of the model against unobserved data and it's categorized as exhaustive and non-exhaustive approaches [19, 20]. Figure 3 depicts the study methodology of mentioned two categories in line with a combination of training and testing datasets or creating a randomized partition of training and subset data test.

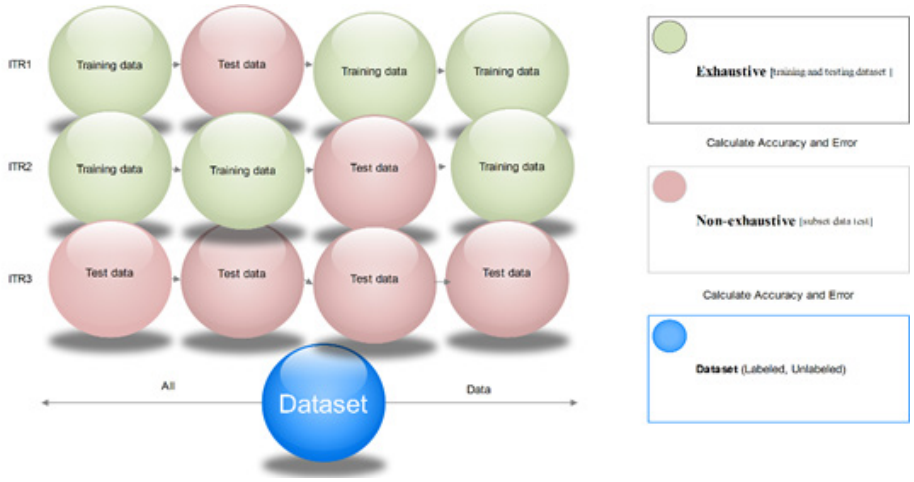

Fig. 3. Cross validation categorizes

Due to the type of cloud style, CLR is based on two categories through validation or splitting preparation process by predicting and computing current and historical CPU utilization. Then, detecting the overload and underload for cloud hosts and VMs.

Evaluating model. This part is responsible for measuring the prediction for future incoming data is applicable or not [21]. strives to increase the flexibility and scalability of computers by offering a variety of resources and standard services[22]. Since the future instances have unknown target values, hence, this part checks the data accuracy based on the target answer. This assessment-based as a proxy for predictive accuracy of future data. This part deals with the sample of labeled data with a target from the training data source. Due to the model nature work" *remembering characteristics*", the accuracy of evaluating the model for training the same data is not useful. Therefore, once finished the ML model training, the model the held-out observations for the target values, then, compares the predictions returned against the target value. Finally, calculate a metric obtained by the predicted and true values match.

Optimization model. The machine learning Optimization model (MLOM) is an essential part of achieving accuracy in a live open environment when building a machine-learning model. The subsequent investigations improved the results using ML algo-

rithm[23]. This part aims to tweak model configuration to improve accuracy and efficiency. Continuously, MLOM strives to improve the ML in order to fit accurate targets, tasks, and results. In general, ML has a rate of error; MLOM is the process of lowering this rate. The process of MLOM contains two hyperparameters, the assessment and model reconfiguration [21]. These hyperparameters aren't learned or developed by the model, nevertheless, their configurations and set by the model designer. The structure of the model, learning rate, and a number of clusters are famous examples of hyperparameters. Thus, in a cloud environment, the problem can be described as approximating a function (AF) that maps inputs to outputs and solved any conflict by framing it as function optimization [18]. The operation summarizes to define a parameterized mapping function (like.$\sum W(i)$): summation of inputs weighted). The optimization algorithm is used to account for the values of the parameters (coefficients model) that minimize the error of the function when used to map inputs to outputs. Finally, the following steps show the work methodology of this part:

- ML algorithms $\overleftrightarrow{\text{preform}}$ AF $\overleftrightarrow{\text{solved}}$ by function optimization (FO).
- FO $\overleftrightarrow{\text{minimize}}$ error, cost, loss.
- FO $\overleftrightarrow{\text{choose}}$ data preparation model, hyperparameter tuning (configured to tailor the algorithm), model selection.
- FO $\overleftrightarrow{\text{trnasfrom}}$ apply data prior modeling.
- FO $\overleftrightarrow{\text{choose}}$ modeling pipeline → Build final model.

At CLR framework, MLOM strives to achieve a map between current and historical cloud CP Utilization, and also, perform the exchange and placement between cloud-VMs after summation of the workloads in cloud-host.

Deployment model. The deployment model is the last step of the CLR building model, it is generally developed and proved in an online or offline cloud environment using training and testing datasets see Figure 3. In the online environment, the deployment model deals with new and unseen data, this matter helps the cloud owner to get accurate results because it is performing the task with live data on the cloud data center. While in offline mode, the deployment model executes the ML model locally on the cloud user's-client and the data never leaves the user-client. This type of deployment presents the service provider with the same level of control and security. On the other hand, containerization is a basic tool of the deployment model [21]. Container is a popular scalable environment for deploying ML models as the approach makes updating or deploying different parts of the model more straightforward.

4.2 Data processing part

This section shows in detail the type of operations that are performed on CLR-data entry.

Feature selection. Feature selection (FS) [21] is faithfully the process of selecting a subset of specific characteristics from a large volume of available share features. Feature selection is one of the most important parts of the learning building model and is used to build actionable intuitions and it is important to be able to select a subset of

important features from the massive number. It can constitute an entirely new area of ML studies where intense efforts are geared toward devising new innovative algorithms and tactics. Currently, many feature selection algorithms have been deployed such as evolutionary algorithms (e.g. Particle Swarms Optimization, Ant Colony Optimization) and stochastic approaches [20, 21]. Some of them are based on classical methods like simulated annealing and genetic algorithm. CLR utilizes the simulated annealing style for the feature selection and optimization matters.

Historical data. Learning from a large set of source data with numerous properties often necessitates a lot of memory and processing capacity, which might affect learning accuracy [17]. As a result, the unnecessary characteristics may be deleted without causing serious data quality loss. Historical data support enables the learning process to work faster and more effectively by providing numerous extrapolations to the ML designer. The goal of feature selection is to find and eliminate as many unimportant qualities as feasible.

As a result, preprocessing is essential to remove any incorrect or duplicate feature vectors, after that, all of the remaining feature vectors are gathered into a massive dataset. In addition, each feature vector in the dataset is split into a training and a test set at random. The training set typically contains 70-90 percent of the feature vectors [9].

To uncover the commonalities concealed in historical data, a supervised learning method in the CLR framework is used with the training set. Moreover, a predictive model may be developed, which will be utilized to make resource allocation decisions in the event of a future unforeseen event. More precisely, modern computing approaches may be employed to seek answers to the optimization problem using cloud computing. Finally, a high-performance resource allocation solution may be found offline and connected to each training feature vector.

Predictive model. The predictive model is a statistical technique that depends on ML and data mining to predict upcoming data outcomes with the assistance of historical and current data. Algorithms and techniques for machine learning (ML) help in pattern recognition and prediction[23]. The study methodology of the predictive model is completely based on the data analysis (current and historical), then, projecting a learned-model to generate the forecast learning environment. Due to the unsuitable state of the raw data (diversify and change), predictive analytics is an optimal solution for cloud-owner to tap data-driven by predictive modeling. Generally, classification models and regression models are two types of predictive models, these models are made up of algorithms [18, 21]. Classification models predict class membership, while the Regression models predict a number. Data mining, statistical analysis, and data patterns are famous operations performed by algorithms. Mostly, the predictive model is not static; it is regularly dynamic and validated to blend changes in the underlying data. Therefore, predictive models make expectations based on data behavior (what has happened in the past and what is happening now) and accordingly recalculated the relation between historical and new data. That means the data needs more training overtime to access the target and is convenient with a selected predictive model. In a cloud environment, the CLR model is based on regression algorithms and statistics to estimate the probability of task load and distribution in cloud virtual machines.

5 Cloud linear regression steps and algorithm [18, 21, 24]

CLR framework entails both cloud technology and machine learning concept and it's based on previous research studies such as [13, 17, 18, 24]. The vital idea of the CLR framework is to accept that the model goes through a straight line, a mean point, and a dynamic random search. CLR is based on the philosophy of running the maximum number of cloud-VMs in a host to increase performance and reduce the EC in the cloud data center. VM Consolidation (VM-C), VM Migration (VM-M), VM placement (VM-P), and building a linear regression prediction model are =\+the main operations in the CLR framework, see Figure 4.

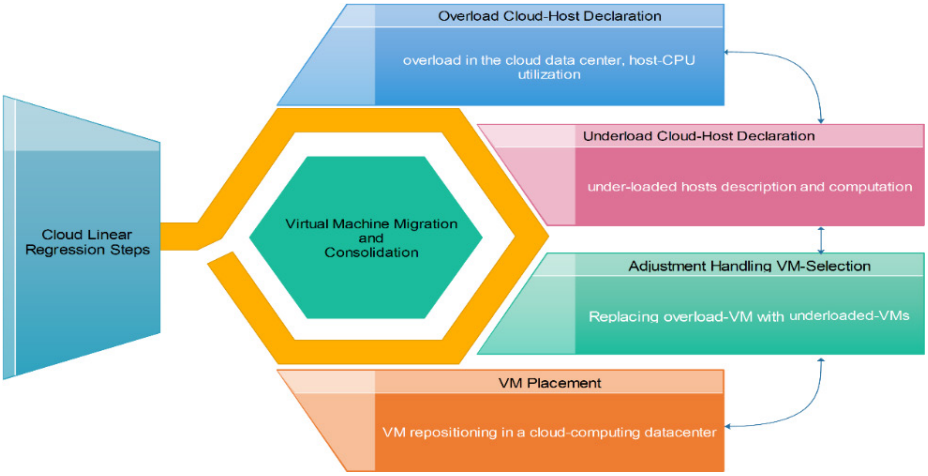

Fig. 4. Virtual machine migration parts

VM-C increase the number of VM execution under the cloud host, which in turn enhance the performance (request, response time). While VM-M is used to exchange and swap among VMs and task distribution in line with resource allocation and provisioning methodology. As shown in Figure 4, four main steps are important to achieve the VM-M process, listed as follows:

i) **Overload Cloud-Host Declaration:** in order to compute and divulge the overload in the cloud datacenter, host-CPU utilization should be predicated on the reverse history of host-CPU utilization. CLR will solve this prediction matter by recording the host-CPU utilization history. Accordingly, Figure 5 shows the proposed algorithm proposed for overload host detection and prediction of host-CPU utilization.

Algorithm	**Overload Host Declaration and Predicate Host-CPU Utilization**

Input: $H = Host$
n= length of host-CPU utilization.
rv = reverse Value of Host-CPU utilization history
$W_{vm} = [W_{\text{vm1}}, W_{\text{vm2}}, \dots, W_{vmn}]$ // workloads for cloud VMs.
Calculation: compute the CPU utilization prediction.
Output: Two values: the value of CPU utilization predication and Boolean

1. Initialize:
- Assign the value to the reverse utilization length (n ← rv).
- Set the number of virtual machines (vm).
- Virtual Machine Migration Intervals times (MT).

2. Begin
3. Compute the workload parameters for each v_m (W_{vm}, T_{vm}(length), L_{vm})
4. While (i ≠ 0) do
5. {
6. x[i] ← i + 1;
7. y[i] ← rv[i];
8. };
9. $\bar{x} \leftarrow \frac{\sum_{i-1}^{n} x_i}{n}$;
10. $\bar{y} \leftarrow \frac{\sum_{i-1}^{n} y_i}{n}$

Based Eq.1. & Eq.2. //linear regression applied.

11. Compute the value of β_0, β_1 basedEq. 1, Eq. 2, and Eq. 10, Eq11 in [18].
12. Calculate (MT ← $\frac{\text{VMMigration Time}}{\text{scheduling period}}$).
13. CPuUtilization ← $\beta_0 + \beta_1$(n + MT);
14. PredUtilization ← CPuUtilization × workload parameters
15. Return PredUtilization
16. If (CPuUtilization < n) then
17. Return total host request.
18. Else
19. for i ← 1to n do
20. {
21. rv[i] ← CPuUtilization[n − i + 1];
22. }
23. Else
24. PredUtilization ← CLR(rv);
25. return rv ≥ 1;
26. end
27. End.

Fig. 5. Overload host declaration and predicate host-CPU utilization algorithm

As shown in Figure 5, Overload Host Declaration and Host-CPU Utilization computing completely depends on one assignment loop. This loop covers the computing of the value of historical CPU utilization through cloud-VMs. Costly, this algorithm needs $O(n^2+n)$ as a cost of execution stemming from Figure 4. The algorithm computes the

value of x and y in line numbers 6 and 7 respectively and the value of $\beta 0$ and $\beta 1$ is calculated in line number 11 using Eq. 10 and Eq. 11 [18]. The migration interval is obtained in line number 12. Regarding the overload cloud-host declaration process, the algorithm describes the relation between current host-CPU utilization and the host-CPU utilization History. Then, compute the reverse host-CPU utilization history using line numbers 16, 17, and 21. The host-CPU utilization is predicted using this algorithm by line number 24, while, a host is overloaded or not in line 25. Finally, Current utilization is calculated in line number 13 and the final predicate utilization is computed using line number 14.

ii) **Underload Cloud-Host Declaration:** one of the basic steps of VM-M and VM-C that is responsible for the under-loaded hosts' description and computation. If the value of under-loaded hosts is high and delays the efficiency of the cloud datacenter, CLR starts to migrate the tasks to other hosts. Then, shut down all under-loaded hosts in order to minimize EC and increase resource utilization and availability.

iii) **Adjustment Handling VM-Selection:** this step adjusts the cloud-VMs by handling the VMs from overload cloud-host and replacing them with VMs from cloud-hosts that are underloaded. In this context, the methodology of VM selection is fully dependent on the policy of minimum migration time. The replacement and selection process used the minimum migration time policy to select a VM that necessitates at least an optimal opportunity to migrate in contrast with different VMs. Usually, the migration time is calculated by the division between the measure of RAM utilization for each VM and the network bandwidth for a host.

iv) **VM Placement :**the process of VM-P is a classical operation used for VM repositioning in a cloud-computing data center. Regularly, in this step, all VMs will be sorted decreasingly as per the CPU utilization. Then, each VM will rotate the suitable cloud-hosts in which the EC is least. Lastly, this step assigns the VM to the minimum power consumed cloud-host to increase energy efficiency.

Finally, the steps of the CLR framework summarizes in the following practical steps:

Step1: cloud-users' submit a new task to the cloud-host.

Step2: Computing a total estimated value for four parameters to specify workloads under VMs configuration (level of users, time, cost, and load).

Step3: Create a parameter vector for each submitted task (put the level of parameters first).

Step4: Computing the weight for each submitted task.

Step 5: Calculating the Host-CPU utilization (Current, Historical).

Step6: Calculating the predicate host-CPU utilization based on Step 5.

Step7: According to VM-migration and consolidation, the CLR model starts to build a learning model like as mentioned in the **Main Building Section** to compute the prediction of CPU utilization, RAM utilization, and execution cost.

Steps8: Check the Task sample is finished, if **yes** go to step6, else, return to Step4.

Step9: After the repetitions terminate, CLR reconfigures the task distribution, VM migration, VM placement, and resource allocation.

Step10: Finally, CLR reveals optimal sharing of resources, reveals an optimal execution time, and cost.

6 CLR analytical analysis

A huge number of researches and studies on cloud optimization performance through the enactment of task scheduling, distribution, resource allocation, and availability [22]. All optimization studies classified a category based on some criteria such as (priority, reduced Make-span, Energy Efficient, improved cost, QoS, improved consistency, and improved fairness). In this study, the author presents a new idea related to machine learning regression by converting all events in the cloud's data center to numeric and statistical values. This section discusses the theoretical analysis for the performance and the effectiveness of the cloud linear regression. Workload balancing, cost and performance evaluation, and execution time are the main criteria for study analysis and discussions.

6.1 Workload balancing

This study aims to accomplish that the CLR illustrates an optimal solution for task scheduling and distribution in line with resource availability and efficiency. According to the mentioned illustration about CLR methodology, CLR is well-organized in identifying the workloads, which reflects the fair distribution of the shared resource. In a nutshell, the CLR achieves the greatest result when it works in an online open environment by implementing VM-migration, replacement, and consolidation. Predictably, the advent of CLR helps to solve the problem of load balancing by the possibility of a combination between machine learning algorithms and cloud computing profiling techniques for random workloads proprieties. This point helps to provide a continuous perception of the intensive workloads. CLR is proposed to measure the randomness of the discrepancy of the dynamic tasks by regularly distributing, implementing, and periodically varying the resource allocation in each duration time. According to logical analysis, CLR promises to achieve better resources allocation, task scheduling in terms of the total cost at the runtime.

6.2 Cost and performance evaluation

One of the most important measures in the cloud environment is resource and task scheduling cost. Hence, all studies take into account this point very seriously. CLR is presented to find an effective way of the dynamic shared parameter to discover an optimal intelligent resource allocation model for different tasks at the same time. CLR assurances to provide the best results at a cost of resources, because, the CLR is used to find the value of the shared parameter in the online data center and escape from the traditional local environment. Scientifically, performance measurement is an integral part of a work. The performance evaluation CLR will be done practically in line with a comparative study with new trends of proposed algorithms. CLR is considered an intelligent mining model, thus, it works satisfactorily through the dynamic exploration environment parameters (processor, memory) for each VM. In addition, it also works fine with re-scheduling, reconfiguring tasks and VMs to make space for a newly arrived query.

6.3 Execution time

All proposed renewed scheduling models are good in some characteristics; however, none of them achieves perfect results. CLR Model promises to provide an efficient solution for speeding up the execution time and high response for assigning tasks.

The efficient feature is coming from the study style, and dynamic search in the open live environment, of the proposed model. Attempting to move or exchange a set of tasks from the overloaded VM to another underload VM, and calculating the total execution time for the task scheduling and CPU utilization compared to the makespan of the task solution are two basic features for the present optimal execution time. Substantially, the optimization matter is executed on each overload VM and continues until there is no more enhancement in the configuration value. Theoretically, cloud computing performance is seriously based on the effectiveness of the type-training model, quality training, data size, and VM-C steps. On the other hand, Total execution time (ET) is the essential time to execute the model and linked data set. Theoretically, the CLR model will show that the execution time is much faster than other benchmark prediction models. Generally, CLR will significantly defeat the well-known prediction models, and it significantly reduces EC. Finally, CLR can be used to build a smart and sustainable cloud environment.

7 Conclusion and future's plan

This study demonstrates the need for machine learning models to enhance cloud resource scheduling, provisioning, and service planning. CLR acts as an innovative cloud machine learning model that utilized the linear regression methodology, which entails both cloud technology and machine learning aspects. The main goal of CLR summarizes in optimizing priority the of task scheduling and resource allocation. it's focused on giving classification for each cloud-user with different levels to prioritize their tasks during arranging the tasks in the task queue. Consequently, CLR is based to create a sub-optimal resource allocation system for cloud computing and measure the reaction time in the next measurement period. However, regarding resources, it's reallocated depending on the current state of all host-virtual machines (over and under load) that are deployed in the physical host. Theoretically, when comparing CLR to another model, the proposed linear regression model is convenient to provide better prediction results. The method may bring the viable solution extremely near to the ideal solution in terms of resource usage by learning the training data. However, a lot of challenges are expected to encounter the CLR that are needed to be solved such as portability, pricing and execution, and architecture.

- **Portability:** cloud architecture should have the capability to softly and scalable transfer VMs from one cloud-host to another.
- **Pricing and Execution time:** it is important for the cloud-user to have levels of time to achieve the requests.

- **Open Architecture:** application or service template must be considered through planning the provisioning scheme, and constructing a deployment plan in the cloud data center.
- **Unsuitability:** CLR applied to independent random unsuitable tasks. In the future, we plan to develop and implement CLR in a real cloud environment using windows server-Hyper V.

8 References

[1] Omer K. Jasim Mohammad, GALO: A New Intelligent Task Scheduling Algorithm in Cloud Computing Environment, International Journal of Engineering & Technology, 7 (4) (2018) 2088-2094. https://doi.org/10.14419/ijet.v7i4.16486

[2] Shieny, J. A Survey on Cloud Computing: Architectures, Data Storage, Services, Security and Applications-manager's Journal on Cloud Computing, vol. 4, issue 2, pp 30-38, 2017.

[3] C.G. Ralha, A.H.D. Mendes, L.A. Laranjeira et al., Multiagent system for dynamic resource provisioning in cloud computing platforms, Future Generation Computer Systems (2018). https://doi.org/10.1016/j.future.2018.09.050

[4] Piotr Nawrocki, Mikolaj Grzywacz, Bartlomiej Sniezynski, Adaptive resource planning for cloud-based services using machine learning, Journal of Parallel and Distributed Computing 152 (2021) 88–97. https://doi.org/10.1016/j.jpdc.2021.02.018

[5] Mehmet Demirci, A Survey of Machine Learning Applications for Energy-Efficient Resource Management in Cloud Computing Environments, TSINGHUA SCIENCE AND TECHNOLOGY, Volume 21, Number 6, pp 660-667, 2016.

[6] Galton, F. (1894), Natural Inheritance (5th ed.), New York: Macmillan and Company.

[7] Cleveland WS. Robust locally weighted regression and smoothing scatterplots. J Am Stat Assoc 1979;74(368):829–36. https://doi.org/10.1080/01621459.1979.10481038

[8] Altaf Hussain and Muhammad Aleem, GoCJ: Google Cloud Jobs Dataset for Distributed and Cloud Computing Infrastructures, Data 2018, 3, 38. https://doi.org/10.3390/data3040038

[9] K. Phaneendra and M. Babu Reddy, Linear Regression based Aggressive Resource Provisioning for Cloud Computing, International Journal of Research, Volume 6, Issue 10, OCT/2017.

[10] Lei Shi, Jing Xu, Lunfei Wang, Jie Chen, Zhifeng Jin, Tao Ouyang, Juan Xu, and Yuqi Fan, Multijob Associated Task Scheduli ng for Cloud Computing Based on Task Dupli cation and Insertion, Wireless Communications and Mobile Computing, Volume 2021, Article ID 6631752, 13 pages. https://doi.org/10.1155/2021/6631752

[11] JosepLl. Beral, RicardGavalda, Jordi Torres. AdaptiveScheduling on Power-Aware Managed Data-Centersusing Machine Learning [R]. ResearchReport number: UPC-LSI-11-7-R, July 2011.

[12] Akindele A. Bankole Samuel A. Ajila, Predicting Cloud Resource Provisioning using Machine Learning Techniques, 2013 26th IEEE Canadian Conference Of Electrical And Computer Engineering (CCECE), Chennai, India. https://doi.org/10.1109/CCECE.2013.6567848

[13] Chenn-Jung Huang, Yu-Wu Wang, Chih-Tai Guan, Heng-Ming Chen, and Jui-Jiun Jian, Applications of Machine Learning to Resource Management in Cloud Computing, International Journal ofModeling and Optimization, Vol. 3, No. 2, April 2013.

[14] Jun-Bo Wang, Junyuan Wang, Yongpeng Wu, Jin-Yuan Wang, Huiling Zhu, Min Lin, Jiangzhou Wang, A Machine Learning Framework for Resource Allocation Assisted by

Cloud Computing, IEEE Network, Volume: 32, Issue: 2, March-April 2018. https://doi.org/10.1109/MNET.2018.1700293

[15] Jixian Zhang, Ning Xie, Xuejie Zhang, Kun Yue, Weidong Li, and Deepesh Kumar, Machine Learning Based Resource Allocation of Cloud Computing in Auction, Computers, Materials and Continua, vol.56, no.1, pp.123-135, 2018.

[16] Jing Chen and Yinglong Wang and Tao Liu, A proactive resource allocation method based on adaptive prediction of resource requests in cloud computing, Wireless Com Network (2021) 2021:24. https://doi.org/10.1186/s13638-021-01912-8

[17] Thang Le Duc, Rafael García Leiva, Paolo Casari, and Per-Olov Östberg. 2019. Machine Learning Methods for Reliable Resource Provisioning in Edge-Cloud Computing: A Survey. ACM Comput. Surv. 52, 5, Article 94, (September 2019), 39 pages. https://doi.org/10.1145/3341145

[18] Nirmal Kr. Biswas, Sourav Banerjee, Utpal Biswas, Uttam Ghosh, An approach towards development of new linear regression prediction model for reduced energy consumption and SLA violation in the domain of green cloud computing, Sustainable Energy Technologies and Assessments 45 (2021) 101087. https://doi.org/10.1016/j.seta.2021.101087

[19] Rafael Moreno-Vozmediano, Rubén S. Montero, Eduardo Huedo and Ignacio M. Llorente, Efficient resource provisioning for elastic Cloud services based on machine learning techniques, Journal of Cloud Computing: Advances, Systems and Applications (2019) 8:5. https://doi.org/10.1186/s13677-019-0128-9

[20] Piotr Nawrocki and Patryk Osypanka, Cloud Resource Demand Prediction using Machine Learning in the Context of QoS Parameters, Journal of Grid Computing (2021) 19: 20. https://doi.org/10.1007/s10723-021-09561-3

[21] Akbar Telikani, Amirhessam Tahmassebi, Wolfgang Banzhaf, and Amir H. Gandomi. 2021. Evolutionary Machine Learning: A Survey. ACM Comput. Surv. 54, 8, Article 161 (October 2021), 35 pages. https://doi.org/10.1145/3467477

[22] A. Edinat, R. Al-Sayyed, and A. Hudaib, "A Survey on Improving QoS in Service Level Agreement for Cloud Computing Environment," *Int. J. Interact. Mob. Technol.*, vol. 15, no. 21, pp. 119–143, 2021. https://doi.org/10.3991/ijim.v15i21.26379

[23] A. Dirin and C. A. Saballe, "Machine Learning Models to Predict Students' Study Path Selection," *Int. J. Interact. Mob. Technol.*, vol. 16, no. 1, pp. 158–183, 2022. https://doi.org/10.3991/ijim.v16i01.20121

[24] Jinn-Tsong Tsai, Jia-Cen Fang, Jyh-Horng Chou, Optimized Task Scheduling and Resource Allocation on Cloud Computing Environment Using Improved Differential Evolution Algorithm, Computers & Operations Researc.

[25] F. E. F. Samann, A. M. Abdulazeez, and S. Askar, "Fog Computing Based on Machine Learning: A Review," Int. J. Interact. Mob. Technol., vol. 15, no. 12, pp. 21–46, 2021. https://doi.org/10.3991/ijim.v15i12.21313

9 Authors

Mohammed E. Seno received M.Sc. (2015), in Computer Science from College of Computer- BAMU university, India, and get B.Sc. (2011) in Computer Science from College of Computer- university of Anbar. His work experience includes 5 years as an academic in Iraq: Private Sector "Alma'arif University College". He can be contacted at email: mohammed.e.seno@uoa.edu.iq

Omer K. Jasim is Head of Quality Assurance and Accridition and an Associate Professor at the Universiti of Fallujah, Iraq, he received his Ph.D. (2015-followership)

in cloud computing Security from Faculty of Computer and Information Sciences- Ain Shams University -EGY, M.Sc. (2009), in Computer Science from College of Computer- Uni. of Anbar, Iraq, and get B.Sc. (2007) in Information Systems from same college. He got the first ranking in three degrees. His work experience includes 9 years as an academic in Iraq: Private Sector "Alma'arif University College" as Head of Computer Science Department (2010-2012)- Deputy Dean at same college. He is assistant professor of cloud computing and he worked at University of Fallujah, Director of computer center. Currently, he is head of quality assurance and accreditation. He is the author/co-author of over 30 research publications. He can be contacted at email: omerk.jasim@uofallujah.org.

Ban N. Dhannoon, PhD holder since 2001, currently, a member of teaching staff in Computer Science Dept. / College of Science/ Al-Nahrain University, Iraq. My main research concerns are: 1. Artificial Intelligent (machine learning, Multiagent, Deep Learning) 2. Digital Image Processing 3. Coding (encryption, data compression, representation) 4. Pattern Recognition & Classification 5. Bioinformatics she can be contacted at email: ban.n.dhannoon@nahrainuniv.edu.iq.

Article submitted 2022-09-03. Resubmitted 2022-10-11. Final acceptance 2022-10-12. Final version published as submitted by the authors.