National College of Ireland

Resource Management in a Cloud Computing Environment using Generative Adversarial Networks (GANs)

MSc Research Project
Cloud Computing

Kelechukwu Chima
Student ID: 19202181

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Kelechukwu Chima
Student ID:	19202181
Programme:	Cloud Computing
Year:	2021
Module:	MSc Research Project
Supervisor:	Vikas Sahni
Submission Due Date:	16/08/2021
Project Title:	Resource Management in a Cloud Computing Environment using Generative Adversarial Networks (GANs)
Word Count:	6558
Page Count:	21

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Kelechukwu Chima
Date:	16th August 2021

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Resource Management in a Cloud Computing Environment using Generative Adversarial Networks (GANs)

Kelechukwu Chima
19202181

Abstract

Automating any resource allocation technique allows implementation of elastic cloud services by providing available cloud resources on-demand. This is important for decreasing energy consumption and meeting the SLAs and QoS, mostly for services that the QoS is dependent on latency and response time (Web servers, Big data analytics in real time). Elasticity is one of the key feature of the cloud computing platform, where the resources can be managed based on the user demand. Because the user requests/demand varies and fluctuates in a cloud environment, there is the issue of managing resources. This paper presents and evaluates a Generative Adversarial Network (GANs) based predictive resource allocation mechanism based on machine learning techniques. The GAN based mechanism aims at accurately predicting the load of any server and adequately provision the optimal number of resources required to optimise the response time and satisfy the SLA while also mitigating the issue of resource over-provisioning. Two other techniques (Rule-based and Linear Regression Model) are also implemented in this paper,the results show that the proposed model has the capability of achieving an optimal forecasting accuracy than the other two techniques implemented. The results are evaluated using the following metrics; latency,scaling response, idle cycles and task processed by the cloud system.

1 Introduction

In recent years, cloud computing has become the most cutting-edge technology, It involves the connection of a set of servers and networks to provide a virtual working environment on a "pay-as-you-go" basis. While there are multiple advantages attached to cloud computing, there are certain issues that needs to be addressed, one of those issues is Resource Management. According to Mustafa et al. (2015), effective management of resources is one of the most complicated procedures in the cloud environment, and should be addressed with priority.

The three distribution models in cloud computing are ; 1)Softare as a Service (SaaS) 2)Platform as a Service (PaaS) and 3) Infrastructure as a Service (IaaS). The Saas environment includes a variety of software applications that users can access, the users can access this environment concurrently without interfering with each other. PaaS provides its users with cloud platform (hardware, software and infrastructure) for developing and

managing their applications without the cost and complexity of having and maintaining the same platform on an on-premise infrastructure.

In an IaaS environment, compute, networking services and storage are offered on-demand to its users and it is the most closely related to resource management.Moreno-Vozmediano et al. (2019). The figure below depicts the resource management taxonomy Bermejo et al. (2017).

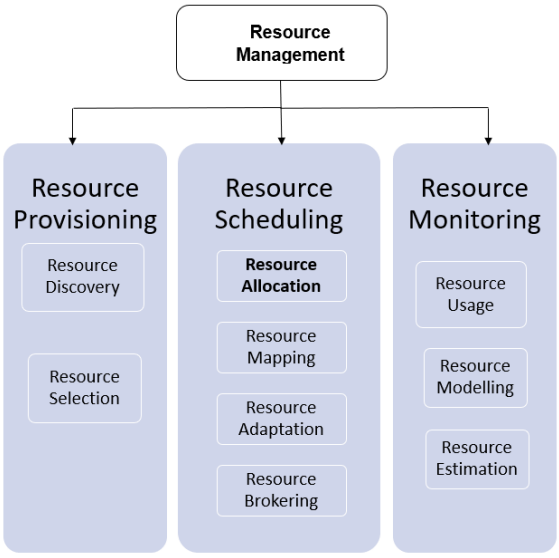

Figure 1: Resource Management Processes

Resource management can be defined as the process that involves assigning on-demand resources (storage, servers) where users can implement their projects, configure computer networks and virtual machines to an array of cloud applicationsMadni et al. (2017). Resource management involves meeting the user expectations by distributing cloud resources effectively.

The centralized resource manager manages cloud computing resources. The tasks were assigned to the appropriate VMs by centralized resource management. The resources of the cloud data center are made available to users/applications via Virtual Machines (VMs). Virtual machines are used to satisfy the resource requirements of programs and to provide run-time assistance. Virtual machines (VMs) run on a server and provide a multi-OS environment with application support. One or more VM(s) can be placed or deployed on a physical machine that meets the requirement for the VM. The work can be scheduled dynamic load balancing across hosts in cloud computing settings utilizing visualization technologies.

From figure 1 above 1, Resource Management is divided to three main sections. This research focuses on Resource Scheduling section where Resource Allocation falls under . Resource allocation involves efficiently allocation resources to its users. Predicting workloads and in extension managing cloud computing resources can complement efficiency in cloud computing environment. The cloud computing resources are provisioned based on the forecast of workload and thus scales to match the workload just in time. The most common methods for workload forecasting is machine learning methods and statistical methods. Statistical forecasting methods predicts the future based on the data collected

from the past by performing trend analysis, customer behaviour etc. Due to the inability to predict for long term, researchers applied the use of machine learning.

To prevent over-supply and under-provisioning problems, resources should be accessible for end-users with minimum administration and an effective mechanism for allocating resources. The mechanism of elasticity also functions as a cloud. As a result, static scheduling methods cannot be used in cloud workload scheduling approaches, and dynamic scheduling will play a significant role in maximizing the use of cloud resources. Various particle swarm optimization (PSO) techniques may be used to compute the dynamics of resource availability and job allocation. The resources should be made accessible under the service level agreement (SLA) agreed upon between end-users and cloud service providers (CSP) when the services request is established. To provide resources to end-users, several policies and indicators related to the SLA will be employed. The advantages of good resource use can improve the CSP's income.

Resource Allocation is a subset of resource management that seeks to allocate available resources in the most cost-effective way possible. Resource allocation is one of the most critical concerns, and if it is not managed effectively, it will famish the services. Only enabling the service provider to handle the resources for each module solves this problem. The issue of resource allocation management is exacerbated by two major factors: the first is dynamically changing demands of cloud users, and the second is the necessity to meet resource requirements in diverse settings. There are two ways to do this: rule-based and theory-based. The present methods for the process of resource techniques function well with little amounts of data, but when huge amounts of data are examined, these methodologies fail to perform efficiently. As a result, a machine learning algorithm is utilized. Many studies have shown that employing a machine learning approach for allocation and deallocation performs better than traditional techniques. One of the primary benefits of employing machine learning methods for resource allocation in cloud computing is that it can manage both static and dynamic workloads.

In Machine Learning, an algorithm is applied to historical dataset and it predicts the prospective outcome(workload). In this research, simple machine learning algorithms such as linear regression, will be used in this paper. The dataset is tested using a threshold-based technique and then compared to the suggested Generative Adversarial Network (GAN) algorithm. However, the accuracy of these predictions is not only dependent on the algorithms utilized but also, on the data provided for training. A Generative Adversarial Networks (GANs) is a system that uses an adversarial process to learn a function with an unknown distribution. Image creation is one use of GANs, which can help with dataset constraints by offering augmentation. GAN is a machine learning paradigm in which two neural networks compete to improve their prediction accuracy. The two neural networks that comprise a GANs are known as the generator (convolutional network)-generates output and discriminator(deconvolutional network)-determine which result were generated intentionally. As the feedback loop between the adversarial networks continues, the generator produces higher-quality output, while the discriminator will grow more adept at detecting falsely generated data.

Some machine learning algorithms applied by researchers for workload forecasting are; Linear Regression Jaykrushna et al. (2018),Sood (2016) Support Vector Regression Huang et al. (2013). Resource mechanism continuously tracks a system and request for a scaling activity if a specific demand of resource is met;provisioning a number of cloud resources if a certain parameter is higher or lower than the set benchmark. This mechanisms are threshold-based mechanisms Lin et al. (2011). The issue with this mechanism is, the

time taken from the event trigger till the resources are provisioned may be insufficient and may cause over loading of resources to the machine. This can cause resource waste and instability in the system. In contrast, when using predictive mechanisms, they try to estimate number of resources needed before the workload arrives based on previous evaluated workloads. Research has been carried out on predictive models based on time-series analysis Kavitha et al. (2016).

This research aims to improve the cloud computing environment by efficiently predicting workload and provisioning resources needed just in time with the help of machine learning techniques. In this research paper, resource allocation and workload prediction is implemented by using Linear regression to generate time-series data and resources are allocated with GANs algorithm. This research paper proposes a novel method for predicting workload and allocating resources required in a cloud environment. Methodology is divided as follows, and uses Threshold based method, Linear Regression and GANs algorithms to predict cloud workload and allocate resources accurately.

This research paper is structured further as follows: section 2 covers Related Works where research on resource allocation and auto-scaling using machine learning performed by different authors are discussed. section 3 covers the proposed methodology and a detailed approach of resource allocation, section 4 covers Design Specification, section 5 shows the implementation, section 6 gives the evaluation and finally section 7, Conclusion.

1.1 Research Question

- Does threshold based algorithm can adapt to the uncertain and dynamic behaviour of Cloud computing environment for allocation of the resources ?
- Can a Machine Learning based predictive algorithm improve workload prediction and provide optimal allocation of resources required to prevent over provisioning or under provisioning in a cloud computing environment?
- How efficiently can GANs performs as compared to the other machine learning algorithms ?

2 Related Work

2.1 Introduction

This section shows review of research papers related to resource allocation and management in cloud setting.

2.2 Literature Review

Murthy et al. (2014) implemented a threshold based approach for dynamic scaling of resources, the scaling was performed for both CPU and RAM utilization. In order to implement their proposed algorithm, the authors set up a cloud environment using Xen Cloud Platform (XCP) and considered two different kind of scaling methods for both CPU and Memory scaling. They implemented a threshold based auto scaling of virtual machines in which VMs will be dynamically scaled based on the application resource utilization (CPU and Memory). In threshold based auto scaling the resource utilization

of the Virtual Machine is monitored. If they exceed the predefined threshold values then VM capacity will be increased or decreased dynamically according to the need without shutting down the VMs, which minimizes resource wastage and VM idleness.

Fallah et.al (Mahallat; 2015) utilizes "automata" for scalability in web applications, they called their algorithm ASTAW. In order to examine the correct threshold based approach they used multiple threshold values with regards to the higher capacity. ASTAW combines virtual machine clusters and the learning automata in order to provide the best possible way for the scaling up and scaling down of the virtual machines. The steps were performed in order to maintain the SLA (Service Level Agreement). CloudSim is used as the simulator for performing the experiment with 20 different hosts and 4 clusters. For performance evaluation, the following metrics were considered; SLA violations and Overhead scaling . Resource allocation in cloud computing centers has become an important research area. Efficient VM allocation can reduce average response time which can benefit both the end users as well as the cloud vendors.

Sharma et al Sharma and Reddy (2016) evaluated a distance based multi-objective energy efficient Virtual Machine placement on servers in a cloud data center. The authors proposed a hybrid algorithm by combing Genetic Algorithm (GA) and Particle Swam Optimization (PSO) to reduce resource idleness thereby minimizing SLA violations during VM allocation. The GA is helps in migrating the VMs from source to target and the PSO algorithm helps the GA to select the optimal target server by allowing VM placement and replacement. The PSO algorithm encodes the VM allocations as a vector where 0 represents sleep mode and 1 represents active server hosting VMs.

Genetic algorithm based workload predictive resource management was implemented in Tseng et al. (2017)and it aimed at improving energy consumption. Their proposed method combines both prediction and VM allocation approaches. The proposed Genetic Algorithm forecasts the resource requirement of next time slot according to the historical data in previous time slots and a VM placement algorithm to allocate Virtual Machines for next time slot based on the prediction results of Genetic Algorithm.

The paper Chen et al Chen et al. (2019), proposes an advantage actor-critic reinforcement learning framework for allocation of VMs in the cloud. To make the predictions two features are important: The actor and critic. The actor is responsible for selecting actions based on probability and then the critic evaluates the action, then the actor changes its probability for selection based on the scores from the critic. The critic guides the actor to make the right prediction. The proposed model was implemented based on TensorFlow library and dataset from Google cluster-usage traces.

A study conducted by Rahman et al. (2018) presented a VNF (Virtual Network Functions) auto-scaling machine learning approach. The ML classification is based on historical VNF auto-scaling choices and has a good rate of accuracy of 96%. The illustrative results demonstrate the effect on the proposed ML classifier of the number of features and the quantity of training data. The influence of start-up time on QoS was also investigated in four distinct virtualization systems. They examined an actual SD-WAN application scenario using optical backbone networks that showed that our suggested technique produces lower rental costs than previous efforts. Future studies should consider exploring the detailed analysis of operational and leasing costs for more such use-cases. In our study, we explored only the horizontal scaling of VNFs. Vertical scaling (i.e., adding/removing CPU/memory resources to the same virtual instance) for VNFs is an important direction yet to be explored. As network services are often deployed as service chains, future studies should explore auto-scaling methods for such scenarios. Future studies should

also explore operational costs and cost models exploring electricity usage, VNF lifecycle management, degradation of service, and other operational activities.

Rafael Moreno-Vozmediano1, Ruben S. Montero1,in Moreno-Vozmediano et al. (2019) evaluate and presents an auto scaling mechanism based on the prediction. Machine learning techniques are used by the authors to predict the cloud workload and this workload prediction is then used to predict the processing load in distributed servers and the correct number of resources (Virtual Machine, Memory) are estimated and provisioned accordingly in order to avoid SLA Violations while optimising response time, minimising over provisioning and under provisioning which helps reduce energy consumption. The authors used SVM regression model for accurately predict server's processing load. Furthermore, the prediction results are used to build queue based performance model to determine actual number of resources that should be provisioned. The experiment and evaluation was carried out using real workload traces and produced the acceptable prediction result closer to the optimal case.

Moghaddam et al Moghaddam et al. (2020) put forward a machine learning approach for VM prediction. The authors applied Linear Regression, Support Vector Regression, Decision Tree and Multi-layer perception for detecting over or under-utilized hosts, migrating all the VMs from their host while utilizing energy. The authors applied cross-validation to enable them to select the best prediction model for each VM. This model, however efficient and gave good prediction, it is an extremely time-consuming process for a real-world setting.

In paper Yu et al. (2018), authors implemented job-pool, where the awareness of the workloads in the pool of tasks is used as basis for the prediction of workload of new tasks.The pool of workloads are put in clusters and Neural net is used to learn the workload characteristics in the job pool and enable prediction. Preliminary workload pattern and submission parameters are used to discover the cluster the job belongs to when it arrives. The neural net for each cluster is used to predict workload of the new job when it arrives.

Prachitmutita et al. (2018) developed a new autoscale framework for predicting workload with models from the Artificial Neural Networks (ANN) and the Recurrent Neural Network (RNN). Then they changed to the necessary RAM and CPU core, based on the predicted workload, so services would continue to operate within the service level agreement (SLA). The performance was assessed using a multi-step forward forecasts that utilizes access records from the 1998 World Cup website. With more anticipated steps forward, the accuracy of the ARIMA model is poorer. In addition, greater accuracy was archived by the LSTM model than the MLP model.

In Dabbagh et al. (2016), the framework implemented uses the Wiener filter approach for predicting resources to be allocated for dynamic workloads in the cloud. This approach can also detect overload in the server. In this paper, different prediction models were employed to predict the resource utilization at VM.However, this predictor system will be installed on every VM instance for it to function well which is not the most optimal approach. Real traces of VM request submitted to a Google cluster is used.

In Zhang et al. (2018)The authors suggested that heuristic and approximate algorithms are not sufficient for resource allocation. They proposed the use of classification to model and analyze dynamic resource allocation, they implemented two resource allocation prediction algorithms based on linear and logistic regression. The model learns a small-scale training dataset and guarantees that the allocation accuracy and resource utilization is very close to optimal allocation solution. The results show that their proposed

scheme showed a good effect on resource allocation in cloud computing.

Savitha et al Savitha and Salvi (2021) evaluates and proposes a priority aware VM allocation policy named P-PAVA algorithm, which considers the priority of an application along with its compute, memory, and bandwidth requirement. This algorithm is responsible for VM allocation prediction based on machine learning model. To limit and further remove workload overhead, the authors employed parallelization before assigning workloads by using first fit technique as a basis for the requests with low priority, parallelization is achieved. Support Vector Mechanism, Neural Networks and Logistic Regression are the algorithms used in the novel approach for prediction of resources for the cloud.

An ensemble learning based VM resource request prediction was presented in Buyya et al Kumar et al. (2020), where the authors of the paper have employed the use of Blackhole learning based algorithm for training a feed-forward neural network. The algorithm which is based on the blackhole theory helps in selecting the optimal weights for the output layer (where the input data is transformed). This approach that was implemented allows the use of more than one prediction model to predict future outcome of an event.

A novel deep learning based approach for VM workload prediction is implemented by Feng et al Qiu et al. (2016). The implemented architecture consists of a deep learning prediction model that is designed with a deep belief network (DBN) that has Boltzmann machines (RBMs) in multiple layers and a regression layer. The DBN functions as a tool for extraction of high level features from all the VMs workload data and prediction of the workload of the Vms in the future is done by the regression layer. The deep learning model has the ability of learning the historical features from the workload data of all VMs ,and these features learned are used in workload prediction . Based on this architecture, the regression layer added performs the fine-tuning of the model and the workload prediction in supervised learning.

The research papers reviewed have shown,rule-based method, machine and deep learning based solutions implemented for a wide variety of resource management issues in the Cloud. According to Demirci (2015) Demand and forecasting of resources is still an issue and this issue requires an optimal technique for forecasting/prediction in a cloud environment.

2.3 Review on Gaps in Literature

To summarise, some research work reviewed used existing approaches or try to streamline the machine learning models to improve the accuracy. The proposed method (GANs) learns deep representations without extensively training the data, the network learns by deriving back propagation signals through processes involving two networks. According to Alqahtani et al. (2019), GANs have made very significant advancements and tremendous accuracy in image processing and editing, classification and image synthesis. Considering fluctuations in cloud workloads, it is essential to have an efficient prediction technique that can be utilised over a long term period. Because of the strides of the GANs algorithm in other predictive applications, it was implemented for workload prediction in this research paper.

Table 1: Summary of Literature Review

Paper Title	Methodology	Advantage	Gaps
Moreno-Vozmediano et al. (2019), Efficient resource provisioning for elastic cloud services based on machine learning techniques	Prediction of the load was done by using the SVM regression model. In the implemented methodology, the results from the SVM model is used to build a queue based model to get the exact number of resources that should be allocated	The metrics for evaluation were; CPU usage, memory requirement and time taken for job executions	This method is capable of only handling one job at a time, jobs are not processed simultaneously
Tseng et al. (2017), Dynamic Resource Prediction and Allocation for Cloud Data Center using the multiobjective genetic algorithm	The Algorithm used in this research predicts the resource requirement of next time slot based on historical data in previous time slots and a VM allocation algorithm is implemented to allocate Virtual Machines for next time slot	Their implemented method combines both prediction and VM allocation approaches	Because Genetic Algorithm is computationally intensive, It requires high energy consumption.
Mahallat (2015) Auto-scaling threshold-based approach for web application in cloud computing environment	The author utilised automata for scalability in the cloud	SLA Violations were considered in the evaluation	Threshold based approaches can result in resource mismanagement.
Savitha and Salvi (2021) Perceptive VM allocation in Cloud Data Centers for effective resource management	Implemented a VM allocation policy named P-PAVA algorithm. This algorithm is responsible for VM allocation prediction based on machine learning model	The methodology takes into consideration the priority of an application based on its compute, memory, and bandwidth requirement	The authors used a Small Data set and this could be insufficient for accurate results.

3 Methodology

Automatic provisioning of the resources in the cloud computing environment is a challenging task. Over-provisioning and under-provisioning of the resources are two main issues in the scaling mechanism. As the cloud works on the pay as you go model, over-provisioning of the resources will increase the resource expenditure from the users end and if the resources are under-provisioned then Quality of Service may be violated. Therefore, in order to match the customer demand and to establish a balance between these two situations an optimal scaling mechanism is required. In this work, we are proposing a GANs (Generative Adversarial Network) based scaling mechanism in cloud computing environment, the proposed architecture for the same is shown in Figure 2.

Our architecture mainly consists of the two ends Client end and server end. On the client side, there can be any number of clients. The main job of the client will be to generate the job/tasks. Client and server are connected to each other via a secure channel. In order to deal with the real world simulation environment, clients can generate any number of tasks in a second. On a specific timestamp, load can vary from high to low. Overall, the load generated by the client needs to be processed by the server. The server side consists of the cloud components which include; cache memory, node cluster, node scaler, resource monitoring system . Each component in the proposed system plays an important role in order to process the tasks (generated by client) in a systematic way. In the further subsections, a detailed explanation about each component is provided.

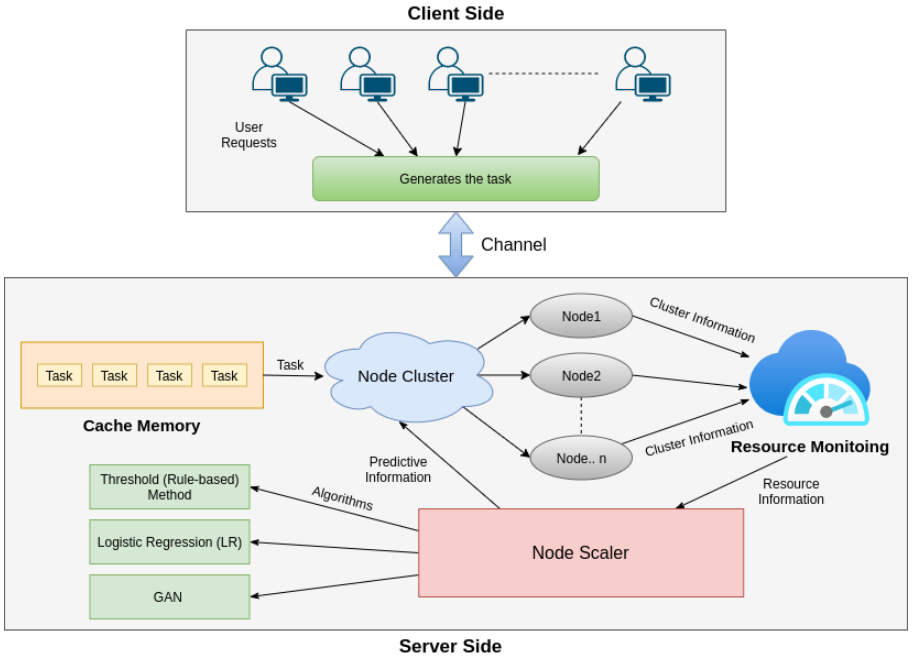

Figure 2: Proposed Architecture for Resource Scalability in Cloud Computing Environment

3.1 Cache Memory

Inside the server system, there is a cache memory which temporally stores the tasks in a queue format. As all the tasks generated by the client can not be processed directly by the server, cache memory acts as a temporary storage for the tasks. Cache memory is connected bi-directionally (client and server). Client performs the push operation in the cache memory as they provide the newly generated task, which are stored by cache memory. On the other hand, server performs the pop operation where the tasks are fetched from the cache memory by the node cluster.

3.2 Node Cluster

Node Cluster acts a main server, where the main functionality of the node cluster is to manage the nodes. Node cluster has the ability to start or stop the nodes at any point of time. The instruction of allocation or Deallocation of the nodes are provided by the node scaler, which is followed by the node cluster. In our proposed work the node cluster has been implemented by an asynchronous python thread. The maximum limit to the add or remove the number of nodes is based on the availability of cloud resources.

3.3 Resource Monitoring

Resource monitoring system keeps track of all activities in the system. It collects, all the cluster related information from the node cluster. Other than that, the resource monitoring system also contains the information about the number of tasks available in the task queue. All the information collected by the resource monitoring system is provided to the Node Scaler, in order to perform the predictive analysis by the algorithms.

3.4 Node Scaler

Node Scaler is the most crucial component, as it is decision making entity in the proposed architecture. The decision taken by the node scaler is mainly based on the algorithms implemented. In this work, we have implemented 3 different types of algorithm, for the node scaler system. The algorithms used in this work are Threshold based algorithm, which follows the rule-based method, Linear regression, which is based on the machine learning method and Generative adversarial networks, which follows the deep neural network architecture. The detailed discussion about these algorithms will be performed in Chapter 4. The information collected from the resource monitoring system are utilized by these algorithms in order to predict the future load on the system and take optimal resource allocation decision accordingly. The performance of the cloud system such as latency, number of task processed, idle time of the nodes are highly dependent on the these algorithms. The utilization of the proposed algorithm automates the process the provisioning the resources.

4 Design Specification

The main objective of this research is to develop an optimal resource allocation method, which can predict the future load on the system and allocate the cloud resources accordingly. As in the chapter 3, the cloud components were extensively discussed, with the Node Scaler being the major component and decision maker in the architecture because it holds the 3 techniques that are implemented in this project.The Techniques used are discussed further in this section.

4.1 Threshold-Based Method

This method can also be regarded as a reactive method. Threshold based methods are implemented by a multiple Cloud Service Providers Moreno-Vozmediano et al. (2012), it allows users to define allocation rules based on a number of metrics. There is a trigger for more allocation of resources(compute and memory), when the specified metrics from the user exceeds or goes below a particular threshold that is previously set. ResearchHasan et al. (2012) has been on to improve this rule-based method. However, because workload in a cloud environment varies and is unpredictable,there is still an issue of wrong triggers which will lead to resource over provisioning.In the simple threshold-based algorithm, no prediction is made based on the historical data. It simply follows the certain set of rules. The Threshold based method has been been implemented in this project. The threshold based algorithm can be represented using the flow diagram in Figure **??**.

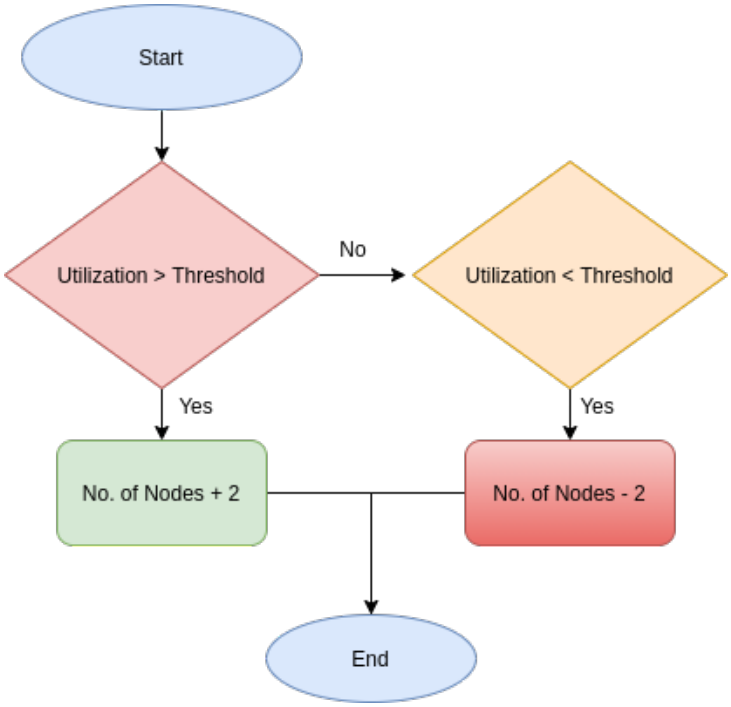

Figure 3: Threshold Based Algorithm Flow Diagram

4.2 Linear Regression

Linear Regression takes a linear approach when performing predictions or forecasting. This method can be argued as the most common and comprehensive machine learning methodsMaulud and Abdulazeez (2020), it finds the relationship between one or various predictors. Linear regression is the most commonly algorithm used for predictive analysis. when implementing a linear regression algorithm,time-series values are increasing or decreasing in a linear fashion. Using linear regression the next 10 hours of data can be predicted by providing the last 10 hours of data. The biggest advantage is that linear regression model requires least amount of training data to perform the prediction.The graph representing the behaviour of linear regression is shown in Figure 4

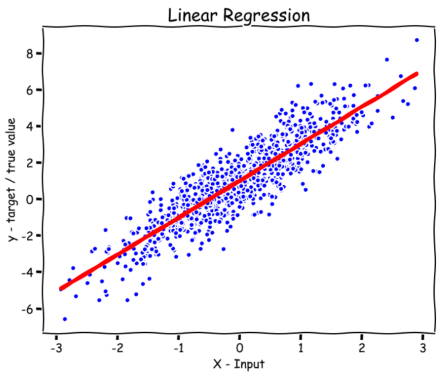

Figure 4: Linear Regression

4.3 Generative Adversarial Networks

Generative Adversarial Networks (GANs) proposed by Goodfellow et al. (2014) estimates generative model through an adversarial network where two models are trained; a generative model- used for data distribution and a discriminative model-calculates the probability that the data sample is part of the training data rather than the generative model. The GANs algorithm have been basically used in image processing, creation and video synthesis tasks where it offers augmentation for dataset constraints. GANs is considered hybrid as it performs supervised and unsupervised learning, in this algorithm, two neural networks (Generator - Convolutional Neural Network and Discriminator - Deconvolutional Network) compete to improve the prediction accuracy of the model. The generator is responsible for generating outputs that could mistaken for actual data, now the discriminator aims to determine which of the results it receives were generated intentionally.

In this project, the pix2pix GAN model was utilised. This architecture of GAN takes the image as the input and generates the another similar image as output. This capability of GANs was utilised and modified to consider 1-D Sequence as input and generate the 1-D sequence as output. After successfully training the 1D-dataset which consists of timestamp data with its associated load, the model was saved for performing predictive analysis.

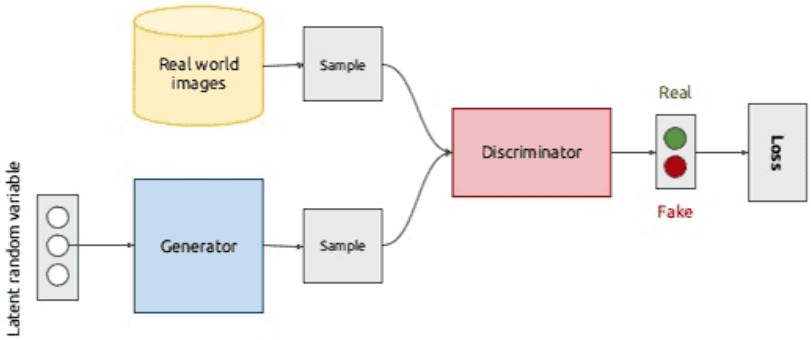

Figure 5: Generator and Discriminator in GAN

5 Implementation

For this research project, a cloud environment is simulated to demonstrate the project objectives using Python programming language(has extensive libraries that are used for machine learning capabilities). In order to implement this project, the system should have python installed and some other libraries (listed below).
The implementation of the proposed framework explained in the chapter 3 was done in this project. The first step was to collect the resource data in order to perform the predictive analysis. In this project, a synthetic time-series data set is generated which contains the timestamp and associated load for each timestamp. In order to simulate the real-world cloud data, sin wave function with different periodicities is implemented. The concept of threads has also been utilized for the start and end operation for instances. Tensorflow 2 library is used in order to implement the GANs architecture. Libraries such

as pandas, numpy, scipy have been utilized application of data manipulation operations and to perform complex mathematical calculations. Matplotlib library is utilized for representing the data with different graphs(data for plotting the graphs has been taken from resource monitoring system) .

Implementation of Node Scaler with different algorithms is another major task in this research. The three techniques used was implemented in the node scaler.The operating system preference is **Ubuntu 20.04**, because major components of this implementation;thread scheduling, memory management and file handling performs efficiently because of Ubuntu's monolithic kernel type. The minimum requirement in order to run the proposed Resource Scalability system is as follows.

- Operating System: Ubuntu 20.04 (LTS)
- Minimum RAM : 8GB
- Hard Drive : 60GB
- No. of CPU Core: 4
- Programming Language: Python3
- Libraries: Tensorflow, pandas, matplotlib, sklearn and numpy.

6 Evaluation

The main purpose of this research is to identify the best resource allocation and scaling algorithm, which can allocate and remove the resources efficiently in a cloud environment. As the different algorithms have been implemented in this research for allocating the resources, it becomes a very important step to analyze the performance of each algorithm/technique. In this research, the proposed cloud system will be executed multiple times using different algorithms. The metrics which will be used to identify the optimal resource allocation mechanisms are Latency, Number of tasks processed by the system and the number of idle cycles. The algorithm having the minimum latency and number of idle cycles and high number of task processes by the system will be considered as an optimal algorithm. In order to calculate the scaling response every algorithm is executed for 100 seconds. There are 3 algorithms/technique have been implemented in the proposed architecture, therefore 3 different experiments will be performed. For each algorithm/technique 3 metrics will be calculated which will be represented by graphs for better visualization.

6.1 Experiment 1/ Threshold Based Algorithm

Discussion about the design and architecture part of the threshold based algorithm has been provided in the Chapter 4. Threshold-based algorithm follows the certain set of rules where the new nodes are added, if the resource utilization is greater than threshold values and vice versa. In the first evaluation, we will calculate the number of task processed and idle cycle using threshold-based algorithm. The bar graph for the same is represented in Figure 6.

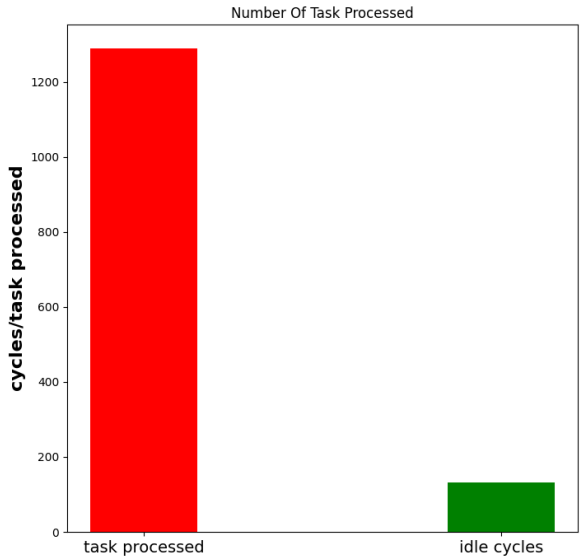

Figure 6: Number of Task processed Vs Idle Cycle

After analyzing the graph shown in Figure 6, it has been observed that number of task processed using threshold based approach is near around 1,100. On the other side, number of idle cycles are found to be approximate 100. Although, the low number of idle cycles represents that threshold-based algorithm utilizes the system efficiently. The next graph shows the Latency Histogram associated with number of tasks. For better performance the latency of the system should be minimum. Graph of latency histogram along with number of tasks is shown in Figure 7.

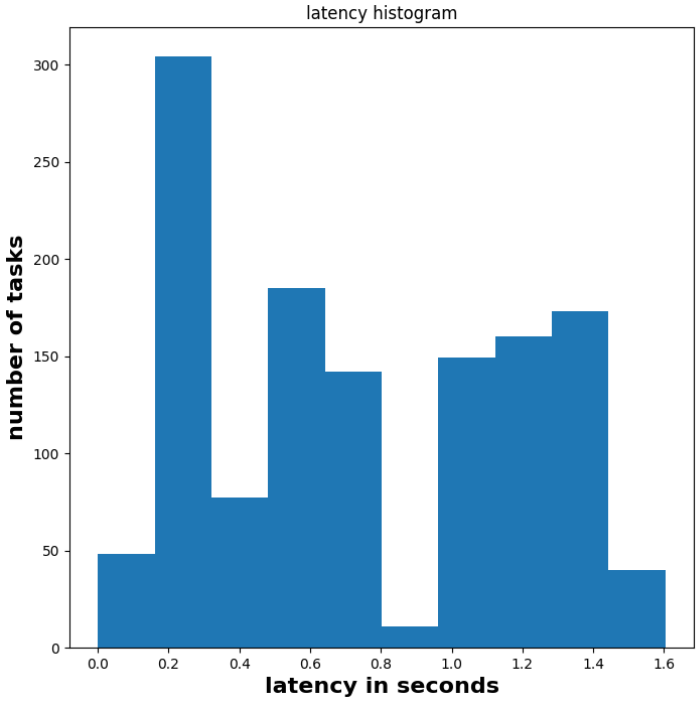

Figure 7: Latency Histogram with Number of Tasks

The highest latency observed in the graph is 1.6 seconds, where the highest number of tasks is around 300 and minimum number of tasks observed are 10. In the cloud system, latency always occurs often . However, the system with a reduced latency can execute the job more efficiently.

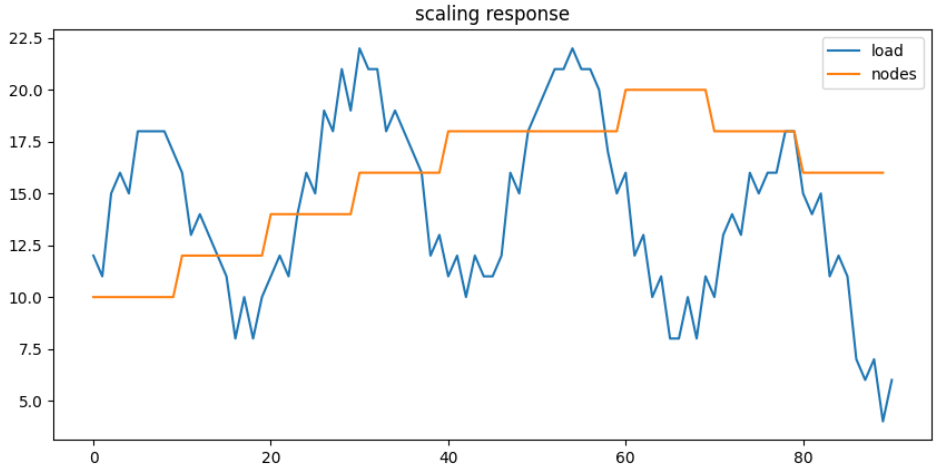

Figure 8: Scaling Response of Threshold-based algorithm

The scaling response of threshold based algorithm is shown in Figure 8. In this graph the X-axis represents the timestamp. Blue line represents the load on the system and the orange line represents the number of nodes allocated in order to process the load. It is clearly visible that, threshold based approach is not optimal when scaling up or scaling down the number of nodes.

6.2 Experiment 2/ Linear Regression Algorithm

In this experiment the evaluation of metrics will be performed using the linear regression algorithm. The linear regression represents the simple linear behaviour and is considered as better model for time-series prediction. Linear regression algorithm analyses the historical load of past 10 seconds of data and can predict the load for next 10 seconds. On analysing the task processed and number of idle cycles it has been observed that number of tasks processes by the threshold based approach and linear regression are very close to each other. However, the number of idle cycles are found to be very low using linear regression algorithm, as compared to threshold based approach. The number of idle cycles and task processed by linear regression algorithm is shown in Figure 9.

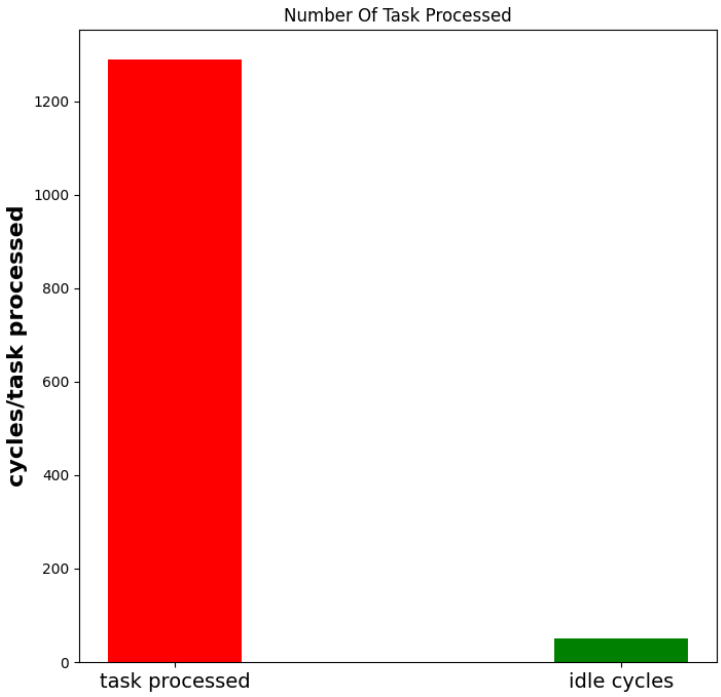

Figure 9: Number of Task processed Vs Idle Cycle

The latency associated with number of tasks obtained using linear regression algorithm is shown in Figure 10. On analysing the graph it has been observed that maximum latency obtained using linear regression is 1.2 seconds. However, the highest number of tasks obtained are approximately 500. As compare to the threshold based approach the latency and number of tasks are found to be very minimal.

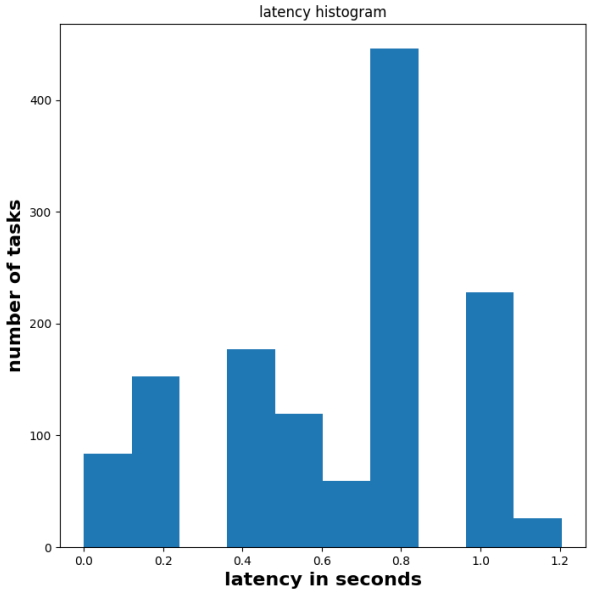

Figure 10: Latency Histogram with Number of Tasks

The graph shown in Figure 11 represents the scaling response obtained using linear

regression algorithm. The scaling response of linear regression algorithm is found to be better when compared to the threshold based algorithm. However, scaling response obtained using linear regression is minimum as there are still variations in load and the number of nodes.

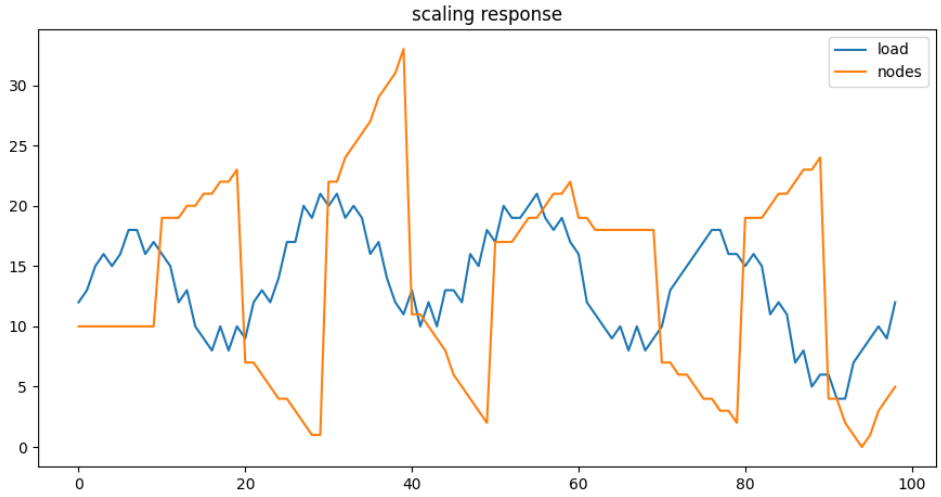

Figure 11: Scaling Response of Linear Regression algorithm

6.3 Experiment 3/ Generative adversarial networks (GANs)

Generative adversarial networks (GANs) is our proposed approach for this research. The main advantage of using the GANs, it can learn from small subset of dataset, GANs is found to be a very effective approach for generating similar fake images from the input data and mainly used by the researchers for image processing purposes. However, in this work we are utilizing the GANs algorithm as Resource scaling algorithm. On performing comparative analysis between the threshold based approach and linear regression algorithm GANs is able to process much higher number of tasks. However, the idle cycle using GANs are found to be very high.

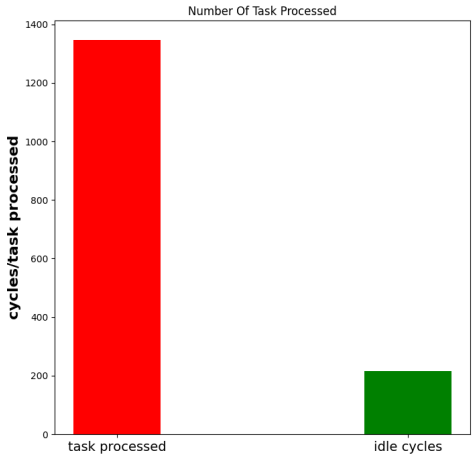

Figure 12: Number of Task processed Vs Idle Cycle

In the other graph, latency along with the number of tasks has been captured using GANs algorithm, the graph for the same is shown in Figure 13. The latency using the GANs algorithm is also very minimum and the number of tasks associated with the latency are quite high in numbers. The highest latency observed using the GANs algorithm is 0.8 seconds and the maximum number of tasks are approximately 750 in numbers. The latency obtained while using GANs is almost half of the latency obtained using threshold based approach. Latency highly impacts on the performance of the model.

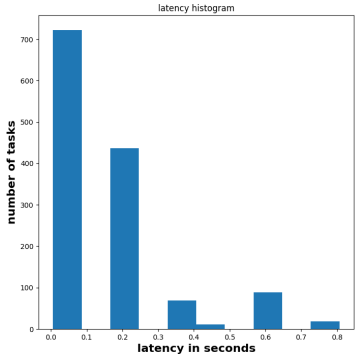

Figure 13: Latency Histogram with Number of Tasks

The final metrics that needs to be analyzed using GANs is the scaling response. The following graph shown in Figure 14, represents the scaling response of GANs algorithm. Using GANs, the number of nodes are able to adapt to the dynamic changes . As already discussed low latency produces a better scaling response. Using GANs, the system is able to scale-up and scale-down the resources in an efficient way. Scaling response using GANs is optimal, when compared to the threshold based algorithm and linear regression algorithm.

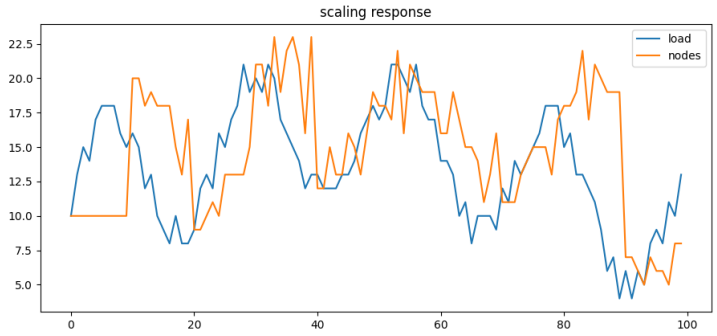

Figure 14: Scaling Response of GANs algorithm

6.4 Discussion

Using machine learning or deep learning algorithm for resource scaling mechanism can be a complex task. In this research, we have performed experiments using multiple algorithms which includes the threshold based algorithm, linear regression algorithm and Generative adversarial networks (GANs) algorithm. After calculating the multiple metrics for each algorithm, it is discovered that the latency plays an important role in

order to improve the scaling response of any algorithm. The Algorithm with minimum latency has the ability to process more number of tasks and can easily adapt to the ever unstable and changing behaviour of user requests in cloud computing environment. Based on the findings, the threshold based and linear regression algorithm can process nearly same number of tasks. However, a major difference has been observed in terms of latency and scaling response. Latency gotten using threshold based algorithm as compared to GANs algorithm is almost double. Therefore, threshold based algorithm may not be an efficient technique for allocation and deallocation of the resources. In terms of idle cycles, the minimum number of idle cycle has been obtained using linear regression algorithm. However, it is also seen that number of tasks processed by the GANs algorithm is much higher. In such cases, it is possible that some nodes may remain in idle state. There is no doubt that, the scaling mechanism in the cloud computing environment can be improved to an great extent by using Generative adversarial networks (GANs).

7 Conclusion and Future Work

After experiments, findings and analysis it can be said that the Generative adversarial networks (GANs) can be used as an optimal scaling mechanism. It not only reduces the latency occurred in the cloud system, but, it also improves the scaling response in the cloud computing environment. Accurate predictive analysis behaviour of the GANs algorithm makes it more useful for various applications. On the other hand, it can also be said that threshold based approach generates the poor results in terms of scaling response and latency, the threshold based approach fails to adapt the dynamically changing environment of cloud computing. Linear regression algorithm outperforms the threshold based algorithm in terms of idle cycles, number of tasks processes and latency. However, due to its linear behaviour this algorithm can not utilized in the heterogeneous cloud computing environment. The main use-case of the GANs algorithm is to generate the similar image from the input image,however in this project it was utilised for scaling mechanism. The data was converted into 1D array. Although, the GANs algorithm can be computationally intensive and consume high bandwidth. In the present architecture that is proposed, there is no component that assigns the tasks to correct node/virtual instance and because of that the number of idle cycle when using GANs are still high.

For future work, more components in the proposed system can be added such task scheduler, load balancer etc. Adding these components can help to improve the response time and execution time of GANs. Also, using real workload traces can be used in the architecture for further work.

References

Alqahtani, H., Kavakli-Thorne, M. and Kumar, G. (2019). Applications of generative adversarial networks (gans): An updated review, *Archives of Computational Methods in Engineering* pp. 1–28.

Bermejo, B., Filiposka, S., Juiz, C., Gómez, B. and Guerrero, C. (2017). Improving the energy efficiency in cloud computing data centres through resource allocation techniques, *Research Advances in Cloud Computing*, Springer, pp. 211–236.

Chen, Z., Hu, J. and Min, G. (2019). Learning-based resource allocation in cloud data center using advantage actor-critic, *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, IEEE, pp. 1–6.

Dabbagh, M., Hamdaoui, B., Guizani, M. and Rayes, A. (2016). An energy-efficient vm prediction and migration framework for overcommitted clouds, *IEEE Transactions on Cloud Computing* **6**(4): 955–966.

Demirci, M. (2015). A survey of machine learning applications for energy-efficient resource management in cloud computing environments, *2015 IEEE 14th international conference on machine learning and applications (ICMLA)*, IEEE, pp. 1185–1190.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014). Generative adversarial nets, *Advances in neural information processing systems* **27**.

Hasan, M. Z., Magana, E., Clemm, A., Tucker, L. and Gudreddi, S. L. D. (2012). Integrated and autonomic cloud resource scaling, *2012 IEEE network operations and management symposium*, IEEE, pp. 1327–1334.

Huang, C.-J., Guan, C.-T., Chen, H.-M., Wang, Y.-W., Chang, S.-C., Li, C.-Y. and Weng, C.-H. (2013). An adaptive resource management scheme in cloud computing, *Engineering Applications of Artificial Intelligence* **26**(1): 382–389.

Jaykrushna, A., Patel, P., Trivedi, H. and Bhatia, J. (2018). Linear regression assisted prediction based load balancer for cloud computing, *2018 IEEE Punecon*, IEEE, pp. 1–3.

Kavitha, S., Varuna, S. and Ramya, R. (2016). A comparative analysis on linear regression and support vector regression, *2016 online international conference on green engineering and technologies (IC-GET)*, IEEE, pp. 1–5.

Kumar, J., Singh, A. K. and Buyya, R. (2020). Ensemble learning based predictive framework for virtual machine resource request prediction, *Neurocomputing* **397**: 20–30.

Lin, W., Wang, J. Z., Liang, C. and Qi, D. (2011). A threshold-based dynamic resource allocation scheme for cloud computing, *Procedia Engineering* **23**: 695–703.

Madni, S. H. H., Abd Latiff, M. S., Coulibaly, Y. et al. (2017). Recent advancements in resource allocation techniques for cloud computing environment: a systematic review, *Cluster Computing* **20**(3): 2489–2533.

Mahallat, I. (2015). Astaw: Auto-scaling threshold-based approach for web application in cloud computing environment, *Int J u-Serv Sci Technol* **8**(3): 221–230.

Maulud, D. and Abdulazeez, A. M. (2020). A review on linear regression comprehensive in machine learning, *Journal of Applied Science and Technology Trends* **1**(4): 140–147.

Moghaddam, S. M., O'Sullivan, M., Walker, C., Piraghaj, S. F. and Unsworth, C. P. (2020). Embedding individualized machine learning prediction models for energy efficient vm consolidation within cloud data centers, *Future Generation Computer Systems* **106**: 221–233.

Moreno-Vozmediano, R., Montero, R. S., Huedo, E. and Llorente, I. M. (2019). Efficient resource provisioning for elastic cloud services based on machine learning techniques, *Journal of Cloud Computing* **8**(1): 1–18.

Moreno-Vozmediano, R., Montero, R. S. and Llorente, I. M. (2012). Iaas cloud architecture: From virtualized datacenters to federated cloud infrastructures, *Computer* **45**(12): 65–72.

Murthy, M. M., Sanjay, H. and Anand, J. (2014). Threshold based auto scaling of virtual machines in cloud environment, *IFIP International Conference on Network and Parallel Computing*, Springer, pp. 247–256.

Mustafa, S., Nazir, B., Hayat, A., ur Rehman Khan, A. and Madani, S. A. (2015). Resource management in cloud computing: Taxonomy, prospects, and challenges, *Computers Electrical Engineering* **47**: 186–203.
URL: *https://www.sciencedirect.com/science/article/pii/S004579061500275X*

Prachitmutita, A., Pojjanasuksakul, S. and Padungweang (2018). Auto-scaling microserviceson iaas under sla with cost-effective framework, *In Proceedings of the 2018 Tenth International Conference on Advanced Computational Intelligence (ICACI)* pp. 583–588.

Qiu, F., Zhang, B. and Guo, J. (2016). A deep learning approach for vm workload prediction in the cloud, *2016 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, IEEE, pp. 319–324.

Rahman, S., Ahmed, T., Huynh, M., Tornatore, M. and Mukherjee, B. (2018). Auto-scaling vnfs using machine learning to improve qos and reduce cost.

Savitha, S. and Salvi, S. (2021). Perceptive vm allocation in cloud data centers for effective resource management, *2021 6th International Conference for Convergence in Technology (I2CT)*, IEEE, pp. 1–5.

Sharma, N. K. and Reddy, G. R. M. (2016). Multi-objective energy efficient virtual machines allocation at the cloud data center, *IEEE Transactions on Services Computing* **12**(1): 158–171.

Sood, S. K. (2016). Function points-based resource prediction in cloud computing, *Concurrency and Computation: Practice and Experience* **28**(10): 2781–2794.

Tseng, F.-H., Wang, X., Chou, L.-D., Chao, H.-C. and Leung, V. C. (2017). Dynamic resource prediction and allocation for cloud data center using the multiobjective genetic algorithm, *IEEE Systems Journal* **12**(2): 1688–1699.

Yu, Y., Jindal, V., Bastani, F., Li, F. and Yen, I.-L. (2018). Improving the smartness of cloud management via machine learning based workload prediction, *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, Vol. 2, IEEE, pp. 38–44.

Zhang, J., Xie, N., Zhang, X., Yue, K., Li, W. and Kumar, D. (2018). Machine learning based resource allocation of cloud computing in auction, *Comput. Mater. Continua* **56**(1): 123–135.