National College of Ireland

Resource Optimization in Cloud Data Centers using Machine Learning

MSc Research Project
MSc in Cloud Computing

Jnanashree Arkalgud Guruvachari
Student ID: 23174528

School of Computing
National College of Ireland

Supervisor: Yasantha Samarawickrama

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Jnanashree Arkalgud Guruvachari		
Student ID:	23174528		
Programme:	MSc in Cloud Computing	**Year:**	2024
Module:	MSc Research Project		
Supervisor:	Yasantha Samarawickrama		
Submission Due Date:	12-12-2024		
Project Title:	Resource Optimization in Cloud Data Centers Using Machine Learning		
Word Count:	7415	**Page Count:**	22

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: Jnanashree A G

Date: 11-12-2024

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 Optimization in Cloud Data Centers Using Machine Learning

Jnanashree Arkalgud Guruvachari
23174528

Abstract

In this digital era, everything is considered as data. Hence, it is very important to protect and manage these data in an optimized manner. We have data centers across the globe that are used to store, manage, and protect the data. Many operations are performed within the data center such as workload distribution, power consumption, auto-scaling, resource allocation, etc. It is a crucial job to optimize these operations to avoid higher operational costs and degraded service quality. In this study, we have implemented three machine learning models that can be used to optimize the data center resources to achieve better performance. The three models are- Linear Regression, Random Forest, and Generative Adversarial Networks (GANs). Out of three models, we have achieved excellent results for Linear Regression and Random Forest compared to GANs. These three models are trained on the historical information and predict future data based on the data flow and knowledge of the historical data. We obtain an output from these models which displays the action that needs to be taken. The output indicates if the resources need to be scaled up or down based on the workflow and if the cooling system needs to be activated based on CPU usage. By implementing this solution, we can achieve optimized resource allocation, better workload distribution, reduced power consumption, and lower operational cost which in turn achieves better performance of the data center.
Keywords: Data Center, Linear Regression, Random Forest, GANs, Optimization.

1 Introduction

Data centers are infrastructures that host multiple IT operations and provide data storage, processing, and data management. Some of the components of the data center are physical infrastructure which includes servers, storage systems, and networking equipment, Cooling systems, Security features such as firewalls and encryption, and Power supply which includes generators for backup. These Data centers are capable of storing and handling vast amounts of data and facilitating real-time access to the processed data. Proper maintenance of these data centers is crucial for better performance and reliability. There are some incidents that have disrupted the data center operations such as Power outages., In 2020[1] a major service provider experienced a power outage which disrupted the services for several hours, affecting several businesses. Natural disasters., for instance in 2012 Hurricane Sandy[2] resulted in downtime for several data centers in the northeastern United States. Cyber-attacks[3] are the most crucial which leads to data breaches, service interruptions, and compromise on data center operations. To avoid all the above-mentioned issues, we should ensure that the data centers are regularly checked and perform scheduled maintenance of hardware and software is necessary. We can set up alerts in the monitoring systems to help detect any deviations in

the normal behavior of the data center operations. We should have disaster recovery strategies to ensure data integrity and availability.

Machine Learning is a field under the umbrella of artificial intelligence that enables the system to learn from data, detect patterns, and make decisions with minimal manual involvement. There are three types of Machine Learning algorithms:

Supervised Learning: This involves making the model learn based on categorized data allowing it to forecast the results for new data. Frequent algorithms include Linear Regression, Decision Trees, and Support Vector Machines.

Unsupervised Learning: Involves unlabelled data to identify patterns and groupings. Algorithms include Clustering algorithms like k-means and hierarchical clustering.

Reinforcement Learning: Involves learning optimal actions through trial and error which is often used in dynamic environments. Algorithms include deep learning and neural network algorithms.

Some of the applications of Machine Learning algorithms in Data center operations are:

Energy Efficiency: ML algorithms can be used to predict CPU usage and optimize resource allocation, reducing energy consumption by 88.5% compared to traditional methods [4].

Predictive Analysis: ML algorithms help in minimizing over-provisioning or underutilization of resources leading to better energy management [5].

Dynamic Cooling Systems: ML algorithms enhance the cooling systems based on environmental data, significantly lowering energy costs [6].

Load Balancing: ML algorithms help improve workload distribution across servers, ensuring optimization of performance and operational costs [7].

In this project, machine learning algorithms are utilized to optimize the data center resources such as workload distribution and activating cooling systems. Different algorithms are used to anticipate the workload based on the historical information to decide if the future data is above the threshold set then resources should be upscaled and if it is below the threshold set, resources should be downscaled. The threshold is set based on various parameters in the dataset. Cooling agent will be activated if the CPU utilization goes above a threshold set. In contrast, ML offers significant potential for optimizing data center operations, it is important to consider the restrictions and risks associated with the implementation. Cooling systems were implemented just for the Linear regression algorithm due to the complex nature of other algorithms. Further section explains how we have implemented the algorithms along with the results. We will also discuss some of the limitations and future work.

1.1 Research Question

"Can implementing a predictive model using the dataset help in training and testing the machine learning model to optimize the efficiency of the data center which results in the betterment of power consumption and workload distribution for Public Data Centers and Enterprise Data Centers".

Machine Learning Predictive Model is employed to enhance the proficiency of the data center resources like workload distribution and CPU usage. This Model will help in

providing a comprehensive decision based on the traffic flow if the resources need to be allocated or deallocated in the data center. This will result in resources not being underused or overused which helps the users access the resources without any delay. Additionally, an efficient data center can improve productivity in terms of processing speed, system reliability, and system delivery. It can help reduce downtime, enhance system performance, and ensure the capability of meeting the increasing demands of the service.

2 Related Work

This section consists of the previous work done in the field of Machine Learning Algorithms in data centers along with discussing some of the pros and cons of the existing work. This section can help us understand the different aspects such as how different kinds of algorithms work under different data and provide us a different insight into the common problems.

2.1 Dynamic Load Balancing Using Machine Learning in Cloud Data Centers

The papers [8][9] address the critical issue of efficiently broadcasting the workload across multiple servers in cloud computing environments. Uneven distribution of workloads in cloud computing can lead to server overloads, resulting in inefficient performance and higher operational costs.

The paper [8] aims to develop a dynamic resource management system that effectively balances the load across multiple servers, ensuring optimal resource management and performance. The author utilizes two algorithms Artificial Neural Network (ANNs) and Linear Regression. These algorithms are chosen for their capability to predict future data based on historical information. Both techniques are implemented using a Python programming language in Jupyter Notebook. The implementation involves collecting data from online sources which contains parameters like server ID, timestamp, and current loads. The data is pre-processed to avoid any kind of noise or null values in the dataset and ensure that the models are trained on clean data. The model is then used to predict server loads and redistribute the load from overloaded servers to those with lower loads. The results indicate that the Linear Regression algorithm yields a higher accuracy as the R square value is closest to 1 compared to ANN. This suggests that Linear Regression is more effective in predicting server loads in this project.

The paper [9] addresses challenges in cloud resource management specifically focusing on load balancing, resource utilization, and power consumption. The approaches used to resolve the problem are an Online VM Prediction system which incorporates an online prediction system that learns and predicts future resource demand at each VM, Multi-Objective Load balancing which employs multi-objective algorithms such as Heuristic Algorithms for Comparison, Neural Network-based predictions, Genetic algorithm components, etc for VM placement and migration focusing on maximizing resource utilization with lower power consumption and communication costs. Evaluation Against Benchmark which indicates that proposed framework is evaluated using three real-world datasets (Google Cluster dataset, Planet Lab, and Bitbrains VM traces)

While the approach taken in the papers is innovative and leverages Machine Learning algorithms effectively, there are some considerations to note:

Data Dependency: The performance of both ANN, Linear Regression, VM prediction system, and Multi-Objective load balancing depends on the quality and quantity of the data. Any insufficient or biased data can lead to poor prediction.

Complexity of ANN: Although ANN is used to model complex relationships, it requires more computation power compared to Linear Regression which could be a drawback when working with real-time applications.

Generalization: The developed models may have to be tested across multiple platforms to ensure their applicability in various scenarios.

Overall, the papers present a solid foundation for using machine learning algorithms to optimize the resources in the cloud data center but further research could enhance the robustness and applicability of the proposed models.

2.2 Enhanced Autoscaling Mechanism of Cloud Computing Resources

The papers [10][11][12] focus on the challenges of auto-scaling in cloud computing environments, particularly focusing on addressing the need for proficient resource management for fluctuating workloads.

The paper [10] aims to develop a model that can efficiently allocate resources to cope with fluctuating workloads. Traditional models often lead to over-employing or under-employing of resources with fluctuating demands resulting in increased costs and degraded service quality to the users. The approach used in this research is a hybrid approach combining two algorithms Linear Regression and Artificial Neural Networks (ANNs). Implementation involves creating a virtual system that incorporates the proposed model. By using hybrid models, the system can adapt to fluctuating workloads more effectively than traditional methods. This adaptability is important to maintain service level agreements SLAs and quality of service. The results of the models show a significant balance between reduced complexity and acceptable accuracy. While the study shows promising results, it is important to consider the trade-offs involved like any significant change in the input could impact the performance in highly dynamic environments.

Overall, the paper presents a good approach to addressing the difficulties of auto-scaling in cloud computing, utilizing a hybrid model combining linear regression and ANN. The implementation shows that this method can improve resource allocation efficiently.

The paper [11] discusses the crucial aspect of maintaining the balance between resource utilization and performance. When resources are over-provisioned, it leads to higher costs and when they are under-provisioned, it leads to poor quality of service. The paper aims to provide a solution that effectively manages these challenges through the auto-scaling mechanism. The paper involves using deep learning techniques specifically Long short-term memory, to predict the workload patterns which allows the capability to proactively allocate resources. The paper also discusses the integration of fuzzy logic and reinforcement learning to enhance the auto-scaling process. The implementation involves setting up an auto-scaling group on AWS, where the LSTM is trained on historical data. An SSH connection is established to upload the required files and dependencies on the Elastic Block Store (EBS) volume. The auto-scaling is enabled based on CPU load predictions, ensuring resources are allocated dynamically as needed. The proposed methods are relevant to address the dual challenges of cost efficiency and performance optimization. The results indicate that the proposed method significantly improves resource utilization and application performance

during workload fluctuations. However, there are some considerations like implementing a deep learning technique can cost higher computational costs, and resources which could be a barrier for small companies.
In conclusion, the paper presents a extensive approach to address the challenges of resource utilization using advanced algorithms in the cloud computing environment. The proposed methods show significant potential, although practical challenges remain in the implementation and scalability.
In the paper [12] focuses on addressing the challenges of auto-scaling in cloud computing environments. As the demands for computational resources increase with technological advancements, the need for efficient resource management becomes critical. The existing traditional method is unable to cope with the rising load on the data centers, leading to increased latency and inefficient resource utilization. The paper aims to develop an effective auto-scaling mechanism that can dynamically adjust the resources based on real-time demand. The author proposed a hybrid model combining machine learning techniques with deep learning techniques to optimize the auto-scaling of cloud resources. This model takes the historical data to train on and predict future resource demands. The proposed model is implemented through simulations that assess the performance against traditional methods. By using DRL (Deep Reinforcement Learning) with time-series forecasting, the framework can dynamically allocate resources, thereby reducing latency and improving overall system efficiency. The results of the study show that the proposed algorithm exceeds the traditional method in terms of average delay time, task distribution, and congestion management. The implementation of the DRL method shows the improvement in resource utilization achieving reduced idle cycles and enhanced productivity.
While the proposed model shows improvements, there are some considerations to note:
Complexity: The use of Bi-LSTM may introduce delays in prediction.
Data dependency: The effectiveness of the machine learning models greatly depends on the quality of the data.
Scalability: As the cloud environments grow, the scalability of the solution must be evaluated. The approach should be tested in larger, more diverse environments to ensure its robustness.
Overall, the paper presents a comprehensive approach to improving auto-scaling mechanisms in cloud environments through innovative use of machine learning and deep learning techniques. While the results are promising, further research is needed to ensure the scalability of the proposed solution.

2.3 A Proactive Mechanism to Improve Workload Prediction and Resource Management for Cloud Services Using Machine Learning

The papers [13][14][15] focus on improving workload prediction and resource management in cloud computing environments, addressing the challenges posed by fluctuating workloads that can impact system performance and resource allocation. Cloud environments often experience higher workloads making it difficult to predict future resource needs accurately. This unpredictability leads to increased operational costs, inefficient resource allocation, and degraded performance.
In this paper [13] discusses the use of a hybrid approach that combines multiple algorithms to enhance the accuracy of workload distribution. The methods used are Support Vector Regression (SVR), Artificial Neural Network (ANN), Wavelet Transformation, and Long Short-Term Memory (LSTM). The implementation involves pre-processing the data using wavelet transformation, followed by the application of SVR and ANN to the transformed data. This approach allows the system to adapt to lower and higher workloads improving

overall prediction accuracy. The paper aims to create a robust prediction model by combining the two models that can handle the complexities of fluctuating workloads. This approach is designed to improve resource allocation decisions leading to better performance and cost efficiency in cloud environments. The results show that the proposed methods outperform traditional methods in predicting workloads. The hybrid method leverages the strength of multiple algorithms eliminating the weakness of individual algorithms. The use of wavelet transformation allows for better handling of the fluctuating data which enhances the model's adjustability. The complexity of the hybrid model may lead to increased computational requirements leading to resource-constrained environments. While the results are promising, testing the model on a diverse dataset may help demonstrate the model's generalizability. Overall, the paper discusses the comprehensive approach to improving workload prediction in cloud environments. The hybrid approach promises to enhance prediction accuracy, although further research is required to optimize the implementation across various scenarios.
The paper [14] aims to develop a more effective approach to predict resources and allocate VMs accordingly. This paper develops a hybrid model that combines machine learning techniques with traditional resource allocation methods. The methods employed are
Threshold-Based Algorithm, Linear Regression, and Generative Adversarial Networks (GANs). The implementation involves collecting historical data and applying these algorithms to predict future resource needs. The system continuously monitors the usage and allocates resources based on the predictions made by the algorithms. By utilizing machine learning techniques, the system can adapt to changing workloads and ensure resources are allocated optimally which reduces the operation cost and improves performance. This hybrid model improves resource allocation efficiently. The paper shows reduced latency, an increased number of tasks processed, and minimized idle cycles which shows the effectiveness of the proposed model. While the proposed model shows promise, there are some ethical considerations:
Complexity: The involvement of multiple algorithms can increase system complexity, making it harder to maintain and troubleshoot.
Data Dependency: The performance of machine learning models heavily relies on the quality and quantity of the data. Hence, in scenarios with limited data, the predictions may not be accurate.
Scalability: As the workload grows, the model should be able to scale accordingly. The paper does not mention how the model operates under significantly larger datasets or complex workloads.
Overall, the paper presents a well-structured approach to resource allocation using the combination of both traditional methods and machine learning techniques. While the results are promising, further research is required to address the complexity and scalability of the proposed solutions.
The paper [15] focuses on addressing the challenges of resource allocation in green computing, particularly in optimizing energy efficiency while maintaining performance. The approach taken in this paper involves developing a resource allocation algorithm that combines multiple optimization techniques. The proposed technique aims to enhance energy efficiency while ensuring that performance is not compromised. The methods used in this research are the Grasshopper Optimization Algorithm, Genetic Algorithm, and CloudSim Simulation. The implementation involves creating a simulation environment using the CloudSim tool, where the algorithms are tested against various workloads. The proposed method aims to achieve a balance between performance and energy efficiency by dynamically allocating resources based on current needs. The proposed methods show improvement in energy efficiency and reduce carbon footprints compared to traditional methods. While the proposed methods show good results in a simulated environment, their

scalability in real-world applications remains to be tested. Although, the approach aims for dynamic resource allocation, the effectiveness of the algorithm in a rapidly changing environment needs further exploration. Real-time adaptability is important for maintaining performance under changing workloads.
Overall, the paper presents a promising approach to resource allocation in green computing, leveraging advanced optimized techniques to address energy efficiency challenges. However, further research is needed to validate the algorithms in real-world scenarios.

2.4 Comparison Table

The table 1 provides the comparison of techniques from related work against this study.

References	Load Balancing	Auto-Scaling	Resource Allocation	Workload Prediction	ML Algorithms	Datasets
[8]	✓		✓		✓	
[9]	✓		✓		✓	
[10]		✓			✓	✓
[11]		✓	✓		✓	✓
[12]		✓			✓	✓
[13]			✓	✓	✓	✓
[14]				✓	✓	✓
[15]			✓		✓	
This Study	✓	✓	✓	✓	✓	✓

Table 1: Table of multiple related work comparison

3 Research Methodology

In this section, the equipment used in the research along with how we gathered the dataset and how we cleaned and transformed the data into a pre-processed dataset are discussed. The details of how the measurements were done along with how the performance was calculated of the pre-processed dataset are explained.

3.1 Block Diagram

This section shows the flow of the model, starting from how the model was created, data collection, and data pre-processing to algorithms used as shown in Figure 1.

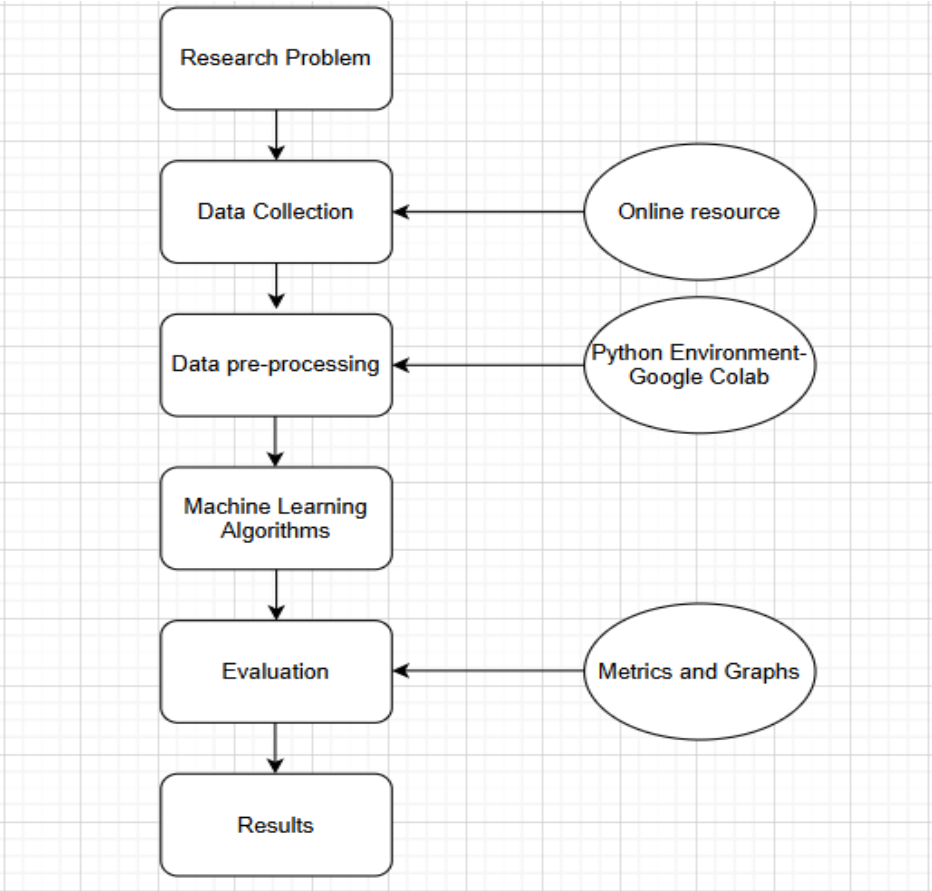


Figure 1: Schematic represntation of the Model

The above diagram shows the components of the research involved. Initially, the research problem is formulated which indicates the problem trying to solve. Next is the data collection, the dataset from Kaggle is used which is a public dataset known as Cloud datacenter Workload. The dataset collected online must be cleaned to ensure there are no null or missing values and we need to make sure in the pre-processed data all the values are of the same datatype. A publicly available platform is used known as Google Collab to pre-process and apply machine learning algorithms for our research problem. For the clean data, machine learning algorithms are applied to train and test the model on the dataset to obtain the performance metrics. Then evaluation of the model will be done based on performance metrics such as Mean Squared error, R squared error, Mean absolute error, Accuracy, Precision, Recall, etc. Graphs are used from Matplotlib and Seaborn. Finally, the results will be displayed to identify which algorithms best suit the research problem and provide optimized results.

3.2 Data Collection

This section explains the details of data collection. The next step was to find the dataset that would contain all the required parameters for our research. The dataset from Kaggle is taken which is a public domain that contains several datasets based on specific domains. The dataset known as Cloud datacenter workload is used which contains the parameters such as timestamp, CPU cores, CPU capacity provisioned [MHz], CPU Usage [MHz], CPU usage [%], Memory capacity provisioned [KB], Memory Usage [KB], Disk read throughput [KB/s], Disk write throughput [KB/s], Network received throughput [KB/s], Network transmitted

throughput [KB/s]. These parameters are used to identify a threshold and decide if the resources need to be upscaled or downscaled. In the downloaded datasets, there are 14 datasets in which we have used one dataset named “3” for our research.

3.3 Data Pre-processing

The obtained dataset may contain null values or missing values and the values and columns in the dataset might not be in the same datatype. To make them symmetric, several techniques are used. The techniques we have used to clean the data in our research are: First Outliers are identified and removed from the dataset. Outliers are the values or points in space that are outside a specific range. The purpose of doing this is to obtain the point values that are within the same data range which leads to a more accurate dataset. The next step is to check if there are any missing values in the dataset, the dataset in this research does not contain any missing values. The value in the dataset contains a lot of decimal values, so it is better to round off the decimal values to simplify the data further. To make the dataset much simpler transformation of the floating value of the dataset to the integer values is required which can be processed easily compared to floating point numbers. Since all the values in the dataset are of the same data type, the transformational method is not used to convert the values from one form to another in the dataset.

3.4 Algorithms Implemented

This section explains the algorithms used in our research. Three algorithms are used in the study and each algorithm is from a different learning technique in the machine learning field such as supervised and reinforcement learning techniques. The reason for using these algorithms is they reliable predicting models. The algorithms used are:

Linear Regression: It is a statistical technique used to model the relationship between a dependent variable (target) and one or more independent variables (features). It is one of the simplest and most widely used algorithms to predict future data. It works well with datasets with linear relationships between the variables. It provides the exact predictions for the given datasets.

Random forest: It is an ensemble learning technique used for both classification and regression techniques. It builds multiple decision trees and combines the output from all of them to provide more accurate and robust predictions. Unlike linear regression models, random forests can capture complex non-linear relationships in the dataset and still provide accurate predictions.

Generative Adversarial Networks (GANs): GANs generate new data points that resemble the original dataset. GANs consist of two neural networks known as generator and discriminator where the generator’s job is to generate fake samples similar to the original data and the discriminator’s job is to identify which data samples are fake and which are real.

3.5 Workflow

This section explains how the data flows from data loading point till the evaluation phase. The Figure 2 below helps us understand the data flow of our research.

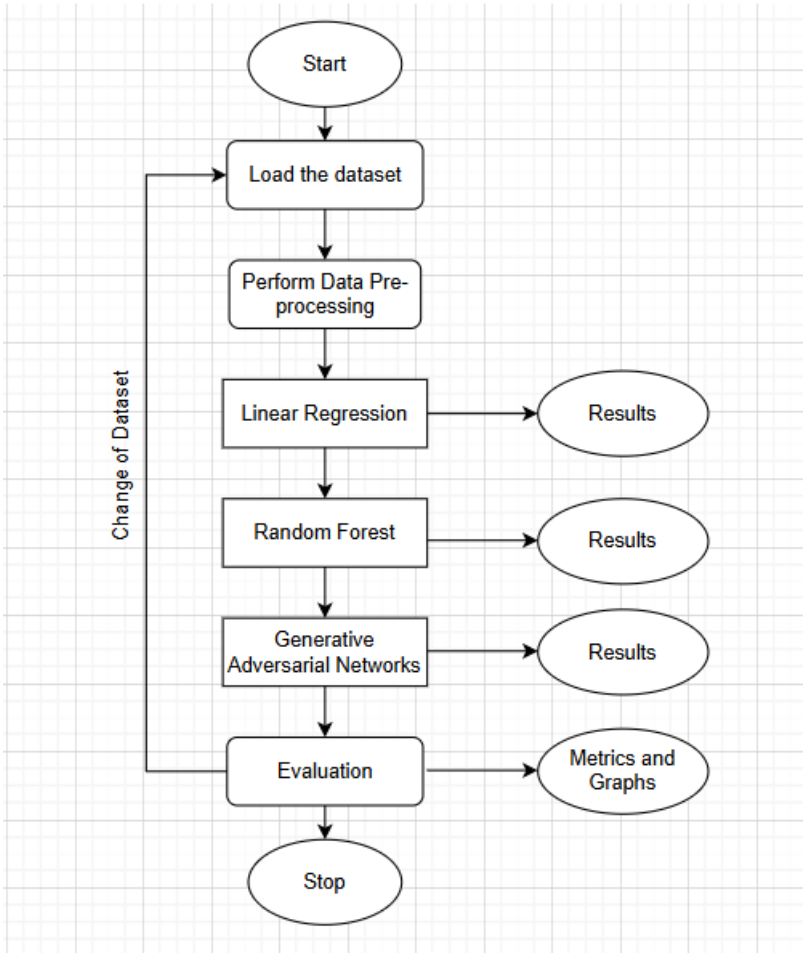


Figure 2: System architecture diagram

The above figure shows the flow of the research. Initially, the dataset is loaded from local into the Google Collab environment. The next step is to perform the data pre-processing to obtain a clean and accurate dataset. The next step is to implement all three algorithms one after the other and get the results from each of the algorithms. Then evaluate the results obtained from each algorithm using several metrics to identify the best-suited algorithm that provides optimized resource allocation and efficiency of the data center resources. Each algorithm has its own performance metrics to measure. The flow stops after obtaining the performance metrics for each algorithm.

4 Design Specification

This section explains the details of the implementation concerning the tools, algorithms, and methods used specifically for the research. The techniques used are as follows:

Datasets: A dataset is required that has the required parameters to train and test the model to achieve good results and implement the best model to optimize the resources in the cloud data center. A publicly available dataset can be used from a genuine source or any private dataset can be used specifically to train the model, once the required consent is received from the owner of the dataset. Own data can be created by using a simulation tool providing the required parameters. Many simulator platforms can be used to generate the data, for instance,

ifog simulator. Once the data is obtained, we must ensure that the data is cleaned to eliminate any noise or null values and transform the values of the dataset into a single datatype which makes the processing of the dataset much easier.

Processing Platforms: Any notebook can be used to process the machine learning algorithms and pre-process the datasets. In this research, the Google Colab platform is used which is free and requires no installation and setup. There are other notebooks such as, Juypter Notebook, and Microsoft Visual Studio, and if we have a subscription to AWS, Amazon SageMaker can be used. Any of these platforms can be used to train and test the machine learning models.

Algorithms: Once the dataset is pre-processed into a clean dataset, algorithms must be identified as per the requirements. In this research, three algorithms are used that are most widely used to predict future data and take appropriate action. One or more algorithms must be chosen to train and test the model and verify we are getting the expected output to solve our problem. In this research, the algorithms Linear Regression, Random Forest, and Generative Adversarial Networks (GANs) are used to solve the cloud problem. The model can be retrained as many times as needed until good results are achieved.

Performance Metrics: Each algorithm has a measure of performance to identify if the model can achieve good results or not. Many performance metrics can be used to identify the working measure of the model. The performance metrics vary with the class of machine learning, for example, classification algorithms have the performance metrics known as Mean square error, R squared error, and Mean Absolute error, and regression algorithms have the same set of metrics as classification along with Accuracy measurements, and Neural network algorithms has the metrics known as Precision, Recall, and Accuracy. Graphs can be used to visually represent the evaluation metrics of the models using Matplotlib, Seaborn, and confusion matrix utilities in Python language. Confusion Metrix is a useful matrix that shows the true positive, true negative, false positive, and false negative prediction of the algorithm.

5 Implementation

The implementation of the final steps in the research includes training and testing the machine learning models on the dataset. It includes the details of the logic used to obtain the required output and perform evaluation on each model to identify the best model for the research.

5.1 Tools and Languages Used

Google Colab: Google Colab[16] is a hosted Juypter Notebook service that requires no setup or configuration and provides free access to computing resources and utilities including GPUs and TPUs. The Colab platform is well-suited for machine learning, and data science projects. It is easy to understand the workings of the platform and helps in achieving our requirements.

Python Programming Language: Python[17] is a high-level, general-purpose programming language. Python is dynamically typed and garbage collected. Python is most commonly used for developing websites and software, task automation, data analysis, and data visualization. Python is an interpreted language and easy to understand the syntax of the program which is similar to pseudo-codes.

5.2 Final Stage of the Implementation Process and Output Produced

Linear Regression: After the data is pre-processed, the dataset is used to train and test the Linear Regression model. In the Linear Regression[18] model, a scaling factor is utilized for the features CPU Usage, Memory Usage, Disk read, Disk write, Network transmitted, and Network received where a particular threshold is specified for greater than and lesser than. This scaling factor is then used in a for loop to determine if the value of each row is greater than the threshold value then a scaling status upscale will be displayed and if the value of each row is lesser than the threshold value then a scaling status of downscale will be displayed and if there is no change in the value then the scaling status of no change will be displayed. There is a logic implemented to check the CPU Usage and activate the cooling system. Here, a CPU Usage threshold value is specified and if the current value of CPU usage is more than the threshold then the status will be displayed as Cooling Activated. The flow diagrams [1][2] below show the logic of the function implemented in the study.

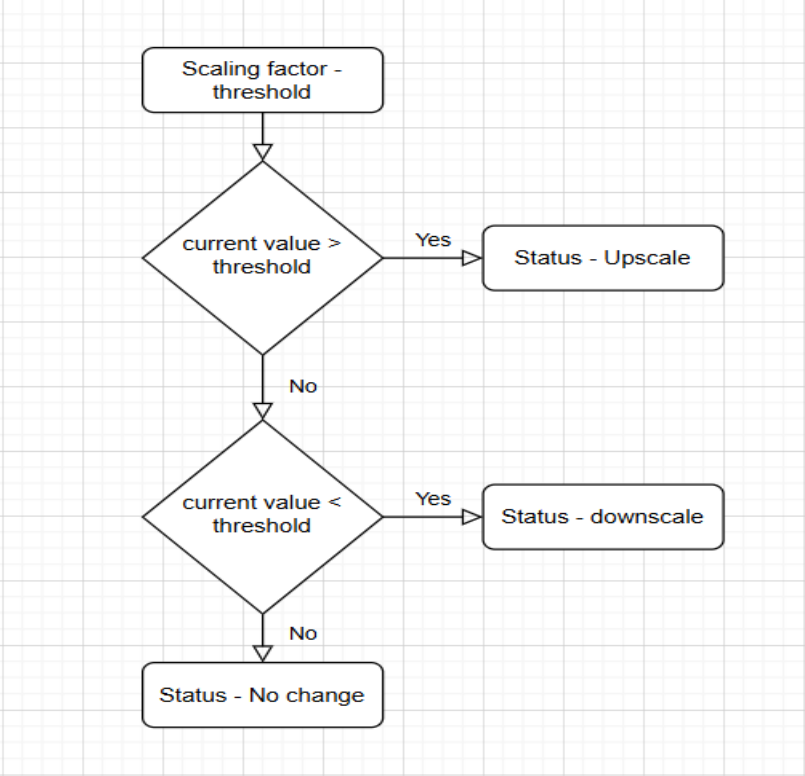


Figure 1: Flow diagram of the linear regression function

The diagram [2] below shows the flowchart of the CPU usage logic implemented in the study.

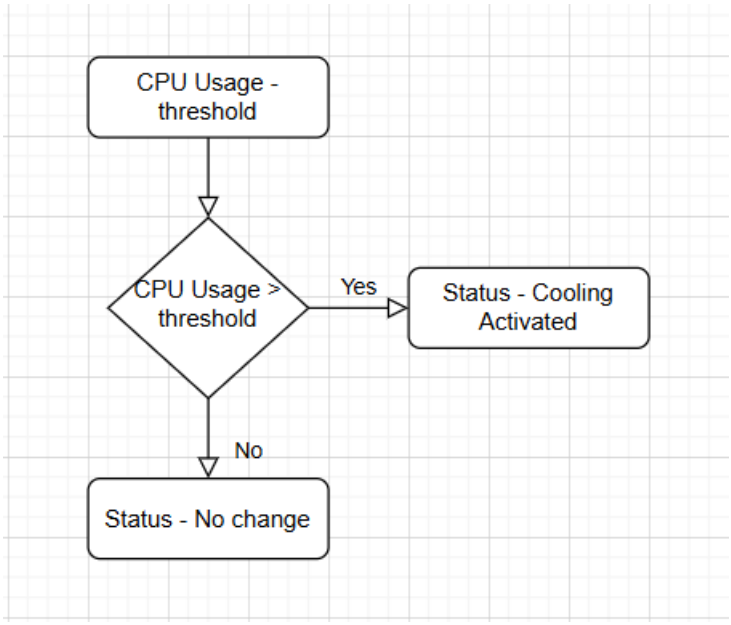


Figure 2: Flow diagram of CPU usage function

Random Forest Algorithm: After the dataset is pre-processed, the dataset is used to train and test the model[19]. A threshold is created for the features CPU Usage, Memory Usage, Disk read, Disk write, Network transmitted, and Network received in the dataset. Then the code is implemented to check values in each row and if that value is greater than the threshold value then the scaling status is set to upscale and if the value is lesser than the threshold value then the scaling status is set to downscale and if the value is same as the threshold value, then the scaling status will be set to no change. An additional column will be added at the end of the output which is the scaling status. The flow diagram [3] shows the logic of the function implemented.

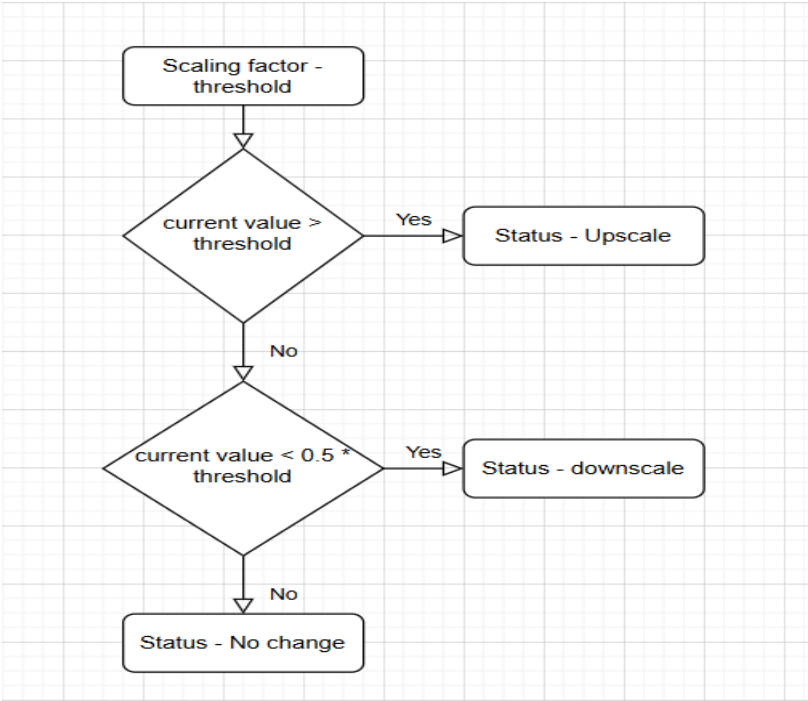


Figure 3: Flow diagram of the Random Forest function

Generative Adversarial Networks (GANs): After the dataset is pre-processed, the dataset is used to train and test the model[20]. In this model, there are two neural networks one is a generator, and another one is a discriminator. The Generator will generate fake data samples similar to real data and the discriminator's job is to identify the real and fake samples. The scaling factors are defined for the features CPU Usage, Memory Usage, Disk read, Disk write, Network transmitted, and Network received. Based on the threshold value, if the current value exceeds then it will display as a max threshold, scaling up. If the current value falls below the threshold value, then it will be displayed as a min threshold, scaling down. The flow diagram [4] shows the logic of the function implemented.

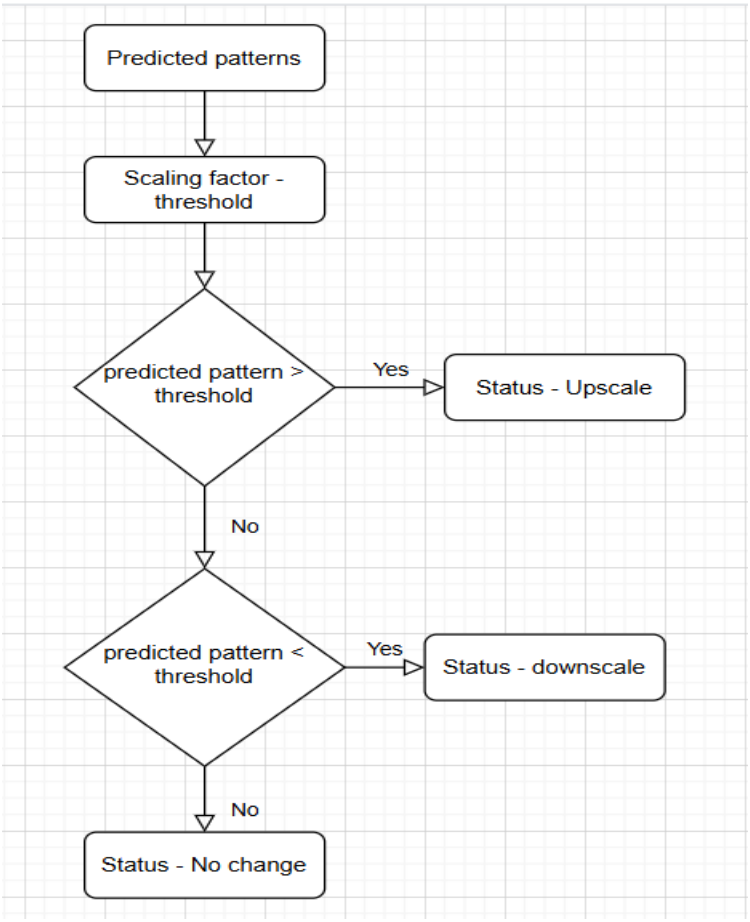


Figure 4: Flow diagram of the GANs function

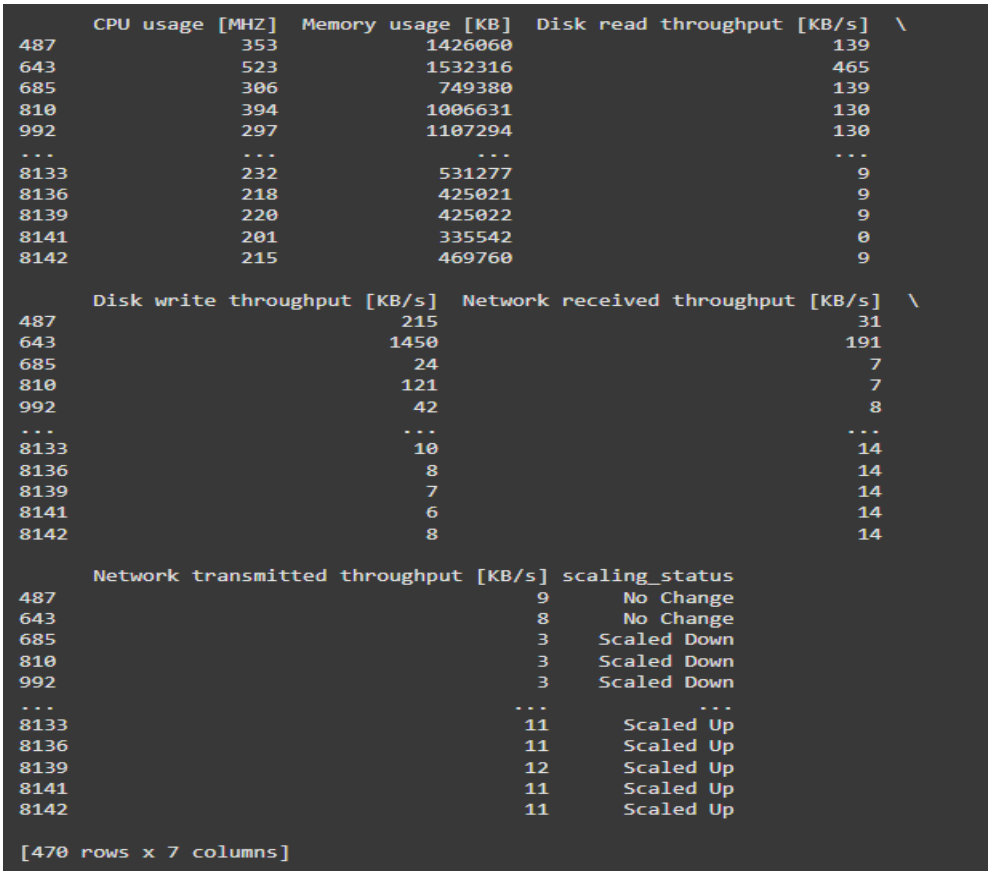
During the Implementation phase, the Google Colab platform is used and Python programming language to solve our research problem. Three algorithms are used that are most widely used for predicting future data based on historical data. We have implemented the logic of threshold-based algorithms, regression and neural network algorithms to achieve the best model to optimize the data center resources efficiently.

6 Evaluation

In this section, there will be a detailed review of the performance metrics of all the algorithms used in the research to solve the problem. We will look at the output of individual algorithms along with performance review and what each of the performance evaluation metrics defines.

6.1 Linear Regression

The output of the Linear Regression algorithms shows the scaling status that is defined for each row of the dataset for the features included in the study. Figure 1 shows the output of the algorithm after training the model and testing the model on the dataset. The last column added known as scaling status shows the values that are upscaled, downscaled and no change are present.



	CPU usage [MHZ]	Memory usage [KB]	Disk read throughput [KB/s]	\
487	353	1426060	139	
643	523	1532316	465	
685	306	749380	139	
810	394	1006631	130	
992	297	1107294	130	
...	...	...	...	
8133	232	531277	9	
8136	218	425021	9	
8139	220	425022	9	
8141	201	335542	0	
8142	215	469760	9	

	Disk write throughput [KB/s]	Network received throughput [KB/s]	\
487	215	31	
643	1450	191	
685	24	7	
810	121	7	
992	42	8	
...	...	...	
8133	10	14	
8136	8	14	
8139	7	14	
8141	6	14	
8142	8	14	

	Network transmitted throughput [KB/s]	scaling_status
487	9	No Change
643	8	No Change
685	3	Scaled Down
810	3	Scaled Down
992	3	Scaled Down
...	...	...
8133	11	Scaled Up
8136	11	Scaled Up
8139	12	Scaled Up
8141	11	Scaled Up
8142	11	Scaled Up

[470 rows x 7 columns]

Figure 1: Output of the Linear Regression algorithms dataset after the function implementation

From the below figures 2 and 3 we can see how the CPU cooling systems are activated when the CPU usage goes beyond the threshold. When the CPU usage is within the range then it shows the percentage of CPU usage and status as Normal.

Function Parameter	Output
CPU Usage	3.5%
Cooling Status	Normal

Figure 2: Output of the CPU usage function

Function Parameter	Output
CPU Usage	42.1%
Cooling Status	Cooling Activated

Figure 3: Output of the CPU usage function

The evaluation metrics known as Mean Squared Error (MSE), R squared error, and Mean Absolute Error (MAE) are used as shown in below figure 4.
Mean Squared Error indicates on average the squared distance between the predicted value and the actual value. Lower values indicate a better fit.
R-squared error indicates how well the model explains the variance in the target variable. 0.82 approximately is 82.36 % which is a good fit and indicates that the model was able to capture most of the model's pattern.
Mean Absolute Error indicates the average of absolute errors between the predicted values and actual values. The MAE score is 0.13 indicates on an average, predictions are off by 0.13 units.

Evaluation Metrics	Output
Mean Squared Error	0.04343300028780921
R Square Score	0.823612599814694
Mean Absolute Error (MAE)	0.13

Figure 4: The Evaluation metrics of Linear Regression Algorithm

From the below figure 5, there is a diagonal line going through two blue points. The blue points are the predictions made by the model and the diagonal line presents the ideal situation where the predictions perfectly match the true values. Points that lie on this line indicate that the model made an accurate prediction. Overall, the Linear Regression algorithms provided good results for the dataset.

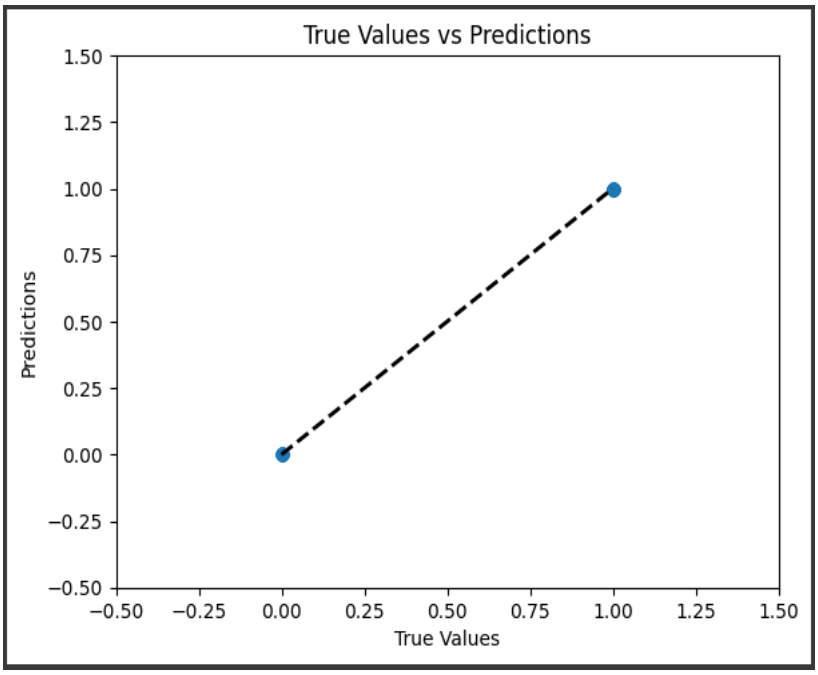


Figure 5: Visual representation of true vs predicted values of the output of Linear Regression algorithm

6.2 Random Forest

The output of the Random Forest algorithm shows the number of resources that are upscaled and downscaled and the number of resources that are not changed. Figure 1 below shows the out of the algorithm.

Scaling Status	Count of resources
Scale up	283
No change	180
Scale down	7

Figure 1: Output of Random Forest algorithm showing the scaling status

The below figure 2, classification report indicates the performance evaluation of the Random Forest algorithm. The precision is 1.0 for no change and scaled-down status which indicates that when the model predicted “no change” and “scaled down” it was always right. Whereas, for scaled-up status, the precision is 0.98 which indicates the model has predicted the status “scaled-up” 98% of the time. Recall indicates that the system successfully identified the status correctly based on the performance. In this output, “no change” status is 0.97 which indicates that the model identified the status “no-change” 97% of the time. For the status “scaled-down” and “scaled-up” it indicates that 100% of the predictions are successfully identified correctly.

	Precision	Recall	F1-score	Support
No change	1.00	0.97	0.99	36
Scale down	1.00	1.00	1.00	1
Scale up	0.98	1.00	0.99	57
Accuracy			0.99	94
Macro avg	0.99	0.99	0.99	94
Weighted avg	0.99	0.99	0.99	94

Figure 2: Classification report showing the evaluation measurements of Random Forest algorithm

The below figure 3, shows the confusion matrix which shows the graph of actual and predicted values. In the output, 35 predictions are made for the status “no change” and 1 prediction is made for the status “scaled-down” and 57 predictions are made for the status “scaled-up”. It can be noticed in the output that one of the predictions is made as “scaled-up” when the actual prediction is “no-change”. A Confusion matrix[21] is a great visual presentation of the predicted and actual values which helps us understand how the model is working easily.

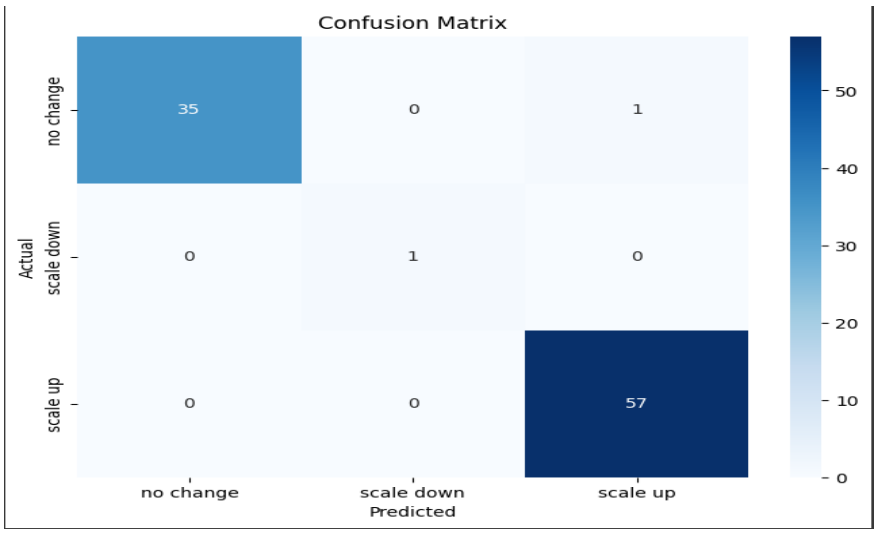


Figure 3: Confusion Matrix of the output of Random Forest algorithm

Figure 4, which indicates the important features of the dataset. The figure below shows that Network transmitted throughput, and Memory Usage features have high importance while features like Disk read throughput and Network received throughput have relatively low importance. This importance of the features shows a strong correlation with the target variable.

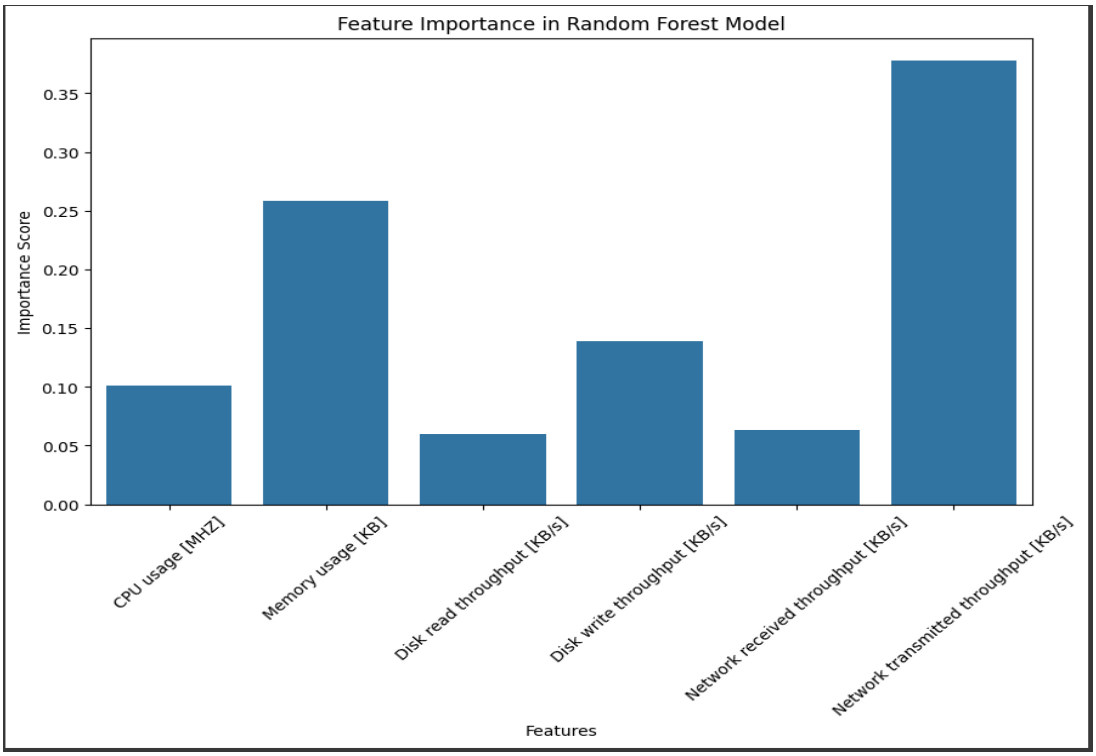


Figure 4: Importance graph indicating the features which has a strong correlation in the dataset

6.3 Generative Adversarial Networks (GANs)

The output of GANs shows the features which are scaled-up, scaled-down, and no-change. From the below figure 1, we can see the output of the model.

Features	Scaling status
CPU Usage [MHz] below min threshold	Scaling down
Memory Usage [KB] below min threshold	Scaling down
Disk read throughput [KB/s] below min threshold	Scaling down
Disk write throughput [KB/s] below min threshold	Scaling down
Network received throughput [KB/s] within normal range	No change
Network transmitted throughput [KB/s] exceeds max threshold	Scaling up

Figure 1: Output of the GANs algorithm showing the scaling status

The below figure 2, indicates the performance evaluation of the GAN model. From the figure, it can be seen that the Final Discriminator Accuracy is 28% which is the accuracy of the discriminator. This value shows how well the discriminator can distinguish between real and fake samples of data. It is noticeable that the other metrics such as precision, recall, and accuracy are the overall model accuracy.

Evaluation Metrics	Output
Final Discriminator Accuracy	28.47%
Epoch	179
D Loss	4.5783
D Acc	28.47%
G Loss	4.5783
Precision	0.4000
Recall	0.6667
F1-Score	0.5000
Accuracy	33.33%

Figure 2: Performance Evaluation metrics output of GANs

Figure 3, shows the confusion matrix which indicates how many samples were properly predicted as fake and real. From the below figure it is seen that 0 fake samples were predicted as fake samples, 6 of the fake samples were predicted as real samples, and 2 of the real samples were predicted as fake samples and 4 of the real samples were correctly predicted as real samples.

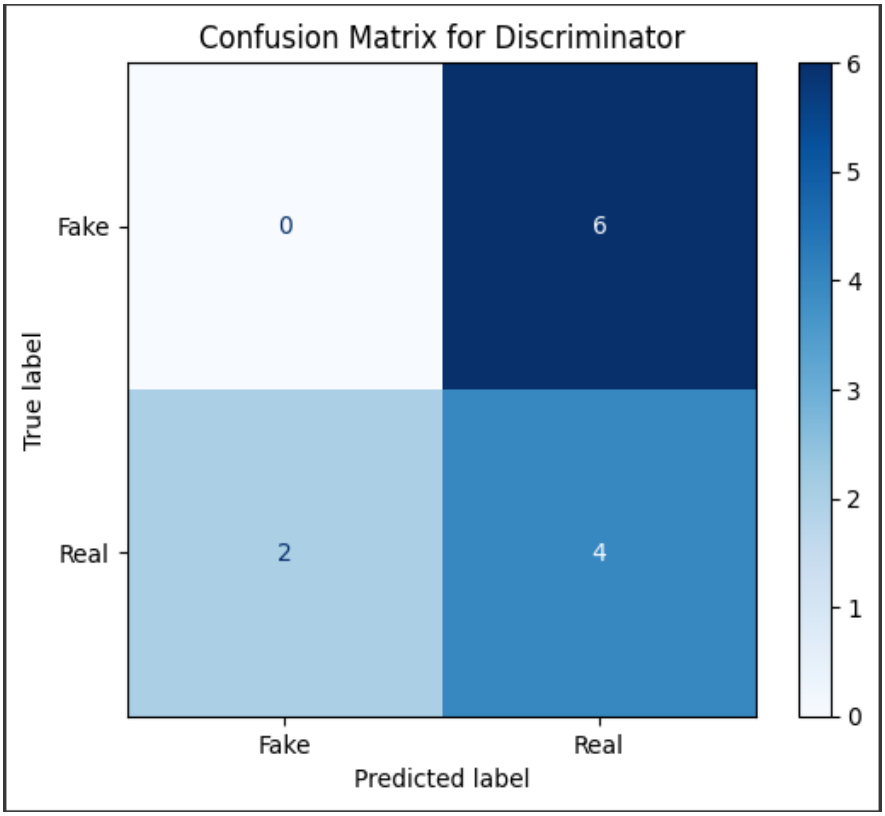


Figure 3: Confusion Matrix showing the True and Predicted values by GANs

7 Conclusion and Future Work

In this study, machine learning models are used to solve the cloud problem where the resources in the data center can be optimized avoiding overutilization or underutilization and more power consumption which results in higher operational costs and violation of SLAs. This study shows the successful implementation of the machine learning working model which can be used to reduce operational costs by optimizing the workload distribution, load balancing, auto-scaling, and resource allocation. Three algorithms are utilized in this study which are, Linear Regression, Random Forest, and Generative Adversarial Networks (GANs). Linear Regression and Random Forest algorithms have provided exceptionally good results compared to GANs. The performance evaluation of the Linear Regression and Random Forest algorithms indicates the model's behavior in predicting the workloads and taking appropriate actions such as upscaling or downscaling the resources. This can be used to optimize the resources in the datacenter for better performance. Every algorithm performs differently with different datasets, so it would be better to train the model on real-time parameters and data and choose the model that better fits the requirement.

As a part of future work, the model can be tested on other datasets to understand the behavior and performance. It would be better to use the parameters from the datacenter to understand the working of the model in real-time scenarios. GANs can further be optimized by using a complex dataset which can help in improving the performance of the model. Using complex algorithms such as GANs can increase the computational cost affecting the environment. Hence, we can find a way out to use the complex models and make sure green computing is achieved at the same time.

References

[1] Konstantin, Pilz., Lennart, Heim. (2023). Compute at Scale - A Broad Investigation into the Data Center Industry. arXiv.org, abs/2311.02651 doi: 10.48550/arxiv.2311.02651

[2] (2023). Introduction. 1-3. doi: 10.1002/9781119898849.ch1

[3] Sahana, Shetty., H., K., Shashikala. (2023). An Innovation Development of New Perspective of Efficient Approaches, Techniques and Challenges for Data Centers. 1-6. doi: 10.1109/ICDCECE57866.2023.10151008

[4] Suraj, Singh, Panwar., M., M., S., Rauthan., Varun, Barthwal., Nidhi, Mehra., Ashish, Semwal. (2024). Machine learning approaches for efficient energy utilization in cloud data centers. Procedia Computer Science, 235:1782-1792. doi: 10.1016/j.procs.2024.04.169

[5] Niranchana, Radhakrishnan., T., R., Vedhavathy., Marxim, Rahula., Bharathi., S., Chakaravarthi., T., Ramesh. (2024). Optimizing Data Centre Energy Efficiency with Dynamic Resource Allocation and Intelligent Cooling Management through Machine Learning. Journal of Electrical Systems, doi: 10.52783/jes.1287

[6] Khush, Patel., Nishant, Mehta., Pranshu, Oza., Jignesh, Thaker., Ashlesha, Bhise. (2024). Revolutionizing Data Centre Sustainability: The Role of Machine Learning in Energy Efficiency. 2:1-6. doi: 10.1109/iatmsi60426.2024.10503107

[7] Khush, Patel., Nishant, Mehta., Pranshu, Oza., Jignesh, Thaker., Ashlesha, Bhise. (2024). Revolutionizing Data Centre Sustainability: The Role of Machine Learning in Energy Efficiency. 2:1-6. doi: 10.1109/iatmsi60426.2024.10503107

[8] Goswami, Amit (2023) Dynamic Load Balancing and Resource Management Using Machine Learning. Masters thesis, Dublin, National College of Ireland.

[9] D. Saxena, A. K. Singh and R. Buyya, "OP-MLB: An Online VM Prediction-Based Multi-Objective Load Balancing Framework for Resource Management at Cloud Data Center," in IEEE Transactions on Cloud Computing, vol. 10, no. 4, pp. 2804-2816, 1 Oct.-Dec. 2022, doi: 10.1109/TCC.2021.3059096.
keywords: {Servers;Resource management;Cloud computing;Task analysis;Load management;Data centers;Power demand;Cloud computing;communication cost;load balancing;online-prediction;oversubscription;server;virtual machine},

[10] Ekhande, Ashish (2020) *Improvement in auto scaling mechanism of cloud computing resources using Composite ANN*. Masters thesis, Dublin, National College of Ireland.

[11] Ravikumaraiah, Mohith Srivathsa (2022) *Enhanced Auto-scaling by using Dynamic scaling policy with Deep learning :LSTM*. Masters thesis, Dublin, National College of Ireland.

[12] Peter, Jackson (2022) *Improving the Auto scaling mechanism in Cloud computing environment using Support Vector regression and Bi-LSTM.* Masters thesis, Dublin, National College of Ireland.

[13] Gursale, Sumedh (2020) *A Proactive Mechanism To Improve Workload Prediction For Cloud ServicesUsing Machine Learning.* Masters thesis, Dublin, National College of Ireland.

[14] Chima, Kelechukwu (2021) Resource Management in a Cloud Computing Environment using Generative Adversarial Networks (GANs). Masters thesis, Dublin, National College of Ireland.

[15] Sonawane, Tushar Pramod (2023) Energy Efficiency & Consumption in Data Centre by Dynamic Resource Allocation Technique for Green Cloud Computing. Masters thesis, Dublin, National College of Ireland.

[16] https://research.google.com/colaboratory/faq.html
[17] https://www.python.org/doc/essays/blurb/
[18] https://www.ibm.com/topics/linear-regression
[19] https://www.ibm.com/topics/random-forest
[20] https://aws.amazon.com/what-is/gan/
[21] https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62