iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** V **Month of publication:** May 2023

DOI: https://doi.org/10.22214/ijraset.2023.51489

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

Review on Machine Learning for Resource Usage Cost Optimization in Cloud Computing

Navdeep Tanwar[1], Dr. Praveen Kumar K V[2]
[1]*Student, Dept. Of CSE, Sapthagiri College of Engineering Bangalore, Karnataka-560057*
[2]*Professor, Sapthagiri College of Engineering Bangalore, Karnataka-560057*

Abstract: ***Small and medium-sized enterprises are increasingly adopting cloud computing, and optimizing the cost of cloud resources has become a crucial concern for them. Although several methods have been proposed to optimize cloud computing resources, these methods mainly focus on a single factor, such as compute power, which may not yield satisfactory results in real-world cloud workloads that are multi-factor, dynamic, and irregular.***
This paper proposes a new approach that utilizes anomaly detection, machine learning, and particle swarm optimization to achieve a cost-optimal cloud resource configuration.
The proposed solution works in a closed loop and does not require external supervision or initialization, learns about the system's usage patterns, and filters out anomalous situations on the fly.
Additionally, the solution can adapt to changes in both system load and the cloud provider's pricing plan. The proposed solution was tested on Microsoft's Azure cloud environment using data collected from a real-life system, and the results show that it achieved an 85% cost reduction over a ten-month period..
Index Terms: cloud resource usage prediction, anomaly detection, machine learning, particle swarm optimization, resource cost optimization. .

I. INTRODUCTION

Cloud computing providers like Amazon Web Services (operated by Amazon), Azure (operated by Microsoft), and Google Cloud Platform (operated by Google) are popular locations for computer systems.
These clouds offer storage, network, and computing resources to users who need them. Different cloud usage models, such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), reduce management effort and downtime risk while providing high scalability possibilities compared to on-premise solutions.
Scalability allows for the addition of new instances of services (PaaS), virtual machines (IaaS), or databases (which are partially SaaS and partially PaaS) as needed. However, it can be challenging to predict load beforehand, making it difficult to meet accessibility and responsiveness requirements.
Therefore, the system must be scaled up with a margin for unforeseen load spikes and long-term load changes, resulting in considerable power and storage overprovisioning and unnecessary spending To reduce costs and protect the environment, it is crucial to optimize cloud resource usage by predicting demand for different resources, such as CPU, memory, storage, and input/output operations per second (IOPS), and adjusting cloud components accordingly. Our proposed solution automates the process of scaling system components while taking into account the predicted usage level, including virtual machines, application services, and databases.
We use machine learning interpolation combined with anomaly detection to predict demand and optimize cloud components that meet the demand and are financially optimal. To achieve the optimal configuration, we use a particle swarm optimization (PSO) algorithm tailored to solving discrete problems.
The traditional approach to cloud resource optimization either focuses on a single resource, such as CPU, and scaling parameter, like the number of machines, or creates resource utilization models that ignore potential unexpected changes. Our proposed solution takes a more comprehensive approach that considers all resources, predicts demand, and adjusts cloud components accordingly, leading to significant cost reductions and environmental protection..

II. LITERATURE REVIEW

1) The proposed model involves data set from Google cluster usage trace, Animoto and IBM Global Services. Pre-processing is the processing steps they applied on dataset before performing the prediction. Different preprocessing technique are applied on data to support their mechanism. Evaluation method is the evaluation metric used to validate the prediction results. Mean Square Error (MSE), Root Mean Square Error (RMSE), Min Max and Average are most commonly used techniques for validating the mechanism prediction results. Comparison is list of existing techniques with which the results are compared. Focus is the goal for which they perform the workload prediction. Workload prediction is critical for efficient resource allocation. Thus, goal of this research is to help cloud provider to improve resource utilization. In this research, Workload trace data provided by Google is being analyzed. Google publicly released Google cluster usage trace data for research purposes, to make the researchers aware of real data and actual complexity faced by cloud provider. The dataset is a trace of production workloads running on Google clusters collected. “Ensemble based workload predictor” based on stacking mechanism is proposed as a prediction mechanism using Over Produce and Choose (OPC) approach . Stacked KNN & DT is stacking ensemble of K-Nearest Neighbour and Decision Tree. Stacking works by pruning heterogeneous learning algorithms. Stacking is one of the ensemble techniques well known for reducing error of classifiers. It includes combination of classification algorithms that reduces biasness of one algorithm. Stacking procedure is explained is been divided into two steps level 0 and level 1. Level 0 learners are base learner and level 1 is the Meta learner. K Nearest Neighbour and Decision Tree are base learners. These base learner are used to train on the dataset and test in the second step. Level 1 learner DT that combines the output of level 0 learners. It combines the output by correcting mistakes to improve classification result. Training dataset is passed to level 0 base classifiers KNN and DT..
2) The methodology of this paper uses a pool of multiple type of instances which differ by their computing power, physical memory or additional hardware such as GPU. The number of different instance differ by their configurations is derived by using historic data which includes assets processing jobs and the corresponding resource utilization by assigned instance type. Each instance from a pool of instances will be monitoring a unique and individual queue such as AWS Simple Queue Service from a pool of queues where size of queue-pool is equal to size of instance –pool. When an input job request to generate rendition of a cloud asset is sent to Central Service (SvC) from client application, SvC's 'Asset Property Analysers' fetches necessary information is passed through 'Performance Monitoring Model' which predicts CPU and memory utilization to process such input job-request. As per the output from 'Performance Monitoring Model' , SvC construct a message (JSON object) and sends message to corresponding queues. A corresponding monitoring instance fetches the message from queue and process further . Hence, the job – request is processed by an appropriate instance-type which optimizes resod to other state-of-the-art algorithms.
3) The methodology of the paper involves collecting historical usage data from cloud service providers. This data includes information on resource utilization and cost for different types of cloud services. The authors then extract features from the data that can be used to predict future resource demands. These features may include application characteristics, such as the number of users or the type of workload, as well as resource usage patterns, such as the amount of CPU or memory used. Then use supervised learning algorithms to predict future resource demands based on the extracted features. They also use unsupervised learning algorithms to identify patterns in the data and group similar applications together. This helps to improve the accuracy of the predictions and allows the system to learn from past behaviour. Once the system has predicted future resource demands, it uses a cost model to evaluate the cost-effectiveness of different resource allocation strategies. The cost model estimates the cost of cloud services based on resource usage and pricing information from cloud providers. Then the cost model to make recommendations on how to optimize costs, such as by adjusting the allocation of resources or switching to a different type of cloud service. Finally to evaluate the proposed system using a real-world dataset and compare it to other cost optimization techniques. They measure the effectiveness of the system in optimizing costs while maintaining the same level of service quality. The experimental results show that the proposed system achieves better cost savings compared to other approaches while maintaining the same level of service quality. This paper involves collecting historical usage data, extracting features, using machine learning algorithms to predict future resource demands, developing a cost model to optimize resource allocation and minimize costs, and evaluating the system's performance using a real-world dataset . This paper uses An autoregressive model with polynomial coefficient is a type of time series model that can be used to forecast the demand for cloud computing resources. This model assumes that the future values of the time series are linearly related to past values, and that the relationship between past and future values can be described by a polynomial function. To use this model to forecast the demand for cloud computing resources, you would first collect historical data on resource usage, such as CPU utilization,

network traffic, and storage usage. You would then fit an autoregressive model with polynomial coefficient to this data, using techniques such as least squares regression or maximum likelihood estimation. Once you have fitted the model, you can use it to make predictions about future resource usage. For example, you might use the model to predict how much CPU capacity will be needed in the next hour, based on past CPU utilization and other factors that affect demand, such as time of day or day of the week. It's important to note that while autoregressive models with polynomial coefficients can be effective at forecasting time series data, they may not be suitable for all types of demand forecasting problems. Other models, such as neural networks or decision trees, may be more appropriate depending on the specific characteristics of the data and the problem you are trying to solve.

4) The methodology for this paper involves proposing and evaluating three algorithms for cost optimization in cloud data centers. The first algorithm is an optimal offline algorithm that leverages dynamic and linear programming techniques to minimize cost under the assumption of available exact knowledge of workload on objects. The second and third algorithms are online algorithms that dynamically select storage classes across CSPs while making trade-offs between residential and migration costs. The methodology for this paper involves proposing and evaluating three algorithms for cost optimization in cloud data centers. The first algorithm is an optimal offline algorithm that leverages dynamic and linear programming techniques to minimize cost under the assumption of available exact knowledge of workload on objects. The second and third algorithms are online algorithms that dynamically select storage classes across CSPs while making trade-offs between residential and migration costs. A deterministic online algorithm is an algorithm that makes decisions without knowledge of future input. It operates in an online setting where input arrives in a sequential order and decisions must be made immediately based on the available information at the time. In a deterministic online algorithm, the decisions made are based solely on the input received up to that point in time, without any consideration of future input. This is in contrast to a stochastic online algorithm, which may use probabilistic methods to make decisions based on uncertain future input.Deterministic online algorithms are often used in real-time systems, where decisions must be made quickly and efficiently based on real-time data. A randomized online algorithm is an algorithm that makes decisions in an online setting using randomization. Like deterministic online algorithms, randomized online algorithms operate in a sequential, online setting, where input arrives in a sequential order and decisions must be made immediately based on the available information at the time

5) The methodology for this paper involves the use of a deep neural network as a function approximator for the Q-function in the reinforcement learning algorithm. The authors use the experience replay technique to store and sample experiences for training the neural network. They also use an epsilon-greedy policy for exploration and exploitation during training. The sensitivity analysis involves varying different parameters such as the number of VMs, the workload intensity, and the learning rate to evaluate their impact on the performance of the DRL approach. An intelligent resource management architecture, mainly contains two components: a intelligent resource manager, which is composed of controller, monitor, and allocator and an IT resource, which consists of extensive resource pools.12 Clients first communicate with the controller to submit application requests with various demands. Based on application demands and current resource utilization information, the controller implements the algorithm chosen from its resource schedule algorithm pool to meet application demands, while respecting system resource constraint. The resource schedule algorithm pool, which plays an important role in intelligent resource management architecture, includes different kinds of algorithms, such as offline and online algorithms and algorithms combining both online and offline parts. The monitor is responsible for gathering information of system resource utilization and application quality of service (QoS) to update the controller periodically, and the allocator is in charge of mapping applications to resource pools according to the configuration negotiated by the controller. The controller is the key part of a resource management architecture, as it not only figures out the (near-) optimal configuration policy but also coordinates with the monitor and allocator to allocate resources intelligently. The heart of the controller is a resource schedule algorithm pool, which contains plenty of control algorithms. The DRL algorithm presented in this paper is an online algorithm, which connects reinforcement learning with deep learning to generate the (near-) optimal resource configuration in limited iterations directly from raw application demands, especially for high dimensional demands.. The deep neural network is pretrained through the stacked autoencoder (SA), followed by using reinforcement learning experiences for optimization.

III. CONSOLIDATED TABLE

S.No	AUTHOR	YEAR	DESCRIPTION	LIMITATION
1.	Tajwar Mehmood, Dr.Seemab Latif ,Dr. Sheheryaar Malik	2018	The paper proposes an ensemble-based workload predictor using stacking mechanism to predict cloud computing resource utilization. The dataset used is the Google cluster usage trace data. The proposed model aims to improve resource utilization. Stacking is used to prune heterogeneous learning algorithms and reduce error of classifiers. KNN and DT are base learners, and DT is the Meta learner.	• Computational Overhead • Limited data set • No real world application
2.	Nirmal Kumawat, Nikhil Handa, Avinash Kharbanda	2020	The paper propose a system to predict cloud computing resources to serve input asset processing request . The method And system are designed in such a way to maximize resource utilization, minimize cost spent and minimize processing time by such computational resources. The method includes training of supervised learning based predictive model with historic data which includes asset processing requests, asset properties..	• Scalability • Resource utilization • Not Optimal long term strategy
3.	Quan Ding, Bo Tang, Prakash Manden, Jin Ren	2018	The paper proposes using an autoregressive model with polynomial coefficient to forecast the demand for cloud computing resources. This model assumes a linear relationship between past and future values and a polynomial function to describe the relationship. Historical data on resource usage would be collected, and the model would be fitted to the data. The fitted model can then be used to predict future resource usage.	• Lack of real-world testing • Limited Scope • Lack of Comparison
4.	Yaser Mansouri, Adel Nadjaran Toosi , Rajkumar Buyya	2019	The methodology for this paper involves proposing and evaluating three algorithms for cost optimization in cloud data centers. The first algorithm is an optimal offline algorithm that leverages dynamic and linear programming techniques to minimize cost under the assumption of available exact knowledge of workload on objects. The second and third algorithms are online algorithms that dynamically select storage classes across CSPs while making trade-offs between residential and migration costs.	• Limited Scope • Lack of real world validation • Complexity
5.	Yu Zhang, Jianguo Yao, Haibing Guan	2018	This paper proposes a methodology for intelligent cloud resource management using deep reinforcement learning. The authors formulate the problem as a Markov decision process and develop a deep Q-network algorithm to learn an optimal resource allocation policy. They evaluate the performance using experiments on a cloud simulation platform.	• Large amount of Computational Resources • Complexity • Large amount of training Data

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

IV. CONCLUSION AND FUTURE SCOPE

The paper proposes a machine learning-based approach for optimizing resource usage costs in cloud computing. It involves collecting data and conducting tests in a mock environment. Four different prediction types were implemented and compared, showing effectiveness in optimizing resource usage and reducing costs. The approach also detects anomalies and generates accurate predictions. Overall, it presents a promising approach for cost optimization in cloud computing using machine learning techniques

Machine learning can enhance the optimization of resource usage cost in cloud computing in several ways. It can enable predictive resource allocation, detect anomalies in workload patterns, develop automated optimization algorithms, offer customer-specific optimization, and integrate with billing systems to provide recommendations for cost optimization. These enhancements can help cloud providers allocate resources accurately and proactively, reduce resource wastage, and offer personalized services to customers.

V. ACKNOWLEDGEMENT

One's success cannot be solely attributed to their individual efforts as it is also influenced by the guidance, encouragement, and cooperation of mentors, seniors, and companions. We express our gratitude to **Dr. Praveen Kumar K V**, a Professor in the Computer Science and Engineering Department at Sapthagiri College of Engineering, and **Dr. Kamalakshi Naganna**, the Head of the Computer Science and Engineering Department at Sapthagiri College of Engineering, for their unwavering backing, direction, and aid during our project. Additionally, we extend our appreciation to our parents and friends for providing us with emotional support throughout the journey.

REFERENCES

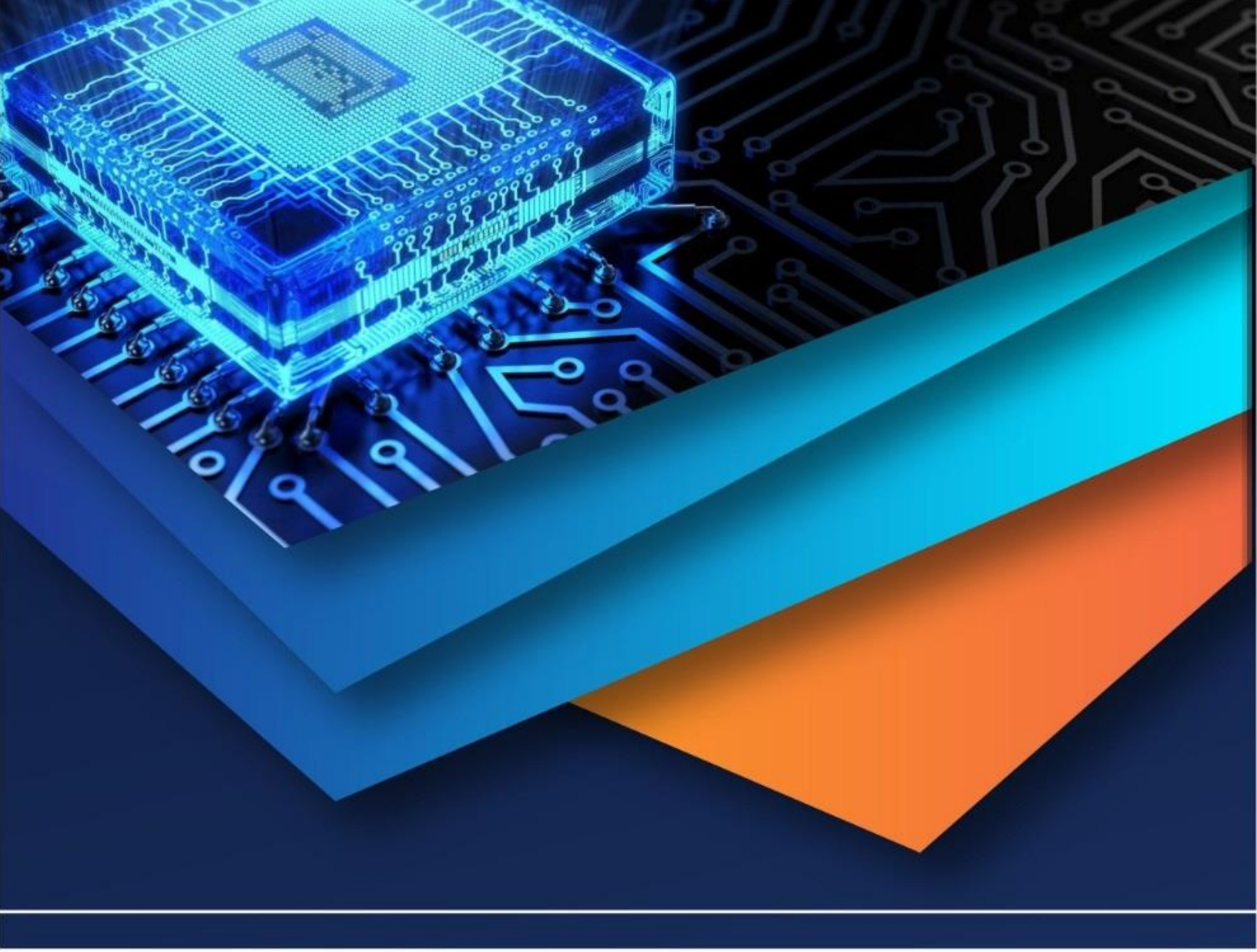
[1] T. Mehmood, S. Latif and S. Malik, "Prediction Of Cloud Computing Resource Utilization," 2018 15th International Conference on Smart Cities: Improving Quality of Life Using ICT & IoT (HONET-ICT), Islamabad, Pakistan, 2018, pp. 38-42, doi: 10.1109/HONET.2018.8551339..

[2] N. Kumawat, N. Handa and A. Kharbanda, "Cloud Computing Resources Utilization and Cost Optimization for Processing Cloud Assets," 2020 IEEE International Conference on Smart Cloud (SmartCloud), Washington, DC, USA, 2020, pp. 41-48, doi: 10.1109/SmartCloud49737.2020.00017.

[3] Q. Ding, B. Tang, P. Manden and J. Ren, "A learning-based cost management system for cloud computing," 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2018, pp. 362-367, doi: 10.1109/CCWC.2018.8301738.

[4] Y. Mansouri, A. N. Toosi and R. Buyya, "Cost Optimization for Dynamic Replication and Migration of Data in Cloud Data Centers," in IEEE Transactions on Cloud Computing, vol. 7, no. 3, pp. 705-718, 1 July-Sept. 2019, doi: 10.1109/TCC.2017.2659728.

[5] Y. Zhang, J. Yao and H. Guan, "Intelligent Cloud Resource Management with Deep Reinforcement Learning," in IEEE Cloud Computing, vol. 4, no. 6, pp. 60-69, November/December 2017, doi: 10.1109/MCC.2018.1081063.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY