# A Regression Learning Based Model For Performance Forecasting In Cloud Environments

[1]Rachita Rathore, [2]Dr. Rajesh Kumar Chakrawarti
[1]Research Scholar, [2]Professor, CSE, SDBCT, Indore
[1]Department of CSE,
[1]SDBCE, Indore, India

***Abstract :*** Data driven cloud computing model have resulted in unprecedented paradigm shifts in cloud application development. Many applications have found data driven cloud computing models indispensable due to the need for high performance computing. Performance prediction is essential for both cloud service providers and users. Providers rely on accurate predictions to manage resources effectively, prevent over-provisioning or under-provisioning, and maintain service-level agreements (SLAs). Users, on the other hand, benefit from performance prediction when selecting cloud services that meet their application requirements. Inadequate performance prediction can lead to increased operational costs, degraded service quality, and customer dissatisfaction. Thus, robust prediction mechanisms are indispensable in ensuring the efficient operation of cloud systems. This work presents a regression learning based model for performance prediction in cloud environments. In this research work, an optimized regression learning model has been developed based on the Bayesian Momentum Based Regularization approach. The essence of this approach is the fact that the model damps the gradient oscillations transverse to the convergence plane and accelerates the gradient momentum along the convergence plane. A comparative analysis with existing work shows that the proposed model obtains improved performance than existing research in the domain.

**Index Terms -** Cloud Computing, service-level agreements (SLAs). Regression Learning, Neural Network, Bayesian Momentum Based Regularization Mean Absolute Percentage Error.

## I. INTRODUCTION

Cloud computing has emerged as a transformative paradigm in modern information technology, enabling organizations to access computing resources over the internet without owning or managing physical infrastructure. Instead of deploying local servers or storage systems, users can leverage remote datacenters operated by cloud service providers [1]. This shift offers significant advantages, including cost savings, rapid scalability, global accessibility, and reduced management overhead. Cloud computing fundamentally changes how individuals, enterprises, and governments build, deploy, and maintain digital services, making it a key enabler of digital transformation [2].

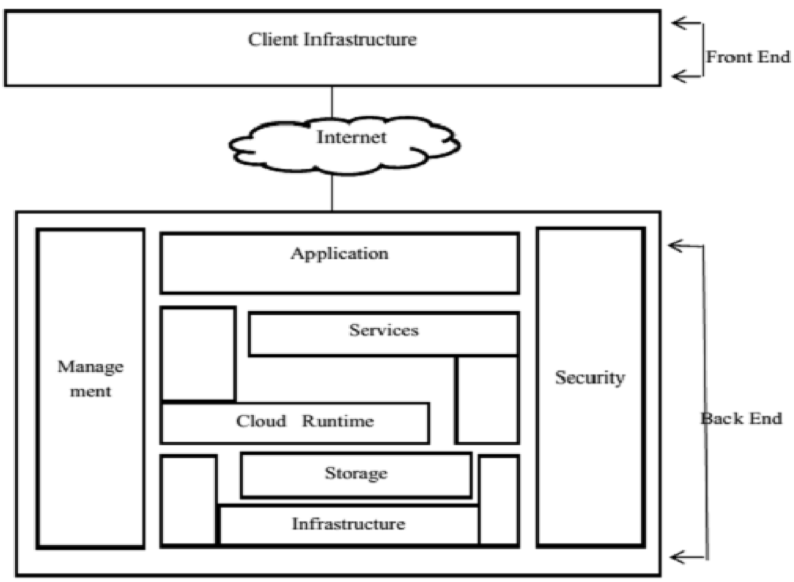


Fig.1. The Cloud Computing Architecture

Figure 1 presents the cloud computing architecture. One of the core motivations behind cloud computing is its ability to provide on-demand resources with minimal management effort. Users can provision computing power, storage, or applications in minutes rather than weeks, allowing businesses to innovate faster and respond to market dynamics with agility [3]. Cloud infrastructures are also designed with built-in redundancy and failover capabilities, enhancing service reliability and minimizing downtime. The pay-as-you-go billing approach further allows organizations to optimize costs by paying only for the resources they consume [4]. The foundation of cloud computing lies in its service delivery models, which categorize how resources are packaged and delivered. The three primary cloud service models—Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS)—offer varying degrees of control, flexibility, and management responsibility. This layered model ensures that different users, from developers to business end-users, can select the appropriate level of abstraction and convenience to meet their specific needs [5]

## II. NEED OF THE STUDY

The need for the presents study stems from the fact that different cloud services are needed by different clients. Typically, cloud services are categorized as:

### 2.1 Cloud Services

Infrastructure as a Service (IaaS) is the most fundamental level, offering virtualized computing resources such as virtual machines, storage, and networking [6]. It gives users maximum control over the operating system, runtime environment, and applications while offloading the hardware management to the cloud provider. IaaS is suitable for organizations that require customizable infrastructure or wish to migrate legacy applications to the cloud. Popular examples include Amazon EC2, Google Compute Engine, and Microsoft Azure Virtual Machines [7]

Platform as a Service (PaaS) offers a higher level of abstraction by providing a managed environment for application development, deployment, and scaling. Developers can focus on writing code while the provider handles infrastructure provisioning, load balancing, and runtime management. PaaS is ideal for rapid application development and DevOps workflows, helping teams reduce complexity and accelerate time-to-market. Services like Google App Engine, AWS Elastic Beanstalk, and Azure App Service exemplify this model [8].

Software as a Service (SaaS) delivers fully managed applications accessible through web browsers or mobile apps. Users do not need to install or maintain software; instead, updates, security, and data management are overseen by the provider. SaaS is widely used in business productivity, customer relationship management, and collaboration tools. Common examples include Google Workspace, Microsoft 365, Salesforce, and Dropbox. This model significantly reduces operational effort and ensures users always have access to the latest features [9].

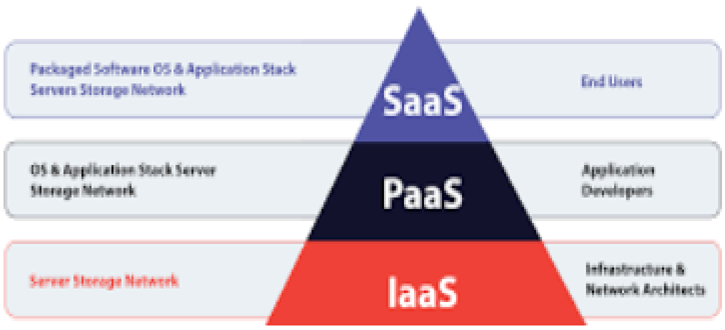


Fig.2. Different Cloud Services

Figure 2 presents the different cloud services. In addition to service models, cloud computing also defines deployment models such as public, private, hybrid, and multi-cloud. Public clouds offer shared infrastructure accessible to multiple users, while private clouds provide dedicated environments for organizations requiring strict security and customization. Hybrid cloud combines both models, allowing data and applications to move seamlessly between environments, whereas multi-cloud leverages multiple providers to avoid vendor lock-in and enhance resilience [10].

### 2.2 Cloud Performance

Cloud environments are inherently dynamic, with fluctuating workloads, shared resource usage, and complex virtualization layers. These uncertainties make it difficult to guarantee predictable performance, thereby creating a strong need for accurate cloud performance prediction mechanisms. At the core of this need is the variability of cloud resources. In cloud infrastructures, multiple tenants share CPU cores, memory, storage, and network bandwidth. This multi-tenancy often leads to performance interference, where one user's workload affects another's performance. Moreover, providers deploy load balancing and auto-scaling algorithms that constantly reallocate resources based on demand. Without performance prediction, users struggle to understand how their applications will behave under different resource allocations, leading to suboptimal decisions during deployment [11].

Cloud performance prediction is also essential for cost optimization. Since most cloud services operate on a pay-as-you-go model, users must carefully select the right combination of virtual machine types, storage classes, and network configurations. Over-provisioning resources can lead to unnecessary spending, while under-provisioning may cause degraded application performance or service outages [12]. Predictive models help estimate the performance-to-cost ratio of various configurations, enabling organizations to choose the most economical options without compromising quality of service (QoS). Another important driver is the need to guarantee Service Level Agreements (SLAs). Businesses that deliver services on the cloud must meet specific latency, throughput, and availability requirements defined by SLAs. Violations can lead to financial penalties, reputational damage, and user dissatisfaction. By predicting performance ahead of time, organizations can proactively detect potential bottlenecks, allocate sufficient resources, and maintain compliance with SLAs even under peak workloads [13]

Cloud performance prediction is also crucial for scaling applications efficiently. Auto-scaling mechanisms depend on accurate forecasts of future workload demands. If the scaling decision is delayed or inaccurate, it may result in performance drops, increased response times, or sudden spikes in resource usage. Predictive models based on machine learning or statistical analysis allow systems to anticipate workload patterns, enabling proactive scaling and more efficient resource management across distributed cloud environments [14]

## III. RESEARCH METHODOLOGY

The methodology of the proposed work is presented in this section.

### 3.1 Machine Learning Models for Estimating Cloud Performance

Cloud performance prediction is essential for maintaining reliability, optimizing costs, meeting SLAs, ensuring efficient scaling, and supporting workload migration. As cloud environments continue to grow in complexity, predictive models powered by machine learning, performance monitoring, and data analytics will become increasingly vital. They enable businesses to navigate the uncertainty of cloud systems, make informed decisions, and deliver high-quality services in a rapidly evolving digital landscape. Modern cloud environments are complex, dynamic, and highly heterogeneous, making traditional rule-based performance management insufficient. To address this challenge, machine learning (ML) and deep learning (DL) models have emerged as powerful tools for predicting cloud performance with high accuracy. These models learn patterns from historical workload data, resource metrics, and system behavior, enabling proactive decision-making and automated optimization. As cloud systems scale and diversify, data-driven performance prediction has become essential for ensuring reliability, efficiency, and cost-effectiveness [15].

Machine learning models provide foundational capabilities for predicting key performance indicators (KPIs) such as response time, CPU utilization, network latency, throughput, and energy consumption. Techniques like linear regression, decision trees, random forests, and support vector machines (SVMs) are widely used for modeling cloud workloads because they handle diverse features and nonlinear relationships. These models can analyze past performance data to forecast resource demands, detect bottlenecks, and recommend optimal configurations. Their interpretability also helps administrators understand which factors influence application performance the most [16]. As cloud environments generate large volumes of time-series and high-dimensional data, deep learning models offer improved prediction accuracy by capturing complex temporal and spatial patterns. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks excel in modeling sequential cloud metrics such as CPU load or network traffic [17]. They are particularly effective for forecasting workloads that follow periodic or unpredictable trends. Convolutional Neural Networks (CNNs) are also used to extract features from resource usage matrices or system logs, improving prediction performance in multi-metric analysis [18]

More advanced architectures like Deep Reinforcement Learning (DRL) extend cloud performance prediction into intelligent decision-making. DRL agents learn optimal actions—such as when to scale instances or migrate workloads—by interacting with the cloud environment. This makes them suitable for dynamic autoscaling, energy-aware scheduling, and QoS optimization [19]. Techniques like Deep Q-Networks (DQNs) and policy gradient methods allow cloud systems to autonomously adapt to workload variations while maintaining performance guarantees. In multi-cloud and heterogeneous environments, ML and DL models support resource allocation and workload placement by predicting how applications will behave across different platforms. Ensemble learning techniques, such as gradient boosting and stacking, combine multiple models to improve reliability and reduce prediction errors. Meanwhile, unsupervised learning methods like clustering and anomaly detection identify abnormal performance patterns, enabling early diagnosis of failures or SLA violations [20]. The integration of ML and DL models into cloud monitoring frameworks enhances cost optimization, energy efficiency, and fault tolerance. Predictive autoscaling minimizes over-provisioning by allocating resources only when required, while accurate forecasting prevents under-provisioning that could degrade application performance. Deep learning models also help balance performance and operational expenditures by identifying the most cost-effective VM types or storage configurations for specific load conditions [21]

### 3.2 Proposed Algorithm

In the proposed approach the neural network model is used. In this approach the Bayesian Momentum Based optimization is used to train the neural network model. Predicting these metrics accurately is essential for autoscaling, SLA management, and resource allocation. Traditional regression techniques often struggle with the stochastic and dynamic nature of cloud workloads, especially when data exhibits noise, sudden bursts, or nonlinear dependencies. A Bayesian momentum-based approach provides an efficient and probabilistically grounded solution, improving both prediction accuracy and uncertainty estimation for cloud performance analysis. At the core of this method is Bayesian regression, which models cloud performance metrics as random variables with prior probability distributions. Instead of producing fixed parameter estimates, Bayesian regression generates posterior distributions that represent uncertainty in the model. This is particularly valuable in cloud environments where workload demand fluctuates unpredictably. The Bayesian framework allows the incorporation of prior knowledge—such as historical performance trends or resource behavior—which refines predictions and improves robustness against noisy or incomplete data. The weights of the network are updated such that the condition for maximization is satisfied of a new sample bearing a conditional probability defined as [22]:

$$\boldsymbol{P}\left(\frac{\boldsymbol{X}}{\boldsymbol{X_{i},k_1,k_2,M}}\right)=\frac{\boldsymbol{P}\left(\frac{\boldsymbol{X_i}}{\boldsymbol{X,k_2,M}}\right)\boldsymbol{P}\left(\frac{\boldsymbol{X_i}}{\boldsymbol{k_1,M}}\right)}{\boldsymbol{P}\left(\frac{\boldsymbol{X}}{\boldsymbol{k_1,k_2,M}}\right)} \quad \textbf{(1)}$$

Here,

$P$ denotes the probability of occurrence of an event.

$X_i$ denotes the vector corresponding to the bias and weight values of the network.

$X$ denotes the training data set

The training rule for the approach is based on the Bayes theorem of conditional probability which is effective training, based on a penalty $\boldsymbol{\rho}=\frac{\boldsymbol{\mu}}{\boldsymbol{v}}$ . The weights are updated based on the modified regularized cost function:

$$F(w) = \mu w^T w + v[\frac{1}{n}\sum_{i=1}^{n}(p_i - a_i)^2] \tag{2}$$

If $(\pi \ll v)$: Network error are generally low.

else if $(\pi \geq v)$: Network errors tend to increase, in which case the weight magnitude should be reduced so as to limit errors (Penalty).

To optimize the regression parameters efficiently, the approach incorporates a momentum-based gradient update mechanism. In traditional gradient descent, parameter updates can oscillate when the loss surface is irregular, as is common in high-dimensional cloud performance data. Momentum reduces these oscillations by accumulating a moving average of past gradients, enabling smoother and faster convergence. When combined with Bayesian inference, momentum helps explore the posterior distribution more effectively, reducing computational overhead while preserving probabilistic accuracy. The Bayesian momentum-based algorithm typically uses stochastic gradient updates to approximate the posterior, similar to methods like Stochastic Gradient Langevin Dynamics (SGLD). However, adding momentum enhances sampling efficiency, especially when dealing with complex likelihood surfaces common in multi-metric cloud datasets. The momentum term boosts the parameter updates in consistent gradient directions and dampens randomness caused by noisy performance measurements. This ability to stabilize posterior sampling makes the approach well-suited for real-time cloud performance prediction. A key advantage of this technique is the ability to quantify prediction uncertainty, which is crucial for cloud resource management. Performance predictions are not just point estimates but distributions, allowing system controllers to assess the probability of SLA violations or resource saturation. For example, if the posterior variance for predicted CPU utilization is high, the autoscaler may proactively allocate additional VMs to prevent overload. This uncertainty-aware decision-making leads to more reliable and risk-sensitive cloud operations [23].

**Algorithm:**

**Start**
**{**
**Step.1** Identify benchmark datasets with cloud attributes.

**Step.2** Demarcate independent and dependent variables.

**Step.3** Split data into training and testing samples, and apply data filtration.

**Step.4** Initialize weights randomly and start training.

**Step.5** Update weights as:

$$\mathbf{w_{k+1} = w_k - \left[J_k J_k^T + \mu I\right]^{-1} J_k^T e_k}$$

**Step.6** Check for condition:

(Iterations $==$ Max. iterations or cost function stabilizes)
{
Truncate Training
}
else
{
Continue retraining as per step 5

**Step.7** Compute performance metrics.
**}**
**Stop.**

The next section presents the experimental results obtained.

## IV. RESULTS AND DISCUSSION

The results have been simulated using MATLAB.

| | vm_id | timestamp | cpu_usage | memory_usage | network_traffic | power_consun | num_executed_in | execution_time | energy_efficiency | task_type | task_priori | task_status |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | c5215826- | 25-01-2023 09:10 | 54.88135039 | 78.95086102 | 164.7759727 | 287.8089861 | 7527 | 69.34557502 | 0.553589014 | network | medium | waiting |
| 3 | 29690bc6- | 26-01-2023 04:46 | 71.51893664 | 29.90188259 | | 362.2735688 | 5348 | 41.39603965 | 0.349856212 | io | high | completed |
| 4 | 2e55abc3- | 13-01-2023 23:39 | | 92.70919512 | 203.6748471 | 231.4679027 | 5483 | 24.60254888 | 0.796276974 | io | medium | completed |
| 5 | e672e32f- | 09-02-2023 11:45 | 54.4883183 | 88.10095963 | | 195.6399538 | 5876 | 16.45666986 | 0.529510782 | compute | high | completed |
| 6 | f38b8b50- | 14-06-2023 08:27 | 42.36547993 | | | 359.451537 | 3361 | 55.30799162 | 0.351907264 | | medium | waiting |
| 7 | ad14d5d1- | 06-02-2023 16:44 | 64.58941131 | 62.00801573 | 580.5698365 | 115.9316105 | 4766 | 63.13767144 | | io | low | completed |
| 8 | 5651505d- | 15-06-2023 18:04 | 43.75872113 | 22.45642925 | 429.1396396 | 272.9604167 | 9008 | 60.15390382 | 0.461246388 | compute | high | completed |
| 9 | e5970e5c-0350-4131-bd6c-4bd7e2c98df1 | | | 85.43815472 | 685.2828128 | | | 14.19135553 | 0.242538428 | network | high | running |
| 10 | 7fc95f4c-f | 20-05-2023 19:09 | 96.36627605 | 4.388441255 | 902.8277169 | 367.9790559 | 9984 | 21.87669373 | | io | medium | running |
| 11 | 7fd9a664- | 10-07-2023 11:32 | 38.34415188 | 16.44188144 | 779.7913446 | 382.7566963 | 2989 | 42.16084672 | 0.139187233 | io | medium | running |
| 12 | ea8455b0- | 14-01-2023 19:59 | 79.17250381 | 2.972252292 | 926.3740297 | 173.5593254 | 8644 | 55.70267258 | 0.779498827 | compute | medium | completed |
| 13 | eb9c26c6-d99f-40a4-8640-7631ad451034 | | | 20.07657809 | 941.0693854 | 435.0137737 | 5185 | 66.37989278 | 0.700144664 | | medium | completed |
| 14 | 0ddae1eb- | 11-04-2023 22:45 | 56.80445611 | 2.355626169 | 722.552146 | 143.3403992 | 9788 | 79.70075631 | 0.944387475 | io | medium | waiting |
| 15 | 196e5fec- | 20-04-2023 06:07 | 92.55966383 | | 34.3785391 | 433.958848 | 886 | 54.13118611 | 0.204644266 | network | | waiting |
| 16 | bb9d0b5b- | 22-04-2023 16:00 | 7.10360582 | 96.51663852 | 919.1721699 | 275.6299116 | 9117 | 39.96762536 | 0.854724947 | io | low | running |
| 17 | 194210b1- | 07-03-2023 01:40 | | 44.40411245 | 664.6856505 | 245.0042472 | 7358 | 86.68051052 | 0.860580676 | io | medium | waiting |
| 18 | 3b85d1e1- | 07-02-2023 19:00 | 2.021839744 | 89.33690236 | 208.4192736 | 199.2585076 | 1224 | 61.85013513 | 0.697445192 | compute | high | completed |
| 19 | | 28-06-2023 06:55 | 83.26198455 | 41.71619077 | 493.1490075 | 103.741907 | 3187 | 5.677628555 | | compute | high | |
| 20 | 00a10e10- | 12-04-2023 05:05 | 77.81567509 | | 887.4445039 | | 2010 | 32.82460714 | 0.578290513 | io | low | |
| 21 | fd28310b- | 18-01-2023 11:22 | 87.00121482 | 46.62017593 | 437.1881151 | 77.05470159 | 9427 | | 0.207164345 | | low | waiting |
| 22 | f65e1c9b- | 18-06-2023 16:06 | 97.86183422 | 70.71200063 | | 264.7233293 | 3345 | 89.00994248 | 0.482367481 | io | low | waiting |
| 23 | 5da32ae8- | 07-02-2023 17:21 | 79.91585642 | 76.10323679 | | 361.6567231 | 9825 | 52.18702939 | 0.520830534 | | medium | running |
| 24 | 8ca4165d- | 01-02-2023 07:27 | 46.14793623 | | 88.57367902 | 348.3596744 | 1912 | 58.10186507 | 0.42384871 | io | low | running |
| 25 | 74d08dd7- | 13-05-2023 18:43 | 78.05291763 | 75.0869324 | 387.449284 | 288.4658048 | 9174 | 50.00729168 | | compute | low | waiting |
| 26 | ee09578f- | 19-07-2023 04:48 | 11.82744259 | 17.4946851 | 433.6812055 | 214.9120962 | 1147 | 12.59030733 | 0.107757509 | compute | medium | running |
| 27 | d95eef15- | 25-02-2023 05:22 | 63.99210213 | 87.71007666 | 940.2623028 | | 6643 | | 0.166331988 | compute | low | running |
| 28 | b07941d5- | 09-07-2023 22:40 | 14.33532874 | 4.481789361 | 296.7721211 | 386.128954 | | 5.859939367 | 0.633418682 | network | high | completed |

vmCloud_data

Fig. 3. Raw Data

Figure above shows the raw data which is used for the simulation.

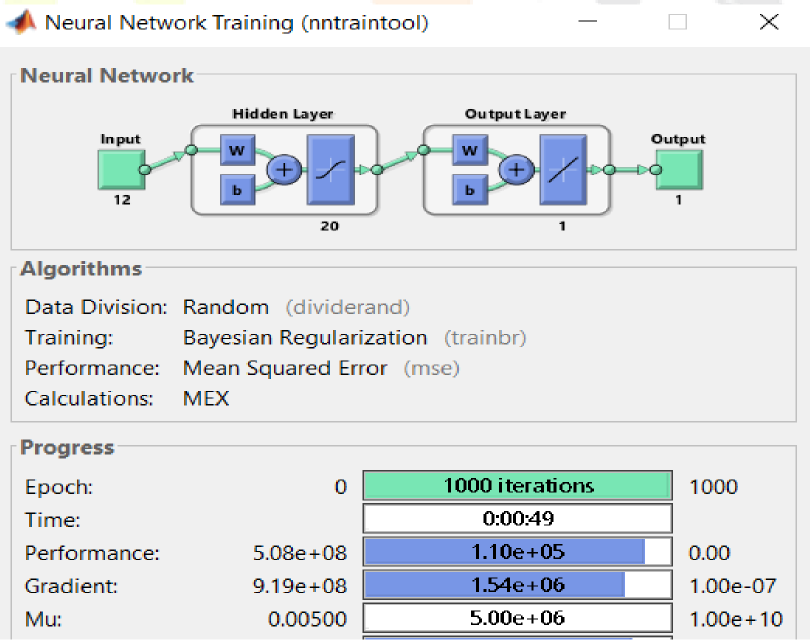


Fig. 4. Neural Network Model

Figure above shows the design of the neural network which is used for pattern recognition. It can be observed that the proposed model has 12 neurons in the input layer and 1 neuron in the output layer.

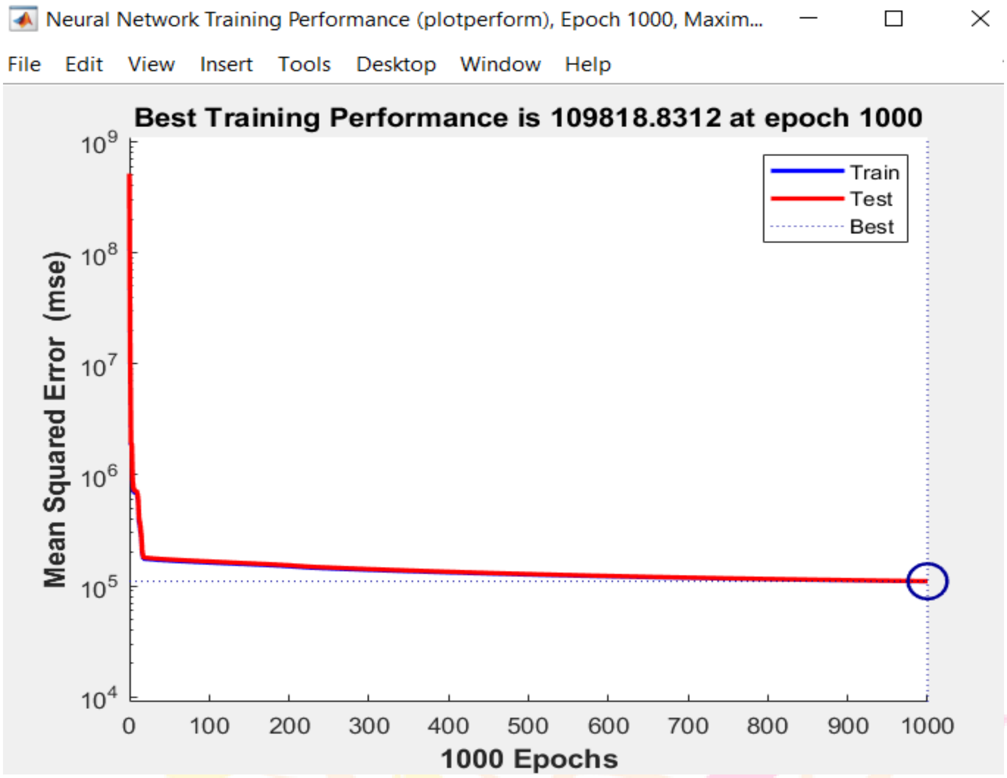


Fig. 5. Training Convergence

Figure above shows the training convergence for the model designed. It can be observed that the training converges at 1000 iterations without any spikes in the cost function exhibiting the fact that the model is stable.

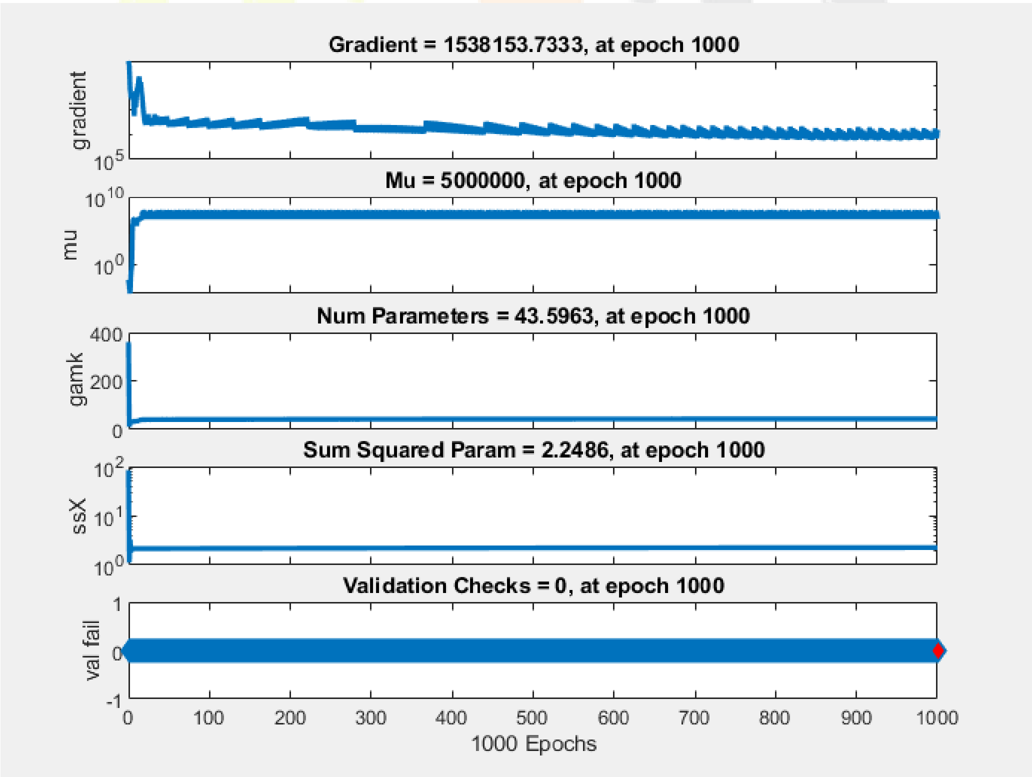


Fig. 5. Training States

Figure above shows the training states for the models and variation of important parameters such as gradient, learning rate, sum squared parameters and validation checks.

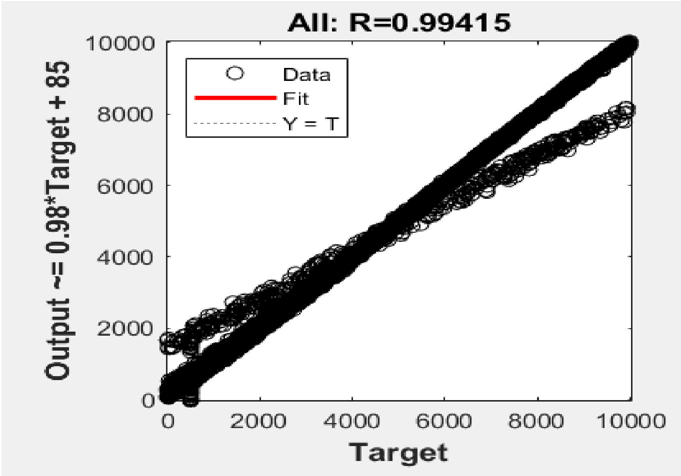


Fig. 6. Model Regression

Figure above shows the overall model regression which is 0.99415 showing close similarity between the predicted and actual values of the variable.

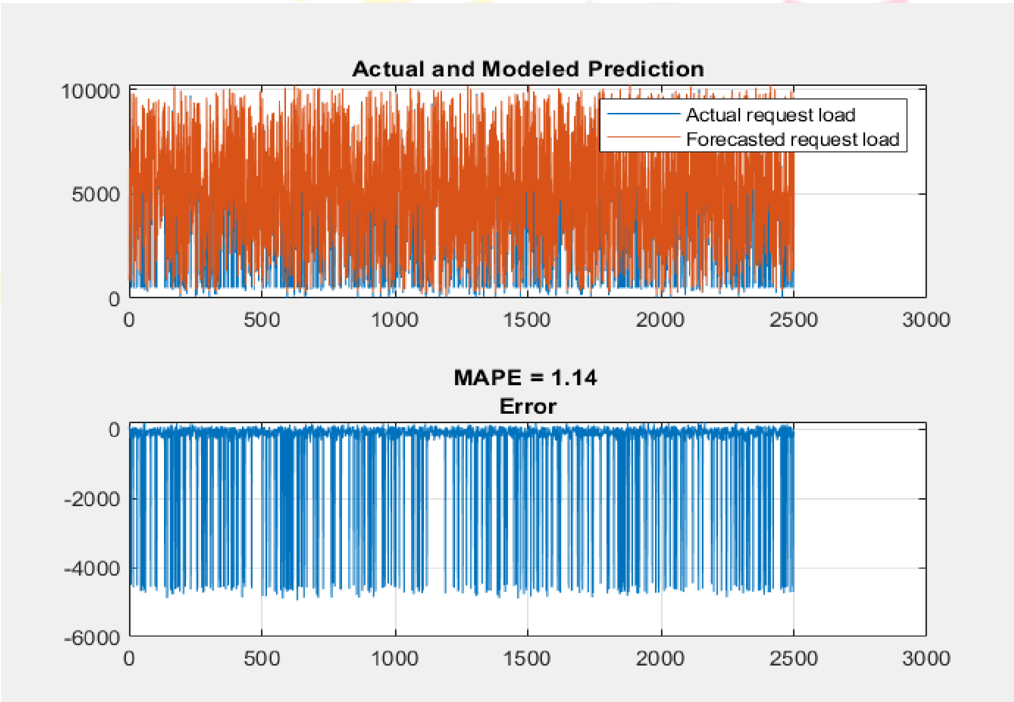


Fig. 7. Model Regression

Figure above shows the MAPE of the designed model which is 1.14% rendering an accuracy of 98.86%.

Table 1 Summary of Results

| S.No. | Parameter | Value |
|---|---|---|
| 1 | Model | Neural Networks |
| 2 | Algorithm | Bayesian Optimization |
| 3 | Iterations | 1000 |
| 4 | Regression | 0.99 |
| 5 | MAPE (Previous Work) Ref [11]: | 6.8% |
| **8.** | **MAPE (Proposed Approach)** | **1.14%** |

The summary of results obtained is presented in table 1. It can be observed that the proposed approach attains improved MAPE % compared to existing work in the domain.

## IV. Conclusion

As more applications resort to cloud services due to increasing data size and more computational complexity due to rising paradigms such as machine learning and big data, performance prediction in cloud platforms will become even more crucial. Presently, cloud computing has become the backbone of modern IT infrastructure, supporting applications ranging from data analytics to artificial intelligence. However, the dynamic and distributed nature of cloud environments introduces significant uncertainty in performance metrics such as latency, throughput, and resource utilization. Accurate machine learning (ML) models are therefore essential for predicting cloud performance and ensuring optimal resource management. In this work, a Bayesian Momentum Based Optimization or Regularization neural network model has been developed for cloud performance prediction in which the dependent variable has been chosen as the number of instructions executed successfully. The overall percentage error is significantly improved compared to previous research in the domain.

## References


1. F. Qiu, B. Zhang and J. Guo, "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), Shanghai, China, 2016, pp. 319-324.
2. D. Saxena and A. K. Singh, "A High Availability Management Model Based on VM Significance Ranking and Resource Estimation for Cloud Applications," in IEEE Transactions on Services Computing, vol. 16, no. 3, pp. 1604-1615, 1 May-June 2023.
3. R Keller, L Häfner, T Sachs, G Fridgen," Scheduling flexible demand in cloud computing spot markets: A real options approach", Business and Information Systems Engineering, Springer 2020, vol.62., pp. 25–39.
4. D. Kong, S. Liu and L. Pan, "Amazon Spot Instance Price Prediction with GRU Network," 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Dalian, China, 2021, pp. 31-36.
5. D. Katayama, K. Kasai and T. Koita, "Migration Destination Selection Algorithm for Spot Instances using SPS," 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan, 2022, pp. 6690-6692
6. A. C. Zhou, J. Lao, Z. Ke, Y. Wang and R. Mao, "FarSpot: Optimizing Monetary Cost for HPC Applications in the Cloud Spot Market," in IEEE Transactions on Parallel and Distributed Systems, 2021, vol. 33, no. 11, pp. 2955-2967
7. M. Kandpal, M. Gahlawat and K. Patel, "Role of predictive modeling in cloud services pricing: A survey" 2017 7th IEEE International Conference on Cloud Computing, Data Science & Engineering - Confluence, Noida, 2017, pp. 249-254.
8. Z Li, H Zhang, L O'Brien, S Jiang, Y Zhou, M Kihl and R. Ranjan, "Spot pricing in the Cloud ecosystem: A comparative investigation", Journal of Systems and Software, Elsevier 2018, Vol-14, pp: 1-19.
9. V Chittora and CP Gupta, "Dynamic Spot Price Forecasting Using Stacked LSTM Networks", Proceedings in 3rd IEEE International Conference on Intelligent Sustainable Systems (ICISS), India, 2020, pp. 1080-1085.
10. Z. Amekraz and M. Youssef, "Prediction of Amazon spot price based on chaos theory using ANFIS model" 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), Agadir, 2016, pp. 1-6.
11. D. Huang, L. Costero, A. Pahlevan, M. Zapater and D. Atienza, "CloudProphet: A Machine Learning-Based Performance Prediction for Public Clouds," in IEEE Transactions on Sustainable Computing, 2024, vol. 9, no. 4, pp. 661-676..
12. D Saxena, AK Singh, "Auto-adaptive learning-based workload forecasting in dynamic cloud environment", International Journal of Computers and Applications, Taylor and Francis 2022, vol.44. no.6., pp.541-551.
13. Z. Amekraz and M. Y. Hadi, "CANFIS: A Chaos Adaptive Neural Fuzzy Inference System for Workload Prediction in the Cloud," in IEEE Access, 2022, vol. 10, pp. 49808-49828
14. A. C. Zhou, J. Lao, Z. Ke, Y. Wang and R. Mao, "FarSpot: Optimizing Monetary Cost for HPC Applications in the Cloud Spot Market," in IEEE Transactions on Parallel and Distributed Systems, 2021, vol. 33, no. 11, pp. 2955-2967

15. G. J. Portella, E. Nakano, G. N. Rodrigues, A. Boukerche and A. C. M. A. Melo, "A Novel Statistical and Neural Network Combined Approach for the Cloud Spot Market," in IEEE Transactions on Cloud Computing, 2023 vol. 11, no. 1, pp. 278-290.
16. 6. Liu D, Cai Z, Lu Y (2019) Spot price prediction based dynamic resource scheduling for web applications. In: 2019 Seventh International Conference on Advanced Cloud and Big Data (CBD). IEEE, pp 78–83 7.
17. Varshney P, Simmhan Y (2019) AutoBot: Resilient and cost-effective scheduling of a bag of tasks on spot VMs. IEEE Trans Parallel Distrib Syst 30(7):1512-1527
18. Sharma P, Lee S, Guo T, Irwin D, Shenoy P (2017) Managing risk in a derivative IaaS cloud. IEEE Trans Parallel Distrib Syst 29(8):1750-1765
19. Mishra AK, Yadav DK (2017) Analysis and prediction of Amazon EC2 spot instance prices. Int J Appl Eng Res 12(21):11205– 11212
20. Teylo L, Arantes L, Sens P, Drummond LM (2019) A bag-of-tasks scheduler tolerant to temporal failures in clouds. In: 2019 31st International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD). IEEE, pp. 144–151
21. Khandelwal V, Chaturvedi AK, Gupta CP (2020) Amazon EC2 spot price prediction using regression random forests. IEEE Trans Cloud Comput 8(1):59–72
22. Y. Huang, J. Sun and Y. Tian, "A Bayesian Optimization Method for Finding the Worst-Case Scenarios of Autonomous Vehicles," in IEEE Transactions on Intelligent Transportation Systems, 2024, vol. 26, no. 1, pp. 529-543
23. S. Amini, I. Vannieuwenhuyse and A. Morales-Hernández, "Constrained Bayesian Optimization: A Review," in IEEE Access, 2025, vol. 13, pp. 1581-1593.