International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

# Forecasting Cloud Requests Using Machine Learning

*Nosheen Pathan*[1], *Megha Jat*[2]

[1]**M.Tech Scholar (CSE), Department of CSE, PCST, Indore, India**
[2]**Assistant Professor (CSE), Department of CSE, PCST, Indore, India**

**ABSTRACT**

Cloud Computing has emerged as one of the most sought after fields in computer science. Several applications which need high computational complexity but cannot be performed on conventional hardware prefer to leverage cloud based platforms. Hence with increasing traffic and load on cloud servers or cloud based platforms, there seems to be a natural need for cloud workload prediction so as to estimate and manage cloud based resources. The present paper presents a neural network based approach for cloud workload prediction. The proposed model uses the PolakRebiere algorithm for workload prediction. It has been shown that the proposed technique outperforms previously existing technique [1].The performance evaluation parameters have been chosen as mean absolute percentage error (MAPE) and regression. It has been found that the proposed system attains an MAPE of 3.65% only. The number of hidden layers taken is 10.

Keywords—Cloud Workload Prediction, Artificial Neural Network (ANN), Polak Rebiere Restarts Algorithm, Mean Absolute Percentage Error (MAPE).

## I. Introduction

Cloud Computing has revolutionized computational technology with cloud based platforms catering to the needs of systems unable to run complex processes on available hardware. Cloud computing has drastically transformed the means of computing in recent years. In spite of numerous benefits, it suffers from some challenges too.[2] Major challenges of cloud computing include dynamic resource scaling and power consumption. These factors lead a cloud system to become inefficient and costly. The workload prediction is one of the variables by which the efficiency and operational cost of a cloud can be improved. However, the data being large and complex needs the aid of Artificial Intelligence for the prediction for the prediction purpose.[4]

## II. ARTIFICIAL NEURAL NETWORKS

The architecture of artificial intelligence can be practically implemented by designing artificial neural networks. The biological-mathematical counterpart of artificial neural networks has been shown below.

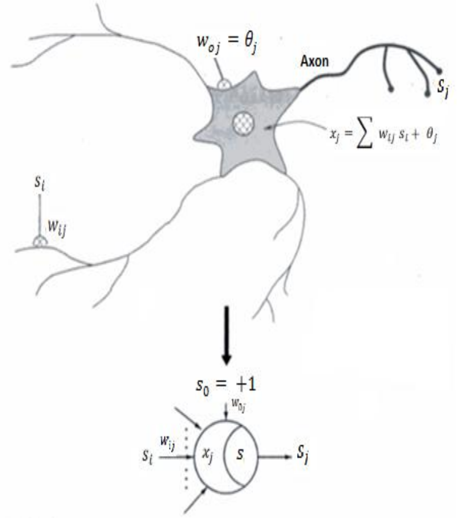


**Fig.1 Biological-Mathematical Counterpart of ANN**

The mathematical conversion of the ANN can be done by analyzing the biological structure of ANN. In the above example, the enunciated properties of the ANN that have been emphasized upon are:

**Strength to process information in parallel way.**

**The power to grasp and learn from weights**

**Searching for patterned sets in complex models of data.**

To see how the ANN really works, a mathematical model has been devised here, to indicate the functions mathematically.[7]. Here it is to be noted that the inputs of information parallel goes on into the input layer as specified whereas the end result analysis is marked from the output layer.

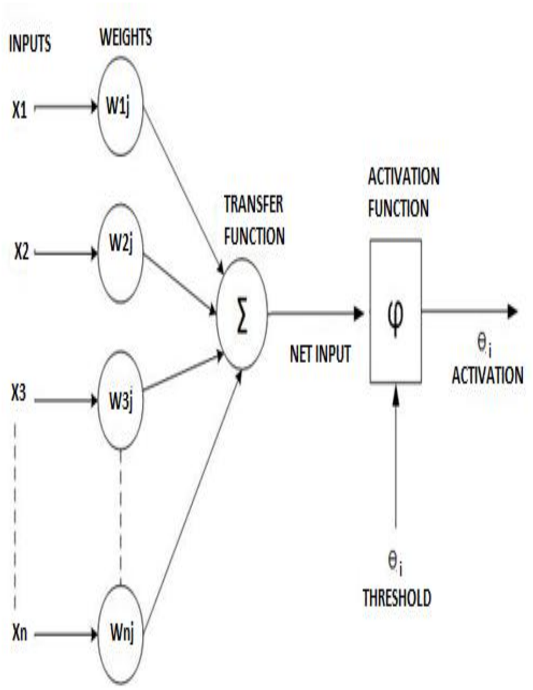


**Fig.2 Mathematical Modeling of ANN**

The figure above illustrates the ANN mathematical model.

The feature of parallel acceptance and processing of data by the neural network serves a vital role. This ensures efficient and quicker mode of operation by the neural network. Also adding to it, the power to learn and adapt flexibly by the neural network aids in processing of data at a faster speed. **[2]**These great features and attributes make the ANN self dependent without requiring much intervention from humans. The ANN output can be put forth like:

$$\boldsymbol{Y} = \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{X_i} . \mathbf{W_i} \; + \; \boldsymbol{\theta_i} \qquad \mathbf{(1)}$$

Here,

Output by ANN marked by y

x signifies the inputs to the ANN

The weights of the ANN shown by w

ϕ denotes the bias.

Training of ANN is of major importance before it can be used to predict the outcome of the data inputs.

## III. POLAK REBIERE RESTARTS ALGORITHM

The Polak-Rebiere algorithms, is considered as one class of optimization methods, are much more efficient than gradient descent algorithms (GDAs) having a low memory requirement and providing fast convergence. Also, its practical for minimizing functions of very many variables since its space complexity is relatively less.
The training rule for the Polak-Rebiere algorithm is given below:

$$\boldsymbol{p_0 = -g_0}\textbf{(2)}$$

Where,

$p_0$represents the negative gradient given by $\frac{\partial e}{\partial w}$

The search direction vector $p_k$ for iteration k (representing adaptive learning rate gradient descent) is given by:

$$\boldsymbol{p_k = -g_k + \beta_k p_{k-1}}\ \textbf{(3)}$$

The constant $\beta_k$is computed by:

$$\boldsymbol{\beta_{k-1} = \frac{g_k^T + g_k}{g_{k-1}^T + g_{k-1}}} \qquad \textbf{(4)}$$

The weight adaptation is given by:

$$\boldsymbol{w_{k+1} = w_k + \beta_k p_k}\textbf{(5)}$$

Here,

$w_{k+1}$is the weight of the next iteration

$w_k$is the weight of the present iteration

The activation function used by the Polak-Rebiere algorithm is the tan-sig function mathematically defined as:

$$\boldsymbol{tansig(x) = \frac{2}{1+e^{-2x}} - 1} \qquad \textbf{(6)}$$

ANN, which has a powerful connection between the input and output variables, is a mathematical model that reflects learning and generalization ability of human neural architecture. ANNs consist of the input layer and the output layer, furthermore, the layer(s) between input and output layers are referred to hidden layer that may be one or more, helps to capture nonlinearity and is not directly observed. In theory, ANNs can be contained an arbitrary number of input and output variables. However, it must be noted that the number of variables and computational cost is entirely proportional. The number of neurons per layers, training algorithms, epochs, maximum training time, performance values, gradient, and validation checks can be set before training of an ANN.

## IV. System Design

This prevalent system of data utilizes data from the data sets of NASA and Saskatchewan server's data for different prediction intervals. The parameters used for training the neural network are:[1]

1) No. of servers
2) No. of uers
3) Response time
4) Deviation delay value
5) Cloud Storage value
6) Mean Deviation value
7) Job Queueing value
8) Number of Operational Nodes
9) No. of Requests

The data is then broadly divided into a training data set and another is the testing data set. 80% training & 20% testing; of the data is deployed for testing.
Following illustrate the performance metrics pertinent to the system designed based on the ANN topology:
1) Mean Absolute Percentage Error (MAPE)

$$MAPE = \frac{100}{M}\sum_{t=1}^{N}\frac{E-E_t|}{E_t} \qquad (7)$$

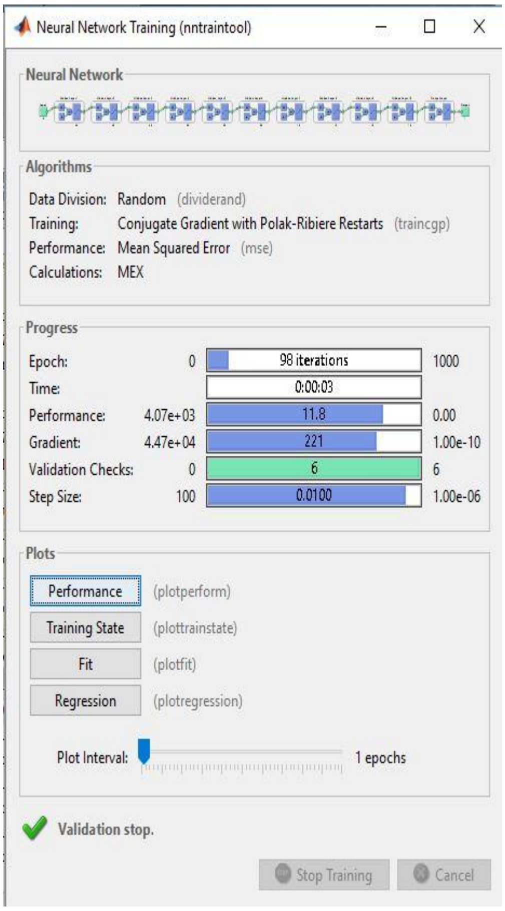
Here $E_t$and $E_t^{\sim}$ stand for the predicted and actual values respectively.
The number of predicted samples is indicated by M.

2) Regression
The amount of similarity between the predicted and actual value set is referred as Regression. The maximum regression value is 1 signifying complete similarity whereas the minimum value is 0 that shows no similarity.

## V. Results

The results have been evaluated based on the following parameters:

1. (MAPE)
2. Regression
3. MSE w.r.t. the number of epochs



Neural Network Training (nntraintool)

Neural Network

Algorithms

Data Division: Random (dividerand)
Training: Conjugate Gradient with Polak-Ribiere Restarts (traincgp)
Performance: Mean Squared Error (mse)
Calculations: MEX

Progress

| | | | |
|---|---|---|---|
| Epoch: | 0 | 98 iterations | 1000 |
| Time: | | 0:00:03 | |
| Performance: | 4.07e+03 | 11.8 | 0.00 |
| Gradient: | 4.47e+04 | 221 | 1.00e-10 |
| Validation Checks: | 0 | 6 | 6 |
| Step Size: | 100 | 0.0100 | 1.00e-06 |

Plots

Performance (plotperform)
Training State (plottrainstate)
Fit (plotfit)
Regression (plotregression)

Plot Interval: 1 epochs

✓ Validation stop.

**Fig.3 Designed ANN Structure**

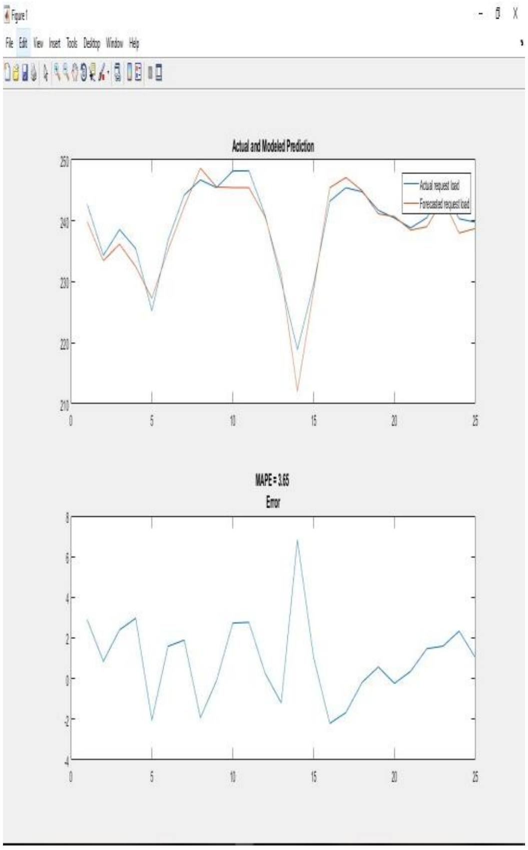


**Fig.4 Predicted and Actual Cloud Workload**

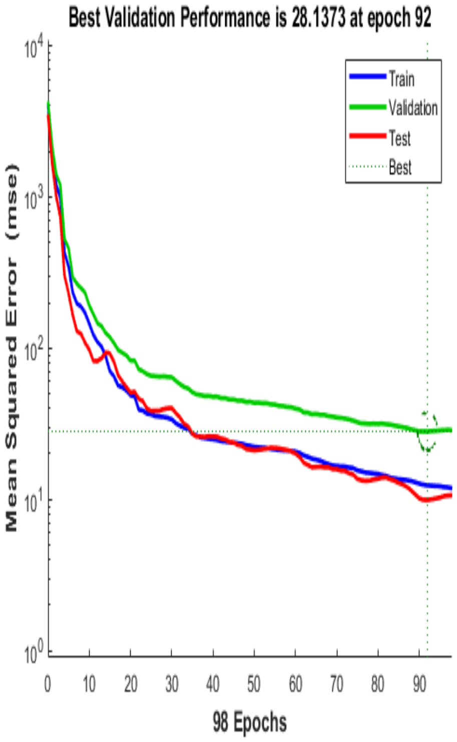


**Fig.5 Variation of MSE with respect to epochs**

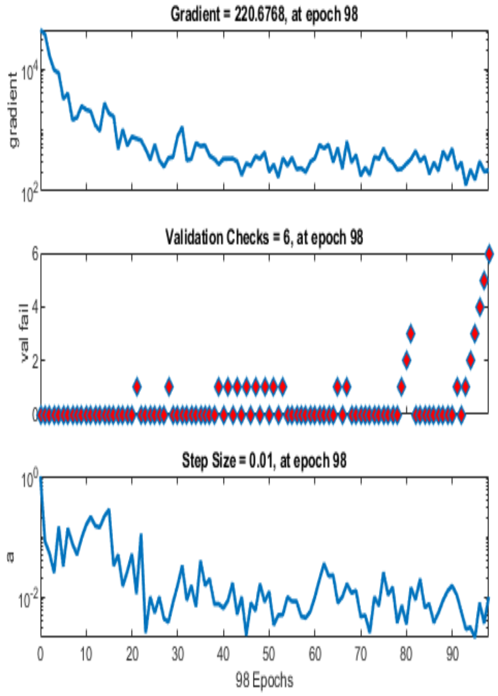


**Fig.6 Training Parameters**

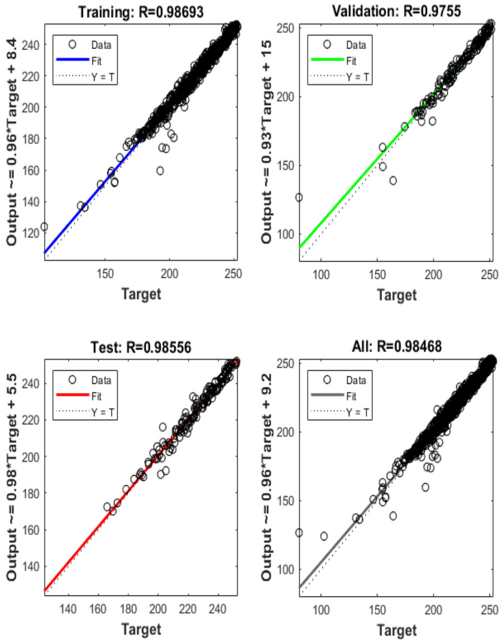


**Fig.7 Regression Analysis**

From the above figures, it can be concluded that the proposed system attains the following results:

1) MAPE of 3.65%

2) Regression of 0.98 (overall)

A comparison in terms of the prediction accuracy with respect to previous work is given below is given below:

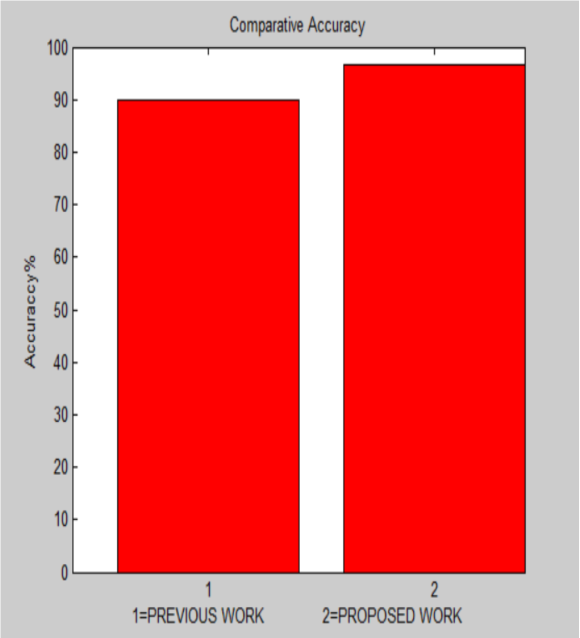


**Fig.8 Comparative Accuracy Analysis**

## Conclusion

The proposed work uses the Polak-Rebiere Restarts algorithm for cloud workload prediction. The structure of the neural network uses a 1-10-1 configuration. It has been shown that the proposed work attains a mean absolute percentage error of 3.65% only. This is significantly less than the previous work [1] which attains a mean absolute percentage error of 10.26%. The prediction mechanism can be useful for data centers using the cloud platform.

## REFERENCES

[1] P Yazdanian, S Sharifian, E2LG: a multiscale ensemble of LSTM/GAN deep learning architecture for multistep-ahead cloud workload prediction", Journal of Supercomputing, Springer 2021, vol. 77, pp.11052–11082.

[2] J. Gao, H. Wang and H. Shen, "Machine Learning Based Workload Prediction in Cloud Computing," 2020 29th International Conference on Computer Communications and Networks (ICCCN), 2020, pp. 1-9

[3] Z. Chen, J. Hu, G. Min, A. Y. Zomaya and T. El-Ghazawi, "Towards Accurate Prediction for High-Dimensional and Highly-Variable Cloud Workloads with Deep Learning," in IEEE Transactions on Parallel and Distributed Systems, 2020, vol. 31, no. 4, pp. 923-934.

[4] L. Wang and E. Gelenbe, "Adaptive Dispatching of Tasks in the Cloud," in IEEE Transactions on Cloud Computing, vol. 6, no. 1, pp. 33-45, 1 Jan.-March 2018

[5] Martin Duggan, Karl Mason, Jim Duggan, Enda Howley, Enda Barrett, "Predicting Host CPU Utilization in Cloud Computing using Recurrent Neural Networks", 2017 IEEE.

[6] Ning Liu, Zhe Li, Jielong Xu, Zhiyuan Xu, Sheng Lin, QinruQiu, Jian Tang, Yanzhi Wang, "A Hierarchical Framework of Cloud Resource Allocation and Power Management Using Deep Reinforcement Learning", 2017 IEEE.

[7] LiyunZuo, Shoubin Dong, Lei Shu, Senior Member, IEEE, Chunsheng Zhu, Student Member, IEEE, and Guangjie Han, Member, IEEE, "A Multiqueue Interlacing Peak Scheduling Method Based on Tasks' Classification in Cloud Computing", 2016 IEEE.

[8] Yazhou Hu, Bo Deng, Fuyang Peng and Dongxia Wang, "Workload Prediction for Cloud Computing Elasticity Mechanism", 2016 IEEE.

[9] Ji Xue, Feng Yan, Robert Birke, Lydia Y. Chen, Thomas Scherer, and EvgeniaSmirni, "PRACTISE: Robust Prediction of Data Center Time Series", 2015 IEEE.

[10] Mehmet Demirci, "A Survey of Machine Learning Applications for Energy-Efficient Resource Management in Cloud Computing Environments", 2015 IEEE.

[11] SherifAbdelwahab, Member, IEEE, BechirHamdaoui, Senior Member, IEEE, Mohsen Guizani, Fellow, IEEE, and Ammar Rayes, "Enabling Smart Cloud Services Through Remote Sensing: An Internet of Everything Enabler", 2014 IEEE.

[12] Chin-Feng Lai, Member, IEEE, Min Chen, Senior Member, IEEE, Jeng-Shyang Pan, Chan-Hyun Youn, Member, IEEE, and Han-Chieh Chao, Senior Member, IEEE, "A Collaborative Computing Framework of Cloud Network and WBSN Applied to Fall Detection and 3-D Motion Reconstruction", 2014 IEEE.

[13] Ian Davis, HadiHemmati, Ric Holt, Mike Godfrey, Douglas Neuse, Serge Mankovskii, “Storm Prediction in a Cloud”, 2013 IEEE.

[14] Abul Bashar, “Autonomic Scaling of Cloud Computing Resources using BN-based Prediction Models”, 2013 IEEE.

[15] Sadeka Islam , Jacky Keunga, Kevin Lee, Anna Liu, “Autonomic Scaling of Cloud Computing Resources using BN-based Prediction Models”, 2012 ELSEVIER.

[16] ErolGelenbe, Ricardo Lent and Markos Douratsos, “Choosing a Local or Remote Cloud”, 2012 IEEE.