A SELF-ADAPTIVE DEEP LEARNING-BASED MODEL TO PREDICT CLOUD WORKLOAD

K. Borna,* *R. Ghanbari**

Abstract: Predicting cloud workload is a problematic issue for cloud providers. Recent research has led us to a significant improvement in workload prediction. Although self-adaptive systems have an imperative impact on lowering the number of cloud resources, those still have to be more accurate, detailed and accelerated. A new self-adaptive technique based on a deep learning model to optimize and decrease the use of cloud resources is proposed. It is also demonstrated how to prognosticate incoming workload and how to manage available resources. The PlanetLab dataset in this research is used. The obtained results have been compared to other relevant designs. According to these comparisons with the state-of-the-art deep learning methods, our proposed model encompasses a better prediction efficiency and enhances productivity by 5%.

Key words: *cloud workload, deep learning, prediction, resource, self-adaptive*

Received: September 26, 2022 **DOI:** 10.14311/NNW.2023.33.010
Revised and accepted: June 30, 2023

1. Introduction

Determining the required resources for a cloud-based application helps providers enhance resource distribution and allocation. Due to the escalating growth of cloud computing services, there is a prevailing tendency for using cloud-native applications. Recently various resource management techniques in cloud computing have been introduced, developed and commercialized. In these methods, the application will allocate the CPU for a short period and as a result, it will reduce server maintenance costs. Running these type of methods in a large cluster of servers, need a robust management system. The particular reason for the circumstance is various situations need different environment variables and controlling a lot of system variables is beyond the control of the human administrator. With the help of self-adaptive systems, handling the required configurations in different situations would be possible. A self-adaptive system automatically estimates and interprets its performance and behaviour. It also modifies the efficiency and applied algorithms according to the management plan and system condition. Having all the salient benefits of a self-adaptive system depends on the environmental settings and

*Keivan Borna – corresponding author; Reza Ghanbari; Faculty of Mathematical Sciences and Computer, Kharazmi University, Tehran, Iran, E-mail: borna@khu.ac.ir, reza91@aut.ac.ir

optimized adjustments. In past years, most of these developments had done manually by a human operator [1]. Self-adaptation in cloud resource administration is a modern and gradual start towards optimizing resources and reducing costs. Most research in recent years has studied supervised and unsupervised methods to extract the features using auto-encoders, RNN and CNN to heighten the prediction correctness for received requests per time. Data pre-processing, feature engineering, model selection, training, prediction, and evaluation are time-consuming activities. This paper proposes a novel self-adaptive approach based on a deep learning model to manage resources in a cloud environment considering the impact of this intelligent method in modifying the amount of processing and predetermined scheduling. The canonical polyadic decomposition has a considerable role in our proposed deep learning model. With the help of this method, an effective model that tries to predict cloud requests is developed, user activities and workload. In the stated model, a stacked auto-encoder neural network is used. The canonical polyadic decomposition has a considerable role in our proposed deep learning model. It has been ranked as the highest compression rate among all other decomposition methods [2]. Before applying canonical polyadic decomposition, one needs a tensor format version of the stacked auto-encoder. This tensor format is achievable by bijection. Our model predicts the cloud services' requests and enhancing resource distribution. The required dataset for our research is collected from PlanetLab. It was a global research system that encouraged the production of modern network services and distributed computing. In May 2020, the PlanetLab project has been shut down. Finally, our model has been compared and analyzed with a recently published method based on this dataset. The examined dataset includes more than 1000 nodes placed at different 700 stations around the world. In this dataset, the detailed information of the virtual servers around the world continuously gathered every 5 minutes. The proposed model introduces an innovative way based on deep learning. A cloud provider can predict the users' requests fraction on each cluster. In the following, the research method has been reviewed and subsequently the achieved results have been evaluated.

2. Research method

Matrices are specific examples of tensors. Tensors are particularly suitable for displaying heterogeneous data. One can use tensor decomposition to reduce dimension in big data. Canonical polyadic decomposition of tensors, as one of the most effective tensor decomposition schemes, would have the highest compression rate for high-dimensional data compared to other methods. A tensor is approximated by the sum of the tensors with a rank of 1, with the available computational errors. If the rank-1 tensors are minimum, then the existing decomposition is a canonical polyadic decomposition. Assume that $A \otimes B$ represents the cross product of two tensors, A and B. Tensors orders are n and m, respectively. The tensor resulting form of this product is $n + m$ order. Lets briefly define tensor cross product:

$$(A \otimes B)_{i_1,\ldots,i_N,j_1,\ldots,j_M} = a_{i_1,\ldots,i_N} \cdot b_{j_1,\ldots,j_M}. \tag{1}$$

According to Fig. 1 to Fig. 4, the approximation error, classification accuracy drop, parameters reduction, and speedup values of our model to Tucker-SAE and estimated the training efficiency are compared. The CP-SAE is the abbreviation form of the canonical polyadic decomposition stacked auto-encoder.

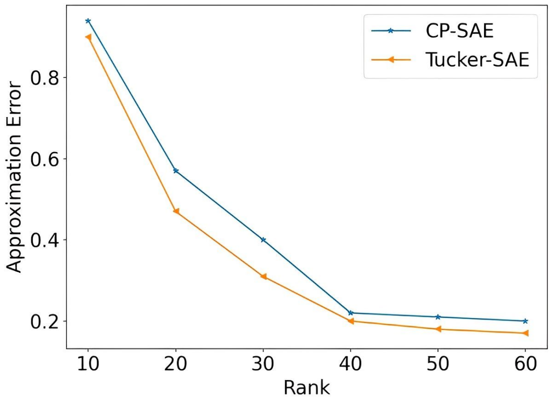

Fig. 1 *Comparison of approximation error, CP-SAE and Tucker-SAE.*

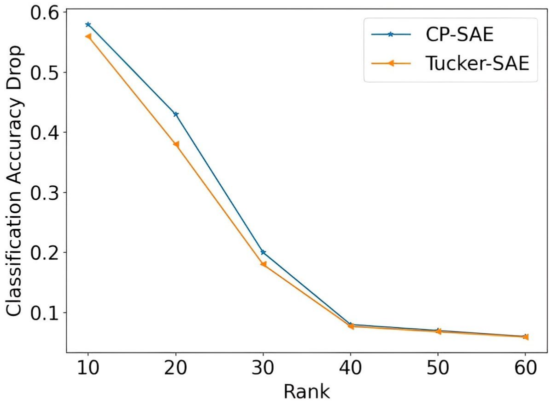

Fig. 2 *Comparison of classification accuracy drop, CP-SAE and Tucker-SAE.*

In this section, the canonical polyadic decomposition form of the preceding auto-encoder is explained. Additionally, one can elaborate on this by bringing up the stacked auto-encoders mechanism. Then the auto-encoder representation form to the tensor model is converted. The input vector is an N-dimensional vector and is converted to a d-order tensor using the bijection function that is named f. There would be a similar correlation among all components. According to Zhang et al. [3], the multi-dot product is qualified for transforming the base model into a tensor construction. Then, each part is an N-order tensor, instead of a matrix. The forward pass of the original auto-encoder in tensor form is also updated.

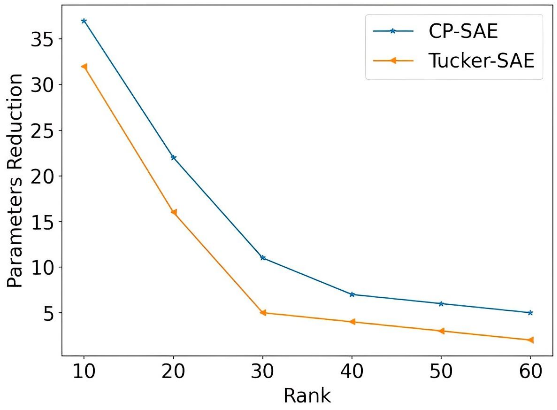

Fig. 3 *Comparison of parameters reduction, CP-SAE and Tucker-SAE.*

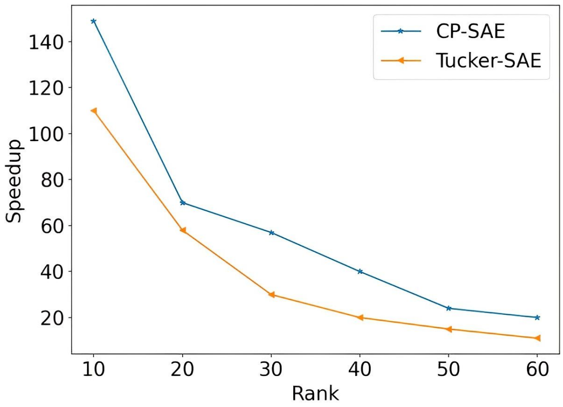

Fig. 4 *Comparison of speedup, CP-SAE and Tucker-SAE.*

The dot product format for the functions have been represented in the following:

$$H = f(W^{(1)} \odot X + b^1), \tag{2}$$

$$Y = g(W^{(2)} \odot H + b^2). \tag{3}$$

And the loss function is:

$$\begin{aligned} J(\theta) \quad = \quad & \frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}\sum_{i_1=1}^{I_1}\cdots\sum_{i_N=1}^{I_N}((Y^i_{i_1,i_2,\ldots,i_N}) \right. \\ & \left. -(X^i_{i_1,i_2,\ldots,i_N}))^2 + \frac{\lambda}{2}(W^{(1)^2} + W^{(2)^2})\right). \end{aligned} \tag{4}$$

The parameters are converted into the tensor format. This conversion also includes existing weighted matrices. From Eq. 2 and Eq. 3, the tensor format of the auto-encoder is not dissimilar from the base model; nevertheless, the loss functions are a bit contradictory. Our model uses Euclidean distance for the loss function and tensor weight are in the canonical polyadic decomposition format. The reformatted weights equations are in the following:

$$W^{(1)}_{\alpha i_1,i_2,\ldots,i_N} = \sum_{r=1}^{R} X^1_{(i_1,r)} X^2_{(i_2,r)} \cdots X^N_{(i_N,r)}, \tag{5}$$

$$W^{(2)}_{\beta j_1,j_2,\ldots,j_N} = \sum_{q=1}^{Q} X^1_{(j_1,q)} X^2_{(j_2,q)} \cdots X^N_{(j_N,q)}. \tag{6}$$

In the proposed model, the ReLU function as the activation function is used. In Zhang and et al. [4], they advised sigmoid as an activation function. ReLU is more computationally effective than sigmoid. This activation function obliges to pick positive values with no expensive exponential operations as in sigmoid. Considering the stated model is based on gradient descent, sigmoid tends to vanish gradient because there is a tendency to reduce the gradient as the input of a sigmoid function increases and also, networks with ReLU expose a more reliable convergence production than sigmoid [5].

3. Numerical results

Experiments are conveyed to assess the production of the proposed deep learning model based on the canonical polyadic decomposition (CP-SAE) in preparation performance and cloud services' request prediction. Our model in a self-adaptive system is used and it efficiently allocates the demanded resources. Therefore, one can distribute available resources to different services to reduce costs in a cloud service system. By using mean absolute percentage error (MAPE) and the root mean squared error (RMSE), the highest number of requests of 20 virtual machines is predicted. Tab. I and Tab. II explain the prediction accuracy in terms of the MAPE and the RMSE of three models for 20 virtual machines in four distinct time intervals. The proposed model yields a lower mean absolute percentage error and root mean squared error than the deep belief network and the traditional neural network [6].

Model	5 min	15 min	30 min	60 min
The CP-SAE	0.2089	0.2094	0.2489	0.2612
Deep belief	0.2212	0.2441	0.2701	0.3204
Neural network	0.2610	0.2698	0.3098	0.3309

Tab. I *MAPE of models.*

Tabs. I and II show the accuracy of predicting the workload and resource consumption of virtual servers in terms of the average absolute percentage error and the mean square error of the three models for 20 virtual machines in the four

Model	5 min	15 min	30 min	60 min
The CP-SAE	0.2089	0.2094	0.2489	0.2612
Deep belief	0.2212	0.2441	0.2701	0.3204
Neural network	0.2610	0.2698	0.3098	0.3309

Tab. II *RMSE of models.*

future time frames. From the results, the presented model has a lower mean absolute percentage error and mean square error than the deep belief network and the traditional neural network. For example, the proposed model for predicting the workload and requests in 15 time intervals achieves a value of 0.22 for the mean absolute percentage error, while the deep belief network as well as the traditional neural network perform 0.24 and 0.27, respectively. These observations show that the proposed model achieves higher accuracy than the deep belief network and the traditional neural network for predicting the workload of multiple virtual machines. In addition, the proposed model can improve the prediction accuracy for longer periods of time. For a period of 60 minutes, it can have an accurate prediction of the actual value, which implies that the proposed model can learn the salient features from the available data about the load rate effectively. The amount of server load is directly related to the amount of resources consumed by them. For example, if the load on a group of virtual servers increases in a period of time, that group of servers should use more power for processing and analysis. This leads to the use of available resources such as the processor and data storage disks for a longer period of time.

Fig. 5 shows the prediction results. These results are based on the maximum usage of virtual servers in the desired periods. These results help us to predict request numbers on virtual servers more correctly and to have more well-defined

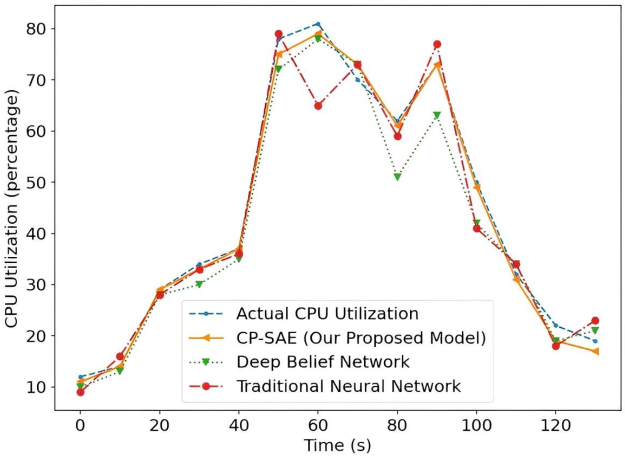

Fig. 5 *Prediction results based on CP-SAE model.*

administration over the number of resources allocated to each service by using self-adaptive systems. Since predicting the number of requests for cloud services helps cloud providers to distribute their resources better, our model can predict the amount of load requested on servers with more accuracy. Since the amount of load requested on the servers also determines the number of resources consumed, cloud providers can distribute the resources required by the services at regular intervals. In the experiments performed to compare the proposed method with other methods, performance improvements are visible. The numbers obtained in each of the four cases, namely approximation error, classification accuracy drop, parameters reduction, and speed up are compared. Considering that the ReLU activation function in this model is used, better results than the similar methods which use the sigmoid activation function are achieved. Efficiency and speed in providing predictions are essential requirements in this model. The calculations required for sigmoid require more time and more resources, and one can also use ReLU to speak with more confidence in the results of the gradients produced. As a result, ReLU as our activation function is used. Increasing the speed of training deep learning models, especially in the problem ahead in the cloud, is an integral part of this section. Future research and development are possible on the decomposition part. For this purpose, one can present a more accurate model using novel methods for compressing data and studying more specific tools to measure the learning time. With the help of these models, self-adaptive systems can operate an influential role in decreasing the human factor and errors.

4. Conclusion

In this research, the data obtained from Bitbrains company by analyzing the virtual machines and the data in a self-consistent system using the presented algorithm. The correlation level in different sources and described the relationship of each with the amount of demand and consumption. The effect of the proposed model in predicting the workload of the target data set is analyzed and compared with other known methods in this field, and this is done by using statistical analysis tools. Specifically, it can be said that this research is in three parts: recognizing the workload of the desired cloud servers, presenting the algorithm and model based on deep learning using multivariate analysis, and using the characteristics of self-adaptive systems to modify and change the management default values to analyze and checks the necessary items.

Cloud computing is the most likely infrastructure for complex tasks in information technology. In this research, an efficient deep learning model based on focal multivariate analysis has been presented for use in self-adaptive systems, which can be used to predict the workload of the cloud and the number of available requests. One of the features of this proposed method is the use of deep learning model to learn the salient features of the data used for training. In addition, the multivariate decomposition used for parameter compression significantly improves the training efficiency. In the experiments, the proposed model was evaluated and investigated in the training efficiency and workload prediction effectiveness. In particular, the classification of decreasing accuracy and increasing speed has been used to measure the training quality of the presented model by comparing it with other deep learning-based methods. The average absolute percentage error as well as the root mean square error are presented to evaluate the prediction accuracy of the model and by comparing two other learning models, i.e. deep belief network and traditional artificial neural network, the analysis of the relevant model has been done. The experimental results clearly show the following three points:

- The presented model can achieve a high speed in the training section, because by using multivariate decomposition for compression, the parameters are significantly reduced in classification accuracy. Specifically, when the rank is 32, the presented model achieves 21 times faster for the deep learning model with a 4.6% decrease in accuracy.

- The proposed model predicts the CPU usage of the highest volume and workload more accurately than the other two methods based on machine learning because the proposed deep learning model can learn the salient features for complex workload data more effectively.

- The presented model achieves the lowest average percentage error and root mean square error. This means that the presented model has the best performance in predicting the load for several virtual machines on the cloud.

In general, since the training of a deep learning model is relatively time-consuming due to the large number of parameters, how to improve the test efficiency has become a challenging issue in the field of deep learning. The multivariate decomposition used in order to improve the training efficiency of the stacked autoencoder model was investigated. Therefore, in this research, a new idea and solution has been provided to researchers to accelerate the training of deep learning models in self-adaptive systems. More precisely, researchers can further investigate tensor decomposition schemes such as ordinary polynomial decomposition and Tucker decomposition to improve training effectiveness for other deep learning models including deep belief network, convolutional neural networks, and recurrent neural networks. Cloud workload forecasting is an important issue that cloud providers can provide quality services to meet consumer needs and reduce resource consumption. In particular, preliminary experiments confirm the performance of the proposed model in this research for cloud workload forecasting. The results indicate that cloud service provider engineers can use the presented model to allocate virtual machine resources by predicting workload. The main goal of this research is to provide an efficient deep learning model for self-adaptive systems for forecasting with cloud computing. In using this deep learning model in the cloud space, the issue of privacy of users whose data is collected from their activities is also observed. In other words, all the programs and servers are like a black box and there is no interference in the type of processing and its working model. Therefore, we can use the efficient deep learning model with privacy protection in future works. The presented model can also be used in the form of reinforcement learning for better learning in the cloud. Certainly, using the main features of self-adaptive systems will help to change and modify the consumption pattern and predict resources in the cloud space. In future research, a more accurate model based on multivariate analysis can also be provided by using reinforcement learning with the help of self-adaptive systems. It is also important to understand the workload in the cloud data center.

References

[1] DE LEMOS R., GIESE H., MÜLLER H.A., SHAW M., ANDERSSON J., LITOIU M., SCHMERL B., TAMURA G., VILLEGAS N.M., VOGEL T., WEYNS D. *Software Engineering for Self-Adaptive Systems: A Second Research Roadmap*. In: de Lemos R., Giese H., Müller H.A., Shaw M. (eds) Software Engineering for Self-Adaptive Systems II. Lecture Notes in Computer Science, vol. 7475. Springer, Berlin, Heidelberg, 2013, doi: 10.1007/978-3-642-35813-5_1.

[2] LIU G., BAO H., HAN B. *A Stacked Autoencoder-Based Deep Neural Network for Achieving Gearbox Fault Diagnosis.*, Mathematical Problems in Engineering, 2018, pp. 1–10.

[3] ZHANG Q., LIN M., YANG L.T., CHEN Z., LI P. *Energy-Efficient Scheduling for Real-Time Systems Based on Deep Q-Learning Model.* In: IEEE Transactions on Sustainable Computing, 4(1), pp. 132–141, doi: 10.1109/TSUSC.2017.2743704.

[4] ZHANG Q., YANG L.T., YAN Z., CHEN Z., LI P. *An Efficient Deep Learning Model to Predict Cloud Workload for Industry Informatics.* In: IEEE Transactions on Industrial Informatics, 14(7), pp. 3170–3178, doi: 10.1109/TII.2018.2808910.

[5] WANG Y., LI Y., SONG Y., RONG X. *The Influence of the Activation Function in a Convolution Neural Network Model of Facial Expression Recognition.* Appl. Sci. 2020, 10, 1897, doi: 10.3390/app10051897.

[6] GHANBARI R., BORNA K. Multivariate Time-Series Prediction Using LSTM Neural Networks, 2021 26th International Computer Conference, Computer Society of Iran (CSICC), 2021, pp. 1–5, doi: 10.1109/CSICC52343.2021.9420543.