Workload Prediction Using VMD and TCN in Cloud Computing

Amine Mrhari and Youssef Hadi
Research in Computer Science Laboratory, Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco
Email: {amine.mrhari, hadiyoussef}@gmail.com

***Abstract*—Workload prediction becomes a key major to improve the management of resources in cloud computing, recent studies have shown that predicting workload in data center positively affect the quality of service, elasticity, Service-Level Agreement (SLA) and power consuming, etc. In this paper, we design an efficient model to predict workload demand in dynamic cloud computing, which is a combination of Variational Mode Decomposition (VMD) and Temporal Convolutional Network (TCN). First, we use VMD to decompose workload extracted from history traces, into multiple cloud workload sequences. Second, the decomposed sequences are fed in a Temporal Convolutional Network. By using Alibaba workload dataset as a case study, the results show that the proposed model outperforms the compared deep learning based model in term of accuracy and achieve the state-of-art.**

***Index Terms*—workload prediction, temporal convolutional networks, variational mode decomposition, dilated causal convolution, deep learning, cloud computing**

I. Introduction

Recently, there is a tendency in firms to migrate their tasks in cloud, where cloud computing proposes many services and technologies via the Internet, such as virtualization of resources, software as services, and platforms as services [1]. By employing the concept "Pay-as-you-go" in cloud computing, the on-demand services offer the end users, the elasticity, scalability, mobility, availability, reliability and security of services [2] and reduce the cost associated with services and the power consumption [3], [4]. However, to provide a high availability of services, service providers are facing difficulties for managing resources in high or low demand periods, when end users allocate resources simultaneously (under-provisioning) [5], or when they offer an excess of resources (over-provisioning) [6]. In order to overcome these difficulties, service providers can choose between two approaches to manage resources, there are the reactive and proactive approaches [7]. In the reactive approach, the end users allocate more resources initially or at a specific threshold. In the proactive approach, the service providers predict resource usage ahead of time and provide it dynamically on demand. Therefore, workload prediction is crucial to predict resources usages in short and long term by changing the allocation strategy and task scheduling in high and low resources demand by scaling resources up or down according to usages.

In this paper, we design an efficient model to predict workloads demand in dynamic cloud computing, which is a combination of Variational Mode Decomposition (VMD) and Temporal Convolutional Network (TCN). First, we use VMD to decompose workload extracted from history traces, into multiple cloud workload sequences. Second, the decomposed sequences are fed in a Temporal Convolutional Network. Finally, the sum of the output results of all decomposed sequences is the predicted workloads demand.

This paper is organized as follows. Section II gives a review of related works. Section III describes the details of the proposed model. Section IV, presents and discusses the experimental results. Finally, Section V conclude the paper and propose future work.

II. Related Works

In literature, workload predicting methods is divided into statistical methods, machine learning methods and deep learning methods.

A. Statistical Methods

The authors in [8] proposed a workload prediction model using ARIMA model. The prediction results achieve accuracy up to 91%, with minimal impact in QoS for users. [9] compared the resource prediction using AR, MA and ARIMA methods. The results conclude that AR is the best predicting model. [10] based on the second-order ARMA and exponential smoothing prediction algorithms to estimate the workload prediction.

B. Machine Learning Methods

Paper cited in [11] predicted resources using Linear Regression (LR), Neural Network (NN) and Support Vector Machines (SVM) in the multi-tier web application environment. Results showed that SVM predicts better compared to NN and LR. [12] predicted workload resources by using Markov and Bayesian techniques in the cloud environment. [13] uses K-Nearest Neighbor regression (KNN) for predicting short-term resources demands. The authors of [14] predicted CPU usage workload based on Deep Belief Network (DBN) with multiple-layered Restricted Boltzmann Machines (RBMs) and a regression layer. Results showed that the model is appreciable.

Manuscript received November 17, 2021; revised April 20, 2022.

doi: 10.12720/jait.13.3.284-289

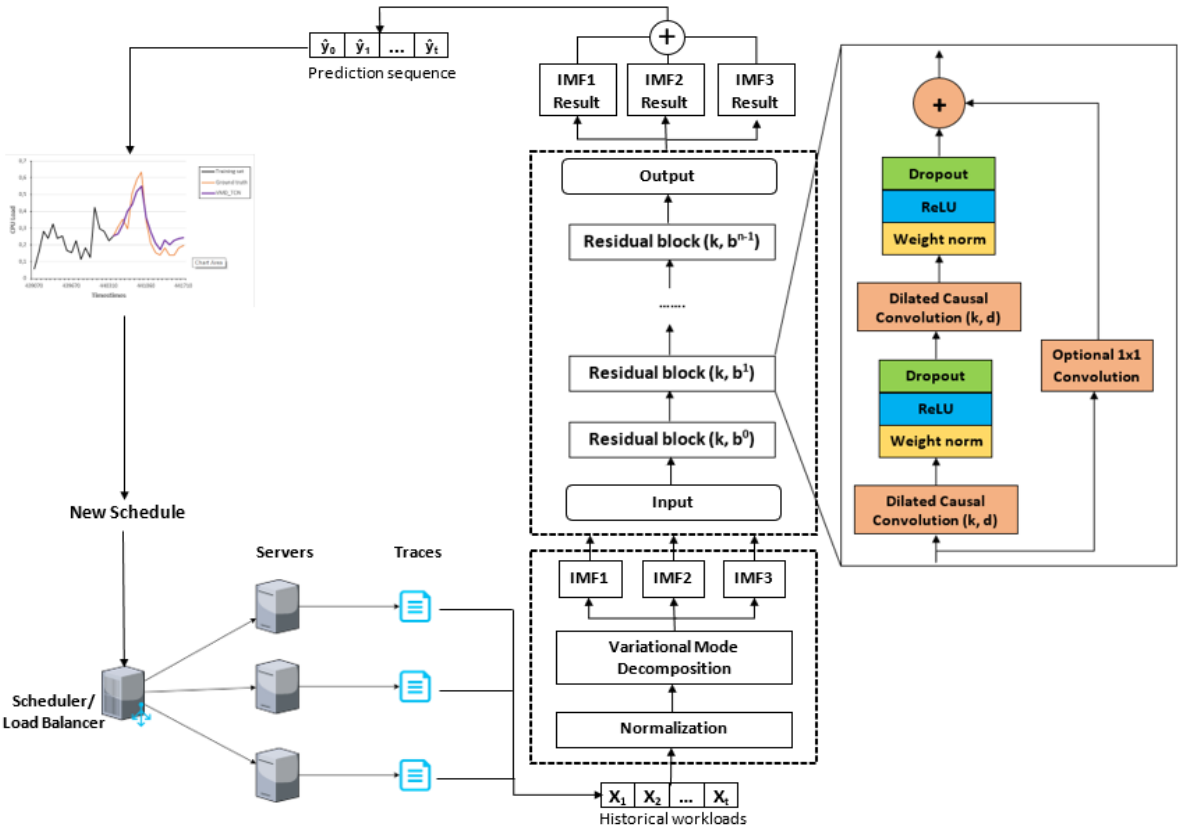

Figure 1. The full architecture of resources scheduling and load balancing using the proposed approach to predict workload in the cloud computing.

C. Deep Learning Methods

[15] proposed a model based on Recurrent Neural Network (RNN) to predict cloud resource requests by using an Orthogonal Experimental Design (OED) to find the most influential parameters. [16] compared the Seasonal Autoregressive Integrated Moving Average (SARIMA) and long short-term memory LSTM models to forecast CPU usage. The SARIMA model outperformed the LSTM for the long-term prediction task, but performed poorer on the short-term task. [17] proposed an approach to predict workload in cloud environment by LSTM encoder-decoder network with attention mechanism, the model outperforms LSTM and Gated Recurrent Units (GRU) encoder-decoder networks. [18] showed that the basic LSTM network achieve good performance in predicting multistep ahead workload. [19] proposed a novel approach to predict workload by stacking Recurrent Neural Networks and Autoencoder on different datasets to compare prediction accuracy. The model outperforms the compared models. [20] uses neural network to create a framework named PRACTISE to predict the future workload of the CPU, memory, disk, and network bandwidth. The shown that the model had reliable accuracy, robustness, and flexibility.

III. Methodology

Fig. 1 shows the full architecture of load balancing and resources scheduling using the proposed approach to predict workload in the cloud computing, by combining [21] and [22] approaches.

First, the system records workloads in real time as traces and supposes them as a time series sequences. Then, process the former using min-max normalization and decompose it by VMD into multiple Intrinsic Mode Functions (IMFs), described in details in Section III-B. Second, the decomposed sequences are fed into a Temporal Convolutional Network to predict IMFs. Then, adding the results of each one to obtain the workloads predicts next step. Finally, the results are inputted into the resource management system to create a new allocation schedule and update the load balancer server.

A. Sequence Modeling

The workload traces recorded can be seen as CPU usage, memory usage, bandwidth or disk space, etc. In cloud data centers, CPU usage is the most important and limited resource [23]. To this end, the workloads history used in this paper is based on CPU usage, and supposed as a time series sequence of $x = (x_{T-n}, x_{T-(n-1)}, \dots, x_T)$, where x_T is the workload value at time T and n is the length of history window. After feeding the model by historical workloads, a min-max normalization is applied on data by scaling all values in range of 0 and 1 to accelerate the convergence of predicting algorithms. The min-max normalization formula is defined by:

$$x_t = L_{max} + \frac{x_t - x_{min}}{x_{max} - x_{min}}(L_{max} - L_{min}) \quad (1)$$

where L_{min} and L_{max} are respectively the minimum and the maximum limits of the target range. x_{min} and x_{max} are the minimum and the maximum values of the dataset.

B. Variational Mode Decomposition

Variational Mode Decomposition (VMD) is a non-recursive signal decomposition algorithm proposed by [24]. The algorithm decomposes signal into a k discrete number of sub-signals (modes), which each mode has a central frequency and limited bandwidth. First, the analytical signal of each mode is obtained by a Hilbert transformation to acquire the unilateral spectrum. Second, for each mode, an exponential term is added to adjust its estimated center frequency with the spectrum of each mode modulated to the corresponding base frequency band. Finally, we will estimate the bandwidth by calculating the Gaussian smoothness of the demodulation signal. A constrained variational problem is constructed as follows:

$$\begin{cases} \min\limits_{\{u_k\},\{\omega_k\}} \left\{ \sum_k \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-i\omega_k t} \right\|_2^2 \right\} \\ s.t \sum_k u_k = f \end{cases} \quad (2)$$

where ∂ is the Dirac distribution, $\{u_k\} := \{u_1, u_2, \dots, u_k\}$ is the set of all modes which the high order k is the low frequency components, $\{\omega_k\} := \{\omega_1, \omega_2, \dots, \omega_k\}$ is the central frequency set of all modes, $\sum_k := \sum_{k=1}^{k}$ is the sum of the modes and f is the original signal.

In this paper, we suppose that the time series sequence x of historical workloads is a signal, then we apply VMD to decompose it into stable and predictable time series called Intrinsic Mode Function (IMF), in order to reduce the instability of x, while each IMF is a new time series that contains a part of x, and the total of IMFs cover the original time series [25]. The IMFs obtained from VMD are independently fed into a TCN to predict the result separately. The sum of the output results of all IMFs is the final predicted workload demands. Fig. 2 shows an example of decomposition of Alibaba dataset [26] into 3 modes, where IMF1 presents, the low frequency part of the original workload contains a smooth trend change of it. IMF2 and IMF3 present the high frequency parts of the original workload, and contain small details of it.

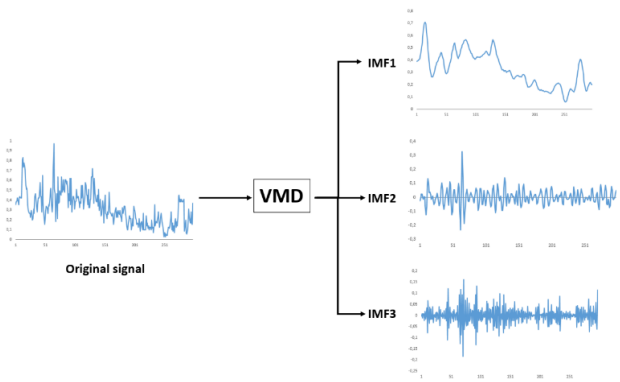

Figure 2. Decomposition of workload demand into 3 IMFs modes.

C. Temporal Convolutional Network

The TCN is a novel prediction approach proposed by [21], based on dilated causal convolutions network. It uses the parallelism concept to speed up training and evaluation, and to avoid the problem of vanishing gradients, it is composed of three parts: causal convolutions, dilated convolutions, and residual connections, described in detail in following sections.

1) Dilated causal convolutions

After feeding TCN model with historical workload, a 1D Fully-Convolutional Network (FCN) [27] and dilated causal convolutions [28] are applied to the sequence with adding a zero padding of length k–1 in each hidden layer. The output of the tth element in sequence is defined as:

$$F(t) = \sum_{i=0}^{k-1} f(i) \, . \, x_{t-d.i} \quad (3)$$

where x is 1D input sequence, $f : \{0,1, \dots\dots, k-1\} \rightarrow R$ is the filter with size k, d is the dilation factor and $t - d.i$ is the accounts for the direction of the past.

Fig. 3 shows an example of dilated causal convolution for a sequence with dilation factors $d = 1, 2, 4$ and filter size $k = 3$.

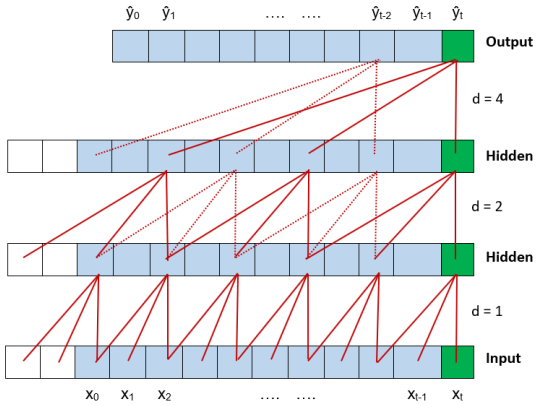

Figure 3. Dilated causal convolution network.

2) Residual connections

A residual block [29] is a combination of two layers $F(x)$, added to the input identity of the block x, a 1x1 convolution shown in Fig. 4. Each layer contains a dilated causal convolution, weight normalization [30] for normalizing the convolutional filters, rectified linear unit (ReLU) [31] for non-linearity and dropout [32] for regularization. We define the output o of residual block as:

$$o = activation(x + F(x)) \quad (4)$$

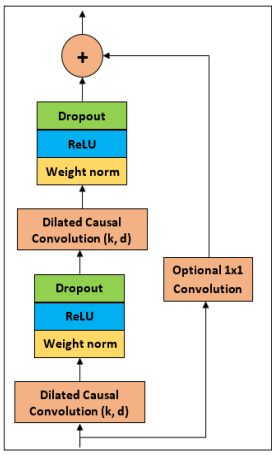

Figure 4. A TCN residual block.

IV. Experiments

In this section, experiments are carried out to evaluate and demonstrate the effectiveness of the proposed model using real-world dataset by comparing existing Recurrent Neural Network (RNN) and TCN methods. First, we introduce the dataset and the preprocessing parameters. Second, we describe the evaluations metrics and introduce other predicting methods. Finally, we present the experiments results and discuss the comparison between them.

A. Dataset

To evaluate our model, we extract a dataset collected from the real-world cloud computing and provided by Alibaba [26], which contains traces for approximately 4000 machines in a period of 8 days. We used the CPU usage as a workload demands for one machine chosen randomly. Then, to accelerate the convergence of predicting algorithms, we preprocess data by the min-max normalization formula presented in (1), where L_{min} and L_{max} are set to 0 and 1 respectively. The preprocessed workload is shown as original signal in Fig. 2. Thereupon, a fixed length sliding time window is applied to the preprocessed data to create pairs of features x and labels y with length of 18 and 15 datapoints respectively. Afterwards, we split 80% of windowed data as training set and 20% as test sets train. Finally, we train the model VMD-TCN on the training set and testing the prediction accuracy on the test set.

B. Evaluation Metrics

To evaluate the performance of the compared methods, we consider three metrics: the mean squared error (MSE), the mean absolute percentage error (MAPE) and the root mean squared error (RMSE), which are calculated as follow:

$$MSE = \frac{1}{n}\sum_{t=0}^{n}(y_t - \hat{y}_t)^2 \quad (5)$$

$$MAPE = \frac{100}{n}\sum_{t=0}^{n}\left|\frac{y_t-\hat{y}_t}{y_t}\right| \quad (6)$$

$$RMSE = \sqrt{\sum_{t=0}^{n}\frac{(y_t-\hat{y}_t)^2}{n}} \quad (7)$$

where y_t and $\hat{y}_t$ are respectively, the observed and the predicted data at time t, n is the length of data.

C. Baseline Methods

To assess the performance of the VMD-TCN model, serval baseline methods are selected to the comparison.

GRU [33]: The basic gated recurrent unit is a recurrent network that uses GRU cell as the computing unit. The multilayer GRU network is fed by the current value of workload, the output of the last layer is the workload predicted value.

LSTM [18]: The basic long short-term memory is similar to GRU network, except in LSTM network uses LSTM cells unlike the GRU cells.

TCN: The basic Temporal Convolutional Network described in Section III-C is used first without VMD decomposition, then with VMD decomposition as presented on our model above.

D. Results and Analysis

We trained the baseline methods and the VMD-TCN method with the same parameters shown in Table I, and the same dataset in the GPU engine proposed by Google Collaboratory [34].

Fig. 5 shows the workload prediction results of the baseline methods and the proposed VMD-TCN for one machine extracted from Alibaba dataset, where the light blue curve is the ground truth. The gray, yellow, orange and blue curves are respectively, the GRU, LSTM, TCN and VMD-TCN, predicted results. By observing the chart, it is clearly that the VMD-TCN has accurate prediction in each step according to the original signal. TCN curve is also close to the original signal and most of directions change is correctly predicted. The GRU curve has failed in predicting some directions change. The LSTM curve has failed to predict most of directions change. LSMT and GRU has succeeded to extract the non-linearity from the sequence locally but failed in globally, which means that LSMT and GRU cannot predict long-term changes. TCN has succeeded to extract the non-linearity from the sequence locally and globally, which means that TCN can predict short- and long-term changes. In addition, by decomposing the sequence into low and high frequencies by VMD, VMD-TCN has succeeded to extract the seasonal and trend from low frequency, and the sequence details from high frequency, and improve the training time by fast converging the IMFs individually. Table II presents a comparison of workload predicting performances, given in term of MSE, MAPE and RMSE. Where the MSE of VMD-TCN is 4×10^{-2} while that of GRU, LSTM and TCN are respectively 12.70×10^{-2}, 16.02×10^{-2} and 11.04×10^{-2}. The MAPE of VMD-TCN is 14.41% while that of GRU, LSTM and TCN are respectively 23.550%, 26.037% and 22.043%. The RMES of VMD-TCN is 10.50% while that of GRU, LSTM and TCN are respectively 11.27%, 12.65% and 10.55%. Note that the lower values are better. The proposed method based on VMD achieves the state-of-art performance and outperforms the compared baseline methods in predicting real-world workload in cloud computing.

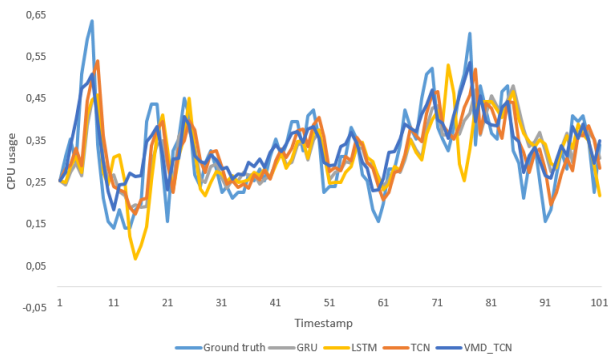

Figure 5. Workload prediction results of the baseline methods and the proposed VMD-TCN on Alibaba dataset.

TABLE I. PARAMETERS USED IN TRAINING MODELS

Parameter	Value	Parameter	Value
Input length	18	Filter size	5
Output length	15	Number of filters	3
Number of epochs	400	Hidden size	128
Dropout	0.2	Batch size	128
Dilation factor	2		

TABLE II. COMPARISON OF PERFORMANCES IN PREDICTING WORKLOAD

Model	$\times 10^{-2}$		
	MSE	MAPE	RMSE
GRU	12.70	23.550%	11.27
LSTM	16.02	26.037%	12.65
TCN	11.04	22.043%	10.55
VMD-TCN	**4.00**	**14.410%**	**10,50**

V. CONCLUSION

In this paper, we have presented a novel approach to predict workload demands in cloud computing based on variational mode decomposition and temporal convolutional network. The VMD is used to decompose workload into IMFs to reduce the variance impact and stabilize the workload predicting accuracy. The TCN is used to predict IMFs separately, where the sum of the predicted IMFs is the final predicted workload demands. The model was compared with the existing RNN methods and the TCN method. The simulation results show that the proposed model achieves the state-of-art performance and outperforms the compared methods in terms of predicting workload in real-world cloud computing. The future work will be combining game theory and VMD-TCM model to approximate the optimal solution for the response time of users in load balancer.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Amine Mrhari conducted the research, analyzed the data, wrote the paper. Youssef Hadi supervised the research.

REFERENCES

[1] M. Masdari, S. ValiKardan, Z. Shahi, and S. I. Azar, "Towards workflow scheduling in cloud computing: A comprehensive analysis," *J. Netw. Comput. Appl.*, vol. 66, pp. 64-82, 2016.

[2] R. Moreno-Vozmediano, *et al.*, "Key challenges in cloud computing: Enabling the future internet of services," *Internet Computing*, vol. 17, no. 4, pp. 18-25, 2013.

[3] Q. Zhang, Z. Chen, A. Lv, L. Zhao, F. Liu, and J. Zou, "A universal storage architecture for big data in cloud environment," in *Proc. IEEE International Conference on Internet of Things*, 2013, pp. 476- 480.

[4] S. Singh and I. Chana, "A survey on resource scheduling in cloud computing: Issues and challenges," *J. Grid Comput.*, vol. 14, no. 2, pp. 217-264, 2016.

[5] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: State-of-the-Art and research challenges," *J. Internet Serv. Appl.*, vol. 1, pp. 7-18, 2010.

[6] M. Amiri and L. Mohammad-Khanli, "Survey on prediction models of applications for resources provisioning in cloud," *Journal of Network and Computer Applications*, vol. 82, pp. 93-113, 2017.

[7] F. Tahir, M. Abdullah, F. Bukhari, K. M. Almustafa, and W. Iqbal, "Online workload burst detection for efficient predictive autoscaling of applications," *IEEE Access*, vol. 8, pp. 73730-73745, 2020.

[8] R. N. Calheiros, E. Masoumi, R. Ranjan, and R. Buyya, "Workload prediction using ARIMA model and its impact on cloud applications," *IEEE Transactions on Cloud Computing*, vol. 3, no. 4, pp. 449-458, 2015.

[9] P. A. Dinda, "Design, implementation, and performance of an extensible toolkit for resource prediction in distributed systems," *IEEE Trans. Parallel Distrib. Systems*, vol. 17, pp. 160-173, 2006.

[10] N. Roy, A. Dubey, and A. Gokhale, "Efficient autoscaling in the cloud using predictive models for workload forecasting," in *Proc. IEEE International Conference on Cloud Computing*, 2011, pp. 500-507.

[11] A. A. Bankole and S. A. Ajila, "Cloud client prediction models for cloud resource provisioning in a multitier web application environment," in *Proc. 7th International Symposium on Service Oriented System Engineering*, 2013, pp. 156-161.

[12] P. S. Doshi, M. Goel, A. Agarwal, and K. Punjabi, "Performance provisioning using machine learning based automated workload classification," US Patent App. 20180024859, 2018.

[13] F. Farahnakian, T. Pahikkala, P. Liljeberg, and J. Plosila, "Energy aware consolidation algorithm based on K-nearest neighbor regression for cloud data centers," in *Proc. 6th International Conference on Utility and Cloud Computing*, 2013, pp. 256-259.

[14] F. Qiu, B. Zhang, and J. Guo, "A deep learning approach for VM workload prediction in the cloud," in *Proc. 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, 2016, pp. 319-324.

[15] W. Zhang, B. Li, D. Zhao, F. Gong, and Q. Lu, "Workload prediction for cloud cluster using a recurrent neural network," in *Proc. International Conference on Identification, Information and Knowledge in the Internet of Things*, 2016.

[16] L. Nashold and R. Krishnan, "Using LSTM and SARIMA models to forecast cluster CPU usage," ArXiv abs/2007.08092, 2020.

[17] Y. Zhu, *et al.*, "A novel approach to workload prediction using attention-based LSTM encoder-decoder network in cloud environment," *EURASIP Journal of Wireless Com. Network.*, article number 274, 2019.

[18] B. Song, Y. Yu, Y. Zhou, Z. Wang, and S. Du, "Host load prediction with long short-term memory in cloud computing," *J. Supercomput.*, vol. 74, pp. 6554-6568, 2018.

[19] H. M. Nguyen, S. Woo, J. Im, T. Jun, and D. Kim, "A workload prediction approach using models stacking based on recurrent neural network and autoencoder," in *Proc. IEEE 18th International Conference on High Performance Computing and Communications*, 2016, pp. 929-936.

[20] J. Xue, F. Yan, R. Birke, L. Y. Chen, T. Scherer, and E. Smirni, "Practise: Robust prediction of data center time series," in *Proc. 15th International Conference on Network and Service Management*, 2015, pp. 126-134.

[21] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," ArXiv abs/1803.01271, 2018.

[22] J. K. Dragomiretskiy and D. Zosso, "Variational mode decomposition," *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 531-544, 2013.

[23] F. J. Baldán, S. Ramírez-Gallego, C. Bergmeir, F. Herrera, and J. M. Benítez, "A forecasting methodology for workload forecasting in cloud systems," *IEEE Transactions on Cloud Computing*, vol. 6, no. 4, pp. 929-941, 2016.

[24] K. Dragomiretskiy and D. Zosso, "Variational mode decomposition," *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 531-544, 2013.

[25] A. A. Abdoos, "A new intelligent method based on combination of VMD and ELM for short term wind power forecasting," *Neurocomputing*, vol. 203, pp. 111-120, 2016.

[26] J. Guo, Z. Chang, S. Wang, H. Ding, Y. Feng, L. Mao, and Y. Bao, "Who limits the re-source efficiency of my datacenter: An analysis of Alibaba datacenter traces," in *Proc. the International Symposium on Quality of Service*, 2019, p. 39.
[27] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2015.
[28] F. Yu and V. Koltun, "Multiscale context aggregation by dilated convolutions," arXiv preprint arXiv:1511.07122, 2015.
[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016, pp. 770-778.
[30] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," *Advances in Neural Information Processing Systems*, vol. 29, 2016.
[31] A. Maas, A. Y. Hannun, and A. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, 2013.
[32] R. J. A. Little, "Modeling the drop-out mechanism in repeated-measures studies," *Journal of the American Statistical Association*, vol. 90, pp. 1112-1121, 1995.
[33] Y. Guo and W. Yao, "Applying gated recurrent units pproaches for workload prediction," in *Proc. IEEE/IFIP Network Operations and Management Symposium*, 2018, pp. 1-6.
[34] Google Collaboratory. [Online]. Available: https://colab.research.google.com/

Amine Mrhari Received his B.Sc. degree in Networks and Telecommunications from University of Moulay Ismail, Errachidia Poly-disciplinary Faculty, Morocco, in 2013, and M.Tech. degree in Computer Software Engineering from Hassan I University, Faculty of Sciences and Technologies, Settat, Morocco, in 2015. He is currently working towards the Ph.D. degree with the Department of Computer Research, Ibn Tofail University, Faculty of Sciences, Kenitra, Morocco.
His research interest includes game theory, deep learning, resources management in Cloud computing.

Youssef Hadi (Member, IEEE) received the Ph.D. degree from Mohammed V University, Rabat, Morocco in 2008. Currently, he is a professor and doctoral supervisor at the Department of Computer Research, in Ibn Tofail University, Faculty of Sciences, Kenitra, Morocco. His research interests include cloud computing, deep learning, big data, video indexing.