Performance Comparison of Deep VM Workload Prediction Approaches for Cloud

Yashwant Singh Patel and Rajiv Misra

Abstract With the exponential growth of distributed devices, the era of cloud computing is continued to expand and the systems are required to be more and more energy-efficient with time. The virtualization in cloud manages a large-scale grid-of-servers to efficiently process the demands while optimizing power consumption and energy efficiency. However, to ensure the overall performance, it is critical to predict and extract the high-level features of the future virtual machines (VMs). To predict its load deeply, this paper investigates the methods of a revolutionary machine-learning technique, i.e., deep learning. It extracts the multiple correlation among VMs based on its past workload trace and predicts their future workload with high accuracy. The VM workload prediction helps the decision makers for capacity planning and to apply the suitable VM placement and migration technique with a more robust scaling decision. The effectiveness of deep learning approaches is extensively evaluated using real workload traces of PlanetLab and optimized with selection of model, granularity of training data, number of layers, activation functions, epochs, batch size, the type of optimizer, etc.

Keywords Cloud computing · Deep learning · Energy efficiency
Physical machine (PM) · Virtual machine (VM) · Workload prediction

1 Introduction

Billions of smart devices, i.e., sensors and smart phones that compose the cyber physical systems (CPS) and Internet of things (IoT), will continuously generate huge amount of data than any individual Web application. The digital universe

Y. S. Patel (✉) · R. Misra
Department of Computer Science and Engineering,
Indian Institute of Technology, Patna 801106, India
e-mail: yashwant.pcs17@iitp.ac.in

R. Misra
e-mail: rajivm@iitp.ac.in

P. K. Pattnaik et al. (eds.), *Progress in Computing, Analytics and Networking*,
Advances in Intelligent Systems and Computing 710,
https://doi.org/10.1007/978-981-10-7871-2_15

resides escalating in a computing cloud, higher than terra firma of huge hardware data centers connected to billions of distributed devices, all monitored and controlled by intelligent softwares [1]. Cloud computing is undoubtedly a fine approach to address these staggering requirements. To address the data boom caused by the devices like IoT requires fully controlled cloud services. The cloud service providers have to guarantee the levels of interoperability, portability, and manageability that are almost far away to achieve with the current solutions. Service providers help the companies to select suitable communication hardware and software to support cloud protocols as well as secure remote upgrades. To offer fully managed private, public, and hybrid cloud solutions from a simple development to resource-intensive applications, the cloud infrastructure and platform technologies have to ensure elastic scalability and high-throughput event processing services. To achieve this, the companies have designed open-source distributed database systems for accumulating, processing, and managing large amount of data across commodity clusters and servers. In order to extract the knowledge from the collected data and to feed users of smart city applications, such system follows a typical three-layer architecture. Firstly, the collection layer is responsible to collect data from individual devices and send it to the gateways. In the transmission layer, data is moved from gateways to distributed cloud platforms. At last, at the processing layer data is convoluted in the platform of cloud where the knowledge is extracted and makes available to applications [2]. During such complex process, cloud has promised the vision of computing resources and advances the faster network with lower latency. As the services of cloud computing become well-liked, more and more data centers persisted to be deployed around the globe to remotely deliver the computational power over the Internet. Such data centers acquired a larger fraction of the planet's computing resources. During its management, the service providers will definitely suffered from critical business challenges such as security, privacy, interoperability, portability, reliability, availability, bandwidth cost, performance, cost management, complexity of building cloud, and its environmental impact. But, the major worries while providing the light-speed transfer of data are the increased carbon emissions due to servers. In this reference, energy-efficient management of data center resources is a critical and challenging task while considering operational costs as well as CO2 emissions to the surroundings. In a data center, the long-term operation of servers will not only wear out the equipment, but will also carry the problems of high temperature and energy consumption [3]. A recent report on power consumption of server farms is of evidence that the electricity consumed by servers around the globe accounts to 3% of the global electricity production and about 2% of total greenhouse gas emissions.

Virtual machine (VM)-based distributed and scalable on-demand resource allocation techniques, load balancing approaches, and energy-performance trade-offs while reducing cost and power consumption at the large-scale data centers are the need of time. VM allocation methods try to deploy multiple heterogeneous VMs on each physical machine (PM). In case of high overload situation, i.e., higher than specified threshold of CPU utilization, more VMs are reallocated from one operating PM to another to avoid the violation of service-level

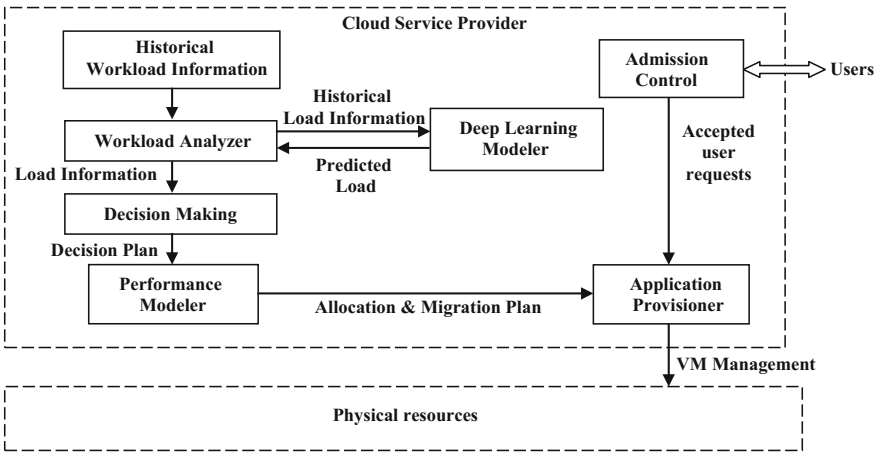


Fig. 1 Utilization-aware workload prediction framework in cloud

agreement (SLA) [4] while the under-utilization, i.e., lower than specified threshold of CPU utilization, will scale down the performance. During such live migration, the overall performance of running applications inside the VM can be impacted negatively [5]. Therefore, by predicting the future workload of VM will not only enhance the overall utilization of resources but also minimize the problem of energy consumption. The prediction of the workload will help the decision makers for capacity planning and applying suitable VM placement as well as migration technique as depicted in Fig. 1 [6].

In this work, we present multi-layer neural networks or popularly known deep learning models to predict the VM workload based on its past workload traces. Deep learning, however, is become a new era of machine learning. It has modernized the machine learning to another level of creating algorithms and to make the system much better analyzer. In recent years, deep learning has reignited the grand challenges of artificial intelligence and become a third boom of AI. It helped the researchers to identify ordinary characteristics of certain objects from the massive amounts of data. The proposed deep learning models will learn the inherent correlated features of VMs workload trace and more effectively predict the workload of future VMs. The predicted load information will be transferred to workload analyzer and then given to the decision-making modeler. It will generate a decision plan of VM management and provide it to performance modeler, and then, the allocation and migration plan choice will be transferred to application provisioner. The application provisioner will receive the accepted user request and apply the suitable VM placement strategy to map the VM to physical servers. In case of overloading, the best migration plans will be selected.

The rest part of this paper is structured as follows: Sect. 2 describes recent works. Section 3 presents performance modeling for utilization-aware workload

prediction. Section 4 elaborates different deep learning techniques for workload prediction. Simulation design and performance evaluations are described in Sect. 5. Finally, the conclusion and future works are discussed in Sect. 6.

2 Background and Related Works

Dynamic VM consolidation approaches are widely known for improving resource utilization and maintain energy efficiency in data centers. In literature, various strategies of VM consolidation have been presented. Bobroff et al. [7] proposed a dynamic server consolidation method for a given workload. In their work, they have integrated bin-packing heuristics and time series-based forecasting to reduce the amount of physical capacity needed to support a specific rate of service-level agreement (SLA) violations. Secron et al. [8] used a threshold value assumption to prevent CPUs from reaching 100 percent utilization that may lead to performance degradation. Beloglazov and Buyya [5] applied a statistical analysis of the historical data and used two thresholds, i.e., upper and lower thresholds. They have divided the VM consolidation technique into detection of (i) host under-load situation, i.e., lower than specified threshold of CPU utilization; (ii) host overload situation, i.e., higher than specified threshold of CPU utilization; followed by (iii) VM selection, i.e., to pick best VMs for migration; (iv) VM placement; and (v) balancing of workload among physical machines, i.e., servers. Therefore, it is superior to predict the future workload rather than monitoring the current workload and applying the migration techniques. Prediction of workload will facilitate the decision maker to plan and deal with the capacity of resources. Such advance load prediction will not only improve the performance of overall system but also make it energy-efficient.

In the field of VM workload prediction, numerous approaches have been proposed. Dorian and Freisleben [9] proposed artificial neural networks (ANNs)-based distributed resource allocation approach to find best VM allocations while optimizing utility function. Bitirgen et al. [10] used ANN-based model to support online model training by predicting the performance. In their work, they have considered resource allocation at the multiprocessor chip level. Zhen et al. [11] presented an exponentially weighted moving average (EWMA) approach for short-term prediction of CPU load. Kousiouris et al. [12] proposed a GA-based approach based on the artificial neural networks (ANNs) for workload prediction. Calheiros et al. [13] proposed a proactive method for dynamic provisioning of resources. It is based on autoregressive integrated moving average (ARIMA) model that employs linear prediction structure and predicts future workload by using real-world traces. Fahimeh et al. [14] proposed k-nearest neighbor regression-based model to predict the future utilization of resources. Their utilization prediction-aware best fit decreasing (UP-BFD) method optimizes the VM placement by considering present and upcoming resource requirements. Feng et al. [15] proposed a deep belief network (DBN) that contains multiple-layered restricted Boltzmann machines (RBMs) and a regression layer to predict the workload of

future VM. Authors have evaluated its performance with the existing literature work such as EWMA and ARIMA method. All of these existing approaches have used a linear prediction model and implemented a very low dimension structure. These approaches give low performance when long-time workload prediction is required and inherent VM features required to be extracted in complex cloud network. To resolve these issues, this work applies and investigates deep learning techniques to identify inherent correlation of VMs from the massive amounts of workload trace and predict the future workload of VMs.

3 Performance Modeling for Utilization-Aware Prediction

Let X(t) be the set of all past CPU utilization trace as the time intervals of every 5 min and Y(t) be the CPU utilization at the next time (t + 1). To predict the CPU utilization at (t + 1), the past information of CPU utilization at previous time intervals (t − 1) and (t − 2) will be used. This problem can be modeled as a regression problem, where the input time values are (t − 2), (t − 1), t is given, and the predicted output value is (t + 1). The mean absolute error can be calculated by Eq. (1):

$$\text{Mean Absolute Error} = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|x_i^a - x_i^p\right|}{x_i^a} \tag{1}$$

where n is the prediction intervals, $\mathbf{x_i^a}$ is the actual CPU utilization value, and $\mathbf{x_i^p}$ is the predicted CPU utilization value.

The performance modeler will use this predicted CPU utilization for allocation or migration of VM at the destination host with the following constraints of Eq. (2) [14]:

$$PU_{cpu}(VM) + CU_{cpu}(PM) \leq TH_{cpu} X\, TU_{cpu}(PM) \tag{2}$$

where $PU_{cpu}(VM)$ is a predicted CPU utilization of VM, $CU_{cpu}(PM)$ is the current CPU utilization of PM, TH_{cpu} is the threshold value, and $TU_{cpu}(PM)$ is the total CPU utilization of PM. If the Eq. (2) is satisfied, the VM placement can be performed at destination PM and the status of PM will be updated as shown in Eq. (3):

$$TU_{cpu}(PM) = PU_{cpu}(VM) + CU_{cpu}(PM) \tag{3}$$

During the case of hot spot mitigation or VM migration, the new PMs are required to be searched (or) the idle PMs are to be switched in active state. The violation can be formulated by Eq. (4):

$$PU_{cpu}(VM) + CU_{cpu}(PM) \geq TH_{cpu} X\, TU_{cpu}(PM) \tag{4}$$

Overall CPU load of PM denoted by $L_{cpu\ (PM)}$ can be calculated by Eq. (5):

$$L_{cpu(PM)} = \frac{CU_{cpu(PM)}}{TU_{cpu(PM)}} \tag{5}$$

where $CU_{cpu}(PM)$ is the current CPU load of PM, and $TU_{cpu}(PM)$ is the total CPU load of PM. Threshold values of PM can be defined though Eq. (6):

$$TH_{Lcpu}(PM) \leq TU_{cpu}(PM) \leq TH_{Ucpu}(PM) \tag{6}$$

where TH_{Lcpu} (PM) denotes lower threshold value of PM, and TH_{Ucpu} (PM) denotes upper threshold value of PM.

Case of host under-load situation, i.e., lower than specified threshold of CPU utilization, is shown in Eq. (7):

$$TU_{cpu}(PM) \leq TH_{Lcpu}(PM) \tag{7}$$

This is also called the case of cold spot; if it is satisfied, then there is a need of VM migration so that the PM can be switched off or switched to sleep state and rest PM can be utilized. It will reduce the number of active PMs and improve the degree of energy efficiency.

Case of host overload situation, i.e., higher than specified threshold of CPU utilization, is shown in Eq. (8):

$$TU_{cpu}(PM) \geq TH_{Ucpu}(PM) \tag{8}$$

This is also called the case of hot spot; if it is satisfied, then there is a need of VM migration so that the unnecessary SLA violation can be avoided and other PMs will be searched to satisfy the increased demand of particular VM.

4 Deep Learning-Based Workload Prediction Techniques

The existing workload prediction approaches apply the statistical analysis of the workload trace, i.e., CPU, memory, disk, and bandwidth and predict the future workload by identifying variations in workload trace. To deeply analyze the workload variations, the depth of layers in a neural network becomes a critical factor and gave birth to "deep learning—a revolutionary machine-learning technique." The problem of workload prediction is assumed to be a time series-based regression problem and solved with powerful deep learning approaches. The input of these models is workload trace of VMs recorded in different time intervals, and output is predicted load of future VMs. These approaches apply the technique of unsupervised learning, where only a little knowledge of resources is provided. Accuracy of prediction can be improved with the number of hidden layers, epochs,

batch size, activation functions, and type of optimizer. Different deep learning models that are used in this work are discussed as follows:

4.1 Recurrent Neural Network (RNN) Model

Recurrent neural networks (RNN) [16, 17] are known to be a complement set of classical neural networks, i.e., feed-forward network. It removes the constraint of passing the information in forward manner and improves the model by providing at least one feed-backward edge.

4.2 Long Short-Term Memory (LSTM) Network Model

LSTM model was proposed by Hochreiter et al. in 1997 [18], Wang and Raj [17]. It tries to contest the vanishing gradient problem with the help of gates and an explicitly defined memory cell. Each neuron has three gates, i.e., input, output, and forget, along with memory cell. The input gate decides that how much information of the previous layer is required to be stored in the cell. The output layer decides how much of cell state to be known by the next layer, and the forget gate is for erasing the few content of previous layer. LSTM has been widely applied in several real-world problems.

4.3 Boltzmann Machine Model

It is also known as a hidden unit version of Hopfield network. It is a fully connected network made by hidden and visible units. In this model, few neurons are marked as input while others are hidden. It initiates with random weights and learns through contrastive divergence. The process of training and running is similar to Hopfield. It is inspired from physics where the rise in temperature causes the state transfer. The energy function of Boltzmann machine can be represented by Eq. (9):

$$E(v,h) = -\sum_i v_i b_i - \sum_k h_k b_k - \frac{1}{2}\sum_{i,j} v_i v_j w_{i,j} - \frac{1}{2}\sum_{i,k} v_i h_k w_{i,k} - \frac{1}{2}\sum_{k,l} h_k h_l w_{k,l} \quad (9)$$

where v defines visible units, h defines hidden units, w defines weights, and b is for bias. The global temperature value controls the activation; if it is minimized, then the energy of the cells decreases. The right temperature to the network achieves an equilibrium state [17].

4.4 Convolution Neural Network (CNN) Model

CNN is also known as LeNet. It is based on traditional multiple layer perceptrons. It was proposed by LeCun et al. in 1998 [19]. It applies convolution and sub-sampling operation alternatively on input data by using different computational units in convolutional and sub-sampling layers. After this, the data represented in higher layers fed to a fully connected network and complete the task.

5 Simulation Design and Performance Evaluation

5.1 Simulation Design

To evaluate the efficiency of deep learning approaches, a real workload trace of PlanetLab [20, 21] is used. It is a widely popular open platform that contains the CPU utilization of over 1000 VMs. This data is collected in every five minutes and stored in different files. For one VM, there are total 288 observations per day [22].

5.2 Performance Evaluation

In this work, the input of model is the CPU utilization of VMs and output is future CPU utilization. The data set is divided into two parts, i.e., training set and test set. The training set includes the CPU utilization of VMs recorded in 7 days, and remaining is kept for test set. For single VM workload prediction, there are total 2880 time intervals out of which 70% is used for training and 30% is used for test. Then, the deep learning models are trained with different sizes and features as shown in Table 1 and tested with unknown data set. The performance of deep learning models, i.e., multi-layer NN, convolutional NN, recurrent NN, Boltzmann, and LSTM NN for single VM workload prediction during long time intervals of testing data, is represented in Fig. 2. It can be observed that the predicted utilization of CPU by LSTM network is too close to actual workload, while the convolutional network gives low performance in comparison with other techniques.

Table 1 Experimental configuration

Layers	Epochs	Batch size	Activation function	Loss function	Optimizer
3	50	10	Relu, SoftMax	MSE	SGD
5	100	50	Relu, SoftMax	MSE	Adam
10	150	100	Relu, SoftMax	MSE	Adamax

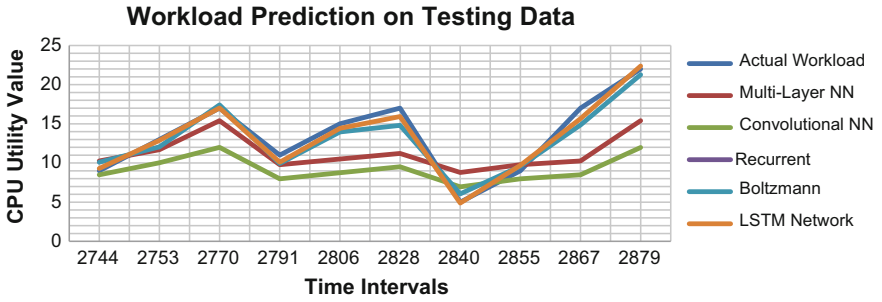
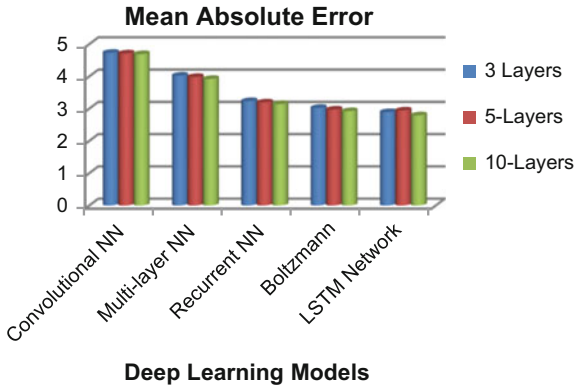


Fig. 2 Performance comparison of deep learning approaches for long-term workload prediction of single VM

Overall analysis of mean absolute error for single VM is represented in Fig. 3. The deep learning models are also depended on the amount of training data. As much as we increase the amount of training data, it will improve the accuracy of prediction. In case of multiple VMs, we have selected 10 continuously running VMs and plotted its average mean absolute error in Fig. 4. The LSTM network model gives minimum average mean absolute error and performs better than other deep learning models. It is advantageous in the case of multiple VM workload prediction during long time intervals. These deep learning models are beneficial during VM management and help the decision makers to pre-plan the VM placement and migration strategies. The overall performance of deep learning models can be arranged as:

$$(\text{Low})\text{Convolutional} < \text{Multi} - \text{layer} < \text{Recurrent} < \text{Boltzmann} < \text{LSTM}(\text{High})$$

Fig. 3 Performance comparison for single VM workload prediction



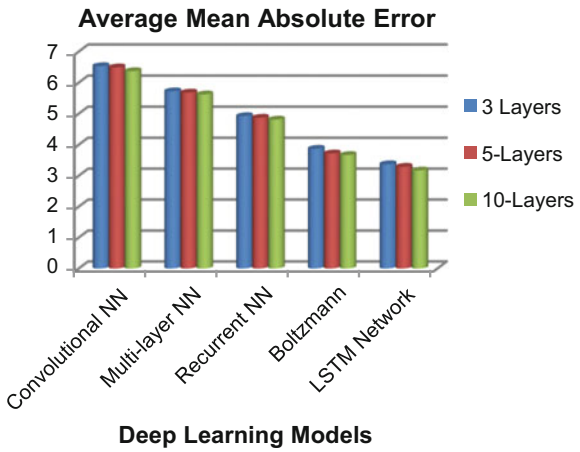


Fig. 4 Performance comparison for 10 VMs workload prediction

6 Conclusion and Future Works

In this work, we have presented different utilization-aware prediction models of deep learning approaches. It extracts the high-level features from the workload trace of past VMs and predicts their future workload with high accuracy. The prediction of the workload can help the decision makers to estimate the overall capacity and to apply the suitable VM placement as well as migration technique. The proposed framework will support the prediction of large-scale data intensive systems for distributed decision making such as hot spot mitigation, cold spot mitigation, threshold violation, and SLA violation. The accuracy of deep learning approaches is evaluated using real workload traces and shown with the help of experimental results. The results are promising and show that the LSTM-based networks improve the performance of workload prediction while convolutional NN gives low performance. Deep learning approaches are suitable for long-term prediction of workloads. The performance of the deep learning approaches can be improved further by increasing size of training data and depth of the model. This will help the model to find more correlation between workload patterns and determine the load with dynamic requirements. In future, we will try to investigate more robust and efficient approaches of workload estimation while coordinating with multi-tier applications and multi-tier VMs running on heterogeneous PMs in real cloud platform.

References

1. The digital universe in 2020, https://www.emc.com/leadership/digital-universe/2012iview/executive-summary-a-universe-of.htm (2012).

2. F. Tao, L. Zhang, V.C. Venkatesh, Y. Luo, and Y. Cheng, Cloud manufacturing: A computing and service-oriented manufacturing model, in Proc. Inst. Mech. Eng. B—J. Eng. Manuf., 225 (10), (2011) 1969–1976.
3. C.C. Lin, P. Liu, and J.J. Wu, Energy-efficient virtual provision algorithms for cloud systems, 4th IEEE International Conference on Utility and Cloud Computing (2011) 81–88.
4. M. Mishra, A. Das, P. Kulkarni, and A. Sahoo, Dynamic resource management using virtual machine migrations, IEEE Communications Magazine (2012) 34–40.
5. A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," Concurrency and Computation: Practice and Experience, 24 (13), (2012) 1397– 1420.
6. R. N. Calheiros, E. Masoumi, R. Ranjan and R. Buyya, Workload Prediction Using ARIMA Model and Its Impact on Cloud Applications' QoS, in IEEE Transactions on Cloud Computing, 3(4), (2015) 449–458.
7. N. Bobroff, A. Kochut, and K. Beaty, Dynamic placement of virtual machines for managing sla violations, in Integrated Network Management, IM'07. 10th IFIP/IEEE International Symposium (2007) 119–128.
8. A. Murtazaev and S. Oh, Sercon: Server consolidation algorithm using live migration of virtual machines for green computing, TE Technical Review, 28(3), (2011) 212–231.
9. D. Minarolli and B. Freisleben, "Distributed Resource Allocation to Virtual Machines via Artificial Neural Networks," 22nd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, Torino (2014) 490–499.
10. R. Bitirgen, E. Ipek, and J.F. Martinez. Coordinated management of multiple interacting resources in chip multiprocessors: A machine learning approach. In Proc. 41st Annual IEEE/ACM International Symposium on Microarchitecture (2008) 318–329.
11. Z. Xiao, W. Song and Q. Chen. Dynamic resource allocation using virtual machines for cloud computing environment, IEEE Transactions on Parallel and Distributed Systems, 24(6), (2013) 1107–1117.
12. G. Kousiouris, A. Menychtas, D. Kyriazis, et al. Parametric design and performance analysis of a decoupled service-oriented prediction framework based on embedded numerical software, IEEE Transactions on Services Computing, 6(4), (2013) 511–524.
13. R. Calheiros, E. Masoumi, R. Ranjan and R. Buyya. Workload Prediction Using ARIMA Model and Its Impact on Cloud Applications' QoS, IEEE Transactions On Cloud Computing, 3 (4), (2015) 449–458.
14. F. Farahnakian, T. Pahikkala, P. Liljeberg, J. Plosila and H. Tenhunen, Utilization Prediction Aware VM Consolidation Approach for Green Cloud Computing, IEEE 8th International Conference on Cloud Computing, New York City, NY, (2015) 381–388.
15. F. Qiu, B. Zhang and J. Guo, "A deep learning approach for VM workload prediction in the cloud, 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Shanghai (2016) 319–324.
16. Ken-ichi Funahashi, Yuichi Nakamura, Approximation of dynamical systems by continuous time recurrent neural networks, Neural Networks, 6(6), (1993) 801–806.
17. H. Wang and B. Raj, A survey: Time travel in deep learning space: An introduction to deep learning models and how deep learning models evolved from the initial ideas, arXiv preprint arXiv:1510.04781 (2015).
18. Sepp Hochreiter and Jürgen Schmidhuber, Long short-term memory. Neural computation 9 (8), (1997) 1735–1780.
19. LeCun, Yann, et al., Gradient-based learning applied to document recognition, Proceedings of the IEEE, 86(11), (1998) 2278–2324.
20. K. Park and V. Pai, CoMon: a mostly-scalable monitoring system for PlanetLab, ACM SIGOPS Operating Systems Review, 40, (2006) 65–74.

21. R.N. Calheiros, R. Ranjan, A. Beloglazov, C. Rose and R. Buyya, CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, Software: Practice and Experience, 41(1), (2011) 23–50.
22. Beloglazov A, Energy-efficient management of virtual machines in data centers for cloud computing, PhD thesis (2013).