2021 International Conference on Information and Communication Technology Convergence (ICTC) | 978-1-6654-2383-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICTC52510.2021.9620931

Multivariate Deep Learning Model For Workload Prediction In Cloud Computing

1st NHAT-MINH DANG-QUANG
Department of Information Communication Convergence Technology
Soongsil University
Seoul, Korea

2nd MYUNGSIK YOO
School of Electronic Engineering
Soongsil University
Seoul, Korea

***Abstract*—Many application developers are now choosing to install their Web applications on cloud data centers because of the attractiveness of cloud computing environment. Predicting future resource workload is critical since it allows cloud service providers to automatically modify resources online in order to meet service level agreements (SLA). This paper proposes a multivariate deep learning prediction model to predict future resource workload for cloud computing environment. The prediction model uses a special type of recurrent neural network (RNN) called Bidirectional long short-term memory (Bi-LSTM). This work also explains and shows the advantage by using multivariate data compared to univariate data in time series forecasting. The experiments, using real world workload dataset, show that the proposed multivariate Bi-LSTM model outperforms the univariate Bi-LSTM model in prediction accuracy.**

***Index Terms*—Workload prediction, time series forecasting, cloud computing.**

I. Introduction

Recently, with the rapid development of cloud computing, virtualization technologies have become a popular topic in both academy and industry due to they enable cloud systems like Google Cloud Platform [1], Amazon Web Services [2], Microsoft Azure [3]. Traditional hypervisor virtualization technology is a technique, which is used to create virtual machines (VMs) with specific resources (CPU, RAM, Disk space, etc.) and operating systems (OSs). ESX, Xen, and Hyper-V are some of the most widely used hypervisors in the world. Container-based virtualization is a lightweight alternative to virtual machines (VMs), which can assist reduce startup time and resource consumption compared to VMs. LXC, Docker, Kubernetes, and Kata are some well-known example of this technology.

The main feature of cloud computing is elasticity, which enables service owners to provision or de-provision resources follow unpredictable workload demands, which are inherent in internet-based services [4]. By predicting workload demands in the near future, service owners can acquire and release resources ahead of time. However, workload demands in cloud computing are hard to predict because it fluctuates continuously by time. Many resource estimation studies have been published [5]–[9] and more. These works commonly use a technique called time series forecasting, which is explained in Section II, to predict resource workload in the near future. Furthermore, these works focus on univariate time series analysis, which means they only analyze series of measurements of a single feature such as CPU utilization or RAM or network throughput, etc., across time and give predictions. However, in a cloud system, we have multiple features to consider. Some of them can have hidden correlations and depend on each other. Multivariate time series analysis can analyze these hidden correlations of temporal data and better understand the whole system, thus give better predictions.

The large dimensionality and spatial-temporal dependency characteristics of multivariate time series data, as well as the presence of noisy data, make it challenging to model effectively using traditional statistical approaches [10]. Artificial intelligence, especially deep learning, has been used in a variety of domains such as computer vision, speech recognition, and biological signal analysis [9]. Although the traditional statistical method can be used for modeling time series data, deep learning-based forecasting time series data are becoming popular [8], [9]. Therefore, this paper proposed a multivariate Bi-LSTM prediction model for analyzing and predicting resource workload in cloud computing environment.

The rest of this paper is organized as follows. Section II describes the background knowledge in time series forecasting. Section III discuss about the related works followed by the proposed prediction model in Section IV. Section V evaluates the experiments and evaluations of the proposed prediction model. Section VI gives the conclusions and future works of this paper.

II. Background

Time series forecasting is a popular technique for analyzing the behavior of temporal data and predict future values. It's widely used in industries [11], including air quality forecasting [12], anomaly detection [13], and medical monitoring [14], etc. Typically, time series data have a natural temporal ordering. Its type consists of univariate time series data and multivariate time series data.

A univariate time series is a series of measurements of a single variable across time. Univariate refers to a single variable or variate. A multivariate time series is a collection of variables measured across time: a number of different variables or variates.

Analyzing multivariate time series data allow researchers to look at relationships between variables in an overall way and to quantify the relationship between variables. They can use cross-tabulation, partial correlation, and multiple regressions to control the relationship between variables, as well as introduce other variables to determine the links between the independent and dependent variables or to indicate the conditions under which the association occurs. Multivariate analysis has the advantage of providing a more realistic picture than single variable analysis. Furthermore, when compared to univariate procedures, multivariate techniques provide a more powerful test of significance [15].

III. Related work

In this section, we review some recent works related to resource estimation in cloud computing environments using time series forecasting.

TABLE I
Review of related works

Paper	Feature	Technique
Li and Xia [6]	CPU usage	ARIMA
Ciptaningtyas et al. [7]	HTTP workload	ARIMA
Imdoukh et al. [8]	HTTP workload	LSTM
Dang-Quang and Yoo [9]	HTTP workload	Bi-LSTM
Tang et al. [16]	CPU usage	Bi-LSTM

Traditional statistical approaches in time series analysis include Auto-Regression (AR), Moving Average (MA), Autoregressive Moving Average (ARMA), and Autoregressive Integrated Moving Average (ARIMA).

Li and Xia suggested a platform for auto-scaling web applications on hybrid clouds in [6]. The proposed prediction model uses ARMA to analyze historical CPU usage data and predict the resource needed in the future to provision ahead of time. However, they did not compare with other works.

ARIMA model was also employed by Ciptaningtyas et al. [7] to forecast the amount of future workload requested, as it has a higher accuracy for short-term forecasting. The used ARIMA model to analyze and predict the HTTP workload coming to the system. They tested the following four ARIMA models, which had the same degree of differencing (d) as 1 and the same order of moving average (q) as 0, but different lag order values (p) ranging from 1 to 4. The model with the lag order values of 4 has the best prediction accuracy.

Although these statistical methods have high accuracy in short-term forecasting. However, they have bad accuracy when dealing with long-term forecasting [7]. This is when machine learning techniques, especially deep learning, are introduced to analyze the time series data.

Imdoukh et al. [8] proposed to use a special type of RNN called LSTM to analyze and predict the HTTP workload in containerized application. The results indicated that the proposed LSTM model has has a slightly higher prediction error than the ARIMA model in one-step and multi-step forecasting. But the computation speed is hundreds of times faster.

Dang-Quang and Yoo [9] proposed to use an extension of LSTM called Bi-LSTM to analyze and predict HTTP workload in Kubernetes [17]. The proposed Bi-LSTM prediction model was evaluated with the LSTM model of the work [8] and ARIMA model of the work [7]. The results show that the Bi-LSTM model has higher prediction accuracy compared to LSTM and ARIMA modelS, while maintaining same computation speed as LSTM.

Tang et al. [16] presented a container load prediction model based on the Bi-LSTM model, which predicts future load based on the container's historical CPU usage load. When compared to the ARIMA and LSTM models, the proposed model had the lowest prediction error. The authors, on the other hand, make no indication of how to set up the parameters of the proposed model.

As mentioned earlier, these works focus on univariate time series analysis, which means they only analyze series of measurements of a single feature such as CPU utilization or RAM or network throughput, etc., across time and give predictions. However, some of these features can have hidden correlations and depend on each other. Thus, by using multivariate time series analysis, the prediction model can analyze these hidden correlations of temporal data and better understand the whole system, thus give better predictions. For those reasons, this paper propose a multivariate Bi-LSTM prediction model to forecast resource workload in cloud computing environment.

IV. Proposed prediction model

This section discuss about the proposed prediction model.

A. Multivariate bidirectional long short-term memory prediction model

The regular RNN or LSTM takes the input of time series data with forward direction. This is a limitation due to it ignores the continuous data changes [9]. Then, LSTM can only capture partial information. Bi-LSTM [18] is an extension of LSTM, which has two LSTM layers applied to the input data. The first LSTM layer is fed with the input sequence in its original direction (forward). The second LSTM layer is fed with the reverse sequence of data. The Bi-LSTM takes input in two directions, one from the past to the future (forward direction) and the other from the future to the past (backward direction). By doing this, it can preserve information from the past to the future as well as the future to the past. By applying two LSTM layers stack on each other, it can help improve learning long-term dependencies and thus consequently will improve the accuracy of the prediction model [19].

A multivariate time series usually consists of multiple variable values at each observation time-step in time series forecasting. Each variable in a multivariate time series is depended on not just by its previous values but also by the values of other variables in some instances. These correlations are important when we model the multivariate time series data. The ability to learn the correlation characteristics of the multiple monitored variables in multivariate time series datasets is critical for modeling temporal data. For those reasons, this paper proposes a multivariate Bi-LSTM prediction

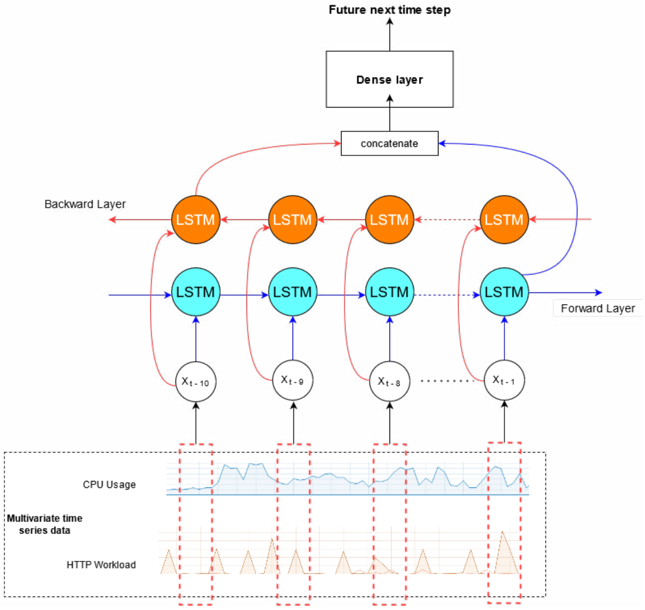

Fig. 1. Proposed Bi-LSTM neural network architecture.

model to forecast the resource workload in cloud computing environment.

B. Neural network architecture

Figure 1 shows the proposed Bi-LSTM neural network architecture. In this paper, for the multivariate data, we choose the CPU usage and HTTP workload as the inputs data at each time step for the prediction model. These parameters are both well used separately in forecasting problems in both academy and industry works. Besides, these parameters may have some correlations. By monitoring, we can observe that if the HTTP workload coming to the system increases, so does the CPU usage because the system has to process more works. Then, combining these parameters as the input of the proposed neural network can help the prediction model learns the correlation for better understanding the system, thus give better prediction decisions.

The proposed prediction model has two LSTM layers move forward and backward. Each layer consists of 10 neural cells for the last 10 time steps, with 30 hidden units for each neural cell. Both outputs of backward and forward layers will be concatenated and used as input the Dense layer. The Dense layer is used to output predictions. The predicted output of the proposed neural network architecture can be the value of CPU usage or HTTP workload or both of them in the next time step.

V. Experiment and Evaluation

In this section, we evaluate the performance of proposed Bi-LSTM multivariate prediction model described in Section IV with the univariate Bi-LSTM prediction model. The models are evaluated using real trace workload dataset GWA-T-12 Bitbrains [20] from the Delft University of Technology.

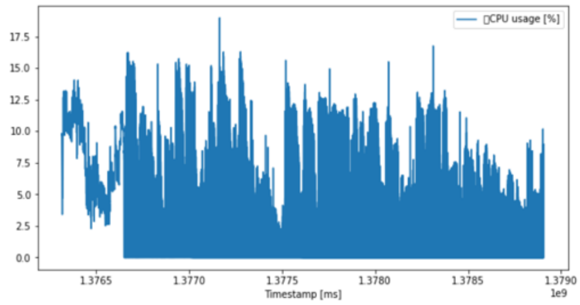

Fig. 2. Dataset CPU usage

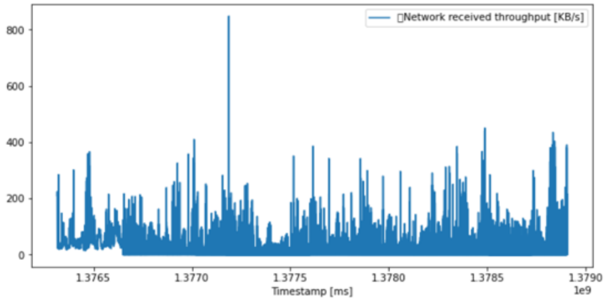

Fig. 3. Dataset Network receive throughput

A. Dataset

The GWA-T-12 Bitbrains contains performance metrics of 1,750 VMs from a distributed datacenter from Bitbrains. In this experiment, we use the first trace, fastStorage, which consists of 1,250 VMs that are connected to fast storage area network (SAN) storage devices. Each file in the dataset is made up of a series of lines, each of which represents a 300-ms observation of a virtual machine's performance data since 2018-01-01. We use the metric of CPU usage and network receive throughput, as shown in Figure 2 and 3, for this experiment.

As described in Section IV, the inputs of the proposed multivariate Bi-LSTM prediction models are network receive throughput and CPU usage. Besides that, the univariate Bi-LSTM will receive the network throughput as its input. Both prediction models will output the future network receive throughput workload. We split the dataset into 80% for training and 20% for testing. We also normalize the dataset to range within [0,1] by using Equation 1.

$$z = \frac{x - min(x)}{max(x) - min(x)} \tag{1}$$

B. Evaluation metric

To evaluate accuracy of the prediction models, we use the popular metric in evaluating regression models, root mean square error (RMSE). The RMSE metric is described in Equation 2. It is the square root error of the differences between the predicted and actual values. $\mathbf{Y_i}$ and $\mathbf{\hat{Y}_i}$ are the actual value and forecast value, respectively.

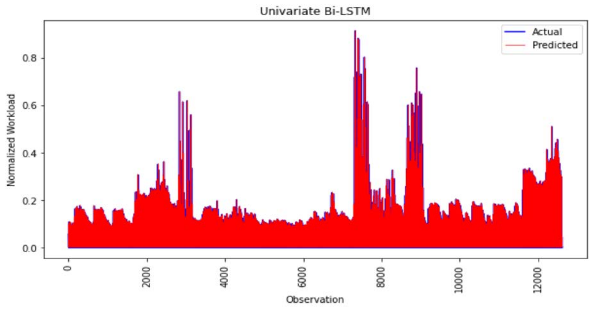

Fig. 4. Result of univariate Bi-LSTM on testing set

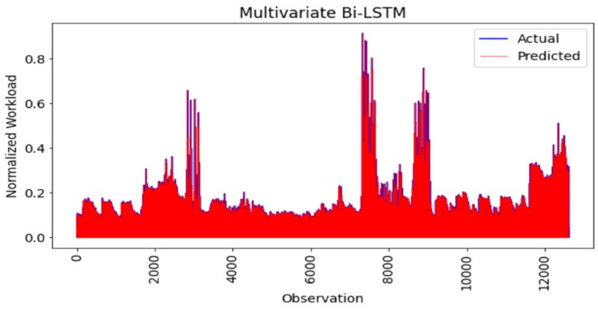

Fig. 5. Result of multivariate Bi-LSTM on testing set

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2} \qquad (2)$$

C. Experiment results

Table II shows the experiment results of the univarite and the proposed multivariate Bi-LSTM prediction models. The proposed multivariate Bi-LSTM prediction model has smaller prediction error than the univariate Bi-LSTM prediction model. It reduces prediction error by 46% on RMSE metric. Figure 4 and 5 present the result of the predicted value versus the observed value. The performance of 2 prediction models appears to be similar. Then we take a closer look by zooming into the result. We show the absolute error of models for the first 100 observations of the testing set in Figure 6. We can observe that univariate Bi-LSTM model has higher error than the proposed multivariate Bi-LSTM model. Thus, the proposed multivariate Bi-LSTM model is better than the univariate Bi-LSTM model.

TABLE II
EXPERIMENT RESULTS

Prediction model	**RMSE**
Univariate Bi-LSTM	0.000260
Multivariate Bi-LSTM	0.000139

VI. Conclusion

Predicting future resource workload is critical since it allows cloud service providers to automatically modify resources online in order to meet SLA. However, workload demands in cloud computing are hard to predict because it fluctuates continuously by time. Besides, most recent works just focusing on univariate time series analysis, which means they only analyze series of measurements of a single feature across time and give predictions. Some of these features can have hidden correlations and depend on each other. Thus, by using multivariate time series analysis, the prediction model can analyze these hidden correlations of temporal data, better understanding the whole system, thus give better predictions. This is shown by experiment results on real world workload dataset of GWA-T-12 Bitbrains. For future works, we plan to do more experiments on different datasets for the proposed prediction model and extend the work to the autoscaling problem.

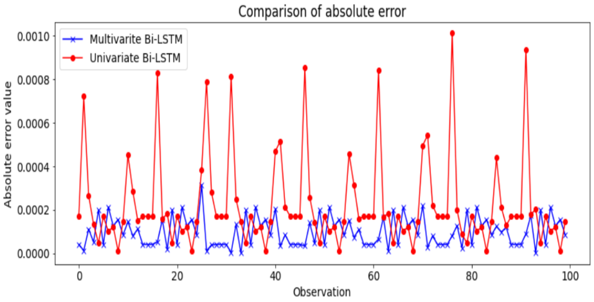

Fig. 6. Absolute error of the first 100 observations

Acknowledgment

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2021-0-02046) supervised by the IITP (Institute of Information & Communications Technology Planning & Evaluation)

References

[1] Google cloud platform. https://cloud.google.com. Accessed on 2021-08-01.

[2] Amazon web services. https://aws.amazon.com. Accessed on 2021-08-01.

[3] Microsoft azure. https://azure.microsoft.com. Accessed on 2021-08-01.

[4] Hector Fernandez, Guillaume Pierre, and Thilo Kielmann. Autoscaling web applications in heterogeneous cloud infrastructures. In *2014 IEEE International Conference on Cloud Engineering*, pages 195–204, 2014.

[5] Issaret Prachitmutita, Wachirawit Aittinonmongkol, Nasoret Pojjanasuksakul, Montri Supattatham, and Praisan Padungweang. Auto-scaling microservices on iaas under sla with cost-effective framework. In *2018 Tenth International Conference on Advanced Computational Intelligence (ICACI)*, pages 583–588, 2018.

[6] Yunchun Li and Yumeng Xia. Auto-scaling web applications in hybrid cloud based on docker. In *2016 5th International Conference on Computer Science and Network Technology (ICCSNT)*, pages 75–79, 2016.

[7] Henning Titi Ciptaningtyas, Bagus Jati Santoso, and Muhammad Fahrur Razi. Resource elasticity controller for docker-based web applications. In *2017 11th International Conference on Information Communication Technology and System (ICTS)*, pages 193–196, 2017.

[8] Mahmoud Imdoukh, Imtiaz Ahmad, and Mohammad Alfailakawi. Machine learning based auto-scaling for containerized applications. *Neural Computing and Applications*, pages http://link.springer.com/article/10.1007/s00521–019, 07 2020.

[9] Nhat-Minh Dang-Quang and Myungsik Yoo. Deep learning-based autoscaling using bidirectional long short-term memory for kubernetes. *Applied Sciences*, 11(9), 2021.

[10] Tak chung Fu. A review on time series data mining. *Engineering Applications of Artificial Intelligence*, 24(1):164–181, 2011.

[11] Jan G. De Gooijer and Rob J. Hyndman. 25 years of time series forecasting. *International Journal of Forecasting*, 22(3):443–473, 2006. Twenty five years of forecasting.

[12] A distributed spatial–temporal weighted model on mapreduce for short-term traffic flow forecasting. *Neurocomputing*, 179:246–263, 2016.

[13] Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, 262:134–147, 2017. Online Real-Time Learning Strategies for Data Streams.

[14] Shengdong Du, Tianrui Li, Yan Yang, and Shi-Jinn Horng. Multivariate time series forecasting via attention-based encoder–decoder framework. *Neurocomputing*, 388:269–279, 2020.

[15] Multivariate techniques: Advantages and disadvantages. https://www.theclassroom.com/multivariate-techniques-advantages-disadvantages-8247893.html. Accessed on 2021-08-01.

[16] Xuehai Tang, Qiuyang Liu, Yangchen Dong, Jizhong Han, and Zhiyuan Zhang. Fisher: An efficient container load prediction model with deep neural network in clouds. pages 199–206, 12 2018.

[17] Kubernetes. https://kubernetes.io/. Accessed on 2021-08-01.

[18] M. Schuster and K.K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[19] Lavanya Ramakrishnan and Beth Plale. A multi-dimensional classification model for scientific workflow characteristics. In *Proceedings of the 1st International Workshop on Workflow Approaches to New Data-Centric Science*, Wands '10, New York, NY, USA, 2010. Association for Computing Machinery.

[20] Gwa-t-12 bitbrains. http://gwa.ewi.tudelft.nl/datasets/gwa-t-12-bitbrains. Accessed on 2021-09-27.