PD-GABP - A Novel Prediction Model Applying for Elastic Applications in Distributed Environment

Dang Tran, Nhuan Tran, Binh Minh Nguyen*, Hieu Le
School of Information and Communication Technologies,
Hanoi University of Science and Technology, Hanoi, Vietnam
dangtv18@gmail.com, tranducnhuan1994@gmail.com,
minhnb@soict.hust.edu.vn, hieulq19@gmail.com

***Abstract*—In comparison with other scaling techniques, forecast of workload and resource consumption brings a great advantage to SaaS operations in cloud environment because system knows early and precisely the number of resources must be increased or decreased. However, the prediction accuracy still needs to be improved further even though there are many research works that have dealt with the problem. In this paper, we present a novel prediction model, which combines periodicity detection technique and neural network trained by genetic-back propagation algorithm to forecast the future values of time series data. The model is experimented with real workload dataset of a web application. The tests proved significant effectiveness of the model in improving the prediction accuracy. Our model thus can enhance the performance of applications running on cloud and distributed environment.**

***Index Terms*—Periodicity detection, neural network, genetic algorithm, adaptive resource management, cloud computing.**

I. Introduction

The Software-as-a-Service (SaaS) is an emerging concept of a software deployment model where the applications are licensed to customers as an on demand service, using a business model of pay-per-use. Even though it has become a mainstream solution, from the view of application developers, several problems still remain. Firstly, the deployment and configurability for the internet applications need to be automatic and flexible in order to support a one-to-many consumer model. After that the frequent change of user quantity and their requirements leads to the hardware system also must have ability to provide elastically by time.

In the meantime, virtualization has greatly improved and become a de-facto standard resolving the problems of SaaS. The technology allows costs reductions (hardware and energy), better resource allocations and centralized management [1]. Some public virtualization providers offer the benefits without the hardware costs for users. These providers are often called cloud service vendors.

However, there are also still concerns in porting applications into clouds. One of them is how to exploit completely the elasticity feature of virtualization for SaaS applications in the manner of controlling effectively and automatically the increase and decrease of resources according to user demands, which usually change continuously by time. Currently, most of cloud providers provide functionalities or services to measure the application resource consumptions, some others provide load balancers [2]. These solutions aim at enabling resource elasticity for the applications. Unfortunately, in some cases, the solutions are not enough to ensure the 100 percent capability to serve the applications with actual operations. It can be seen that the elasticity is a vital point for SaaS application operations. According to their approaches, scaling techniques in cloud and distributed environment can be divided roughly into three types as follows:

- Periodicity: ordinarily, during operations, SaaS appliances have resource consumption periods by time (i.e. hourly, daily or monthly). Based on that, administrators will decide a suitable moment to scale resources for applications. The disadvantage is that using periods to scale, the system cannot meet all sudden requirements from applications. Theoretically, some research works dealt with period detection such as [3] and [4].
- Thresholds: developers can set thresholds that determine when resources will be increased or decreased. These thresholds operate based on resource consumption figures such as usage percent of CPU, memory or connections. The disadvantage is that it is very difficult to set precisely thresholds in the manner of satisfying application requirements but still avoid wasting or lacking resources (since scaling early or tardily). In addition, at the scaling moment, the system status and requirements could be already altered, therefore the scaling also cannot achieve the desired effects.
- Prediction: this scaling type relies on data collected in the past and predicts resource usage or workload in the future in order to determine early and precisely resource amount that system has to provide for applications. There are many research works, which brought out the different theory models for prediction [5], [6] and [7], especially for cloud computing [8], [9]. However, the accuracy of prediction models still need more to improve.

In this paper, we present a novel prediction model applying to SaaS applications. Because almost all SaaS applications produce time series data of resource consumption or workload, and the data often has periodicities, our model is formed by combining periodicity detection and neural network trained by genetic-backpropagation algorithm to forecast the future

*Corresponding author

values. The model is experimented with FIFA World Cup 98 site workload dataset [10]. The data represents one of monitoring factors, which are used to make elasticizing cloud resources decisions for SaaS applications. The achieved results proved that the accuracy of our model is better in comparison with traditional prediction methods such as back-propagation neural network, neural network trained by genetic algorithm.

The rest of this paper is organized as follows. Section 2 gives and discusses related works to time series forecasting. Section 3 presents our design for proposed prediction model with its parameters. Section 4 introduces PD-GABP algorithms, which is core of the prediction model. In Section 5, we provide experiments and evaluations of our model with real data. Section 6 concludes and indicates some future directions.

II. Related work

Recently, a large number of research works related to time series forecasting have been carried out, in which many efforts have been devoted to improving the accuracy and efficiency of forecasting. During the past three decades, several traditional statistical models such as Autoregressive (AR), Moving Average (MA), combination of AR and MA (ARMA), Autoregressive integrated moving average (ARIMA) are popular and widely used as methods for time series forecasting [11]. However, linear models have serious limitations, especially in capturing nonlinear relationships of data, which are common in many practical time series.

Artificial neural networks (ANNs) have been suggested as an alternative to time series forecasting because of their unique characteristics: adaptability, nonlinearity, and arbitrary function mapping ability [5]. The feed-forward neural network with back-propagation algorithm, known as back-propagation neural network (BPNN) is the most popular type of ANNs used for forecasting purpose [5]. Nevertheless, drawbacks also exist in BPNN, which includes easily running into local minimum point, slow convergence speed and sensitivity of initial network weight. The local optimum problem of back-propagation (BP) algorithm has been overcome in [12], [13], that proposed an algorithm combining back-propagation and genetic algorithm for the training of neural network. This algorithm, called GA-BP, can improve the convergence speed of the network and reduce the training failure.

For time series forecasting problem, the inputs of ANN are typically last p consecutive observations $(y_t, y_{t-1}, \ldots, y_{t-p})$ of the data series and the output is the future value y_{t+k}. However, because of data repetition of periodic time series, the future values also have strong relationship with corresponding values in the past cycles. Authors of [6] and [7] proposed architectures of ANN applied to seasonal time series, in which the input of ANN is consists of recent values $y_t, y_{t-1}, \ldots$ and the corresponding observation values in the past seasonal periods $y_{t-s}, y_{t-2s}, \ldots$ (where s is seasonal period). Otherwise, [14] proposed the Seasonal Artificial Neural Network (SANN) in which the i-th seasonal period observations are values of input neurons and the output neural values are the observations in $(i+1)$-th seasonal period. However, in all of above researches, the period of time series is given, but it is common hidden in practice.

Periodicity of time series has been studied in several related works. There are a number of algorithms for periodicity detection in time series data. Existing periodicity detection algorithms can be broadly classified in two groups: time domain methods based on autocorrelation and frequency domain methods using periodogram [3]. However, they still have drawbacks separately such as: using autocorrelation is more difficult to discover important periods while the periods estimated by periodogram can be false. This issue was solved by the AUTOPERIOD method proposed in [4], which exploits the information in both periodogram and autocorrelation to provide very accurate periodicity estimates for time series data. It can be seen that, there is a lot of potential when the AUTOPERIOD method and ANN are combined together to improve forecast accuracy and create a novel prediction model.

In the context of cloud computing, the time series forecasting models are widely used in workload or resource usage prediction. Authors of [15] proposed a resource prediction model based on double exponential smoothing. Otherwise, [8] used ANN with back-propagation algorithm and multiple linear regression method. Roy et al. [9] use a second order autoregressive moving average method (ARMA) for workload prediction, based on the last three observations. In [16], several forecasting methods including AR, MA, exponential smoothing, and ETS, Automated ARIMA and BPNN are employed to predict several real cloud workload. The gained results show that neural network method produced more accurate predictions than most others.

Nevertheless, all the methods used in forecasting cloud application consumption still have drawbacks presented previously, including the limitation of linear model and the local optimum problem of back-propagation neural network. In addition, a lot of real cloud workloads analyzed in [17] have strong periodicity. Hence, the main contributions and differences of our work are the combination of periodicity detection and neural network trained by GA-BP algorithm to forecast the future values of time series data.

III. PD-GABP scaling architecture

In this Section, we provide an overview of scaling architecture based on resource consumption and workload predictions. The architecture can be used to develop a auto-scaling service deployed in cloud environment.

A. Main concepts

The PD-GABP architecture is shown in Fig. 1. It consists of three modules: resource monitoring, PD-GABP prediction and scaling decision. The role of monitoring module is to collect monitoring data from cloud VMs. Usually, monitoring functions are provided by cloud vendors, however users also can deploy other monitoring solutions such as Nagios, Zabbix by themselves.

PD-GABP prediction module contains two components: data analysis and prediction. In the first component, data is preprocessed by the periodicity detection (AUTOPERIOD) and

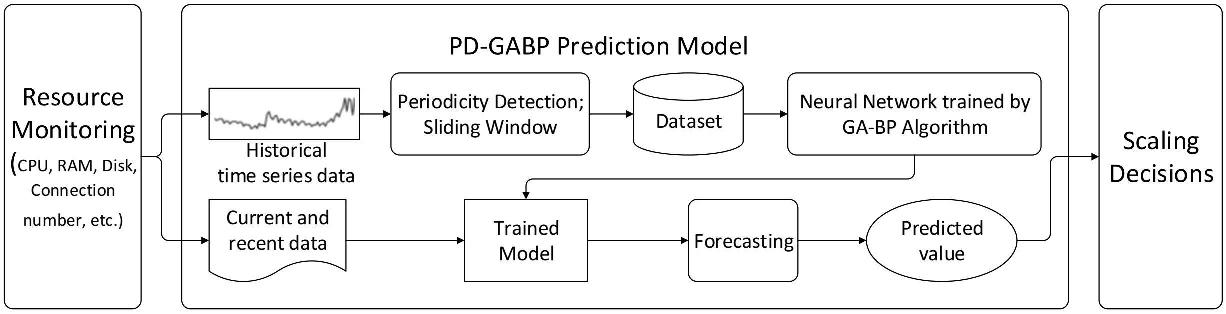

Fig. 1: Prediction-based auto-scaling service

sliding window techniques to construct a dataset, which is used to train genetic-backpropagation neural network (GA-BPNN). After that, the prediction component uses the trained neural network to forecast future consumptions or workload values. The resource metrics may be: CPU and RAM utilizations, Disk I/O, connection number and so on. All of the metrics are monitored and measured to generate historical time series sets. The time series data also is continuously updated by new monitored data. Due to this update, the analysis component is periodically reperformed every time interval t_{update} to guarantee the forecast accuracy.

The last module uses the future values that are generated from the second module to make scaling decisions. It means the module will command cloud infrastructure to increase or decrease resource amount provided to SaaS. In order to perform these scaling decisions, our architecture will use API functions provided by cloud vendors. In fact, there are several types of elastizing resources: scale up or down that changes capabilities of a VM and scale in or out that change the number of VMs provided for SaaS systems.

In this work, we concentrate on building a novel model to improve prediction accurate in the context of SaaS operations. This is the second module in PD-GABP architecture presented above.

B. Challenges addressed by PD-GABP prediction module

Our proposed prediction model addresses two major challenges in development process of a auto-scaling service in clouds, namely:

- Most systems cannot immediately increase or decrease provided resource capabilities with application demands at a time. The systems thus often need a time period to change provided resource capacity. The delay problem will be solved with PD-GABP model through forecasting consumption.
- The hidden periodicity of time series data is exploited to improve the accuracy of prediction model.

IV. PD-GABP ALGORITHM

A. Prediction method - genetic-backpropagation neural network

As model proposed in [8], our prediction approach uses the feedforward neural network with three layers to capture underlying functional relationship between future values and past observations. However, to improve forecasting accuracy, we train the neural network with a hybrid algorithm that combines genetic and back-propagation algorithms. This algorithm, called GA-BP, consists of two stages [13]. At first, genetic algorithm is employed to train neural network in order to find a point in the weight space which is close to the global optimum. Then, back-propagation starts from that point, and conducts an efficient local search. In this way, the approach can solve both problems of local optimum and slow convergence speed.

In the method, inputs of ANN are the last p consecutive observations of the data series $y(t), y(t-1), \ldots, y(t-p+1)$. However, our prediction approach exploits periodicity of time series to determine better inputs for neural network. The detail of PD-GABP algorithm is presented in the next subsection.

B. Periodicity detection

To estimate the periodicity of time series, M. Vlachos et al [4] proposed the AUTOPERIOD method, which has two stages. At first, periodogram is used to detect potential candidates of periodicity called ”hints”. The periodogram can be calculated using DFT (Discrete Fourier Transform) of a sequence as follows:

$$P\left(f_{k/N}\right) = \left\|X(f_{k/N})\right\|^2, \quad k = 0, 1, \ldots, \lceil (N-1)/2 \rceil \quad (1)$$

where P is periodogram, X is the DFT of a sequence $x(n)$, $n = 0, 1, \ldots, N-1$.

After that, because the periodicity hints may be false, all candidate periods are verified and further refined based on the circular auto-correlation function (ACF). The circular autocorrelation function of sequence x(n) with time lags is given by:

$$ACF\left(\tau\right) = \frac{1}{N}\sum_{n=0}^{N-1} x\left(\tau\right).x\left(n+\tau\right), \quad \tau = 0, 1, \ldots, N-1 \quad (2)$$

C. PD-GABP algorithm: a combination of periodicity detection and GA-BPNN

PD-GABP algorithm is composed of two phases: the first phase is to estimate periodicity of time series; the second phase uses the detected periods to determine the inputs vector that consists of several last observations and in some recent cycles for neural network. Then neural network will be trained by the GA-BP algorithm.

Assume that $T_1, T_2, \ldots, T_r$ estimated in the periodicity detection phase, we determine the input vector that is composed of p recent consecutive observations $y(t), y(t-1), \ldots, y(t-p+1)$ - called sliding window, and the values in the m previous cycles $y(t+k-T_1), y(t+k-2T_1), \ldots, y(t+k-mT_1), y(t+k-T_2), y(t+k-2T_2), \ldots, y(t+k-mT_2), \ldots, y(t+k-T_r), y(t+k-2T_r), \ldots, y(t+k-mT_r)$ - called periodic inputs. Hence, the ANN performs a nonlinear functional mapping:

$$\begin{aligned} y(t+k) \quad = \quad & f(y(t), y(t-1), \ldots, y(t-p+1), (t+k-T_1), \\ & y(t+k-2T_1), \ldots, y(t+k-mT_1), \\ & y(t+k-T_2), \ldots, y(t+k-mT_2), \ldots, \\ & y(t+k-T_r), \ldots, y(t+k-mT_r)) \end{aligned}$$

Specifically, the PD-GABP algorithm is described in Algorithm 1.

Algorithm 1 PD-GABP

1: Evaluate periods $T_1, T_2, \ldots, T_r$ of time series by AUTOPERIOD method
2: **if** time series is periodic ($r > 0$) **then**
3: Configure the input vector for ANN: $(y(t), y(t-1), \ldots, y(t-p+1), (t+k-T_1), y(t+k-2T_1), \ldots, y(t+k-mT_1), y(t+k-T_2), y(t+k-2T_2), \ldots, y(t+k-mT_2), \ldots, y(t+k-T_r), y(t+k-2T_r), \ldots, y(t+k-mT_r))$
4: **else**
5: Configure the input vector for ANN: $(y(t), y(t-1), \ldots, y(t-p+1))$
6: **end if**
7: Configure the output values for ANN: $y(t+k)$
8: Training ANN by GA-BP algorithm

D. Algorithm analysis and time constraints

The time consumption of each component in PD-GABP prediction model must satisfy the following constraints:

$$t_{compt\ 1} < t_{update} \tag{3}$$

$$t_{compt\ 2} < t_{prediction\ interval} \tag{4}$$

where $t_{compt\ 1}$ and $t_{compt\ 2}$ is time consumption of two components in prediction model mentioned respectively at subsection III; t_{update} is time interval that the first component is performed again, $t_{prediction\ interval}$ is prediction interval. In the first component, both periodogram and ACF can be computing efficiently through the Fast Fourier transform with complexity $O(NlogN)$ (N is size of historical time series) that much less than time consumption of the training of neural network ($t_{training}$). Therefore, to meet the condition (3), the time interval t_{update} will be increased if $t_{training} > t_{update}$.

The complexity of neural network to compute output from input is $O(m^2)$ where m is number of neurons in network. Because m is not large, the time consumption of second component $t_{compt\ 2}$ is much less than prediction interval, the condition (4) is satisfied.

V. EXPERIMENTS AND EVALUATIONS

In this section, we focus on evaluating the accuracy of PD-GABP prediction model. The entire PD-GABP model with three parts presented in section III-A will be implemented and evaluated in the near future.

A. Experiment Setup

We evaluate the capability of our model in prediction with the 1998 World Cup website workload dataset [10]. Particularly, we predict number of requests to the website. Like resource workload, request number is one of the monitoring metrics for SaaS applications in operation process. The time series data is produced by counting number of requests to the website every 10 minutes. We select 1-step-ahead forecasting that means the number of requests in the next 10 minutes will be predicted based on past observations. The data set from 40th to 46th day is used to train neural network, and then workload from 46th to 47th day will be predicted. The parameters for genetic algorithm and back-propagation are configured as follows: the size of the group $P_{size} = 225$, crossover rate $P_C = 0.9$, mutation rate $P_M = 0.01$, learning rate $\eta = 0.000001$. The input data of neural network is normalized according to the following formula [5]:

$$\widehat{x_i} = \frac{x_i - x_{\min}}{x_{\max} - x_{min}} \tag{5}$$

where x_i and $\widehat{x_i}$ denote the original value and the normalized value separately. The accuracy of prediction model is evaluated by several frequently used metrics defined as follows [5]:

- Root mean square error: $RMSE = \sqrt{\frac{\sum_{i=1}^{n} \left(\widehat{y_i} - y_i\right)^2}{n}}$
- Mean absolute percentage error: $MAPE = \frac{1}{n} \sum_{i=1}^{n} \frac{\left|\widehat{y_i} - y_i\right|}{y_i}$
- Mean absolute error (MAE): $MAE = \frac{1}{n} \sum_{i=1}^{n} \left|\widehat{y_i} - y_i\right|$

B. GA-BPNN

In the first experiment, the GA-BPNN method is evaluated and compared with BPNN method to prove efficiency of GA-BP algorithm. Fig. 2 and Fig. 3 show the prediction results with BPNN and GA-BPNN. The experiments demonstrate evidently that GA-BPNN prediction can fit closely the trend of actual number of request as well as BPNN. Concretely, in Table I, three evaluation metrics of GA-BPNN are smaller than BPNN for different sliding window sizes.

It is noteworthy that different sliding window sizes have a positive effect on accuracy of the models. The lower MAPE and RMSE results are, the prediction is more correct. Fig. 4 compares the responses of the prediction accuracy with different sliding window sizes based on MAE metric. In general, the metric of GA-BPNN is smaller than BPNN. For the GA-BPNN model, the optimal value is 65649.54 at $p = 4$. In contrast, the BPNN model has also an optimal sliding window size at 4 with MAE value = 69521.13.

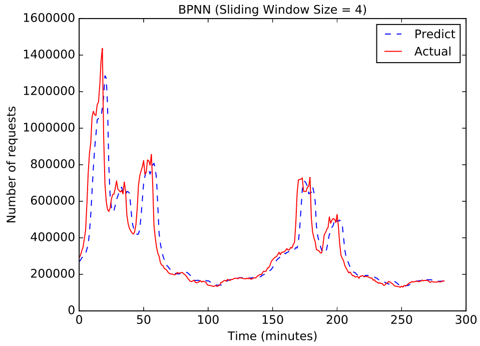

Fig. 2: Prediction results of BPNN model with $p = 4$

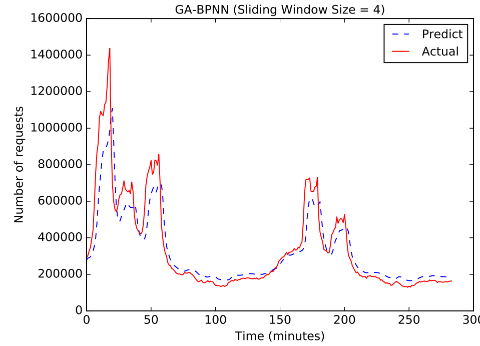

Fig. 3: Prediction results of GA-BPNN model with $p = 4$

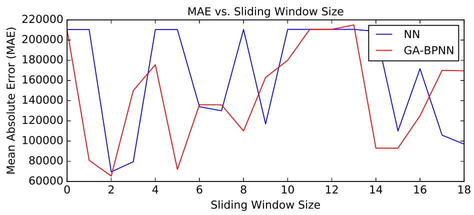

Fig. 4: Mean absolute error against window size in GA-BPNN and BPNN

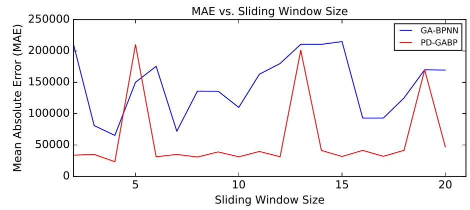

Fig. 5: Mean absolute error against window size in GA-BPNN and PD-GABP

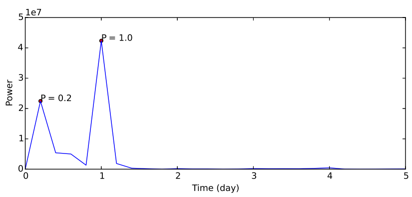

Fig. 6: Periodogram of request time series

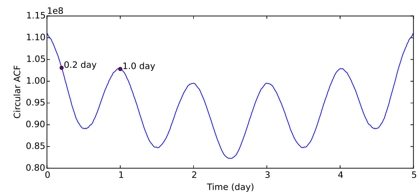

Fig. 7: Circular ACF of request time series

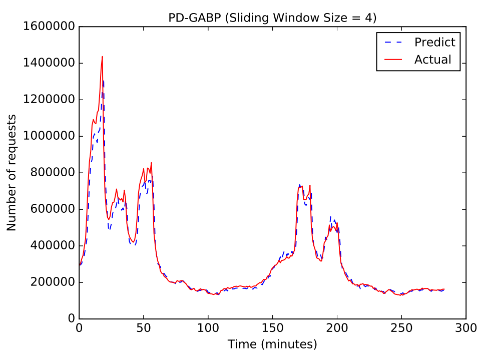

Fig. 8: Prediction results of PD-GABP algorithm with $p = 4$

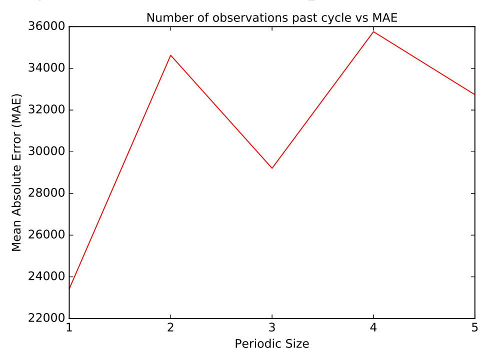

Fig. 9: MAE against number of periodic inputs in PD-GABP

TABLE I: Comparison of prediction accuracy among BPNN, GA-BPNN and PD-GABP models with different sliding window sizes (p)

Metric	RMSE			MAE			MAPE		
	$p = 2$	$p = 4$	$p = 6$	$p = 2$	$p = 4$	$p = 6$	$p = 2$	$p = 4$	$p = 6$
BPNN	328554.3	120275.6	328554.3	210424.8	69521.1	210425.1	1.65	0.13	1.63
GA-BPNN	328552.2	104314.7	297306.0	210383.9	65649.5	175616.0	1.63	0.12	0.98
PD-GABP	70656.1	47109.4	60548.6	33742.8	23425.6	31186.6	0.07	0.06	0.07

C. PD-GABP algorithm

In the second experiment, PD-GABP algorithm is employed to forecast the number of requests. The results is compared with GA-BPNN method to show the ability of periodicity detection in improving the forecasting accuracy.

Fig. 6 and Fig. 7 show the results of AUTOPERIOD method. Two candidate periods are discovered in periodogram in Fig. 6 with $P_1 = 0.2$ and $P_2 = 1.0$. The candidates are validated by using circular ACF and only P2 resides on the hill while P_1 is rejected. The final result of periodicity detection phase is that the request time series has single periodicity with the period p = 1.0 day.

With the period p = 1.0 day, the input vector are determine as follows: $(y(t), y(t-1), \ldots, y(t-p+1), (t+k-T), (t+k-2T), \ldots, (t+k-mT))$, where $T = 144$ is the number of observations in a period (1 day), p is sliding window size and m is number of periodic inputs. Fig. 8 shows the request prediction by PD-GABP method in case p = 4 and m = 1. It can be seen that the prediction resembles actual one. The prediction accuracy is summarized in Table I, in which the best forecasting measures are obtained by using the PD-GABP neural network. The smaller values of three evaluation metrics for PD-GABP demonstrate the efficiency of novel model in improving prediction accuracy.

In addition, the prediction accuracy also is affected by the different sliding window sizes and number of periodic inputs. Fig. 5 illustrates the MAE values of the two models with various sizes of sliding window. It can be seen that achieved MAE values of PD-GABP are lower than that of GA-BPNN.

Fig. 9 shows the change of MAE when $p = 4$ and m increases from 1 to 5. As an overall trend, it is clear that the MAE metric remained below than 120000 except the consideration surge at m = 3. The MAE reached optimal value at m = 1. Obviously, by selecting one input from past cycle with mostly recently consecutive observation in the past makes the better prediction accuracy in our model.

VI. Conclusions and Future Works

This work presents a novel prediction model in time series analytics context. The model is formed based on the combination of periodicity detection and neural network trained by genetic-backpropagation algorithm. In this direction, we use periodic data as input of GA-BPNN. The model was tested to forecast future connections for a real and well-known web application dataset: World Cup 98 site workload. The gained results demonstrated that the prediction accuracy is improved with our model as compared with other prediction methods. The forecast results thus can help systems scale effectively instead of using monitoring data at scaling moment. In the near future, we plan to integrate our prediction model into auto-scaling component of certain cloud middleware like OpenStack to better resource allocation. In addition, we also will apply the model for other monitoring metric data of virtual machines like CPU, memory, read/write drive speed, bandwidth and so on to help make auto-scaling decisions more precisely.

Acknowledgment

This work was supported by Hanoi University of Science and Technology's project "Research and Development of Semantic Interoperability Framework for IaaS Clouds" (No. T2015-227).

References

[1] M. B. Nguyen, V. Tran, and L. Hluchy, "A generic development and deployment framework for cloud computing and distributed applications," *Computing and Informatics*, vol. 32, no. 3, pp. 461–485, 2013.

[2] B. M. Nguyen, D. Tran, and Q. Nguyen, "A Strategy for Server Management to Improve Cloud Service QoS," in *Distributed Simulation and Real Time Applications (DS-RT), 2015 IEEE/ACM 19th International Symposium on*, Oct 2015, pp. 120–127.

[3] C. Berberidis, W. G. Aref, M. Atallah, I. Vlahavas, A. K. Elmagarmid *et al.*, "Multiple and partial periodicity mining in time series databases," in *ECAI*, vol. 2, 2002, pp. 370–374.

[4] M. Vlachos, S. Y. Philip, and V. Castelli, "On periodicity detection and structural periodic similarity." in *SDM*, vol. 5. SIAM, 2005, pp. 449–460.

[5] G. Zhang, B. E. Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks: The state of the art," *International journal of forecasting*, vol. 14, no. 1, pp. 35–62, 1998.

[6] C. C. Julian Faraway, "Time series forecasting with neural networks: A comparative study using the airline data," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 47, no. 2, pp. 231–250, 1998.

[7] J. Kihoro, R. Otieno, and C. Wafula, "Seasonal time series forecasting: A comparative study of ARIMA and ANN models," *AJST*, vol. 5, no. 2, 2004.

[8] S. Islam, J. Keung, K. Lee, and A. Liu, "Empirical prediction models for adaptive resource provisioning in the cloud," *Future Generation Computer Systems*, vol. 28, no. 1, pp. 155–162, 2012.

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

[10] M. Arlitt and T. Jin, "1998 World Cup Web Site Access Logs," accessed: 2016-2-19. [Online]. Available: http://ita.ee.lbl.gov/html/contrib/WorldCup.html

[11] K. W. Hipel and A. I. McLeod, *Time series modelling of water resources and environmental systems*. Elsevier, 1994, vol. 45.

[12] D. Venkatesan, K. Kannan, and R. Saravanan, "A genetic algorithm-based artificial neural network model for the optimization of machining processes," *Neural Computing and Applications*, vol. 18, no. 2, pp. 135–140, 2009.

[13] S. Ding, C. Su, and J. Yu, "An optimizing BP neural network algorithm based on genetic algorithm," *Artificial Intelligence Review*, vol. 36, no. 2, pp. 153–162, 2011.

[14] C. Hamzaçebi, "Improving artificial neural networks performance in seasonal time series forecasting," *Information Sciences*, vol. 178, no. 23, pp. 4550–4559, 2008.

[15] J. Huang, C. Li, and J. Yu, "Resource prediction based on double exponential smoothing in cloud computing," in *Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on*. IEEE, 2012, pp. 2056–2060.

[16] C. Vazquez, R. Krishnan, and E. John, "Time series forecasting of cloud data center workloads for dynamic resource provisioning," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, vol. 6, no. 3, pp. 87–110, 2015.

[17] A. Ali-Eldin, J. Tordsson, E. Elmroth, and M. Kihl, "Workload classification for efficient auto-scaling of cloud resources," Available: http://www8.cs.umu.se/research/uminf/reports/2013/013/part1.pdf, Tech. Rep., 2013.