2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)
October 11-14, 2020. Toronto, Canada

Improved LSTM-based Prediction Method for Highly Variable Workload and Resources in Clouds

Shuang Li[1], Jing Bi[1], Haitao Yuan[2], MengChu Zhou[3] and Jia Zhang[4]

[1]Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China
[2]School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China
[3]Dept. of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102, USA
[4]Dept. of Computer Science, Southern Methodist University, Dallas, TX 75275, USA
*lishuang2018@emails.bjut.edu.cn; bijing@bjut.edu.cn; yuanhaitao@buaa.edu.cn; zhou@njit.edu; jiazhang@smu.edu

***Abstract*—A large number of services provided by cloud/edge computing systems have become the most important part of Internet services. In spite of their numerous benefits, cloud/edge providers face some challenging issues, e.g., inaccurate prediction of large-scale workload and resource usage traces. However, due to the complexity of cloud computing environments, workload and resource usage traces are highly-variable, thus making it difficult for traditional models to predict them accurately. Traditional models fail to deal with nonlinear characteristics and long-term memory dependencies. To solve this problem, this work proposes an integrated prediction method that combines Bi-directional and Grid Long Short-Term Memory network (BG-LSTM) models to predict workload and resource usage traces. In this method, workload and resource usage traces are first smoothed by a Savitzky-Golay filter to eliminate their extreme points and noise interference. Then, an integrated prediction model is established to achieve accurate prediction for highly-variable traces. Using real-world workload and resource usage traces from Google cloud data centers, we have conducted extensive experiments to show the effectiveness and adaptability of BG-LSTM for different traces. The performance results well demonstrate that BG-LSTM achieves better prediction results than some typical prediction methods for highly-variable real-world cloud systems.**

***Index Terms*—Cloud computing systems, hybrid prediction, resource provisioning, BG-LSTM, artificial intelligence, deep learning, Savitzky-Golay filter**

I. INTRODUCTION

In recent years, cloud computing has become growingly in demand and widely adopted by many massive organizations. It integrates data center networks, servers, storage, application software, services and other resources to build a shareable and configurable computing resource pool [1]–[3]. For example, network bandwidth, internal and external storage resources are distributed according to users' demand. Typical cloud providers, *e.g.*, Google, Facebook, Amazon and Alibaba, have built large-scale data centers for users to rent their computing resources [4]–[6]. As the quantity of users sustained to expand, cloud computing providers need to handle a large number of users' requests while ensuring Quality of Services (QoS) of all users, and this dramatically increases their cost.

To ensure the on-demand availability of resources and meet the requirements of Service-Level Agreements (SLAs), cloud data center (CDC) providers have to conduct proactive resource provisioning [7], [8]. They must predict future server load trace conditions and provide appropriate resource provisioning to cope with CDC workload. However, Workload is dynamic and highly fluctuating, and resource usage is constantly changing during the execution of a task, which makes it difficult to predict. Under normal circumstances, most of users suffer unnecessary cost. It also causes huge waste of resources and reduce revenue of CDC providers. In addition, if users select insufficient resources, their tasks may be delayed or even unable to complete. In this way, QoS requirements of users' services cannot be well met, and may lead to the loss of users. If CDC providers can accurately predict the number of resources that users may need to use in future time slots based on historical workload and resource data, they can more effectively manage CDC resources, and obtain greater revenue.

Currently, there are multiple prediction methods in the field of time series. For traditional time series prediction, Back-Propagation Neural Network (BPNN) [9], Support Vector Machine (SVM) [10] and Autoregressive Integrated Moving Average model (ARIMA) [11] are some widely used and typical methods. Calheiros *et al.* [12] apply ARIMA model to solve the workload prediction problem for cloud service providers. However, it fails to capture nonlinear characteristics of workload time series. Our previous work [13] proposes an integrated forecasting method to predict the amount of workload in future time slots. The recent emergence of deep learning methods, *e.g.*, Deep Belief Networks (DBN) [14], especially Long Short-Term Memory (LSTM) [15] neural network model, provides new mechanisms to effectively realize high-accuracy time series prediction, and can effectively alleviate the gradient disappearance problem of traditional Recurrent Neural Networks (RNN) [16]. Zhang *et al.* [17] offer an efficient deep learning model to predict cloud workload for industry informatics. Chen *et al.* [18] propose a deep learning based prediction algorithm for cloud workloads (L-

This work was supported in part by the Major Science and Technology Program for Water Pollution Control and Treatment of China under Grant 2018ZX07111005, and the National Natural Science Foundation of China (NSFC) under Grants 61703011 and 61802015.

PAW). Futhermore, some studies have been conducted to improve the internal cell of an LSTM network structure and transform its external model structure. Among them, Bi-directional Long Short-Term Memory (BiLSTM) [19] and Grid Long Short-Term Memory (GridLSTM) [20] are two variants that change its external model structures. BiLSTM and GridLSTM can capture two directional dependence characteristics and different dimension information, respectively. Due to the highly changing characteristics of workload and resource usage, traditional prediction methods are insufficient to predict the large-scale data. Therefore, different from these studies, our work proposes a novel deep RNN method that integrates advantages of both BiLSTM and GridLSTM to predict the time series of workload and resource usage, and achieve better prediction performance.

The contributions of this work are given as follows:

1) After testing various smoothing methods to eliminate extreme points and noise interference in the original time series, it identifies a Savitzky-Golay (S-G) filter [21] as the most effective one to do so among the tested ones;
2) It integrates BiLSTM and GridLSTM models, referred to as BG-LSTM, to build a prediction model of workload and resource usage time series. It can effectively extract the complex characteristics of such series and achieve high prediction accuracy; and
3) Extensive experiments with real-world datasets demonstrate that BG-LSTM outperforms several baseline methods in terms of prediction accuracy, particularly for the prediction of relatively longer time series.

Section II describes the proposed method. Section III presents experimental results. Section IV concludes this work.

II. Model Framework

A. BG-LSTM

Traditional RNNs like LSTM can only investigate previous context information. To overcome it, Schuster *et al.* [22] introduce a Bi-directional RNN (BRNN) that can simultaneously train a model in two temporal directions, and has forward and backward hidden layers, respectively. Graves *et al.* [19] combine BRNN with LSTM, and propose the BiLSTM. GridLSTM [20] arranges LSTM cells into a grid of one or more dimensions. Different from existing LSTM, a GridLSTM network has recurrent connections along their depth dimension for improving learning characteristics. Fei *et al.* [23] propose a method that takes into account context-sensitivity and gradient problems. They construct a novel bidirectional structure by using GridLSTM named Bidirectional Grid Long Short-Term Memory (BiGridLSTM).

Different from [23], to achieve better prediction accuracy, and to capture features of context and depth dimension, this work stacks BiLSTM and GridLSTM models into a new integrating model called BG-LSTM, with its structure shown in Fig. 1. The output of BG-LSTM is described as follows. BiLSTM and GridLSTM are improved models of LSTM, and their calculations of some intermediate outputs are similar to LSTM. Their repetitive parts are replaced by $\Upsilon(\cdot)$ defined as:

$$\begin{aligned}
&\overleftarrow{O}_t^L = \Upsilon(\overleftarrow{f}_t^L, \overleftarrow{i}_t^L, \overleftarrow{o}_t^L, \overleftarrow{h}_{t-1}^L, I_t)\\
&\overrightarrow{O}_t^L = \Upsilon(\overrightarrow{f}_t^L, \overrightarrow{i}_t^L, \overrightarrow{o}_t^L, \overrightarrow{h}_{t-1}^L, I_t)\\
&O_t^{L+1} = \Upsilon(f_t^{L+1}, i_t^{L+1}, o_t^{L+1}, h_{t-1}^{L+1}, O_t^L)\\
&\overleftarrow{O}_{t+1}^{L+1} = \Upsilon(\overleftarrow{f}_{t+1}^{L+1}, \overleftarrow{i}_{t+1}^{L+1}, \overleftarrow{o}_{t+1}^{L+1}, \overleftarrow{h}_t^{L+1}, O_t^{L+1})\\
&\overrightarrow{O}_{t+1}^{L+1} = \Upsilon(\overrightarrow{f}_{t+1}^{L+1}, \overrightarrow{i}_{t+1}^{L+1}, \overrightarrow{o}_{t+1}^{L+1}, \overrightarrow{h}_t^{L+1}, O_t^{L+1})\\
&y_{t+1} = W_{\overrightarrow{h}y}\overrightarrow{O}_{t+1}^{L+1} + W_{\overleftarrow{h}y}\overleftarrow{O}_{t+1}^{L+1} + b_y
\end{aligned} \tag{1}$$

where f, i and o denote outputs of three LSTM gate units. I denotes the input of BG-LSTM. $\overleftarrow{O}_t^L$ and $\overrightarrow{O}_t^L$ represent the outputs of layer L in BiLSTM. $\overleftarrow{O}_{t+1}^{L+1}$ and $\overrightarrow{O}_{t+1}^{L+1}$ represent the outputs of layer L+1 in BiLSTM. O_t^{L+1} denotes the output of layer L+1 in GridLSTM. W and b denote the weight matrix and bias, respectively. h denotes the recurrent information among models. t denotes the time interval. The superscript $\rightarrow$ denotes the sequence from 1 to T, and $\leftarrow$ the sequence from T to 1. y_{t+1} denotes the output of BG-LSTM.

During the training phase in BG-LSTM, this work adopts a loss function in (2) to achieve the best prediction accuracy. Workload and resources have a large amount of traffic and a large differences in the order of magnitude. Many common network performance function, like Mean Square Error (MSE), cannot reflect the prediction accuracy appropriately. It is obvious that a large differences in the order of magnitude sequences impact more in these performance function than smaller ones. Thus, to diminish the influence caused by the orders of magnitude we take logarithm for both actual and predicted data. The evaluation metric is Root Mean Squared Logarithmic Error (RMSLE). The loss function for sequence $I_t = [I_1, \ldots, I_n]$ is given as:

$$Loss(t) = \frac{1}{n}\sum_{t=1}^{n} |\log \frac{y_t + 1}{\hat{y}_t + 1}| \tag{2}$$

where n is length of input, y_t and $\hat{y}_t$ represent actual and predicted data, respectively. Based on this function, we update weights and bias of BG-LSTM.

B. Framework of Prediction

To seek high prediction accuracy, three methods are used in the stage of data preprocessing. An S-G filter is first adopted to reduce the noise of the original workload and resource usage data. Then, the nature logarithm [24] and Min-Max scaler [25] are adopted to reduce the scale of the original data. After such data preprocessing, BiLSTM and GridLSTM are integrated into BG-LSTM for training and testing time series data. The specific prediction framework is proposed, as shown in Fig. 2. The input of the proposed prediction model is generated from the preprocessed data. BG-LSTM includes a GridLSTM layer, which is in the middle of two BiLSTM layers. After that, the output of BG-LSTM is transmitted to a fully connected layer for the final output.

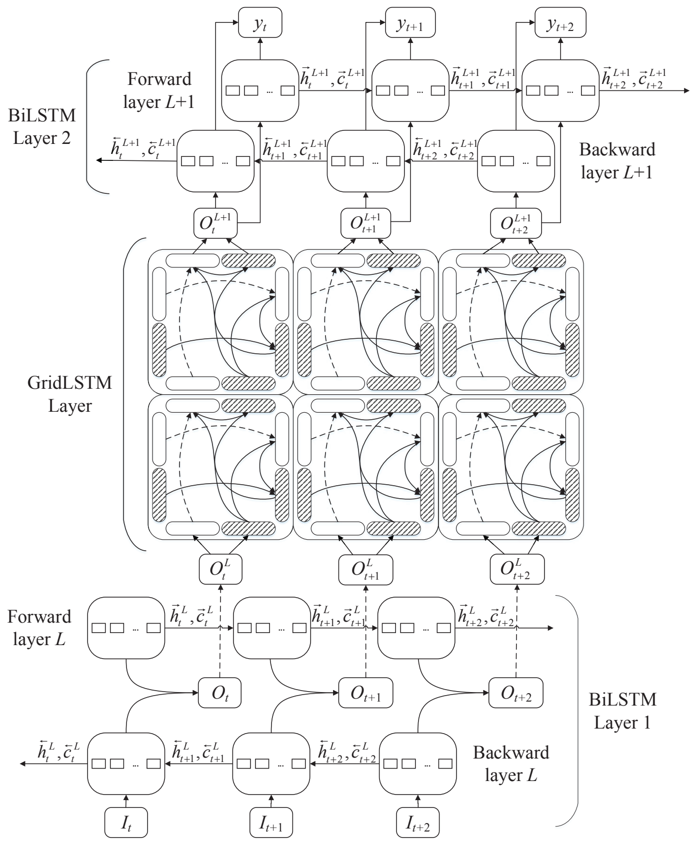


Fig. 1. BG-LSTM model structure.

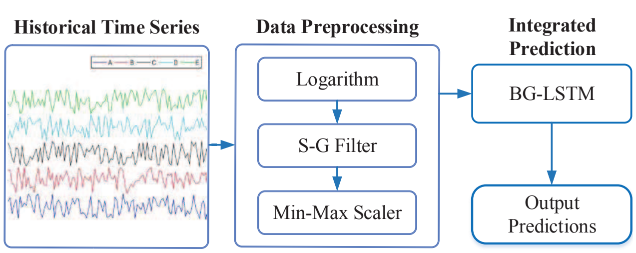


Fig. 2. Prediction framework including data preprocessing and BG-LSTM.

III. PERFORMANCE EVALUATION

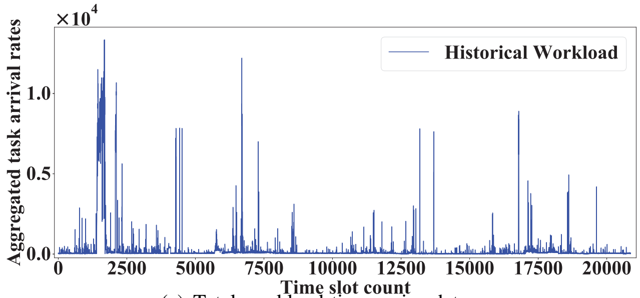
A. Data Preprocessing

To obtain characteristics of realistic arriving task and resource usage data, this section adopts the workload and resource usage trace collected from Google production compute clusters[1], which consists of approximately 12,000 machines. This workload trace contains a total of 672,003 jobs and 25,462,157 tasks over a time span of 29 days. Our work builds a prediction model based on workload and resource usage sequences. We first divide 29 days into 20880 time slots. The length of each time slot is 2 minutes. According to the timestamps of tasks, we count the number of task and record resource usage data, *e.g.*, CPU and Random Access Memory (RAM) usages, which are needed by the tasks during each time slot. Here, the workload time series means all of the whole task trace data in Google clusters in 29 days as shown in Fig. 3(a). CPU and RAM usage time series are illustrated in Figs. 3(b)-(c), respectively.

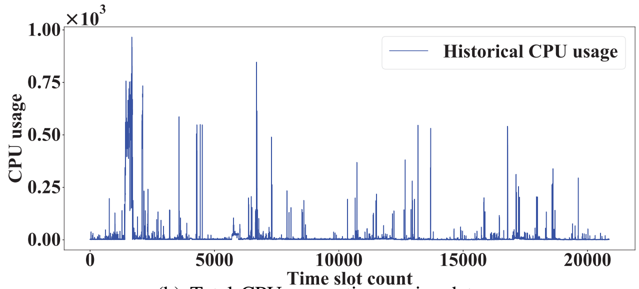
However, original workload and resource usage time series contain much noise caused by physical machine failures in CDCs or other abnormal cases, *e.g.*, the number of abnormal workload and resource usage caused by some unusual activities. Therefore, this makes it difficult to realize accurate prediction. We take the nature logarithm before smoothing such that the magnitude of total workload and resource usage time series is greatly reduced. We further compare several filter algorithms to filter outliers and noise. There are four series in our experiments, including the original sequence without smoothing, two processed by median and average filters, and smoothed one processed by the S-G filter. Here we collect workload, CPU and RAM series from Google cluster trace and perform experiments. The evaluation metric is RMSLE. In median, average and S-G filters, the window size needs to be set first. As shown in Table I, an S-G filter has better performance than commonly used median and average filters in different window (Win.) sizes. In this case, the processed workload time series is smoothed by the S-G filter to eliminate the possible outliers and noises.

Fig. 4(a) shows MSE between the original data and pre-

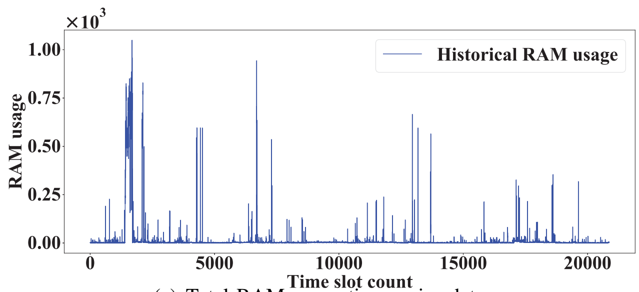
[1]https://github.com/google/cluster-data



(a) Total workload time series data.



(b) Total CPU usage time series data.

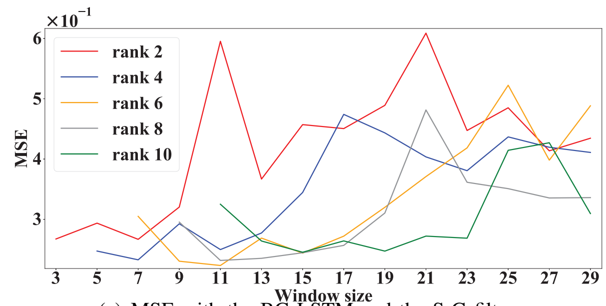


(c) Total RAM usage time series data.

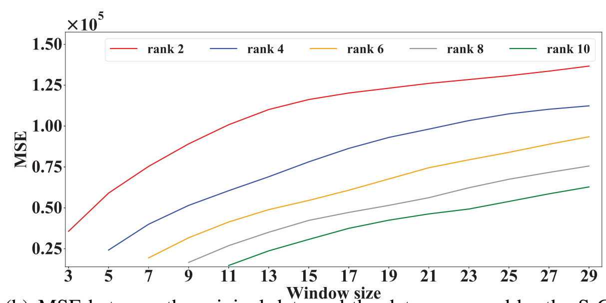
Fig. 3. Workload and resource usage time series data.

TABLE I
PERFORMANCE COMPARISON OF DIFFERENT FILTERS

Win. Size \ Methods	No filter	Median filter	Average filter	S-G filter
	Workload			
3	0.72	0.46	**0.18**	0.20
5		0.51	0.23	**0.17**
7		0.52	0.25	**0.15**
9		0.54	0.29	**0.15**
11		0.76	0.70	**0.16**
	CPU			
3	0.73	0.54	0.57	**0.28**
5		0.55	0.59	**0.23**
7		0.57	0.58	**0.18**
9		0.59	0.62	**0.17**
11		0.62	0.58	**0.16**
	RAM			
3	0.75	0.67	0.70	**0.22**
5		0.61	0.79	**0.18**
7		0.59	0.60	**0.15**
9		0.69	0.68	**0.16**
11		0.68	0.69	**0.14**



(a) MSE with the BG-LSTM and the S-G filter.



(b) MSE between the original data and the data processed by the S-G filter.

Fig. 4. MSE with respect to different window sizes and rank values.

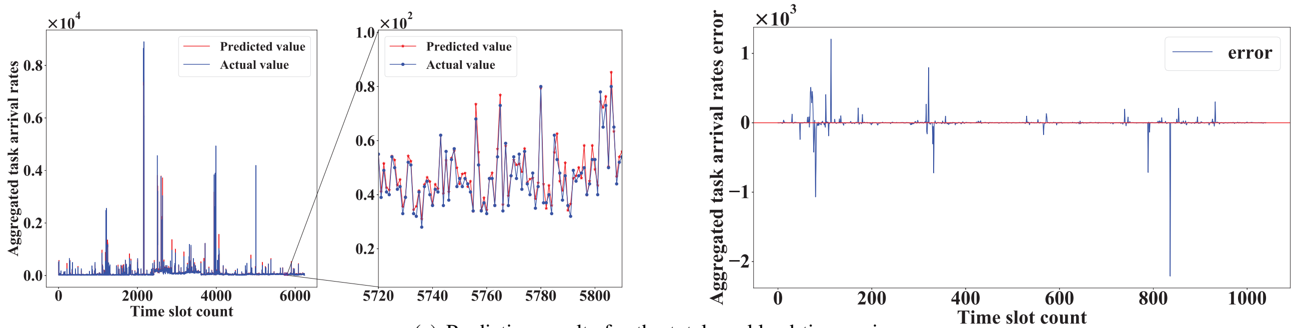
dicted data with BG-LSTM and the S-G filter with respect to different window sizes and rank values. It is shown in Fig. 4(a) that after several experiments, the S-G filter with the window size of 11 and the rank of 6 achieves the minimum change of the original shape of the data, and therefore it is used to establish the model. Fig. 4(b) shows MSE between the original data and data processed by the S-G filter. In Fig. 4(b), MSE is used to calculate the amount of change between the smoothed data and original one. Larger MSE means a greater change between the smoothed data and original one. Based on results shown in Figs. 4(a)-(b), the S-G filter with the window size of 11 and the rank of 6 is finally selected in our method.

B. Prediction Results

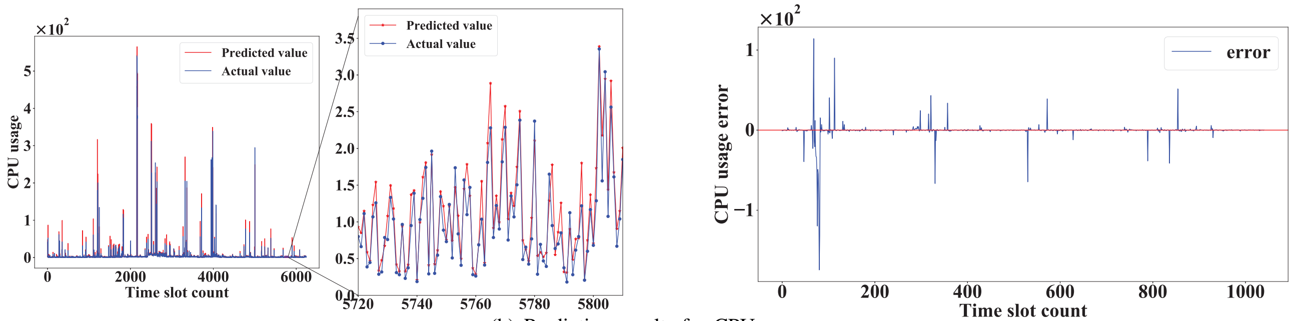
The best comprised of hyper-parameters for BG-LSTM is methodically investigated by conducting multiple trials and experiments. Tables II and III show the parameter setting of BG-LSTM for workload and resource time series.

Fig. 5 shows the performance of BG-LSTM on three different datasets. The left figures show the predicted data and actual data, and the right ones show the errors between them. Fig. 5(a) shows the prediction results for workload. Figs. 5(b)-(c) show the prediction results for CPU and RAM usage, respectively. Table IV shows the performance of BG-LSTM on different data of Google cluster trace in the experimental test set. The three data sets are workload, CPU usage and RAM usage. The evaluation metrics are MSE, RMSLE and R^2.

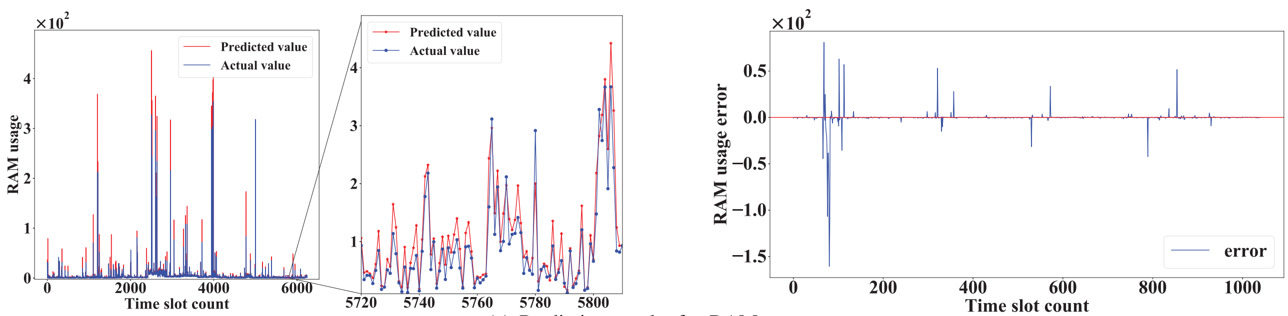
To verify the effectiveness and robustness of BG-LSTM, we conduct repeated experiments on the random data from the workload and resource usage time series, as shown in Table V. RMSLE is used as the evaluation criterion of different models. They include traditional methods, *e.g.*, ARIMA and SVM, and deep learning methods, *e.g.*, LSTM, BiLSTM, GridLSTM, SG-LSTM, SG-BiLSTM, and SG-GridLSTM. Here SG-



(a) Prediction results for the total workload time series.



(b) Prediction results for CPU usage.



(c) Prediction results for RAM usage.

Fig. 5. Prediction results for three data sets with BG-LSTM.

TABLE II
PARAMETER SETTING OF BG-LSTM FOR WORKLOAD

Parameter	Value	Description
X	60	Network input
Y	1	Network output
Structure	[60,45,30,15,1]	Network structure
Optimizer	Adam	Optimization function
Batch_size	5000	Batch size
Epochs	40000	Iteration times

TABLE III
PARAMETER SETTING OF BG-LSTM FOR RESOURCES

Parameter	Value	Description
X	60	Network input
Y	1	Network output
Structure	[60,50,45,20,1]	Network structure
Optimizer	Adam	Optimization function
Batch_size	4000	Batch size
Epochs	40000	Iteration times

TABLE IV
PERFORMANCE COMPARISON OF DIFFERENT DATASETS IN BG-LSTM

Performance	**Workload**	**CPU**	**RAM**
RMSLE	0.15	0.16	0.14
MSE	13934.54	128.89	131.29
R^2	0.9991	0.9997	0.9999

TABLE V
PERFORMANCE COMPARISON OF DIFFERENT METHODS WITH RMSLE

RMSLE / **Methods**	**Workload**	**CPU**	**RAM**
ARIMA	0.93	0.77	0.81
SVM	0.86	0.67	0.78
LSTM	0.83	0.56	0.61
BiLSTM	0.80	0.63	0.75
GridLSTM	0.77	0.58	0.69
SG-LSTM	0.74	0.23	0.22
SG-BiLSTM	0.17	0.20	0.16
SG-GridLSTM	0.19	0.19	0.15
BG-LSTM	**0.15**	**0.16**	**0.14**

means that before using the model to predict data, the S-G filter is first used to process the data. It is observed from Table V that the deep learning methods perform better than the traditional ones. In addition, after applying the S-G filter method, RMSLE of each method is significantly improved. Among them, BG-LSTM that combines BiLSTM and GridLSTM performs the best in terms of RMSLE.

C. Discussion

The combination of BiLSTM and GridLSTM layers in BG-LSTM has better modeling capacities than LSTM layers and other improved LSTM models in Google cluster trace. BiLSTM layers are able to explicitly model the time series directly near the current time interval. While GridLSTM layer can model the time series through the depth dimention. This complements the implicit modeling of LSTMs. The increase in modeling capacities allows BG-LSTM to outperform LSTM or other improved LSTMs with the similar number of parameters.

IV. CONCLUSION

Accurate prediction of complex and varying workload and resource usage traces plays is important in efficient resource provisioning for cloud data centers (CDCs). Due to their complicated characteristics, it is challenging to accurately predict them. In this work, we propose an integrated prediction model, named BG-LSTM, which is composed of a Bi-directional LSTM and a Grid LSTM. This work first adopts a Savitzky-Golay filter to make workload and resource usage traces easier to predict. Furthermore, the constructed BG-LSTM is proposed to extract the characteristics in workload and resource usage traces, and achieve the adaptive and accurate prediction for highly-variable workload and resource usage traces in CDCs. Finally, real-world datasets are used to demonstrate that the proposed model achieves significantly better prediction accuracy than other methods. Next, we plan to extend our work in two aspects: 1) using intelligent optimization methods to train model's parameters for fast training and better performance of the model [26]; and 2) exploring an adaptive method for resource provisioning with reinforcement learning for the dynamic and complex environment of cloud systems.

REFERENCES

[1] M. Ghahramani, M. Zhou, and C. T. Hon, "Toward Cloud Computing QoS Architecture: Analysis of Cloud Systems and Cloud Services," *IEEE/CAA Journal of Aut. Sinica*, vol. 4, no. 1, pp. 5–17, Jan. 2017.

[2] H. Jin, X. Wang, S. Wu, S. Di, and X. Shi, "Towards Optimized Fine-Grained Pricing of IaaS Cloud Platform," *IEEE Trans. on Cloud Computing*, vol. 3, no. 4, pp. 436–448, Dec. 2015.

[3] P. Zhang and M. Zhou, "Dynamic Cloud Task Scheduling Based on a Two-stage Strategy," *IEEE Trans. Automation Science and Engineering*, vol. 15, no. 2, pp. 772-783, Apr. 2018.

[4] H. Yuan, J. Bi, W. Tan, M. Zhou, B. H. Li, and J. Li, "TTSA: An Effective Scheduling Approach for Delay Bounded Tasks in Hybrid Clouds," *IEEE Trans. on Cybernetics*, vol. 47, no. 11, pp. 3658–3668, Nov. 2017.

[5] H. Yuan, J. Bi, and M. Zhou, "Multiqueue Scheduling of Heterogeneous Tasks With Bounded Response Time in Hybrid Green IaaS Clouds," *IEEE Trans. on Industrial Informatics*, vol. 15, no. 10, pp. 5404–5412, Oct. 2019.

[6] H. Yuan, J. Bi, M. Zhou, and A. C. Ammari, "Time-Aware Multi-Application Task Scheduling With Guaranteed Delay Constraints in Green Data Center," *IEEE Trans. on Automation Science and Engineering*, vol. 15, no. 3, pp. 1138–1151, Jul. 2018.

[7] A. Amokrane, R. Langar, M. F. Zhani, R. Boutaba and G. Pujolle, "Greenslater: On Satisfying Green SLAs in Distributed Clouds," *IEEE Trans. on Network and Service Management*, vol. 12, no. 3, pp. 363–376, Sept. 2015.

[8] J. Bi, H. Yuan, W. Tan, M. C. Zhou, Y. Fan, J. Zhang, and J. Li, "Application-Aware Dynamic Fine-Grained Resource Provisioning in a Virtualized Cloud Data Center," *IEEE Trans. on Automation Sci. and Engineering*, vol. 14, no. 2, pp. 1172-1184, 2017.

[9] J. Kumar and A. K. Singh, "Workload Prediction in Cloud Using Artificial Neural Network and Adaptive Differential Evolution," *Future Generation Computer Systems*, vol. 81, pp. 41–52, Apr. 2018.

[10] P. Zhang, S. Shu and M. Zhou, "An online fault detection model and strategies based on SVM-grid in clouds," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 2, pp. 445–456, Mar. 2018.

[11] Y. Weng, X. Wang, J. Hua, H. Wang, M. Kang and F. Wang, "Forecasting Horticultural Products Price Using ARIMA Model and Neural Network Based on a Large-Scale Data Set Collected by Web Crawler," *IEEE Trans. on Computational Social Systems*, vol. 6, no. 3, pp. 547–553, June 2019.

[12] R. N. Calheiros, E. Masoumi, R. Ranjan, and R. Buyya, "Workload Prediction Using ARIMA Model and Its Impact on Cloud Applications' QoS," *IEEE Trans. on Cloud Computing*, vol. 3, no. 4, pp. 449–458, Oct.-Dec. 2015.

[13] J. Bi, H. Yuan, and M. Zhou, "Temporal Prediction of Multiapplication Consolidated Workloads in Distributed Clouds," *IEEE Trans. on Automation Sci. and Eng.*, vol. 16, no. 4, pp. 1763–1773, Oct. 2019.

[14] G. M. Wang, J. F. Qiao, J. Bi, W. J. Li, and M. C. Zhou, "TL-GDBN: Growing Deep Belief Network with Transfer Learning," *IEEE Trans. on Automation Science and Engineering*, vol. 16, no. 2, pp. 874–885, Apr. 2019.

[15] S. Wen, *et al.*, "Memristive LSTM Network for Sentiment Analysis," *IEEE Trans. on Systems, Man, and Cybernetics: Systems*, doi: 10.1109/TSMC.2019.2906098, 2019.

[16] W. Zhang, B. Li, D. Zhao, F. Gong, and Q. Lu, "Workload Prediction for Cloud Cluster Using a Recurrent Neural Network," *2016 International Conference on Identification, Information and Knowledge in the Internet of Things (IIKI)*, pp. 104–109, 2016.

[17] Q. Zhang, L. T. Yang, Z. Yan, Z. Chen, and P. Li, "An Efficient Deep Learning Model to Predict Cloud Workload for Industry Informatics," *IEEE Trans. on Industrial Informatics*, vol. 14, no. 7, pp. 3170–3178, July 2018.

[18] Z. Chen, J. Hu, G. Min, A. Y. Zomaya, and T. El-Ghazawi, "Towards Accurate Prediction for High-Dimensional and Highly-Variable Cloud Workloads with Deep Learning," *IEEE Trans. on Parallel and Distributed Systems*, vol. 31, no. 4, pp. 923–934, April 2020.

[19] A. Yuille J. Wang, "Semantic Part Segmentation Using Compositional Model Combining Shape and Appearance," in *Proc. The IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1788–1797, 2015.

[20] I. Danihelka, N. Kalchbrenner and A. Graves, "Grid Long Short-term Memory," *arXiv:1507.01526*, 2015.

[21] A. Savitzky and M. J. E. Golay, "Smoothing and Differentiation of Data by Simplified Least Squares Procedures," *Anal. Chem.*, vol. 36, pp. 1627–1639, 1964.

[22] M. Schuster, and K. K. Paliwal, "Bidirectional Recurrent Neural Networks," *IEEE Trans. on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[23] H. Fei, and F. Tan, "Bidirectional grid long short-term memory (bigridlstm): a method to address context-sensitivity and vanishing gradient," *Algorithms*, vol. 11, no. 11, pp. 172, 2018.

[24] D. Mozyrska, and D. F. Torres, "The Natural Logarithm on Time Scales," *Journal of Dynamical Systems and Geometric Theories*, vol. 7, no. 1, pp. 41–48, Jun. 2009.

[25] S. Zhang, Z. Kang, Z. Hong, Z. Zhang, C. Wang, and J. Li, "Traffic Flow Prediction Based on Cascaded Artificial Neural Network," *Proc. IEEE International Geoscience and Remote Sensing Symposium*, Valencia, pp. 7232–7235, 2018.

[26] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, and J. Wang, "Dendritic neuron model with effective learning algorithms for classification, approximation and prediction," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 30, no. 2, pp. 601–614, Feb. 2019.