International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

Adaptive LSTM-Based Model for Accurate Forecasting of Workload and Resource Variability in Cloud Computing

Mukund N.Kulkarni[1], Dr.Anil B. Nandgaonkar[1], Dr.Sanjay L. Nalbalwar[1]
[1]Department of E&TC Engineering, Dr. Babasaheb Ambedkar Technological University, Lonere-402103, INDIA.
[1]mnkulkarni@dbatu.ac.in

Abstract – Cloud/edge computing systems play a crucial role in providing a wide range of services for Internet users. Despite their numerous advantages, providers of these systems face certain challenges, such as accurately predicting large-scale workloads and resource usage traces. The complexity of cloud computing environments makes it difficult for traditional models to accurately predict these traces due to their highly variable nature. Traditional models struggle to handle nonlinear characteristics and long-term memory dependencies. To address this issue, this study proposes an integrated prediction method that combines Bi-directional and Grid Long Short-Term Memory network (BGLSTM) models to predict workload and resource usage traces. The proposed method first smooths the traces using a Savitzky-Golay filter to eliminate extreme points and noise interference. Subsequently, an integrated prediction model is established to achieve accurate predictions for highly variable traces. The effectiveness and adaptability of the BG-LSTM model for different traces are demonstrated through extensive experiments using real-world workload and resource usage traces from Google Cloud data centers. The performance results indicate that BG-LSTM outperforms typical prediction methods in accurately predicting highly variable real-world cloud systems.

Keywords – Cloud Computing, Data Center, Load Balancing, Prediction, QoS, Workload Balancing

1. Introduction

In recent times, there has been a surge in the popularity and adoption of cloud computing among numerous large-scale organizations. This technology seamlessly integrates various components such as data center networks, servers, storage, application software, and services to create a flexible and shareable pool of computing resources. For instance, network bandwidth and storage resources, both internal and external, are allocated based on the specific requirements of users. Major players in the cloud computing industry, including Google, Facebook, Amazon, and Alibaba, have established extensive data centers where users can rent computing resources. Cloud computing has experienced a significant surge in demand and has been widely embraced by numerous large-scale organizations in recent years. It encompasses the integration of data center networks, servers, storage, application software, services, and various other resources to establish a pool of computing resources that can be shared and configured [1]–[3]. This entails the distribution of network bandwidth and internal and external storage resources based on the specific requirements of users. Prominent cloud providers such as Google, Facebook, Amazon, and Alibaba have constructed expansive data centers that allow users to rent their computing resources [4]–[6]. However, as the user base continues to expand, cloud computing providers face the challenge of managing a substantial volume of user requests while ensuring the Quality of Services (QoS) for all users, which inevitably leads to a significant increase in costs.

Cloud Data Centre (CDC) providers do proactive resource provisioning [7], [8] to ensure on-demand availability of resources and to meet Service-Level Agreements (SLAs). They need to anticipate future server load behaviour and make adequate resource reservations to satisfy the CDC workload. However, the workload is dynamic and highly volatile, and the consumption of resources changes with task execution making it hard to predict. This way, most users pay unneeded expenses in default. It also entails a colossal wastage of resources thereby reducing the revenue of CDC providers. Additionally, if inadequate resources are chosen by the users, they might experience task delays or even failure to complete. Thus, users' QoS requirements for their services are not adequately met, which may force them out. If CDC providers can estimate how many resources users might need in future time slots based on historical workload and resource data, then they will be better able to control their CDC resources and earn more money.

Various prediction techniques are currently used in the time series domain. While Back-Propagation Neural Network (BPNN) [9], Support Vector Machine (SVM)[10], and

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

Autoregressive Integrated Moving Average model (ARIMA)[11] are common methods for traditional time series prediction, they may fall short in capturing nonlinear characteristics of workload time series.

a) Calheiros [12] employed the ARIMA model to tackle workload prediction in cloud service providers but failed to address the nonlinear aspects. Previous work proposed [13] an integrated forecasting approach for future workload prediction.
b) The introduction of deep learning methods like Deep Belief Networks (DBN) and Long Short-Term Memory (LSTM) neural network models has revolutionized high-accuracy time series prediction, effectively overcoming the gradient disappearance issue experienced in traditional Recurrent Neural Networks (RNN).
c) Recent studies by Zhang et al. and Chen et al. have introduced efficient deep learning models for cloud workload prediction, aiming to enhance prediction algorithms for cloud workloads.
d) Bidirectional Long Short-Term Memory (BiLSTM) and Grid Long Short-Term Memory (GridLSTM) have been developed as variants that alter the external model structures of LSTM, capturing two-directional dependence characteristics and different dimension information.
e) Recognizing the limitations of traditional prediction methods in predicting large-scale data due to changing workload and resource usage characteristics, a novel deep RNN method is proposed in our work to integrate the strengths of BiLSTM and GridLSTM for enhanced prediction performance**.**

The following outlines the contributions made by this study:

- The Savitzky-Golay (S-G) filter [21] is identified as the most effective smoothing method for removing extreme points and noise interference from the original time series. The most effective one to do so among the tested ones.
- The combination of BiLSTM and GridLSTM models, known as BG-LSTM, is utilized to construct a workload and resource usage time series prediction model. This approach enables the extraction of intricate features within the series, resulting in a high level of prediction accuracy.

Numerous empirical investigations conducted on real-world datasets provide compelling evidence that BG-LSTM surpasses various benchmark techniques in terms of prediction accuracy, especially when forecasting relatively longer time series. The rest of the paper is organised as: Section 2 discusses the model framework for the proposed work. The implementation and results along with analysis is reported in section 3. The paper is concluded along with a discussion on future work in section 4.

2. Materials and Methods

2.1. B-LSTM: The Bidirectional Long Short-Term Memory Network

Bidirectional Long Short-Term Memory (B-LSTM) networks are a type of recurrent neural network (RNN) architecture designed to capture dependencies in sequential data in both forward and backward directions. Let us break down the components:

1. Long Short-Term Memory (LSTM): LSTMs are a type of RNN that is well suited for learning long-term dependencies in sequential data. They consist of memory cells and gates that regulate the flow of information. LSTMs can remember information over long sequences, making them effective for tasks like natural language processing (NLP), time series prediction, and speech recognition.
2. Bidirectional: Unlike traditional LSTMs that process input sequences in only one direction (from past to future), B-LSTMs process sequences in both forward and backward directions simultaneously. This allows them to capture information from both past and future contexts. For example, in NLP tasks like sentiment analysis, understanding the context before and after a word can be crucial for determining its meaning.

By processing sequences bidirectional, B-LSTMs can capture dependencies that may not be apparent from a unidirectional processing approach. Each time step in a B-LSTM is computed by concatenating the output of the forward LSTM and the output of the backward LSTM for that time step. Applications of B-LSTMs include tasks such as named entity recognition, part-of-speech tagging, sentiment analysis, and machine translation, where understanding the context in both directions can improve performance.

Here is a summary of how B-LSTMs work:

- Input sequences are fed into both forward and backward LSTMs simultaneously.
- The outputs of both LSTMs at each time step are concatenated.
- The concatenated outputs are then fed into subsequent layers of the neural network for further processing or prediction.

Overall, B-LSTMs are powerful tools for capturing complex dependencies in sequential data and have been widely adopted in various domains due to their effectiveness in capturing bidirectional context information.

2.2. Grid- LSTM

Grid LSTM (GLSTM) is a neural network architecture introduced to address the limitations of traditional LSTM networks in capturing spatial dependencies in data, particularly in tasks such as image processing and language modeling. The Grid LSTM was proposed by Google DeepMind researchers in 2016.

Here's a breakdown of Grid LSTM:

1. Motivation: Traditional LSTM networks are effective for capturing temporal dependencies in sequential data, but they may not be well suited for tasks where spatial dependencies are crucial, such as in images or structured grids of data. Grid LSTM was designed to extend the capabilities of LSTM to better capture spatial dependencies.
2. Architecture: In Grid LSTM, the memory cells are arranged in a grid-like structure, rather than being arranged sequentially as in traditional LSTMs. This grid structure allows the model to capture spatial relationships between adjacent elements in the data.
3. Computation within the Grid: Each cell in the grid computes its hidden state and memory cell using inputs from neighboring cells in addition to inputs from the current time step. This allows the model to capture both local and global dependencies within the data.
4. Applications: Grid LSTM has been primarily applied to tasks where spatial dependencies are important, such as image captioning, scene labeling, and video processing. By leveraging the grid structure, Grid LSTM can effectively model the spatial relationships between pixels or other elements in the data.

Overall, Grid LSTM extends the capabilities of traditional LSTM networks to capture spatial dependencies, making it well-suited for tasks involving structured data or data with spatial relationships. It has shown promising results in various applications and remains an active area of research in deep learning.

2.3. BG- LSTM

Traditional RNNs, such as LSTM, can only analyze past context information. Schuster et al. [22] developed a Bi-directional RNN (BRNN) to address this issue. To train a model in two temporal directions, use forward and backward hidden layers. Graves et al. [19] introduced the B-LSTM, a combination of BRNN and LSTM. Grid-LSTM [20] organizes LSTM cells in one or more dimensions. A Grid-LSTM network differs from traditional LSTMs by utilizing recurrent connections along the depth dimension to enhance learning capabilities. Fei et al. [23] provide a strategy that considers context-sensitivity and gradient issues. The authors developed BiGrid-LSTM, a unique bidirectional structure based on Grid-LSTM.

In contrast to [23], this study combines B-LSTM and Grid-LSTM models to create a new integrating model termed BG-LSTM, as illustrated in Fig. 1. This model improves prediction accuracy and captures context and depth characteristics. The output of the BG-LSTM is detailed below. B-LSTM and Grid-LSTM are upgraded versions of LSTM, with identical computations for intermediate outputs. Eq. 1, below represents the basic structure of computations in the BG-LSTM.

$$\begin{aligned}
\overleftarrow{O}_t^L &= \mathrm{N}(\overleftarrow{f}_t^L, \overleftarrow{i}_t^L, \overleftarrow{o}_t^L, \overleftarrow{h}_{t-1}^L, I_t \\
\overrightarrow{O}_t^L &= \mathrm{N}(\overrightarrow{f}_t^L, \overrightarrow{i}_t^L, \overrightarrow{o}_t^L, \overrightarrow{h}_{t-1}^L, I_t \\
O_t^{L+1} &= \mathrm{N}(f_t^{L+1}, i_t^{L+1}, o_t^{L+1}, h_{t-1}^{L+1}, O_t^L \\
\overleftarrow{O}_{t+1}^{L+1} &= \mathrm{N}(\overleftarrow{f}_{t+1}^{L+1}, \overleftarrow{i}_{t+1}^{L+1}, \overleftarrow{o}_{t+1}^{L+1}, \overleftarrow{h}_t^{L+1}, O_t^{L+1} \\
\overrightarrow{O}_{t+1}^{L+1} &= \mathrm{N}(\overrightarrow{f}_{t+1}^{L+1}, \overrightarrow{i}_{t+1}^{L+1}, \overrightarrow{o}_{t+1}^{L+1}, \overrightarrow{h}_t^{L+1}, O_t^{L+1} \\
y_{t+1} &= W \rightarrow (\overrightarrow{h}_y)\overrightarrow{O}_{t+1}^{L+1} + W \leftarrow (\overleftarrow{h}_y)\overleftarrow{O}_{t+1}^{L+1} + c_y
\end{aligned}$$

This study uses a loss function in the BG-LSTM training phase to optimize prediction accuracy. Workload and resources are highly trafficked and have significant variances in size. Common network performance metrics, such as Mean Square Error (MSE), may not accurately reflect forecast accuracy. Large changes in the order of magnitude sequences have a greater influence on performance functions than smaller ones.

To reduce the impact of magnitude differences, we employ logarithms for both actual and forecasted data. The evaluation metric is the Root Mean Squared Logarithmic Error. The loss function for the sequence [I1;::: ; In] is as follows:

$$Loss(t) = \frac{1}{l}\sum_{t-1}^{n} \left| \log \frac{y_t + 1}{y_t + 1} \right| \qquad (2)$$

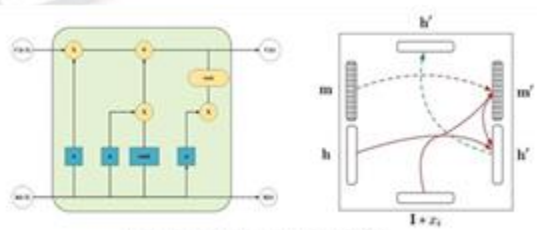


Fig. 1. Long Short-Term Memory (LSTM)

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

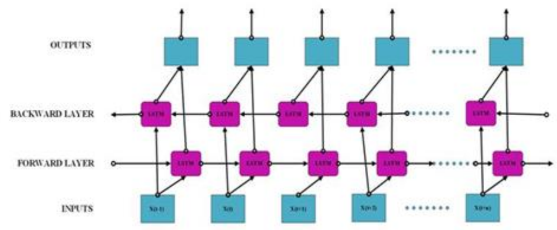


Fig. 2. Bidirectional Long Short-Term Memory (B-LSTM)

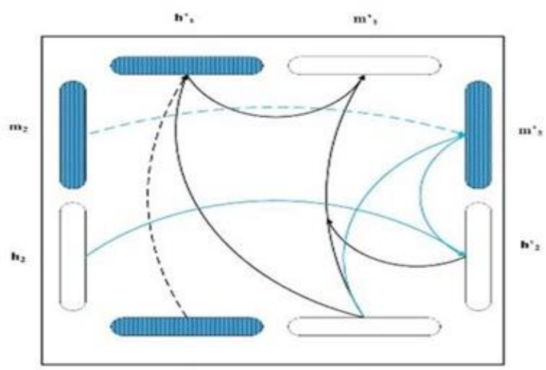


Fig. 3. Grid Long Short-Term Memory (G-LSTM)

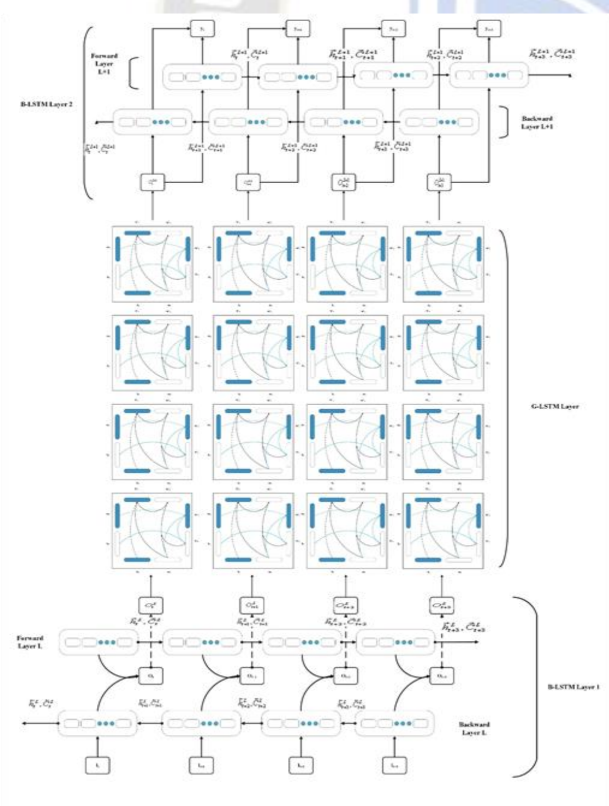
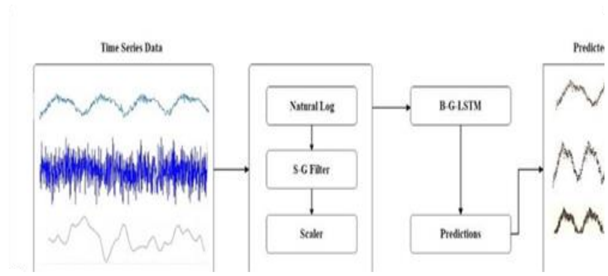


Fig. 4. BG-LSTM structure



Results and Discussions

3.1. Data Processing

This section analyses workload and resource utilization statistics from Google's production compute clusters, which include over 12,000 computers. The workload trace spans 15 days and includes 672,003 jobs and 25,462,157 tasks. Our study creates a prediction model based on workload and resource utilization sequences. We initially split 15 days into 20880 time slots. Each time slot lasts for two minutes. We count the number of processes and record resource utilization statistics, including CPU and RAM consumption, for each time slot based on their timestamps.

The initial workload and resource use time series contain noise due to actual machine failures in CDCs or other atypical instances, such as the number of anomalies. Unexpected actions led to increased effort and resource utilization. This makes it challenging to make reliable predictions. By using the nature logarithm before smoothing, we significantly lower the size of the overall effort and resource utilization time series. We examine several filtering techniques to remove outliers and noise. The studies consist of four series: the original without smoothing, two treated using median and average filters, and smoothed one using the Savitzky–Golay filter. We collect workload, CPU, and RAM data from Google cluster traces and conduct tests. The evaluation metric is RMSLE. For median, average, and Savitzky–Golay filters, the window size must be selected first.

Table I shows that Savitzky–Golay filters outperform median and average filters across different window widths. The Savitzky–Golay filter smoothes the processed workload time series, eliminating outliers and noise. The model is established with a size of 10 and a rank of six, which minimizes the change to the original form of the data. In our procedure, a rank of six is chosen from a list, and the size of the window is 10.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

Table 1. Performance Evaluation of Different Filters used for smoothening data

Window Size	CPU			
	No Filter	Median Filter	Moving Average Filter	Savitzky–Golay Filter
3	0.73	0.54	0.57	0.28
4	0.73	0.545	0.58	0.26
5	0.73	0.55	0.59	0.23
6	0.73	0.56	0.59	0.21
7	0.73	0.57	0.58	0.18
8	0.73	0.58	0.60	0.175
9	0.73	0.59	0.62	0.17
10	0.73	0.61	0.60	0.175
11	0.73	0.62	0.58	0.16

Window Size	Memory			
	No Filter	Median Filter	Moving Average Filter	Savitzky–Golay Filter
3	0.75	0.67	0.70	0.22
4	.075	0.64	0.755	0.20
5	0.75	0.61	0.79	0.18
6	0.75	0.60	0.695	0.165
7	0.75	0.59	0.60	0.15
8	0.75	0.64	0.64	0.155
9	0.75	0.69	0.68	0.16
10	0.75	0.69	0.69	0.15
11	0.75	0.68	0.69	0.14

2.4. Forecasting Results

Multiple trials and experiments are conducted to determine the optimal hyper-parameters for the BG-LSTM. Tables 2 and 3 illustrate the parameter settings for BG-LSTM for workload and resource time series. Figures on the left display expected and actual data, while figures on the right display errors between the two. Figure 5 depicts workload prediction findings. Table 4 displays BG-LSTM's performance on several Google cluster trace datasets in the experimental test set. The three data sets include workload, CPU use, and RAM usage. The assessment criteria are MSE, RMSLE, and R^2.

To test the efficacy and resilience of BG-LSTM, we did tests using random data from workload and resource utilization time series (Table 5). RMSLE is used as an assessment criterion for several models. Traditional approaches like ARIMA and SVM, as well as deep learning techniques like LSTM, Bi-LSTM, Grid-LSTM, SG-LSTM, SG-Bi-LSTM, and SG-Grid-LSTM, are used. The term "SG" refers to employing the Savitzky–Golay filter to analyze data before applying the model for prediction. Table 5 shows that deep learning outperforms standard approaches. The SG filter approach dramatically improves RMSLE for all methods. BG-LSTM, a combination of Bi-LSTM and Grid-LSTM, outperforms other models in terms of RMSLE.

BG-LSTM, which combines Bi-LSTM and Grid-LSTM layers, outperforms LSTM layers and other enhanced LSTM models in Google Cluster Trace. Bi-LSTM layers can explicitly represent time series near the current interval. The Grid-LSTM layer may model time series using the depth dimension. This complements the implicit modeling of LSTMs. BG-LSTM outperforms LSTM and other enhanced LSTMs with similar settings due to its increased modeling capacity.

Table 2. BG-LSTM Parameter Set for Workload

Parameter	Value	Description
Structure	[60,45,30,15,1]	Network Structure
X	60	Network Input
Y	1	Network Output
Batch Size	5000	Batch Size
Epochs	40000	Iteration Time
Optimiser	Adams	Optimization Function

Table 3. BG-LSTM Parameter Set for Resources

Parameter	Value	Description
Structure	[60,45,30,15,1]	Network Structure
X	60	Network Input
Y	1	Network Output
Batch Size	4000	Batch Size
Epochs	40000	Iteration Time
Optimiser	Adams	Optimization Function

Table 4. BG-LSTM Performance Comparison of Google Datasets

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

Table 5. Performance Comparison of Various Methods with RMSLE

Methods	Memory (RAM)	CPU	Workload
ARIMA	0.81	0.77	0.93

SVM	0.78	0.67	0.86
LSTM	0.61	0.56	0.83
Bi-LSTM	0.75	0.63	0.80
G-LSTM	0.69	0.58	0.77
SG-LSTM	0.22	0.23	0.74
SG-Bi-LSTM	0.16	0.20	0.17
SG-G-LSTM	0.15	0.19	0.19
BG-LSTM	0.14	0.16	0.14

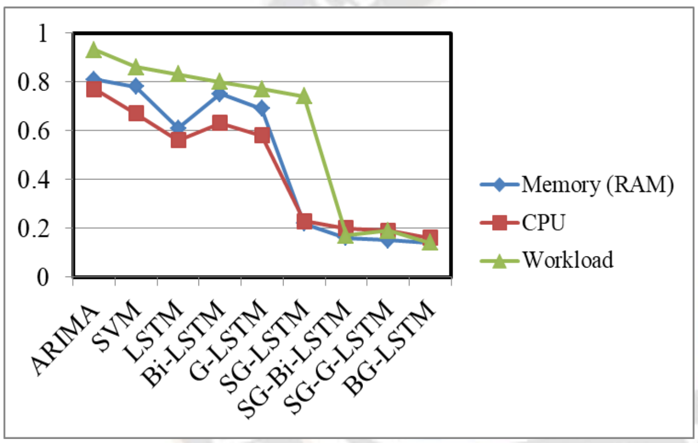


Fig. 6. Performance Analysis of various methods Conclusions

Predicting complicated workloads and resource consumption trails accurately is crucial for effective resource allocation in cloud data centers (CDCs). Accurate prediction is tough due to their complex properties. We introduce BG-LSTM, an integrated prediction model that combines Bi-directional and Grid LSTMs. This study uses a Savitzky-Golay filter to forecast workload and resource utilization. The suggested BG-LSTM extracts feature from the workload and resource utilization traces, enabling adaptive and accurate prediction in cloud data centers (CDCs) with high variability. Using real-world datasets, the suggested model outperforms existing techniques in terms of prediction accuracy. We plan to expand our work in two areas:

1. Using intelligent optimization methods to train model parameters for faster training and improved performance
2. Exploring an adaptive resource provisioning method with reinforcement learning for the dynamic and complex environment of cloud systems.

Conflicts of Interest

The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

Funding Statement

The reported work does not receive any form of funding from any individuals or Institutions.

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.