Information Sciences 481 (2019) 57–68

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

SGW-SCN: An integrated machine learning approach for workload forecasting in geo-distributed cloud data centers*

Jing Bi[a,b,*], Haitao Yuan[b,c], LiBo Zhang[a], Jia Zhang[d]

[a] Faculty of Information Technology, Beijing University of Technology, Beijing, China
[b] Department of Electrical and Computer Engineering, New Jersey Institute of Technology, USA
[c] School of Software Engineering, Beijing Jiaotong University, China
[d] Department of Electrical and Computer Engineering, Carnegie Mellon University, USA

ARTICLE INFO

Article history:
Received 15 June 2018
Revised 11 November 2018
Accepted 13 December 2018
Available online 21 December 2018

Keywords:
Geo-distributed cloud data centers (Geo-2DCs)
Stochastic configuration networks (SCNs)
Wavelet decomposition
Workload forecasting
Savitzky-Golay filter

ABSTRACT

Nowadays, a large number of cloud services have been published and hosted by geo-distributed cloud data centers (Geo-2DCs). In spite of numerous benefits, those Geo-2DCs face significant challenges such as dynamic resource scaling where workload forecasting plays a crucial role in addressing such a challenge. High accuracy and fast learning are key indicators for workload forecasting and the literature has witnessed a lot of efforts. This work proposes an integrated forecasting method, equipped with noise filtering and data frequency representation, named Savitzky-Golay and Wavelet-supported Stochastic Configuration Networks (SGW-SCN), to predict the amount of workload in future time slots. In this approach, the workload time series is first smoothed by a Savitzky-Golay filter and then decomposed into multiple components via wavelet decomposition. With stochastic configuration networks, an integrated model is established to characterize statistical characteristics of both trend and detail components. Extensive results have demonstrated that the proposed method achieves higher forecasting accuracy and faster learning speed than typical forecasting methods.

1. Introduction

Cloud computing has become a new type of Internet service because of its high scalability, flexibility, and cost-efficiency. Geo-**D**istributed **C**loud **D**ata **C**enters (Geo-2DCs) are established, so that customers can deploy applications and store data in the centers, while leveraging the computing powers of the server farms as well as various computing services hosted at the centers [3,21]. As a result, Geo-2DCs and their effective operations have received significant attention in recent years [12]. However, because of the massive computing resources maintained in Geo-2DCs, the power consumption has become a critical problem. Besides, Geo-2DC providers aim to maximize their revenue while meeting various Service Level Agreements (SLAs) for customers. Therefore, recent studies have proposed a collection of dynamic scheduling algorithms to reduce resource consumption as well as to improve energy efficiency in Geo-2DCs [4].

* Corresponding author at: Faculty of Information Technology, Beijing University of Technology, Beijing, China.
E-mail address: bijing@bjut.edu.cn (J. Bi).
* This paper belongs to the special issue special issue name edited by "Prof. W. Pedrycz."
* This paper belongs to the special issue special issue name edited by "Prof. W. Pedrycz."

https://doi.org/10.1016/j.ins.2018.12.027
0020-0255/

However, dynamic resource scaling typically depends on a number of factors including the number of active users, upcoming events, and the current states of a system. Historical usage patterns are usually used to predict the number of resources in future time slots. In recent years, researchers have proposed many workload forecasting schemes [1,5–7,13,22], which mainly measure the maximum or average workload for specified time slots. However, the accuracy of their forecasting results may not always be satisfactory.

In this research, we propose a machine learning-based workload forecasting method, focusing on estimating the number of tasks arriving in the future time slots in Geo-2DCs. Be more specific, we have developed a novel approach called SGW-SCN, which incorporates **S**avitzky-**G**olay filter and **W**avelet decomposition (SGW) [17,19] with **S**tochastic **C**onfiguration **N**etworks (SCNs) [23–26]. Extensive results based on a real-life benchmark dataset[1] have demonstrated that our approach outperforms several common methods with respect to forecasting accuracy and training speed. The main contributions of this work are three-fold:

- It adopts the Savitzky-Golay filter method to eliminate possible outliers and noises in the non-stationary workload time series.
- It applies wavelet decomposition to obtain the trend and detail components for different original workload time series.
- An integrated forecasting model with noise filtering, frequency representation of data and SCNs, named SGW-SCN, is proposed to predict the amount of original workload in future time slots.

The remainder of this paper is organized as follows. Related studies are discussed in Section 2. Section 3 describes the dataset and data preprocessing. Section 4 describes the proposed model. Section 5 presents experimental results. Finally, Section 6 concludes this work.

2. Related work

2.1. Data-Driven approaches

Machine learning approaches have been increasingly adopted for time series forecasting [6,7,9–11,30]. Chen et al. [7] propose a self-adaptive prediction method that uses an ensemble model and subtractive-fuzzy clustering-based fuzzy neural network. Chang et al. [6] propose a workload prediction model by using a neural network and the steepest descent learning algorithm. It improves prediction accuracy over a time-delay neural network and the linear regression methods. Kumar and Singh [11] present a workload forecasting model that uses neural network and the self-adaptive differential evolution algorithm. It can learn the most suitable burst workload along with an optimal crossover rate. Islam et al. [10] present a prediction-based resource provisioning strategy by using a sliding-window-based neural network and the linear regression techniques. It performs better than non-sliding-window-based neural network.

Most of the above methods adopt neural network and the linear regression approaches. However, they only work well in catastrophic and irregular workload bursts. Furthermore, their forecasting performance is limited when dealing with periodic and non-linear cloud-based workload series. To overcome such limitations, this work proposes SGW-SCN to predict cloud workload by considering its periodicity and non-linearity.

2.2. Model-based approaches

Several researches for Geo-2DCs have been proposed to analyze the workload prediction by using model-based approaches [8,14,15,20]. Urgaonkar et al. [20] capture the transient behavior of workloads in shared data centers. They model server resources by using a time-domain queuing model that dynamically maps resource requirements of each application to workload characteristics. Mishra et al. [15] propose a method to understand task resource consumption in Google compute clusters with workload classification models. Based on it, fine-grain task scheduling and capacity planning are realized. Liu et al. [14] present an approach to model the energy consumption in a data center. They predict renewable energy and workload demands one week into the future, and allocate IT resources according to time varying power supply and cooling efficiency.

However, the above methods adopt the queuing models to estimate the average queue length for the next time slot, and they are not suitable for on-line prediction due to performance concerns. Different from them, our forecasting technique aims to predict workload in Geo-2DCs in a shorter time period at a scale of minutes.

2.3. Integrated learning approaches

Integrated learning is a paradigm that jointly integrates multiple related methods and has demonstrated its advantages in many fields [1,5,22,27,29]. Many researchers have achieved workload forecasting through different integrated learning approaches. Calheiros et al. [5] adopt an autoregressive integrated moving average (ARIMA)-based predictor for proactive provisioning of virtual machine instances. Their simulation is conducted through workload trace from Wikimedia. Ardagna

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

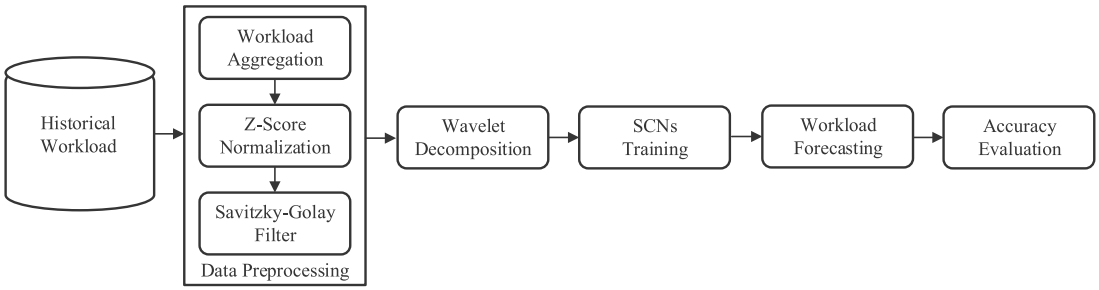


Fig. 1. Workload forecasting approach.

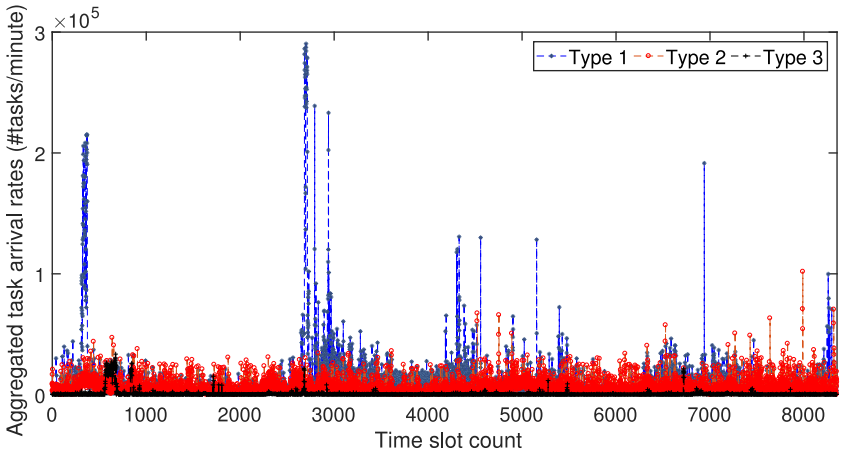


Fig. 2. Aggregated task arriving rates of different types.

et al. [1] propose a distributed solution that integrates a workload prediction model and distributed non-linear optimization techniques. Van den Bossche et al. [22] evaluate the effectiveness of the different workload forecasting techniques including exponential smoothing, Holt-Winters, and ARIMA. They then propose a method to automate procurement decisions of contracts for Infrastructure-as-a-Service (IaaS) providers.

In contrast, our forecasting can quickly react to tasks with optimal resources, which is critical for Geo-2DCs. Besides, our proposed SGW-SCN can better capture transient behaviors of workload and effectively avoid one-step delay occurrence.

3. Data preprocessing

Fig. 1 illustrates our integrated forecasting approach. The data preprocessing step first extracts tasks in each 5-min time slot. The noise points (outliers) in the workload that affect the forecasting accuracy are eliminated, and a stationary sequence is obtained. Then, we decompose the stationary sequence into two parts through wavelet decomposition. One is the trend component of the workload, and the other contains its details. Thus, we obtain two time series with different features. Afterwards, we adopt SCNs to determine optimal parameters of the forecasting model for both time series. The values of the trend and detail components at the subsequent time slot are obtained, respectively. Finally, the number of tasks arriving at the following time slot is obtained via wavelet reconstruction.

To understand the characteristics of tasks, we investigated the workload traces in Google production compute clusters in May 2011, and found that they are highly non-linear, non-stationary with significant noise interference. The traces were collected from an 12.5k-machine cell over 29 days, resulting in a total of 25,462,157 tasks. Each task has an attribute indicating its importance, and Google clusters divide all tasks into different levels. A higher level implies a more important task. Twelve attributes were studied dividing into three groups: gratis (0–1), other (2–8), and production (9–11) [28]. Accordingly, we divided all tasks into three types, and obtained the number of arriving tasks of each type. We then divided 29 days into 8352 5-min time slots, and counted the aggregated arriving rates of three types of tasks as shown in Fig. 2. As explained in Section 4, we adopted data in the first 25 days for training, and used the data in the last 4 days for testing.

The workload time series is highly non-stationary thus it is difficult to realize accurate forecasting. Therefore, before constructing a workload forecasting method, the workload time series needs to be stabilized. Typically, a stationary time series means that its mean and variance are stable around a constant. As shown in Fig. 2, however, the number of tasks varies significantly. It is hard to extract features in such a volatile workload because of the dramatic changes. Thus, we decided to standardize the workload to decrease the fluctuation range, which will also clarify its features and ease their

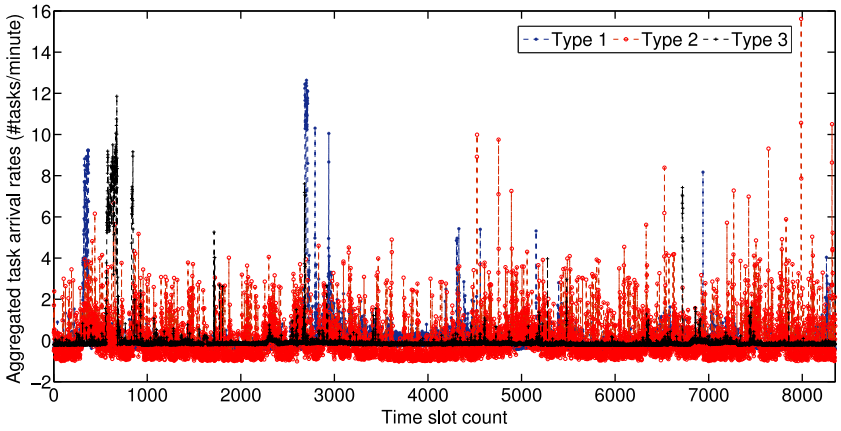


Fig. 3. Aggregated task arriving rates of different types (z-score).

extraction. We first performed the z-score (zero-mean) normalization on the workload to reduce its standard deviation. As shown in Fig. 3, the aggregated task arriving rates of different types become more stable. Here, the aggregated task arriving rates refer to the sum of the number of tasks per minute.

3.1. Savitzky-Golay filter

The non-stationary workload time series was further smoothed to remove outliers and noises. Savitzky-Golay filter [19] is a data smoothing method known for its least square polynomial smoothing. It can eliminate noises while preserving the peak and the width of the signal. In workload collections or transmissions, some data may be abnormal or lost, which may result in interference data. To restore the objective authenticity of data, we experimented three methods, namely, average filter, median filter, and Savitzky-Golay filter to smooth the original workload. We found that the Savitzky-Golay filter with a window size of 5 achieves the best performance.

A workload time series is described as:

$$X=\{x_1, x_2, \cdots, x_t\}, \quad t\in N^+, \tag{1}$$

where X is the workload, and $N^+=\{1, 2, \ldots\}$. x_t is the number of tasks at time slot t. $Y_k(k\in[m+1, t-m])$ is a subsequence of X, and its size is $2m+1$. Y_k is obtained as:

$$Y_k=\{x_{k-m}, \cdots, x_k, \cdots, x_{k+m}\}, \quad k\in[m+1, t-m]. \tag{2}$$

A set of $(2m+1)$ consecutive values were used in the determination of the best mean square fit through these values of a polynomial of degree ϖ (ϖ less than $2m+1$). The coefficients of a polynomial are obtained as:

$$p(n)=\sum_{r=0}^{\varpi} a_r n^r, \quad n\in[-m, m]. \tag{3}$$

Note that, the value of n ranges from $-m$ to m, and that $n=0$ at the central point of the set of $2m+1$ values. Hence, the least squares criterion requires that the sum of the squares of the differences can be a minimum between the observed values x_{k+n}, and the calculated values $p(n)$ over the interval being considered. We obtain

$$\mathcal{E}=\sum_{n=-m}^{m}(p(n)-x_{k+n})^2=\sum_{n=-m}^{m}\left(\sum_{r=0}^{\varpi} a_r n^r - x_{k+n}\right)^2. \tag{4}$$

Thus, the central point of the fitted polynomial is taken as the smoothed data point.

3.2. Augmented Dickey-Fuller test

Augmented Dickey-Fuller (ADF) [16] test is widely adopted to test whether a sequence is stationary. It can evaluate the stationarity of a high-order autoregressive process, by judging whether a unit root exists in a sequence. If there is no unit root, the sequence is considered stationary; otherwise, it is not. We applied the ADF unit root test to Google cluster-usage traces, and the result is summarized in Table 1. It is observed that the p value is less than the significant level of 0.01, so the data can be considered stationary. Besides, the value of white noise is zero. Therefore, the smoothed workload time series has non-white noise, and can be viewed as stationary and predictable.

Table 1
Result of ADF unit root test.

Type	Value
ADF	-9.18045751
p value	2.25931e-15
Critical value (10%)	-2.56695506
Critical value (5%)	-2.86188766
Critical value (1%)	-3.43113669
White noise	0

4. Workload forecasting approach

4.1. Wavelet decomposition

After data preprocessing, we study how to obtain the trend and detail components for different workload time series. Wavelet decomposition is widely adopted to analyze non-stationary and nonlinear signals, since it can reduce non-stationary characteristics of a series and improve the forecasting accuracy. Besides, wavelets can capture the details in data at different scales of resolutions. There are many wavelet algorithms, e.g., Morlet, Mexican Hat and Daubechies wavelets. They can achieve better resolution for a smooth time series. However, they are more time-consuming to calculate than the Haar wavelet. Compared to them, the Haar wavelet requires only additions, instead of miltiplications. Besides, the Haar matrix contains many zero-value elements and therefore its computation time is limited. In addition, it can be used to analyze localized features of a workload time series. Thus, in our work, Haar wavelet decomposition [17] is adopted to extract the characteristics of workload time series.

For a sequence with 2^n numbers, every two adjacent numbers are arranged into a group. Then, the sequence is regarded as a new one with 2^{n-1} groups. We calculate the difference and the sum of two values in each group separately, and obtain two new sequences. Such a process is called one stage of wavelet transformation. The process is repeated recursively, pairing up the sums to prove the next scale and leading to 2^{n-1} differences and a final sum. The mother wavelet of a Haar wavelet is obtained as:

$$\Theta(\gamma) = \begin{cases} 1 & 0 \le \gamma \le 1/2 \\ -1 & 1/2 \le \gamma \le 1 \\ 0 & otherwise, \end{cases} \tag{5}$$

where γ is the support domain of a Haar wavelet. The scaling equation is obtained as:

$$\theta(\gamma) = \begin{cases} 1 & 0 \le \gamma \le 1 \\ 0 & otherwise. \end{cases} \tag{6}$$

For a Haar wavelet, there are Haar matrices. In our work, a Haar matrix $\urcorner$ is obtained as:

$$\urcorner = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{7}$$

With a Haar matrix, we can transform any sequence $(x_1, x_2, x_3, x_4, ..., x_{2t}, x_{2t+1})$ of equal length into a sequence of vectors $((x_1, x_2), (x_3, x_4),..., (x_{2t}, x_{2t+1}))$. The sum s and difference d of each vector are calculated, and the result $((s_1, d_1), (s_2, d_2),..., (s_t, d_t))$ is further processed by the Haar-wavelet transformation.

In our work, we performed one-stage Haar-wavelet decomposition on the time series and obtained two new sequences that reflect different features of the original workload. The first sequence reflects the trend characteristics of the original workload. It treats two successive data as a whole and reflects the overall change characteristics. In the second sequence, the details of the original workload are recorded to calculate the changes between adjacent data. Thus, we obtain change and detail trends of the original workload.

4.2. SCN-based data modeling techniques

In [24], Wang and Li propose a learner model, which is generated incrementally with stochastic configuration networks (SCNs). In our work, we adopt SCNs to construct randomized learner models under a supervised mechanism. Input weights and biases of hidden nodes are randomly assigned in accordance with a supervised mechanism, and output weights are analytically evaluated in a constructive manner. To avoid slow convergence rate in the constructive process, we adopted a calculation method for output weights according to Theorem 7 in [24]. It is demonstrated that the resulting randomized learner models are universal approximators through an SC algorithm, which is obtained as below:

- **Configuration of Hidden Parameters**: Randomly assign input weights and biases to meet

$$\langle e^*_{L-1,q}, g_L \rangle^2 \ge b_g^2 \delta^*_{L,q}, \tag{8}$$

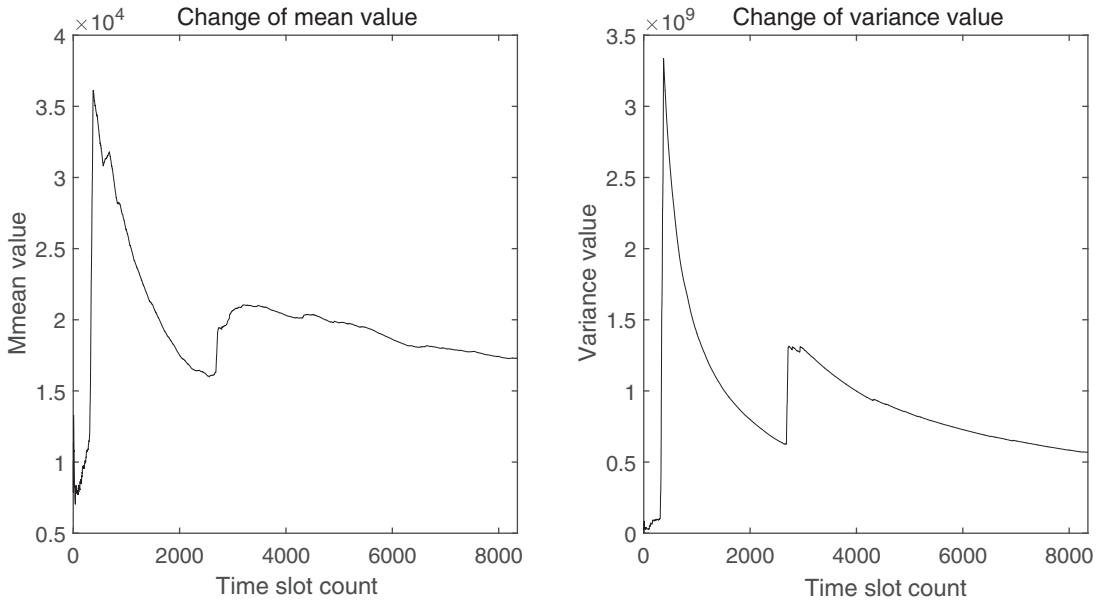


Fig. 4. Changes of mean and variance.

where $L = \{1, 2, \ldots\}$, $q \in \{1, 2, \ldots, u\}$, $e_{L-1,q}$ is the current residual error, L is the number of hidden nodes, g_L is the random basis function, and $\forall g \in \Gamma$, $0 < \|g\| < b_g$ for some $b_g \in \mathbb{R}^+$. It is assumed that span(Γ) is dense in an L_2 space, and a non-negative real number sequence $\{\mu_L\}$ with $lim_{L\to+\infty}\mu_L = 0$ and $\mu_L \le (1-r)$ $(0 < r < 1)$. $\delta_L^* = \sum_{q=1}^{u} \delta_{L,q}^*$ and $\delta_{L,q}^* = (1 - r - \mu_L) \parallel e_{L-1,q}^* \parallel^2$. Then, we generate a new hidden node and add it to the current model.

- **Evaluation of Output Weights**: Selectively determine output weights of the current model, and they are obtained as:

$$[\beta_1^*, \beta_2^*, \ldots, \beta_L^*] = \arg\min_{\beta} \left\| f - \sum_{j=1}^{L} \beta_j g_j \right\|. \tag{9}$$

Then, we have $lim_{L\to+\infty} \parallel f - f_L^* \parallel = 0$. In the training process, our work adopts a method similar to the SC-III Algorithm in [24].

The parameters of SCN are described as:

- $L_{\max}$ is the number of maximum hidden nodes.
- $T_{\max}$ is the number of maximum random configurations.
- ϵ is the expected error tolerance.
- The scope control set $\Upsilon := \{\lambda_1 : \Delta\lambda : \lambda_{\max}\}$ plays an important role in setting ranges of random parameters including weight ω_L and bias b_L.
- r is the learning parameter.
- np is the number of nodes added into each loop of the network.

SC algorithms quickly rebuild a learner model when it is over-fitting in training data, instead of reproducing a new learner model or keeping all weights in each iteration for model retrieval. It provides more confidence and high flexibility to users. After the above steps, a forecasting method is obtained. Then, the one-step forecasting of the workload time series is realized. It is worth noting that it is easy to implement the proposed method in real-life Geo-2DCs once previous workload time series is collected and become available at current time slot. Specifically, the Savitzky-Golay filter is adopted to smooth workload time series that is further decomposed into trend and detail components via wavelet decomposition. Finally, SGW-SCN is established to obtain optimal parameter setting in the forecasting models. The predicted results are reconstructed via wavelet reduction to obtain the number of arriving tasks in current time slot.

5. Performance evaluation

5.1. Data preprocessing

Fig. 4 shows the changes of mean and variance of the original workload time series. As shown in Fig. 4, the values of mean and variance both change greatly. Besides, neither of them stays stable abound a constant, and it is clear that the workload time series does not meet the stationarity constraint.

Table 2
M_{SE} with different window sizes.

Size\M_{SE}	MM Filter	MA Filter	SG Filter
3	0.1137	0.1434	0.2278
5	**0.1144**	**0.1234**	**0.0638**
7	0.1140	0.1480	0.0902
9	0.1138	0.1542	0.1219

Table 3
M_{SE} of four sequences.

Group\M_{SE}	Sample A	Sample B	Sample C
Original	0.2664	0.1878	0.1412
MA Filter	0.1234	0.0984	0.0827
MM Filter	0.1144	0.0881	0.0664
SG Filter	**0.0638**	**0.0446**	**0.0404**

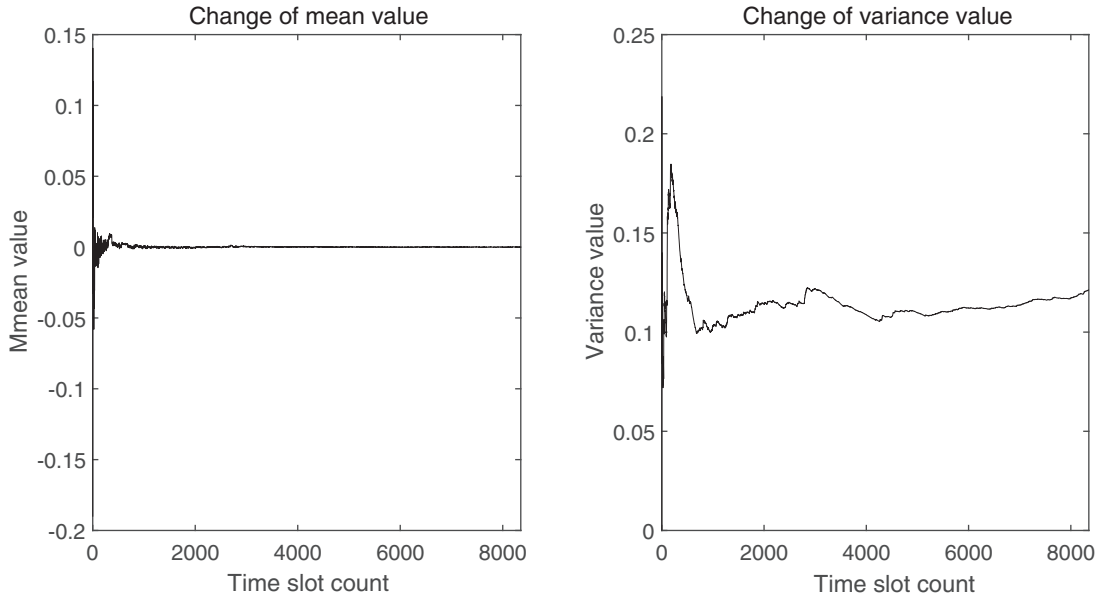


Fig. 5. Mean and variance changes of smoothed sequence.

We further evaluated the performance of the Savitzky-Golay (SG) filter by comparing it with two typical filter methods including the Moving Median (MM) filter and the Moving Average (MA) filter. After extensive tests, we obtained the best filter effect when the window size is 5. Table 2 shows the Mean Squared Error (M_{SE}) with different window sizes. M_{SE} [18] is defined as:

$$M_{SE}=\frac{\sum_{\nu=1}^{\Upsilon}(\widehat{y}_{\nu}-y_{\nu})^2}{\Upsilon}, \tag{10}$$

where y_{ν} and $\widehat{y}_{\nu}$ represent outputs of normalized and predicted data at time slot ν, respectively, and Υ is the total number of observations in the dataset.

We further evaluated the forecasting model with three different sets of test data (samples A, B, and C). Table 3 shows M_{SE} of four sequences including the original, MA filter, MM filter, and SG filter data. It is clearly observed that the Savitzky-Golay filter achieves higher forecasting accuracy than its two peers. Based on the smoothed sequence, the ADF test was adopted to determine whether it is stationary. A differential operation was conducted on non-stationary sequences, and the ADF test was performed repeatedly until the sequence meets the stationarity condition.

Afterwards, we obtained a stationary sequence smoothed by the Savitzky-Golay filter. Fig. 5 shows its mean and variance trends. It is shown that the mean is stable at 0, and the variance is also stable at a constant. This means that the influence of extrema on the stationarity of the sequence is eliminated.

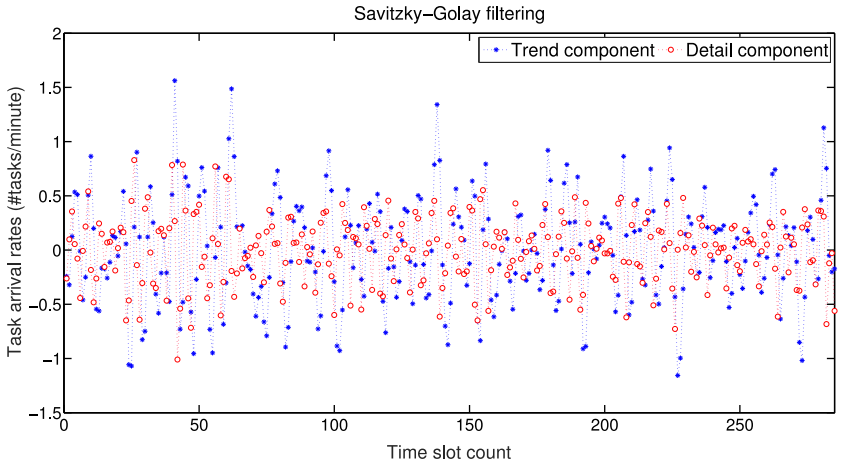


Fig. 6. Trend and detail components with wavelet decomposition.

Table 4
The training process of $L_{\max}$.

$L_{\max}$	M_{SE}	R^2	Training time (s)
10	0.1057	0.5678	0.74
20	0.0901	0.6316	1.53
30	0.0798	0.6738	2.46
40	0.0661	0.7298	3.45
50	0.0681	0.7217	4.86
60	0.0635	0.7404	6.72
70	**0.0627**	**0.7438**	**7.59**
80	0.0647	0.7355	14.25
90	0.0673	0.7248	14.42
100	0.0664	0.7284	27.64
150	0.0688	0.7188	39.56
200	0.0719	0.7062	94.09

Table 5
The training process of $T_{\max}$.

$T_{\max}$	M_{SE}	R^2	Training time (s)
1	0.0633	0.6411	7.05
5	0.0633	0.6982	12.41
10	0.0628	0.7432	8.06
20	0.0635	0.7404	8.59
50	0.0647	0.7356	11.28
100	**0.0627**	**0.7438**	**7.59**
150	0.0646	0.7360	22.07
200	0.0631	0.7422	21.34
300	0.0630	0.7424	28.52
500	0.0623	0.7451	45.94

We further applied wavelet decomposition to the workload time series. The order of Haar wavelet was set to 1. For example, we gained its trend and detail components with wavelet decomposition for workload time series in the 25th day, and the result is shown in Fig. 6.

Furthermore, SCN was adopted to build the forecasting model. In our work, we adopted the parameter setting of SCNs for workload time series. In the Savitzky-Golay filter, m=2 and ϖ=3. We obtained the arriving rate of smoothed tasks of each type in 29 days. We further gave the training process of SCN parameters with Google workload dataset in days 1–25. The test dataset was used in days 26–29. SCN parameters were set as:

- In the training process of SCN, the number of hidden layer nodes is increased gradually. When it increases to $L_{\max}$ or $e_{L-1,q}$ is less than the predefined expected error tolerance ϵ, the training is terminated. The training process of $L_{\max}$ is shown in Table 4 and it is shown that $L_{\max} = 70$ achieves the best performance.
- The training process of $T_{\max}$ is shown in Table 5 and it is shown that $T_{\max} = 100$ achieves the best performance.
- The error tolerance is set to 0.00001, i.e., $\epsilon = 0.00001$.
- $\lambda \in \{0.5, 1, 5, 10, 30, 50, 100, 150, 200, 250\}$.

Table 6
Performance comparison of different methods for type-1 application.

Methods	M_{SE}		R^2		Training
	Original	Smoothed	Original	Smoothed	time (s)
ARIMA	0.0655	N/A	0.0657	N/A	62.44
BPNN	0.0838	N/A	-0.1945	N/A	80.10
SCN	0.0526	N/A	0.2507	N/A	7.30
WARIMA	0.0686	N/A	0.0223	N/A	91.10
WBPNN	0.0768	N/A	-0.0940	N/A	183.52
WSCN	0.0750	N/A	-0.0690	N/A	15.75
SG-ARIMA	0.0203	0.0068	0.7109	0.9326	44.32
SG-BPNN	0.0216	0.0089	0.6921	0.9306	75.50
SG-SCN	0.0187	0.0072	0.7333	0.7377	13.21
SGW-ARIMA	0.0169	0.0077	0.7589	0.8364	94.33
SGW-BPNN	0.0231	0.0073	0.6705	0.9702	172.89
SGW-SCN	**0.0166**	**0.0058**	**0.7627**	**0.8760**	**14.04**

Table 7
Performance comparison of different methods for type-2 application.

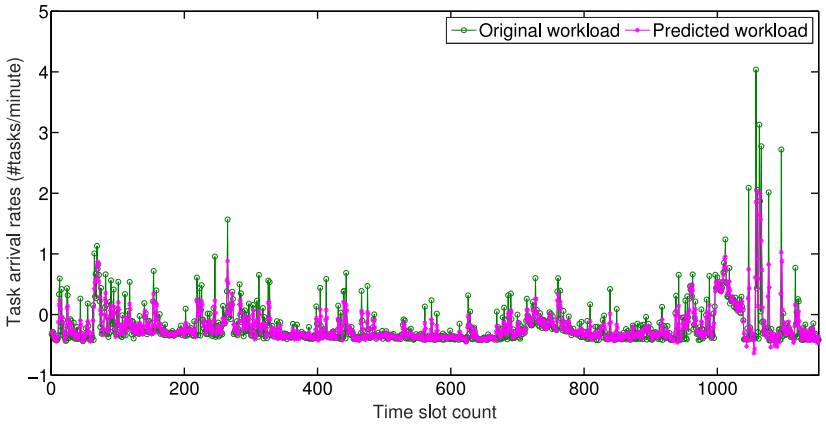
Methods	M_{SE}		R^2		Training
	Original	Smoothed	Original	Smoothed	time (s)
ARIMA	1.5196	N/A	0.1383	N/A	73.16
BPNN	2.4818	N/A	-0.4073	N/A	90.25
SCN	1.4750	N/A	0.1636	N/A	7.28
WARIMA	1.6491	N/A	0.0649	N/A	106.25
WBPNN	1.7197	N/A	0.0249	N/A	173.60
WSCN	1.7049	N/A	0.0332	N/A	14.58
SG-ARIMA	0.5187	0.3075	0.7059	0.7566	58.78
SG-BPNN	0.7504	0.5406	0.5745	0.5722	87.45
SG-SCN	0.5657	0.3487	0.6792	0.7240	13.53
SGW-ARIMA	0.4609	0.1961	0.7654	0.8465	127.32
SGW-BPNN	0.5293	0.2760	0.6998	0.7816	180.63
SGW-SCN	**0.4130**	**0.1928**	**0.7658**	**0.8475**	**14.16**

Table 8
Performance comparison of different methods for type-3 application.

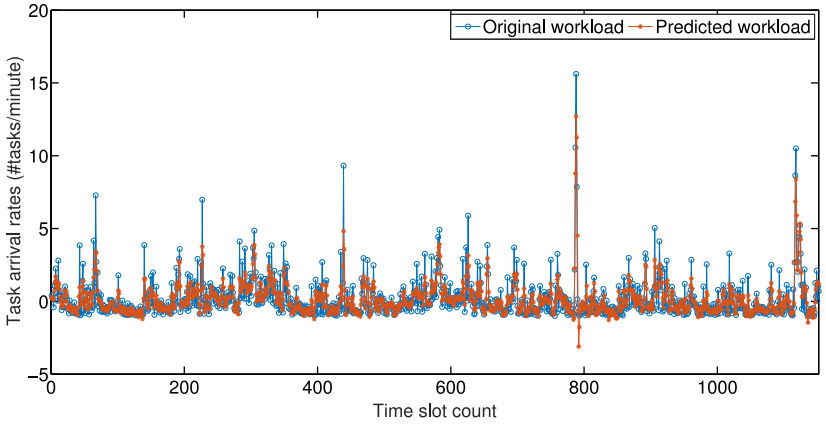
Methods	M_{SE}		R^2		Training
	Original	Smoothed	Original	Smoothed	time (s)
ARIMA	0.0252	N/A	-0.3449	N/A	48.34
BPNN	0.0310	N/A	-0.6538	N/A	78.05
SCN	0.0264	N/A	-0.4096	N/A	-0.80
WARIMA	0.0227	N/A	-0.2114	N/A	110.02
WBPNN	0.0280	N/A	-0.4972	N/A	158.35
WSCN	0.0235	N/A	-0.2535	N/A	14.06
SG-ARIMA	0.0043	0.0050	0.7692	0.5351	54.19
SG-BPNN	0.0070	0.0068	0.6261	0.3736	76.75
SG-SCN	0.0049	0.0055	0.7370	0.4922	12.83
SGW-ARIMA	0.0079	0.0028	0.6261	0.7356	112.38
SGW-BPNN	0.0080	0.0037	0.5744	0.6628	161.33
SGW-SCN	**0.0071**	**0.0024**	**0.6803**	**0.7756**	**13.79**

- $r \in \{0.9, 0.99, 0.999, 0.9999, 0.99999, 0.999999\}$.
- The number of hidden layer nodes is set to 70 and np=1.

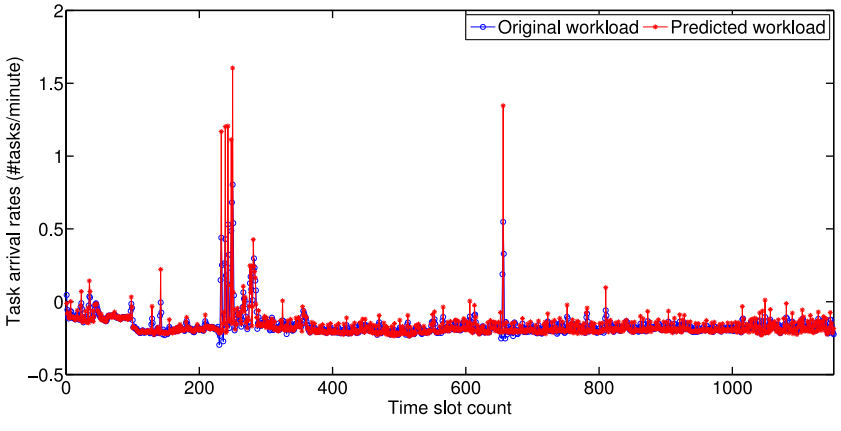
In these experiments, we adopted a control variable method to determine the optimal value of a specific variable. Then, the number of tasks arriving at the next time slot was obtained by wavelet reconstruction.



(a) Type 1



(b) Type 2



(c) Type 3

Fig. 7. Forecasting results of three types of task arriving rates.

Table 9
Robustness analysis of SCN for the training process.

Length of input	$L_{\max}$	$T_{\max}$	M_{SE}	R^2
10	70	100	0.0834	0.6589
20	70	100	0.0742	0.6966
30	70	100	0.0701	0.7135
40	70	100	0.0680	0.7221
50	70	100	0.0674	0.7244
70	**70**	**100**	**0.0627**	**0.7438**
70	100	100	0.0664	0.7284
70	150	100	0.0688	0.7188
70	200	100	0.0719	0.7062
70	300	100	0.0739	0.6979

5.2. Evaluation criteria

We evaluated the accuracy of the forecasting models generated from different learning algorithms with M_{SE} [18], and R^2 [2] that are measures of the goodness-of-fit of a forecasting model:

$$R^2=\frac{\sum_{\nu=1}^{\Upsilon}(\widehat{y}_\nu-\bar{y})^2}{\sum_{\nu=1}^{\Upsilon}(y_\nu-\bar{y})^2}, \tag{11}$$

where $\bar{y}=\frac{1}{\Upsilon}\sum_{\nu=1}^{\Upsilon}y_\nu$, y_ν denotes the output of original normalized data and $\widehat{y}_\nu$ denotes the predicted output, and Υ denotes the number of observations. R^2 describes the fitting ability of a model, and its value falls within the range [0, 1]. $R^2 = 1.0$ indicates a perfect forecasting model. Tables 6–8 show their performance comparison of different models from two aspects. Firstly, we compared the predicted results with the original data that is not smoothed by Savitzky-Golay filter. The results are labeled as Original. Secondly, we compared the predicted results with the data smoothed by the Savitzky-Golay filter. The results are labeled as Smoothed.

Tables 6–8 show that SGW-SCN achieves better forecasting accuracy with shorter training time than other methods. Besides, we applied the Savitzky-Golay filter method to the original workload data series in the performance comparison of the last six methods. However, we did not adopt the Savitzky-Golay filter method in the first six methods. Therefore, we did not show the performance comparison of the smoothed workload data series in the first six methods in Tables 6–8.

Furthermore, we conducted the robustness analysis for SGW-SCN through the adjustment of system parameters. Table 9 shows that more accurate forecasting results for the original workload time series are achieved when $L_{\max}$ and $T_{\max}$ are set to 70 and 100, respectively. In addition, the lengths of input vectors of BPNN and SCN are both set to 70.

5.3. Forecasting result

We further verified whether the proposed model can predict different types of tasks. Fig. 7 shows the forecasting results of three types of task arriving rates. Based on the forecasting results in Fig. 7 and the performance comparison in Tables 6–8, we can draw the following conclusions:

- A more accurate forecasting model is established by the data smoothed with the Savitzky-Golay filter.
- Wavelet decomposition improves the forecasting accuracy of Google workload time series.
- SGW-SCN achieves better forecasting performance than other typical methods when they are applied to the same workload dataset.

6. Conclusions

It is critically important for geo-distributed cloud data centers (Geo-2DCs) to improve their energy efficiency and to reduce energy consumption. Accurate forecasting of arriving tasks plays an important role in achieving so since it can be used to achieve the optimal resource provisioning for tasks. However, it is difficult to accurately predict because of the irregularity and complexity of the workload time series. This paper presents an integrated forecasting method resulting from a novel combination of **S**avitzky-**G**olay filter, **W**avelet decomposition (SGW) techniques and **S**tochastic **C**onfiguration **N**etworks (SCNs). SGW-SCN is established to predict arriving tasks for Geo-2DCs at the following time slot. Our simulation results based on real-life workload have demonstrated that our integrated model achieves more accurate forecasting results and faster learning speed than state-of-the-art forecasting methods. In the future work, we plan to apply deep learning methods to further improve the forecasting accuracy of large-scale workload. In addition, we plan to analyze spatial and temporal characteristics of workload, and use them to further improve the forecasting accuracy.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grants 61703011 and 61802015.

References

[1] D. Ardagna, S. Casolari, M. Colajanni, B. Panicucci, Dual time-scale distributed capacity allocation and load redirect algorithms for cloud systems, J. Parallel Distrib. Comput. 72 (6) (2012) 796–808.
[2] S.B. Achelis, Technical analysis from a to z, McGraw Hill, New York, 2001.
[3] J. Bi, H. Yuan, W. Tan, B.H. Li, TRS: temporal request scheduling with bounded delay assurance in a green cloud data center, Inf. Sci. (Ny) 360 (2016) 57–72.
[4] J. Bi, H. Yuan, W. Tan, M.C. Zhou, Y. Fan, J. Zhang, J. Li, Application-aware dynamic fine-grained resource provisioning in a virtualized cloud data center, IEEE Trans. Autom. Sci. Eng. 2 (14) (2017) 1172–1184.
[5] R.N. Calheiros, E. Masoumi, R. Ranjan, R. Buyya, Workload prediction using ARIMA model and its impact on cloud applications' qos, IEEE Trans. Cloud Comput. 3 (2015) 449–458.
[6] Y.C. Chang, R.S. Chang, F.W. Chuang, A predictive method for workload forecasting in the cloud environment, in: Advanced Technologies, Embedded and Multimedia for Human-Centric Computing, Springer, Dordrecht, 2014, pp. 577–585.
[7] Z. Chen, Y. Zhu, Y. Di, S. Feng, Self-adaptive prediction of cloud resource demands using ensemble model and subtractive-fuzzy clustering based fuzzy neural network, Comput. Intell. Neurosci. 2015 (2015) 1–14.
[8] S. Di, D. Kondo, W. Cirne, Host load prediction in a Google compute cloud with a Bayesian model, in: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, November 10–16, Salt Lake City, Utah, USA, 2012, pp. 1–11.
[9] Y. Hu, J.J. You, J.N. Liu, T. He, An eigenvector based center selection for fast training scheme of RBFNN, Inf. Sci. (Ny) 428 (2018) 62–75.
[10] S. Islam, J. Keung, K. Lee, A. Liu, Empirical prediction models for adaptive resource provisioning in the cloud, Future Gener. Comput. Syst. 28 (1) (2012) 155–162.
[11] J. Kumar, A.K. Singh, Workload prediction in cloud using artificial neural network and adaptive differential evolution, Future Gener. Comput. Syst. 81 (2018) 41–52.
[12] D.S. Linthicum, Cloud computing changes data integration forever: what's needed right now, IEEE Cloud Comput. 4 (2017) 50–53.
[13] Q. Liang, J. Zhang, Y.H. Zhang, J.M. Liang, The placement method of resources and applications based on request prediction in cloud data center, Inf. Sci. (Ny) 279 (2014) 735–745.
[14] Z. Liu, Y. Chen, C. Bash, A. Wierman, D. Gmach, Z. Wang, C. Hyser, Renewable and cooling aware workload management for sustainable data centers, ACM SIGMETRICS Perform. Eval. Rev. 40 (1) (2012) 175–186.
[15] A.K. Mishra, J.L. Hellerstein, W. Cirne, C.R. Das, Towards characterizing cloud backend workload: insights from google compute clusters, ACM SIGMETRICS Perform. Eval. Rev. 37 (4) (2010) 34–41.
[16] H. Michio, Time-Series-Based Econometrics: Unit Roots and Cointegration, Oxford University Press, New York, 1996. 48–49.
[17] J. Nunez, X. Otazu, O. Fors, A. Prades, V. Pala, R. Arbiol, Multiresolution-based image fusion with additive wavelet decomposition, IEEE Trans. Geosci. Remote Sens. 37 (3) (1999) 1204–1211.
[18] S. Ohno, T. Shiraki, M.R. Tariq, M. Nagahara, Mean squared error analysis of quantizers with error feedback, IEEE Trans. Signal Process. 65 (22) (2017) 5970–5981.
[19] A. Savitzky, M.J.E. Golay, Smoothing and differentiation of data by simplified least squares procedures, Anal. Chem. 36 (8) (1964) 1627–1639.
[20] B. Urgaonkar, P. Shenoy, A. Chandra, P. Goyal, T. Wood, Agile dynamic provisioning of multi-tier internet applications, ACM Trans. Auton. Adapt. Syst. 3 (1) (2008) 1–39.
[21] M. Vitali, B. Pernici, U.M. O'Reilly, Learning a goal-oriented model for energy efficient adaptive applications in data centers, Inf. Sci. (Ny) 319 (2015) 152–170.
[22] R.V.d. Bossche, K. Vanmechelen, J. Broeckhove, Iaas reserved contract procurement optimisation with load prediction, Future Gener. Comput. Syst. 53 (2015) 13–24.
[23] D. Wang, C. Cui, Stochastic configuration networks ensemble with heterogeneous features for large-scale data analytics, Inf. Sci. (Ny) 417 (2017) 55–71.
[24] D. Wang, M. Li, Stochastic configuration networks: fundamentals and algorithms, IEEE Trans. Cybern. 47 (10) (2017) 3466–3479.
[25] D. Wang, M. Li, Deep stochastic configuration networks with universal approximation property, in: Proceedings of 2018 International Joint Conference on Neural Networks, July 8–13, Rio de Janeiro, Brazil, 2018, pp. 1–12.
[26] D. Wang, M. Li, Robust stochastic configuration networks with kernel density estimation for uncertain data regression, Inf. Sci. (Ny) 412 (2017) 210–222.
[27] J. Zheng, L.M. Ni, Time-dependent trajectory regression on road networks via multi-task learning, in: Proceedings of the 27th AAAI Conference on Artificial Intelligence, July 14–18, Bellevue, Washington USA, 2013, pp. 1048–1055.
[28] Q. Zhang, M.F. Zhani, R. Boutaba, J.L. Hellerstein, Dynamic heterogeneity-aware resource provisioning in the cloud, IEEE Trans. Cloud Comput. 2 (1) (2014) 14–28.
[29] L. Zhao, Q. Sun, J. Ye, F. Chen, C.T. Lu, N. Ramakrishnan, Multi-task learning for spatio-temporal event forecasting, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 10–13, Sydney, NSW, Australia, 2015, pp. 1503–1512.
[30] J. Zhang, Z. Wei, Z. Yan, M.C. Zhou, A. Pani, Online change-point detection in sparse time series with application to online advertising, IEEE Trans. Syst. Man Cybern. 99 (2017) 1–11.