An Adaptive Short-Term Prediction Algorithm for Resource Demands in Cloud Computing

JING CHEN1,2 AND YINGLONG WANG2
1College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China
2Shandong Provincial Key Laboratory of Computer Networks, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan 250101, China

Corresponding author: Yinglong Wang (wangylscsc@126.com)

This work was supported in part by the Shandong Provincial Natural Science Foundation under Grant ZR2019QF014.

ABSTRACT Cloud computing has been widely applied in various fields with the development of big data and artificial intelligence. The associated resource demands exhibit characteristics such as diversity, large scale, burst and uncertainty. This paper analyzes these characteristics of cloud resource demands based on Alibaba cluster data, and proposes an adaptive short-term prediction algorithm for those demands. The proposed algorithm uses a principal component analysis method to extract the primary types of container demands from a time series of resource demands, and executes outlier detection and replacement to obtain a more stationary sequence. An adaptive short-term prediction strategy is proposed to adaptively select a higher-accuracy short-term prediction method to implement the prediction. Further, an error adjustment factor is proposed to reduce the prediction error. Thus, the short-term prediction accuracy of cloud resource demands is improved via outlier detection and replacement, an adaptive selection strategy and an error adjustment. We evaluated the effectiveness of these improvements, and compared our algorithm with existing algorithms in terms of effectiveness and time cost. The experimental results demonstrate that the proposed algorithm improves short-term prediction accuracy effectively.

INDEX TERMS Cloud computing, cloud resource demand, short-term prediction, adaptive selection strategy, error adjustment.

I. INTRODUCTION

Cloud computing has been widely applied in various fields owing to its rich resources, on-demand resource provision, low resource costs, and elastic resource scaling. A large number of applications have been deployed on cloud platforms. Effective resource allocation and task scheduling methods are keys to guaranteeing the quality of service (QoS) and maximizing the profit of cloud services [1]–[6]. With recent developments in cloud computing, big data, and artificial intelligence, massive virtual machine (VM) or container requests are submitted to a cloud platform synchronously, which causes the diversity, large scale, burst, and uncertainty in resource demands. Therefore, accurately predicting the future VM or container demands becomes crucial to ensure the timely allocation of appropriate resources. More specifically, the rapid and accurate prediction of cloud resource demands can help cloud service providers select suitable physical servers or enable stand-by servers to place a large number of VMs or containers in advance. Thus, in addition to guaranteeing the QoS, the resource utilization can also be enhanced.

A cloud platform provides various resources to automatically create VMs or containers and deploy users’ applications. The platform can also flexibly increase or decrease the number of VMs or containers according to changes in users’ resource demands and resource loads. However, a cloud platform carries numerous applications and services, which requires a diverse selection of VMs or containers. For instance, a log analysis application requires 20 VMs (2-core CPU, 4 G memory) with Spark, and a recommendation application requires hundreds of VMs (4-core CPU, 8 G memory) with Hadoop. Thousands of VMs or containers may be requested per minute. Their creation and start-up generally take tens of minutes or more, which results in a temporal discrepancy between resource allocation and resource requests, denying a guaranteed QoS on the part of the cloud platform. Additionally, increased volumes of resource demands can
lead to rapid changes and strong fluctuations in the server load. If the server load changes at a faster rate than that of the migration of VMs on the server, the performance of applications running on these VMs can be affected, or some unnecessary migrations of VMs may cause the cloud platform to become unstable. These problems can be alleviated via the prediction of future resource demands and server loads.

Time series prediction has been widely studied [7]. The UCR time series archive has become an important archive including a large amount of linear or nonlinear time series [8]. Ensemble Empirical Mode Decomposition (EEMD) is an effective method for handling a nonlinear and non-stationary time series [9], [10]. The EEMD-Autoregressive Integrated Moving Average (ARIMA) and EEMD-runs test (RT)-ARIMA methods reported in our previous studies [11], [12] demonstrate the effectiveness of non-stationary time series prediction. This paper further proposes an adaptive short-term prediction algorithm based on our previous EEMD-ARIMA and EEMD-RT-ARIMA methods. This algorithm aims to further improve the prediction accuracy via the following three steps.

1) Preprocess the burst or abnormal data to improve the prediction accuracy of an unstable sequence.
2) Provide an adaptive selection strategy to select a better prediction algorithm based on a dynamic threshold.
3) Propose an error adjustment factor to improve the prediction accuracy.

This paper first analyzes the characteristics of cloud resource demands and resource loads, then uses some methods to process the non-stationary sequences, and finally proposes an adaptive selection strategy and an error adjustment factor to improve the short-term prediction accuracy of cloud resource demands. The performance of our adaptive short-term prediction algorithm is verified through experiments conducted on cloud cluster data.

A list of the mathematical notations used in this paper is given in Table 1.

### II. RELATED WORKS

Current prediction methods can be categorized into three groups: linear, non-linear, and hybrid methods [13]. The general linear prediction methods include Moving Average (MA) [14], Exponential Smoothing (ES) [15], Autoregressive (AR) [16], Autoregressive Moving Average (ARMA) [17], and ARIMA [18], [19]. These methods are usually applied to prediction in the case of a linear stationary or non-stationary time series. A linear stationary time series exhibits little variance in its average value, whereas a linear non-stationary time series exhibits an obvious change in its average value over time. An existing VM consolidation approach adopts a regression-based model to predict the future CPU and memory utilization of VMs and physical machines (PMs) [20]. ARIMA is a traditional prediction model for non-stationary time series. ARIMA models have been employed in the long-term prediction of server workload [21] and a Gaussian interval type-2 fuzzy set theory has been used to predict the long-term traffic volume [22]. However, short-term prediction in minutes is more difficult than long-term prediction in days or weeks [23]. A short-term prediction model has been proposed, which uses an ARIMA model to predict the future workload to ensure cloud applications’ QoS [24]. Although the ARIMA prediction model is effective, its prediction accuracy needs to be improved, and various studies have combined ARIMA with other methods to achieve such an improvement. A model combining ARIMA and fuzzy regression has been used to predict network traffic in cloud computing, which improves the prediction accuracy by adopting a sliding window [25]. An energy-aware cost prediction framework has also been proposed, which uses ARIMA and linear regression models to predict the VM and PM workloads, PM power consumption, and their total cost [26]. An adaptive workload forecasting method has been proposed to improve the forecasting accuracy, which uses a combination of alternative forecasting methods, such as, Simple Exponential Smoothing (SES), ARIMA, Linear Regression (LR), and a set of training data window sizes, to predict the workload in advance [27]. With this method, a better prediction model is selected to execute the next prediction based on the previous predictions of a set of models and training data window sizes, progressively increasing the prediction cost. Although these combined methods improve the prediction accuracy to a certain degree, they do not consider the impact of outliers on the prediction accuracy of extremely non-stationary time series in the real world.

Unlike a linear time series, there are large numbers of nonlinear datasets that cannot be represented through mathematical expressions. Thus, in this case, an accurate prediction is difficult owing to the randomness and uncertainty. Machine- or deep-learning methods are generally used to solve the unpredictability of nonlinear problems, such as
Genetic Algorithm (GA) [28], Neural Network (NN) [29], Support Vector Machine (SVM) [30], Support Vector Regression (SVR) [31], Deep Belief Network (DBN) [32], and BackPropagation Neural Network (BPNN) [33]. A GA-based prediction method has been proposed for resource utilization of VMs and PMs [28]. A cross-correlation prediction approach is presented based on SVM, which uses the cross relation of VMs running the same application to improve prediction accuracy [30]. In addition, a long short-term memory (LSTM) method has been used to predict the dynamic network traffic to obtain a low transmission latency and power consumption [34]. Further, a prediction method using an LSTM encoder-decoder has been proposed to predict the host load, in which an entire input sequence is regarded as an internal representation to improve the memory capability of LSTM and consequently improve the prediction accuracy [35]. An experimental comparison between one-step-ahead prediction and a multi-time-step-ahead prediction has been conducted, demonstrating that Recurrent Neural Network (RNN) is more effective in improving the multi-time step prediction accuracy of the CPU and bandwidth utilization, reducing service-level agreement (SLA) violations and enhancing the efficiency of a cloud data center [36]. An NN-based regression prediction has been conducted on the energy usage and power source output [37]. A Bayesian-based prediction model of virtual resources has also been proposed, in which the correlations among the parameter variables have been identified to improve the resource prediction accuracy [38]. In addition, a novel Dendritic Neutron Model (DNM) has been proposed to solve the classification, approximation, and prediction problems, which considers the nonlinearity of the synapses and uses effective learning algorithms to train the DNM [39]. Although machine- and deep-learning methods are effective in improving the prediction accuracy, they need to train a large amount of historical data to extract the relevant features and build a complex model, which costs too much time to ensure the QoS of cloud services.

The difficulty in determining a time series as linear or nonlinear provides a further challenge to the selection of an appropriate prediction algorithm. It is clearly preferable to select the most accurate method. However, the selected method is not always the best owing to variances among the different samples. In practice, the time series considered is generally not purely linear or nonlinear but a combination of the two. Thus, a combination of different prediction methods can address the prediction in such a case. A load prediction algorithm has been proposed to improve the prediction accuracy using ARIMA and BP to extract linear and nonlinear features of historical data, respectively [40]. ARIMA and BPNN algorithms have also been combined to predict the load of the edge cloud cluster [41]. Further, an adaptive prediction model using LR, ARIMA, and SVR has been proposed for web applications, where a workload classifier selects the model according to the workload features [42]. The ensemble model ESNemble extracts and combines the features of multiple prediction algorithms to forecast the workload time series based on an echo state network [43]. In addition, a workload prediction algorithm W3PSG has been proposed for dynamic resource allocation in mobile cloud computing. This method uses a wavelet transformation method to decompose the workload into different frequency sub-time-series, and different prediction methods, such as GRACH, PSVR, and SVR, or a combination of the three, are applied in the prediction of the sub-time-series to ensure a high accuracy and low computational costs [44]. An integrated workload forecasting method, Savitzky-Golay and Wavelet-supported Stochastic Configuration Networks (SGW-SCN) has been proposed to predict the future workload. This method first uses a Savitzky-Golay filter to smooth the time series, and then decomposes it into multiple components through wavelets [45]. The EEMD-ARIMA and EEMD-RT-ARIMA methods reported in our previous studies also decompose a non-stationary sequence into multiple stationary components, and then implement the predictions for these components, and compose the various prediction results together to achieve the final prediction results. In this paper, we further propose an adaptive selection strategy to select a higher-accuracy method between them for predicting the number of future containers.

Additionally, an integrated prediction method has been proposed, which smoothens and decomposes a task time series into multiple components using the Savitzky-Golay filter and wavelet decomposition, and then uses stochastic configuration networks and a wavelet reconstruction to predict the number of future tasks [46]. An existing proactive and self-adaptive prediction method for resource demands minimizes under- and over-prediction via a prediction adjustment and padding based on the error probabilities [47]. A self-adaptive resource allocation approach has also been proposed based on the progressive QoS prediction model, which uses a self-tuning control to improve the prediction accuracy based on the runtime data [48]. A cloud workload forecasting model uses an error preventive score from the previous forecasting errors to improve the future forecasts [49]. In this study, the RT values and previous prediction errors are used to select a better prediction method and adjust the deviation between the predicted values and the actual values for an improved prediction accuracy.

III. CHARACTERISTIC ANALYSIS OF CLOUD RESOURCE DEMANDS

Cloud resource demands change dynamically with the deployment of larger numbers of applications and services. We utilize the cluster data to analyze the characteristics of cloud resource demands on Alibaba cloud [50]. The data are gathered from online services and batch jobs. A total of 4,034 physical servers and 71,476 containers are used by online services during an 8-day period. Figures 1 and 2 show the numbers of containers used by online services over intervals of 2 minutes and 5 minutes, respectively.
A. BURST, BATCH, AND UNCERTAINTY OF CLOUD RESOURCE DEMANDS

As illustrated in Figure 2, the number of containers used by online services fluctuates significantly over time. The maximum number of containers used is 2,767, whereas the minimum number is zero. In addition, the number of containers changes continuously over time, with some sudden bursts. During these bursts, the resource demands change from hundreds of containers to thousands. We ignore those containers that are actually created before time 0, and regard the time of first use of a container as its creation time. The number of new containers created for online services is counted over intervals of 2 minutes and 5 minutes, as shown in Figures 3 and 4, respectively.

It can be seen that the number of new containers changes dynamically with the demand of online services and exhibits randomness, uncertainty, and non-uniformity in the distribution, making it difficult to predict the resource demands of a cloud platform. We further analyze the probability density function (PDF) and cumulative distribution function (CDF) of the number of new containers for online services over intervals of 5 minutes, as shown in Figure 5. There is a significant probability regarding the demand of a small number of new containers, which indicates that online services require a small number of new containers in most time periods and a large number of new containers in short time periods. A non-uniformity of the resource demand can also be observed.

B. DIVERSITY AND NON-STATIONARY NATURE OF CLOUD RESOURCE DEMANDS

The Alibaba cluster data consist of 33 valid types of containers, each of which corresponds to a different number of CPU cores and memory size. Figure 6 depicts the resource demands for different types of containers. Each sector represents the percentage of the demand for one type of containers to the total number of containers. For instance, CPU = 400 and mem = 1.56 denote a 4-core CPU and 1.56 memory. Here, 100 indicates one CPU core. Memory is the normalized data within the range of [0, 100]. Online services have the greatest demand for 4-core CPU and 1.56 memory type containers at up to 75%, followed by 8-core CPU and 3.13 memory type containers at 17%, and 8-core CPU and 6.25 memory...
type containers at 2%, with the other 29 types of containers accounting for only 5%.

We further analyze the variation in the number of containers over time for four main types indicated in Figure 7. The number of each type of containers changes drastically over time, creating an extreme instability. The number of containers increases by a factor of hundreds during certain intervals, and quickly decreases during other intervals. However, the number of each type of containers can be observed to increase or decrease almost synchronously.

C. STRONG VOLATILITY AND INSTABILITY OF CLOUD RESOURCE LOAD

These data indicate the resource utilization of 155 servers over a 28 h period. Figure 8 shows the CPU and memory utilization of a server over time. The average CPU utilization fluctuates radically. The average memory utilization is high (over 80% for more than 99% of the total duration), and the average CPU utilization is relatively low (below 50% for 90% of the total duration). The average memory utilization is even over 95% in some time periods. Thus, the data indicate that the average memory utilization of the server is perpetually high, whereas the average CPU utilization is perpetually low. Servers with greater than 95% memory utilization are incapable of creating any new containers owing to insufficient memory, causing CPU resources to be wasted because of an unreasonable resource allocation.

IV. AN ADAPTIVE PREDICTION ALGORITHM FOR CLOUD RESOURCE DEMANDS

Different types of resource demands describe distinct distributions and exhibit varying characteristics, which can cause imbalances in the resource utilization. The variance of the data corresponding to different types of containers indicates that a single prediction method may not be suitable for different types of resource demand sequences. In our previous studies, we proposed two effective short-term prediction methods, EEMD-ARIMA and EEMD-RT-ARIMA. The EEMD-ARIMA method utilizes the EEMD method to decompose a non-stationary time series into stationary components, and then employs the ARIMA model to implement the predictions for these components, and finally sums all the prediction results of these components to obtain the final prediction result. However, we concluded that the EEMD-ARIMA method improves the prediction accuracy but significantly increases the prediction time. Consequently, we proposed another short-term prediction method called EEMD-RT-ARIMA. The EEMD-RT-ARIMA method reduces the prediction time and improves the prediction accuracy through the selection and reconstruction of efficient components. We also found that EEMD-ARIMA and
EEMD-RT-ARIMA exhibit different levels of effectiveness for different sequences under different situations.

This paper proposes an adaptive short-term prediction algorithm (ASTPA) based on the EEMD-ARIMA and EEMD-RT-ARIMA methods. The implementation process of this algorithm is shown schematically in Figure 9.

**A. DATA PREPROCESSING**

An effective feature selection method is critical to the preprocessing of a dataset [51]. A dataset of historical resource demands includes a large number of containers of various types. The prediction time increases if all resource demands are to be predicted, which can lead to a poor performance of the prediction algorithm. Therefore, it is important to extract the primary types of containers based on which the prediction should be implemented. Additionally, there may be missing data points and outliers in the resource demand sequence. To ensure an optimal prediction, such abnormal data must be processed.

We first divide the total sequence of container requests into multiple component sequences based on the types of containers involved, and then select the container requests of the primary types whose total quantity exceeds a predefined threshold. Consider sequentially arranged container requests \( C = < c_1, \ldots, c_i, \ldots, c_k > \), where \( k \) denotes the number of sampling points and \( c_i \) denotes the number of container requests at the \( i \)-th sampling point. This sequence includes \( n \) types of containers.

Each component sequence \( C^l = < c^l_1, \ldots, c^l_i, \ldots, c^l_k > \), comprising only the \( l \)-th type of containers, can be extracted from this sequence, where \( c^l_i \in c_i \). A new sequence \( C^{l+m} = < c^{l+m}_1, \ldots, c^{l+m}_i, \ldots, c^{l+m}_k > \) can then be computed by summing at least \( m \) component sequences, where the element \( c^{l+m}_i \) is the number of container requests at the \( i \)-th sampling point, and the ratio \( c^{l+m}_i / c_i \) exceeds the predefined threshold \( T_p \), that is, \( \forall c^{l+m}_i, c^{l+m}_i / c_i > T_p \). Thus, \( m \) component sequences are extracted as the principal types of container requests to be considered during the prediction of resource demands.

For example, considering a corresponding original sequence of container requests \( < 25, 37, 48, 56, 45 > \), including four types of containers, four component sequences \( < 1, 3, 5, 4, 3 >, < 12, 19, 23, 20, 27 >, < 2, 2, 4, 4, 3 >, \) and \( < 10, 13, 18, 28, 12 > \) are extracted. We can obtain a new sequence \( < 22, 32, 41, 48, 39 > \) by summing up at least two sequences \( < 12, 19, 23, 20, 27 > \) and \( < 10, 13, 18, 28, 12 > \), where the ratio of each element in the new sequence to the corresponding element in the original sequence exceeds the predefined threshold (0.85). Thus, in this example, these two sequences are regarded as the primary component sequences corresponding to the original sequence.

We use the quartile method to check the outliers. All data are first arranged in ascending order. The number corresponding to the first quarter position is called the first quartile \( Q_1 \), and the number corresponding to the three-fourths position is called the third quartile \( Q_3 \). The number corresponding to the middle position is called the second quartile, and is also the median value, denoted by \( Q_2 \). For a sequence \( < x_1, \ldots, x_n > \), the parameters \( Q_1, Q_2, \) and \( Q_3 \) can be calculated as follows:

\[
Q_1 = \frac{(x_k + x_{k+1})}{2} \quad n = 4k \quad \text{or} \quad n = 4k + 1
\]

\[
Q_2 = \frac{x_{k+1}}{2} \quad n = 2k + 1
\]

\[
Q_3 = \frac{(x_{3k} + x_{3k+1})}{2} \quad n = 4k
\]

In addition, the InterQuartile Range (IQR) is calculated as follows:

\[
IQR = Q_3 - Q_1
\]

Outliers refer to those values that exceed \( Q_3 \) by more than 1.5-times \( IQR \) or fall short of \( Q_1 \) by more than 1.5-times \( IQR \). Figure 10 shows the process of outlier detection using the quartile method for a sequence with 80 sampling points.

We compute \( Q_1 = 228.5, Q_3 = 746.5, \) and \( IQR = 518 \). The value 1.817 with the red “′′” is an outlier that exceeds \( Q_3 \) by more than 1.5-times \( IQR \).

Following this, we use the cubic spline interpolation method to replace the outliers in the primary component sequences. The cubic spline interpolation method applies \( G_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i \) to fit the given points, which can satisfy the following four equations for a continuous and smooth fitting.

\[
G_i(x_i) = y_i, \quad G_i(x_{i-1}) = y_{i-1}, \quad G'_i(x_i) = G'_{i+1}(x_i), \quad G''_i(x_i) = G''_{i+1}(x_i)
\]

We use RT to determine the fluctuation of the data. RT is a method for determining the randomness of a sequence based
Outlier detection using quartile method.

In the previous predictions, and then select a highly accurate prediction method with a higher prediction accuracy is appropriate threshold for the RT value based upon which the different RT values. Therefore, we propose an adaptive selection strategy to dynamically select the best prediction method based on the RT values. The crucial step is to ascertain an appropriate threshold for the RT value based upon which the better prediction method with a higher prediction accuracy is selected. The predefined threshold of RT values is unsuitable for certain sequences because they have different fluctuations under different scenarios. The main idea of this strategy is to compare the RT value of a sequence with the closest RT value in the previous predictions, and then select a highly accurate prediction method based on the comparison and execute an error adjustment if the prediction error is significant in the next prediction. This process is described in detail below.

1) If the first sequence is to be predicted, both the EEMD-ARIMA and EEMD-RT-ARIMA methods are used to implement the prediction. After a certain amount of time, the actual resource demands can be collected by a monitoring system. Thus, the mean absolute percentage errors (MAPEs) of these methods can be calculated for this sequence. The prediction method corresponding to the lowest MAPE is selected and saved in the database along with the RT value of this sequence. The sequence corresponding to the next observation period is then predicted based on historical data.

2) If the sequence to be predicted is not the first, the RT value of the sequence is calculated and compared with the closest RT value corresponding to a past predicted sequence. If it is less than the closest value obtained through the EEMD-RT-ARIMA method, then the EEMD-RT-ARIMA method should be used for the next prediction. If it is larger than the value obtained through the EEMD-ARIMA method, then the EEMD-ARIMA method is used for the next prediction.

3) If the RT value of a sequence is less than that obtained through the EEMD-ARIMA method but larger than that obtained through the EEMD-RT-ARIMA method, the better prediction method cannot be selected intuitively based on the RT value of the sequence. In such a case, the sequence is regarded as an initial prediction and steps 1 and 2 are executed.

An example of the adaptive selection process is as follows. We first count the number of container requests from a trace of the monitoring data over intervals of 5 minutes in a cloud platform. We then use the EEMD-ARIMA and EEMD-RT-ARIMA methods to predict 5 future values based on the last 80 historical data points. After some time has passed, the actual values can be accessed and the MAPEs corresponding to these methods can be calculated. Further, the RT value of this sequence (e.g., 30) and the selected method (e.g., the EEMD-ARIMA method) are saved in the database. Let us suppose that the RT value of the second sequence corresponding to the next observation period is calculated as 32. Because this is larger than the closest value (i.e., 30), its RT value is saved to the database and the EEMD-ARIMA method is selected for the next prediction. Similarly, we compute the RT value of the next sequence as 28. Although this is less than the closest RT value corresponding to the past prediction (i.e., 30), it corresponds to step 3 of the aforementioned process. Thus, we should regard this sequence as the first to be predicted and use both the EEMD-ARIMA and EEMD-RT-ARIMA methods for the prediction, selecting the method and prediction results corresponding to a lower MAPE. If the EEMD-RT-ARIMA method is selected at this stage, and the RT value of the next sequence is calculated as 25, then the EEMD-RT-ARIMA method is selected again for the next prediction because 25 is less than the closest previous RT
value (28), corresponding to step 2 of the aforementioned process. It should also be noted that this technique is only suitable for sequences that are under the same scenario or have similar characteristics, such as those from a continuous trace or those corresponding to the same time interval.

C. AN ERROR ADJUSTMENT
MAPE is used to evaluate the effectiveness of the prediction algorithms, which is inversely proportional to the prediction accuracy. If MAPE is less than 10%, the prediction is highly accurate. If MAPE is within the range of 10%–20%, the prediction accuracy is considered to be good. If MAPE is within the range of 20%–50%, then the prediction accuracy is considered satisfactory. If MAPE is greater than 50%, the prediction accuracy is considered unsatisfactory. We define an error parameter (EP) $e_i$, in a manner similar to MAPE, where $e_i$ denotes the mean error of the $m$-step prediction results of the $t$-th sequence using the EEMD-ARIMA or EEMD-RT-ARIMA method.

$$e_i = \frac{1}{m} \sum_{i=1}^{m} \frac{y_i^t - \hat{y}_i^t}{y_i^t}$$

where $\hat{y}_i^t$ and $y_i^t$ represent the predicted and actual values of the $i$-th prediction point of the $t$-th sequence, respectively. The value of EP $e_i$ may be positive or negative. A positive EP value indicates that the predicted values are higher than the actual values, and a negative value indicates the opposite.

A good prediction method should have a low and stationary prediction error. However, the prediction error changes rapidly corresponding to sudden and rapid variations in resource demands, which significantly impacts the effectiveness of the prediction. An error adjustment can be utilized to alleviate this problem. An absolute error adjustment may be not suitable for cases with different magnitudes of resource demands. If the MAPE of a sequence is higher than a predefined threshold (e.g., 50%), the error in the prediction of the next sequence is generally high. Thus, the predicted values should be compensated using a dynamic error adjustment factor, which is defined as follows:

$$e_i^{rt} = \frac{e_{i-1} + e_{i-2}}{2}$$

Therefore, the final prediction result $\hat{y}_{t-1}^{i-f}$ is calculated through the following formula, where $\hat{y}_t^i$ is the $i$-th prediction result of the $t$-th sequence obtained via the method selected based on the adaptive selection strategy.

$$\hat{y}_{t}^{i-f} = \frac{\hat{y}_t^i}{1 + e_t^i}$$

As illustrated in Algorithm ASTPA, a sequence $C$ is the input of the prediction. A component sequence $C^l$ is obtained by extracting the primary components of the sequence $C$ (line 1). A preprocessed sequence $S^l$ is obtained by detecting the outliers of the component sequence $C^l$ and replacing them (line 2). Then, each preprocessed sequence $S^l$ is transformed into the sequence $H^l$ based on two values, and its RT value is computed by executing runs test (lines 3-5). Finally, the adaptive selection strategy is used to implement the RT-based adaptive prediction for each preprocessed sequence $S^l$ (lines 6–23), and the error adjustment is executed to obtain a final prediction result (lines 24–28).

V. EXPERIMENTS AND ANALYSIS
To accurately verify the effectiveness of the proposed ASTPA, we use the cluster data obtained from the Alibaba cloud [50] to conduct a few experiments. The statistical data used in the experiments are shown in the figures in Section II and they are also recorded in the Figsheare database [52]. First, we select four sequences, S1–S4, from Figures 1–4, and they are also recorded in the Figsheare database [52].
experiments on a notebook computer (8 G memory, Intel Core i7 6500U CPU). First, we verify the effectiveness of the principal component analysis, outlier processing, adaptive selection strategy, and error adjustment in the proposed ASTPA for improving the prediction accuracy. We then compare and analyze the performance of our algorithm with that of the other algorithms in terms of the effectiveness and time cost.

A. EVALUATION OF PREDICTION EFFECTIVENESS OF ASTPA

To explain the impact of the principal component analysis on the prediction results, we selected one of the four sequences depicted in Figure 11, which is composed of a non-zero quantity of containers. The sequence S1 includes 90 sampling points and 28 types of container requests. We count 28 types of container sequences and select those sequences with larger numbers of containers. We extract the component sequence S1-type1 with 4-core CPU and a 1.56 memory size (cpu = 400, mem = 1.56) and the component sequence S1-type2 with 8-core CPU and a 3.13 memory size (cpu = 800, mem = 3.13) from the original sequence S1. We sum these two sequences to form a new sequence S and calculate the sequence ratio S/S1. Each element in the sequence ratio S/S1, that is, the ratio of the number of container requests in the new sequence to the total number in the original sequence, exceeds the threshold of 85% at each sampling point, and thus the two component sequences S1-type1 and S1-type2 are extracted as the primary component sequences. It should be noted that the number of CPU cores and the memory size are normalized. It can also be observed that changes in the component sequence S1-type1 are almost identical to those in the original sequence S1, whereas the component sequence S1-type2 exhibits some differences in the detailed fluctuations despite its overall trend being the same as that of S1. Undoubtedly, only the primary component sequences may cause an inaccurate container demand. However, these two sequences not only represent the largest number of container requests but also reflect the characteristics of the original sequence. Further, the principal component analysis greatly reduces the prediction cost compared to running the prediction process for all 28 types of distinct container sequence types.

To verify the effectiveness of outlier processing in improving the prediction accuracy, we use the quartile method to detect the outliers in the three sequences depicted in the boxplot of Figure 12. The values exceeding $Q_3$ by more than 1.5-times $IQR$ are outliers. Subsequently, new data are generated to replace these outliers using the cubic spline interpolation method. The preprocessed sequences are depicted in Figure 13, where the data marked by the red circle have been preprocessed.
TABLE 3. MAPE values of ASTPA for different numbers of prediction points.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>RT value</th>
<th>EEMD-ARIMA(5)</th>
<th>EEMD-RT-ARIMA(5)</th>
<th>EEMD-ARIMA(10)</th>
<th>EEMD-RT-ARIMA(10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>preprocessed-S1</td>
<td>32</td>
<td>13.46%</td>
<td>17.74%</td>
<td>30.23%</td>
<td>26.29%</td>
</tr>
<tr>
<td>preprocessed-S1-type1</td>
<td>32</td>
<td>11.88%</td>
<td>16.77%</td>
<td>27.73%</td>
<td>23.64%</td>
</tr>
<tr>
<td>preprocessed-S1-type2</td>
<td>26</td>
<td>28.64%</td>
<td>28.21%</td>
<td>58.57%</td>
<td>56.40%</td>
</tr>
<tr>
<td>S2</td>
<td>25</td>
<td>18.03%</td>
<td>15.02%</td>
<td>66.49%</td>
<td>47.36%</td>
</tr>
<tr>
<td>preprocessed-S2-type1</td>
<td>23</td>
<td>15.38%</td>
<td>13.83%</td>
<td>63.71%</td>
<td>60.33%</td>
</tr>
<tr>
<td>preprocessed-S2-type2</td>
<td>23</td>
<td>33.27%</td>
<td>33.12%</td>
<td>55.14%</td>
<td>52.34%</td>
</tr>
<tr>
<td>preprocessed-S3</td>
<td>21</td>
<td>7.31%</td>
<td>10.81%</td>
<td>26.98%</td>
<td>28.61%</td>
</tr>
<tr>
<td>preprocessed-S3-type1</td>
<td>23</td>
<td>17.76%</td>
<td>18.64%</td>
<td>36.06%</td>
<td>40.36%</td>
</tr>
<tr>
<td>preprocessed-S3-type2</td>
<td>19</td>
<td>73.95%</td>
<td>88.89%</td>
<td>53.41%</td>
<td>62.26%</td>
</tr>
<tr>
<td>preprocessed-S4</td>
<td>19</td>
<td>11.75%</td>
<td>9.87%</td>
<td>29.23%</td>
<td>29.62%</td>
</tr>
<tr>
<td>preprocessed-S4-type1</td>
<td>21</td>
<td>11.28%</td>
<td>33.29%</td>
<td>38.31%</td>
<td>52.09%</td>
</tr>
<tr>
<td>preprocessed-S4-type2</td>
<td>19</td>
<td>58.93%</td>
<td>82.28%</td>
<td>42.08%</td>
<td>59.09%</td>
</tr>
</tbody>
</table>

Subsequently, we use the EEMD-ARIMA and EEMD-RT-ARIMA methods to implement the predictions. For each sequence, we set the first 80 data points as the training set and the next 5 data points as the testing set. We compare the MAPEs of the prediction results of these preprocessed sequences with those of the original unprocessed sequences. The experimental results are listed in Table 2. The EEMD-ARIMA(5) column lists the MAPEs obtained via the EEMD-ARIMA prediction method corresponding to the five future points. The prediction results of the preprocessed sequences basically precede those of the original sequences, most of which are highly sensitive to reducing the prediction error. For instance, the MAPE of the sequence S1 is 20.40% using the EEMD-ARIMA method, whereas that of the preprocessed-S1 sequence is 13.46%, showing a reduction of 6.94%. The MAPEs of the preprocessed-S3 sequence are 7.31% and 10.81% using the EEMD-ARIMA and EEMD-RT-ARIMA methods, respectively, which significantly reduces the prediction errors compared to MAPEs of 96.49% and 115.59% corresponding to the unprocessed data. The MAPE of the sequence S4 is 76.19% using the EEMD-RT-ARIMA method, whereas that of the preprocessed-S4 is only 9.87%, showing a significant improvement in the prediction accuracy. We further analyzed the dataset and found that the outliers in the S3 and S4 sequences are the last values of each, and the prediction accuracy is significantly improved by preprocessing them.

To verify the effectiveness of the adaptive selection strategy used by the proposed ASTPA, we applied preprocessed sequences to conduct some experiments. For each sequence, the first 80 data points were set as the training set, and the next 5 and 10 data points were set as the testing set. The experimental results are listed in Table 3, where bold and underlined values represent the MAPE of a sequence obtained by adopting ASTPA. The S2 sequence had no outliers. It is clear that the MAPEs corresponding to ASTPA are much higher for the 10-step prediction than those for the 5-step prediction. For instance, the MAPE of the preprocessed-S4 sequence is 29.23% for the 10-step prediction, but only 9.87% for the 5-step prediction, when using the same ASTPA approach. Even for the preprocessed-S2-type1 sequence, the difference between the MAPEs corresponding to the 5- and 10-step predictions is 46.5%. Based on the range of MAPE, we can conclude that 83% of the predictions are satisfactory and only 17% of the predictions are unsatisfactory for the 5-step prediction. However, there are no good predictions for the 10-step prediction, which indicates that ASTPA is not applicable for a long-term prediction based on a small amount of data.

It is clear that the adaptive prediction strategy applies a better prediction method at every stage, thereby achieving a higher prediction accuracy for sequences with similar
FIGURE 14. Number of predicted containers for S1 sequence over intervals of 2 minutes.

FIGURE 15. Number of predicted containers for S2 sequence over intervals of 2 minutes.

FIGURE 16. Number of predicted containers for S3 sequence over intervals of 5 minutes.

characteristics. Further, this generally improves the prediction accuracy for most sequences. For example, the preprocessed-S1 and its component sequences are derived from the same dataset over intervals of 2 minutes and exhibit similar characteristics, as shown in Figure 12. The proposed ASTPA first selects the EEMD-ARIMA method to obtain MAPE values of 13.46% and 11.88% corresponding to the 5-step prediction when the RT values of both the preprocessed-S1 and preprocessed-S1-type1 sequences are equal to 32. Their MAPEs are lower than those obtained via the EEMD-RT-ARIMA method. The EEMD-RT-ARIMA method is selected to implement the prediction owing to its lower MAPE than that of the EEMD-ARIMA method when the RT value of the preprocessed-S1-type2 sequence is equal to 26. Because the RT value of the preprocessed-S2 sequence is lower than that of the preprocessed-S1-type2 sequence, the ASTPA algorithm proceeds by selecting the EEMD-RT-ARIMA method for implementing the 5-step prediction, and obtains a MAPE of 15.02%, which is lower than the 18.03% achieved using the EEMD-ARIMA method. The EEMD-RT-ARIMA method is also applied to predict the preprocessed-S2-type1 and preprocessed-S2-type2 sequences owing to their lower RT values. Similarly, the ASTPA prediction algorithm uses the EEMD-ARIMA method to obtain lower MAPEs of 7.31% and 11.28% compared to the 10.81% and 33.29% values obtained through the EEMD-RT-ARIMA method corresponding to the 5-step prediction of the preprocessed-S3 and preprocessed-S4-type1 sequences with an RT value of 21. The preprocessed-S3-type1 sequence continuously uses the EEMD-ARIMA method to achieve a MAPE of 17.76%, which is lower than that achieved using the EEMD-RT-ARIMA method following the adaptive selection strategy of ASTPA when it has a larger RT value than those of the preprocessed-S3 and preprocessed-S4-type1 sequences. This situation continues until the preprocessed-S4 sequence achieves the smallest RT value of 19. The EEMD-RT-ARIMA method is selected to predict the preprocessed-S4 sequence, and achieves the lowest MAPE of only 9.87%. The EEMD-ARIMA method is selected to achieve a MAPE of 11.28% for the preprocessed-S4-type1 sequence, which greatly improves the prediction accuracy compared to a MAPE of 33.29% obtained through the EEMD-RT-ARIMA method. In the case of a 10-step prediction, the EEMD-RT-ARIMA method is selected for the preprocessed-S1, S2, and component sequences, whereas the EEMD-ARIMA method is selected for the preprocessed-S3, preprocessed-S4, and component sequences. These also follow the adaptive selection strategy. Therefore, ASTPA can rapidly select a better method at every stage to obtain the most accurate prediction results.

Figures 14–17 show a comparison between the predicted values and the real data corresponding to the 5-point prediction of the sequences S1–S4. The trend of the predicted data is almost identical to that of the real data.

To analyze the impact of the combination of data preprocessing and an adaptive selection strategy on the prediction accuracy, we used a partial trace to implement the experiments in Figure 1, set the sliding window size to 200 points, and shifted 5 points to the right each time. It should be noted...
that trace5 is identical to trace6 with regard to verifying the prediction stability of ASTPA. Table 4 lists the prediction results obtained through ASTPA without a prediction adjustment, where the bold and underlined values indicate the MAPE of a sequence obtained through ASTPA without a prediction adjustment. It is clear that ASTPA selects the method with a lower prediction error according to its adaptive selection strategy when operating without an error adjustment, and that these methods process the non-stationary sequence to achieve lower prediction errors compared to those of the EEMD-ARIMA and EEMD-RT-ARIMA methods for the original sequence. For instance, ASTPA without an error adjustment achieves a MAPE of 37.47% for the preprocessed trace2 sequence with an RT value of 50, which is lower than the 41.92% and 49.67% values obtained using the EEMD-ARIMA and EEMD-RT-ARIMA methods for the original sequence. ASTPA selects the EEMD-RT-ARIMA method and achieves a MAPE of 26.86% when the RT value of the trace3 sequence changes to 49. The EEMD-RT-ARIMA method is still selected to implement the prediction and achieves a MAPE of 27.59% when the RT value of the trace4 sequence is lower than that of the trace3 sequence. The trace5 and trace6 sequences proceed in a similar manner.

We compute the EP of each sequence using Formula (5), the results of which are listed in Table 5. We can see that the EP values of trace5-trace7 are higher than those of other sequences, which agrees with their MAPE values. The probability of a large prediction error on their next predictions is high. Hence, an error adjustment is executed on these predictions.

To verify the effectiveness of the error adjustment, we compared the results obtained through ASTPA with and without an error adjustment in Figure 18. The MAPE of the trace5 sequence is 76.19%, which implies that the trace6 sequence may also have a high prediction error. Therefore, we use ASTPA to execute an error adjustment and achieve a lower MAPE for the prediction of the trace6 sequence. The MAPE of the trace6 sequence decreases by 22.36% when using ASTPA with an error adjustment as compared to the case of using ASTPA without such an adjustment. Similarly, the MAPE of the trace7 sequence also decreases through an error adjustment. However, the MAPE of the trace8 sequence increases owing to the significant change in EP from the positive value of the trace7 sequence to the negative value of the trace8 sequence, as shown in Table 5. When the last EP is large, it indicates the presence of a large variation. In such cases, an error adjustment is recommended for the next prediction.

B. COMPARISON AND ANALYSIS OF DIFFERENT ALGORITHMS

To demonstrate the effectiveness and time cost, we further compared our proposed algorithm ASPTA with ARIMA, BPNN, and a combined ARIMA and BPNN (ARIMA-BPNN) based algorithm [40]. The ARIMA-BPNN algorithm uses the ARIMA model to predict the original sequence and obtain a residue sequence. A BPNN is used to predict the residue sequence. Finally, the two prediction results are superposed to obtain the overall prediction. In the experiments on the algorithms involving a BPNN, a network is created with one hidden layer composed of five neurons, and

---

**FIGURE 17.** Number of predicted containers for S4 sequence over intervals of 5 minutes.

**TABLE 4. MAPE values of different algorithms.**

<table>
<thead>
<tr>
<th>RT value</th>
<th>ASTPA without error adjustment</th>
<th>EEMD-ARIMA for original sequence</th>
<th>EEMD-RT-ARIMA for original sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>trace1</td>
<td>50</td>
<td>29.00%</td>
<td>30.58%</td>
</tr>
<tr>
<td>trace2</td>
<td>50</td>
<td>37.47%</td>
<td>46.07%</td>
</tr>
<tr>
<td>trace3</td>
<td>49</td>
<td>33.85%</td>
<td><strong>26.86%</strong></td>
</tr>
<tr>
<td>trace4</td>
<td>48</td>
<td>27.71%</td>
<td><strong>27.59%</strong></td>
</tr>
<tr>
<td>trace5</td>
<td>47</td>
<td>97.40%</td>
<td><strong>76.19%</strong></td>
</tr>
<tr>
<td>trace6</td>
<td>47</td>
<td>88.68%</td>
<td><strong>70.09%</strong></td>
</tr>
<tr>
<td>trace7</td>
<td>51</td>
<td><strong>124.41%</strong></td>
<td>159.96%</td>
</tr>
<tr>
<td>trace8</td>
<td>50</td>
<td><strong>13.32%</strong></td>
<td>16.90%</td>
</tr>
<tr>
<td>trace9</td>
<td>50</td>
<td>25.55%</td>
<td><strong>25.61%</strong></td>
</tr>
<tr>
<td>trace10</td>
<td>50</td>
<td><strong>22.24%</strong></td>
<td>33.26%</td>
</tr>
</tbody>
</table>

**TABLE 5. Mean error of sequences.**

<table>
<thead>
<tr>
<th>Sequence</th>
<th>$e_r$</th>
<th>Sequence</th>
<th>$e_r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>trace1</td>
<td>0.0218</td>
<td>trace6</td>
<td>0.4706</td>
</tr>
<tr>
<td>trace2</td>
<td>-0.1026</td>
<td>trace7</td>
<td>1.1891</td>
</tr>
<tr>
<td>trace3</td>
<td>-0.1253</td>
<td>trace8</td>
<td>-0.0716</td>
</tr>
<tr>
<td>trace4</td>
<td>0.2672</td>
<td>trace9</td>
<td>-0.0777</td>
</tr>
<tr>
<td>trace5</td>
<td>0.5220</td>
<td>trace10</td>
<td>-0.1162</td>
</tr>
</tbody>
</table>
the maximum number of epochs to train is set to 10,000. Figures 19 and 20 show the MAPEs and time costs for these algorithms, respectively. It is clear that the proposed ASPTA algorithm achieves lower MAPEs and is more effective than the other algorithms for all test sequences except trace5, trace6, and trace8, as shown in Figure 19. Corresponding to trace6 and trace8, it also achieves a lower MAPE than the BPNN algorithm. However, it achieves the highest MAPE for trace5. In general, the ASPTA algorithm demonstrates a better prediction effectiveness for these sequences. The ARIMA-BPNN algorithm takes second place, followed by the ARIMA model. The BPNN is shown to be the least effective owing to the use of too few data samples for the neural network. Its MAPEs take values of 62.02%, 66.81%, and 60.44% for trace2, trace6, and trace8, respectively.

However, the BPNN algorithm is the most time-effective, with a run time of less than 2 s. The runtime of the ARIMA-BPNN algorithm is greater than that of the ARIMA model owing to the time taken for the additional BPNN prediction of a residue sequence. The proposed ASPTA algorithm runs for the longest time primarily because of the separate extractions of multiple primary component sequences from the original sequence used to implement the EEMD-ARIMA or EEMD-RT-ARIMA prediction, in which each primary component sequence is decomposed into a few sub-sequences, and where the ARIMA prediction is executed on each sub-sequence. However, its time cost is shown to be acceptable when considering the single-tread simulation program and the performance of the notebook computer used during the experiments. The time cost can be greatly reduced when the simulation program is executed on a physical server or a container with tens of CPU cores.

The prediction accuracy of ASTPA depends primarily on the effectiveness of the EEMD-ARIMA and EEMD-RT-ARIMA methods. If these two methods are unsuitable for the resource demand prediction under certain scenarios, the proposed algorithm will no longer be effective. Therefore, we further studied the effectiveness for data at different scales. A 100-h trace of the container numbers, as shown in Figure 1, was divided into 15 consecutive sequences to verify the effectiveness. Each sequence includes 200 sampling points, where each point represents a 2-minute period. We then predicted the values of the next five points. The experimental results are listed in Table 6, where SD and N_o denote the standard deviation and number of outliers, respectively.

Obviously, both the EEMD-ARIMA and EEMD-RT-ARIMA methods are ineffective in predicting these sequences because they achieve a high MAPE of over 75% for most sequences. It can be concluded that the prediction results may be affected by the RT value, standard variance, and number of outliers. When the other factors are negligible, the RT values are directly proportional to the MAPE. As demonstrated by the D6 sequence, the RT value of 79 causes a high MAPE of over 200% for both the EEMD-ARIMA and EEMD-RT-ARIMA methods. Similarly, the D10 sequence achieves a high MAPE of over 400% for both prediction methods because of its high RT value, which reflects strong fluctuations in amplitude. High values for any other factor can induce high MAPE values even if the RT value of a sequence is small, as indicated by D11 and D15. These sequences involve more non-linear features, whereas our algorithm is more effective for sequences with linear features. It can also be seen that ASTPA selects a poor prediction method to obtain a high MAPE owing to an overly large RT value of D9. In addition, the algorithm is ineffective.
for sequences with too many outliers because the detection and replacement of outliers not only leads to a high cost but also changes the features and trends of the sequences, such as in D13 and D14. Therefore, our proposed ASTPA is ineffective for sequences with extremely high RT values, standard deviations, and numbers of outliers because they may involve more non-linear features and exhibit stronger fluctuations.

VI. DISCUSSION

The proposed ASTPA combines a principal component analysis with the quartile method, RT, EEMD method, and ARIMA model to implement an integrated prediction process. These methods are simpler and easier to rapidly execute for a short-term prediction of sequences with a small amount of sampling data as compared to machine-learning- or neural-network-based prediction methods. ASTPA consists of three parts. First, a primary component sequence is extracted from a sequence with N historical data points. The time complexity of this process is \( O(N) \). Possible outliers in the component sequences are detected using the quartile method. The time complexity of the process is \( O(M \times N) \), where \( M \) is the number of primary component sequences. Finally, the RT values of the preprocessed sequences are computed, and then they are predicted using the EEMD-ARIMA and EEMD-RT-ARIMA methods. The time complexity of the EEMD-ARIMA method is \( O(2M \times N + q \times M \times N) \) because it incorporates the RT, EEMD-decomposition, and ARIMA prediction. The time complexity of the EEMD-RT-ARIMA method is \( O(2M \times N + p \times M \times N) \), where \( p \) is the number of newly reconstructed new sequences \( p < q \). Therefore, the time complexity of the proposed ASTPA is \( O(Q \times M \times N) \) or \( O(P \times M \times N) \).

ASTPA exhibits a high time cost, as depicted in Figure 20. In our previous work [38], we also compared the time costs of ARIMA, EEMD-ARIMA, and EEMD-RT-ARIMA. The EEMD-RT-ARIMA method uses the selection and reconstruction of efficient components to reduce its time cost and achieve a cost-effective trade-off. We intend to achieve such a cost-effective trade-off in the case of the proposed ASTPA in the future.

Our algorithm is also suitable for real-time situations. First, this algorithm uses EEMD-ARIMA and EEMD-RT-ARIMA prediction methods for the prediction process and selects the one with the lower MAPE. It then selects the better prediction method between them to predict the sequence corresponding to the next observation period through an RT-based and adaptive strategy. The algorithm can predict a continuous trace with a time variation by setting the same or different observation periods. It is also simple and easy to implement in real time. This work uses the cluster data from Alibaba cloud to conduct experiments, which has been applied in some other relevant studies [53]–[55]. The experiments were conducted on the number of demanded containers counted from this dataset, and the results demonstrate the effectiveness and time cost problem of our algorithm when evaluating and comparing its performance with that of other algorithms. The effectiveness of our proposed algorithm needs to be further evaluated in a real system. Furthermore, our algorithm focuses primarily on the prediction of the number of containers and does not consider the priority or order of the container requests. In a future study, we intend to consider these problems in a comprehensive manner.

VII. CONCLUSION

Cloud resource demands exhibit characteristics of diversity and uncertainty, which causes inaccuracies in the prediction of resource demands in the real world. This paper proposes an adaptive short-term prediction algorithm for cloud resource demands. This algorithm adopts a principal component analysis method to extract the sequences of the main container types, preprocesses the outliers of these sequences to improve the prediction accuracy, presents an adaptive selection strategy to select the higher-accuracy prediction method between EEMD-ARIMA and EEMD-RT-ARIMA methods at every stage, and defines an error adjustment factor to improve the prediction accuracy of the next sequence when the prediction accuracy of a sequence exceeds a predefined threshold. We implement four groups of experiments to verify and evaluate the effectiveness of our proposed algorithm. The experimental results demonstrate that the proposed algorithm can effectively improve the short-term prediction accuracy for cloud resource demands but has a high time cost that needs to be addressed in the future.


**JING CHEN** received the M.S. degree from the Beijing Institute of Technology. She is currently pursuing the Ph.D. degree with the Shandong University of Science and Technology. She is also an Associate Research Fellow with the Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences). Her research interests include cloud computing and software testing.

**YINGLONG WANG** received the M.S. degree in industrial automation and the Ph.D. degree in communication and information systems from Shandong University, Jinan, China, in 1990 and 2005, respectively. He is currently a Research Fellow with the Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences). His current research interests include wireless networks, information security, and cloud computing.

* * *