An Effective Classification-Based Framework for Predicting Cloud Capacity Demand in Cloud Services

Bin Xia, Tao Li, Qifeng Zhou, Qianmu Li, and Hong Zhang

Abstract—The rapid development of pay-as-you-go cloud services motivates the increasing number of cloud resource demands. However, the volatile demands bring new challenges for current techniques to minimize the cost of cloud capacity planning and VM provisioning while satisfying the customer demands. The service vendors will incur enormous revenue loss within the long-term inappropriate planning, especially when the demands fluctuate abruptly and frequently. In this paper, we cast the cloud capacity planning as a classification problem and propose an integrated framework, which effectively predicts the abrupt changing demands, to reduce the cost of cloud resource provisioning. In this framework, we first apply Piecewise Linear Representation to segment the time series of cloud resource demands for labeling the changing trend of each period. Second, Weighted SVM is leveraged to fit the statistical information and the label of each period and predict the changing trend of the following period. Finally, an incremental learning strategy is utilized to ensure the low cost of updating the model using the upcoming requests. We evaluate our framework on the IBM Smart Cloud Enterprise (SCE) trace data and the experimental results show the effectiveness of our proposed framework.

Index Terms—Cloud computing, capacity planning, piecewise linear representation, support vector machine, incremental learning

1 Introduction

Cloud service has become an increasingly ubiquitous infrastructure for many IT supports of current business campaigns due to its lightweight, convenience, and high efficiency. It can flexibly provide various resources as services, such as SaaS (Software as a Service), PaaS (Platform as a Service), and IaaS (Infrastructure as a Service), over the Internet. Among these cloud services, IaaS aims to provide an integrated and elastic infrastructure that is capable of dynamically providing VM (virtual machine) resources for the IT solutions in business activities. In order to meet the various demands of customers, the *cloud capacity planing* and the *instant provisioning of VMs* are the major strategies to effectively arrange and assign the cloud resource in the paradigms such as IaaS [1].

Due to the flexible pay-as-you-go charging style of current cloud services and the increasing amount of demands, the amount of resource demands in IaaS gradually becomes more unstable than previous IT services. It brings big challenges for the *cloud capacity planing* and the *instant provisioning of VMs* to balance the trade-off between the customer satisfaction and the cloud resource provisioning costs:

- *Cloud Capacity Planing* is a straightforward strategy that optimizes the assignment of cloud resource capacity based on the customer demands. However, the elastic and volatile demands bring more difficulties in planning the cloud capacity appropriately. On the one hand, if the customer demands are underestimated, the overflowed requests cannot be satisfied instantly and the shortage cloud capacity will potentially cause the customer and revenue loss. On the other hand, if the demands are overestimated, the prepared cloud resources cannot be fully leveraged. The early purchase of infrastructures and expensive maintenance cost will cause severe stresses on the operation of data centers. A data center needs to pay about millions of dollars for the installation, maintenance, and energy costs in a large amount of cloud infrastructures (e.g., 1000 server racks) every year [2].
- *Instant Provisioning of VMs* aims to efficiently provide specific types of VMs according to customer requests. Similar to the *cloud capacity planning*, the underestimation and overestimation of demands for specific VMs will cause the customer loss and the cloud resource waste, respectively. Although an increasing number of state-of-the-art VMs provisioning strategies [3], [4]

- *B. Xia is with Jiangsu Key Laboratory of Big Data Security & Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210046, P.R. China; and School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, Jiangsu 210094, P.R. China. E-mail: ben.binxia@gmail.com.*
- *T. Li, deceased, was with School of Computing and Information Sciences, Florida International University, Miami, FL 33199 USA; and School of Computer Science & Technology, Nanjing University of Posts and Telecommunications (NJUPT), Nanjing, Jiangsu 210046, P.R. China.*
- *Q. Zhou is with Automation Department, Xiamen University, Xiamen, Fujian 361005, China. E-mail: zhouqf@xmu.edu.cn.*
- *Q. Li and H. Zhang are with School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, Jiangsu 210094, P.R. China. E-mail: liqianmu@126.com, ben.binxia@gmail.com.*

Manuscript received 2 Nov. 2016; revised 19 Nov. 2017; accepted 6 Feb. 2018. Date of publication 12 Feb. 2018; date of current version 5 Aug. 2021. (Corresponding author: Bin Xia.)
Digital Object Identifier no. 10.1109/TSC.2018.2804916

support preparing a specific virtual machine in minutes, customers still need to wait for the startup of services. For those instant demands, the VMs clone techniques [5], [6] become an appropriate choice that prepares and provides VMs in seconds. However, it is difficult to reduce the time consumption of basic procedures such as VM verification and automatic patching, only if the schedule of customer demands is available [1].

The small difference between the actual demands and the estimation may be affordable, however, the long-term estimations which have large differences from actual demands will cause great financial losses. Therefore, it is imperative to improve the efficacy and efficiency of the cloud resource planning to minimize the cloud resource provisioning costs while satisfying the volatile business requirements.

To address the problems of current limitations in techniques, the effective solution is to predict the cloud capacity demands and provide the appropriate amount of VMs according to the prediction. However, the volatile demands, especially the abrupt changes, bring new challenges where the dependency between the resource demand and its relating factors is highly nonlinear and always changes over time. In our previous works, we transform the prediction to a regression problem and apply the state-of-the-art time series strategies to predict the cloud resource demands [1], [7], [8], [9]. Although the regression is capable of presenting the prediction using specific values which are straightforward for the cloud configuration, the weak prediction of abrupt changing trend (i.e., significant error in the estimation) shows the shortage of this strategy. In addition, the costs of overestimation and underestimation in the abrupt increasing and decreasing demands are totally different. At any time, the increasing trend implies more customer demands than the decreasing one. If the demands are underestimated in the increasing trend, more customer demands cannot be satisfied instantly. To ensure the revenue maximization, the estimation of the increasing trend should be paid more attention. However, existing regression-based strategies ignore this and treat different changing trends all the same.

1.1 Challenges and Proposed Solutions

To bridge the gap, it is imperative to construct an effective model to accurately predict the prospective cloud resource demands. In practical applications, VM service providers always prepare VM resources based on the degree of demands (e.g., on-peak and off-peak) where the prediction of demand degree can be considered as a classification problem. The advantage of regression-based approaches is they can predict the customer demands using a specific scalar. However, the regression-based approaches predict the cloud resource demands using successive values while ignoring the accuracy of identifying the abrupt changing demands during specific periods [1]. Even if the accurate prediction of demands is available, providers should prepare more VMs than the predicted customer demands. This is because the customer loss caused by the under-estimation is more serious than the waste of resources due to the over-estimation. Moreover, the performance of regression-based methods is not good enough to help providers make decisions. To this end, we aim to build a classification-based strategy to model this problem to improve the performance of identifying those abrupt changing demands, while providing the appropriate recommendations of prospective cloud resource demands. However, there exist some challenges if we apply a classification-based model in this scenario.

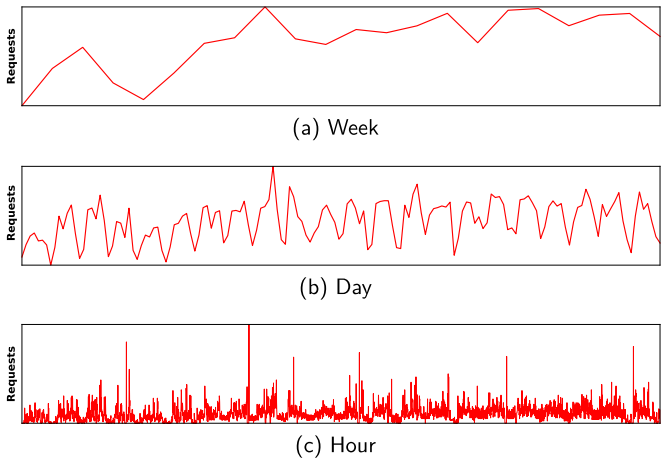


Fig. 1. Time series aggregation granularity selection.

Challenge 1. The original data of cloud resource demands is a time series without the label information (i.e., gradual, abrupt increasing, or abrupt decreasing).

The main idea of a classification-based method is to train the model using samples with corresponding features and label, then predict unknown samples based on their features leveraging the trained model. In the prediction of cloud resource demands, the periods (i.e., sections of time series with the same time scale) are considered as the samples. For each sample, the statistical information collected from the corresponding period are the extracted features, and the label is represented using the changing amount of demands compared to the previous period. The problem is how to select an effective strategy to label each period. To handle the problem in Challenge 1, we employ Piecewise Linear Representation (PLR) [10], which is a time series segmentation strategy, to automatically label the original data according to the predefined threshold. In terms of the predefined threshold, users can customize the degree of abrupt changing demands. More details can be found in Section 3.2.

Challenge 2. The cloud resource demands are extremely volatile, and the relationship between the resource demand and its influencing factors is highly nonlinear.

Actually, the volatile demand is an obstacle to constructing the model. Fig. 1 presents the statistics of cloud capacity provisioning in three different granularities of time series. The coarse granularity (i.e., week) describes the overall trend, the medium granularity (i.e., day) concentrates on the fluctuation in short term, and the fine granularity (i.e., hour) provides more details of the real-time change. As observed in Fig. 1, the time series of cloud resource demands is extremely volatile no matter in which granularity. To deal with Challenge 2, we extract some meaningful statistical information and leverage Weighted SVM (WSVM), a robust classifier, to construct the prediction model (see Section 4.1). As mentioned above, the cost of making wrong decisions during abrupt changing periods is much larger than that during gradual ones. Facing

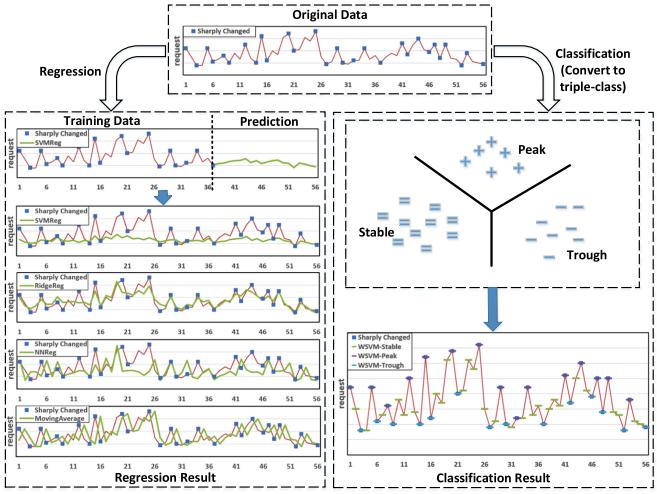


Fig. 2. The illustration of regression-based and classification-based strategies in cloud resource demand prediction. On the top of the figure is the original cloud capacity demands data. The left panel shows the predictions of cloud resource demands to employ several regression-based methods. The right panel presents the idea of our proposed PLR-WSVM that casts the problem as a three-class classification and shows the prediction applying the proposed method. The red line represents real resource demand time series, blue square points are sharply changed demand points, and the green line represents the fitting line using different regression methods.

the volatile demands, WSVM is capable of training the model using different weights of each sample where the abrupt changing periods are added extra weights. Applying this operation, WSVM is able to improve the performance of predicting the abrupt changing demands while maintaining the accuracy of predicting the gradual ones (see Section 3.3).

Challenge 3. Facing the constant demands and the requirement of real-time prediction in practical applications, how to reduce the time cost of training the model?

Compared to the effectiveness of the framework, the efficiency is equally important, especially in practice. Even if the samples are finite and the dimension of features is small, it will still spend the unnecessary cost in re-training the model under the requirement of real-time prediction. However, the relationship between the resource demand and its influencing factors changes over time, and there hardly exist a model that is able to well fit all the data in the future. Therefore, it is imperative to reduce the time consumption of training the model in practical applications. To handle Challenge 3, we propose an incremental learning strategy that reduces the frequency of re-training the model while maintaining the performance of prediction. The details are described in Section 3.3.2.

In this paper, we propose an integrated framework to address the aforementioned limitations of current techniques in predicting the cloud resource demands. As shown in Fig. 2, we cast the problem of predicting the prospective cloud resource demands as a weighted three-class classification where each period is categorized into 'stable'(i.e., gradual), 'peak'(i.e., abrupt increasing), or 'trough'(i.e., abrupt decreasing). In order to train the classification-based model, first, we apply Piecewise Linear Representation (PLR) to segment the changing time series of cloud resource demands and label each period as 'stable', 'peak', or 'trough'. Given the label and statistical information of each period, second, we model the prediction as a weighted three-class classification problem using Weighted SVM (WSVM) that adds the extra weights for the abrupt changing demands (i.e., abrupt increasing and decreasing). Finally, the constructed model will constantly predict the prospective cloud resource demands and be incrementally updated utilizing the upcoming requests. The major contributions of this paper are summarized as follows:

- We cast the prediction of cloud resource demands as a classification-based problem instead of the regression-based approach where the formal method can provide more accurate predictions, especially in the abrupt changing periods.
- We construct the classification-based model which automatically labels the changing time series of cloud resource demands using PLR and effectively predicts the prospective cloud capacity applying WSVM. In the training process of WSVM, we are more concerned with the abrupt increasing and decreasing demands and add the extra weights on these periods that improve the accuracy of prediction.
- We propose an incremental learning strategy for PLR-WSVM that ensures the robustness and effectiveness of our proposed framework in practical applications.

The rest of this paper is organized as follows: In Section 2, we discuss the related work. Section 3 presents the system framework and our proposed strategies in details, while Section 4 discusses the comparison between the regression-based method and our classification-based strategy, and evaluates the effectiveness of our framework. Finally, we summarize this paper in Section 5.

2 Related Work

With the rapid development of modern computing systems, the increasing scale and complexity bring more challenges to manually analyze the real-time behavior of the system. In order to improve the efficiency and efficacy of analysis, many algorithms are proposed to model and monitor the performance of the system. Among these methods, PCM (Partitioned Context Modeling) [11] and EPCM (Extended Partitioned Context Modeling) [12] are the representative caching algorithms to predict the upcoming requests based on the file accesses patterns. For the large scale system, the related work of behavior modeling and prediction is rather limited. In [13], they aim to predict the behavior of VM resource for each individual VM. Gandhi et al. propose AutoScale based on the control policy for the cloud resource management in the data center [14]. In [15], Shen et al. propose a framework named CloudScale that aims to predict the resource at minute level.

The existing works cast the prediction of the cloud capacity planning as a regression problem, and employ some state-of-the-art time series prediction techniques to predict the prospective cloud resource [7], [16]. The sliding window approach [17] and some extensions of Auto Regression [18], [19] have leveraged to fit the time series and provide the recommendations for the cloud capacity planning. Artificial Neural Network (ANN) and Support Vector Machine (SVM) regression are also utilized to predict the cloud capacity

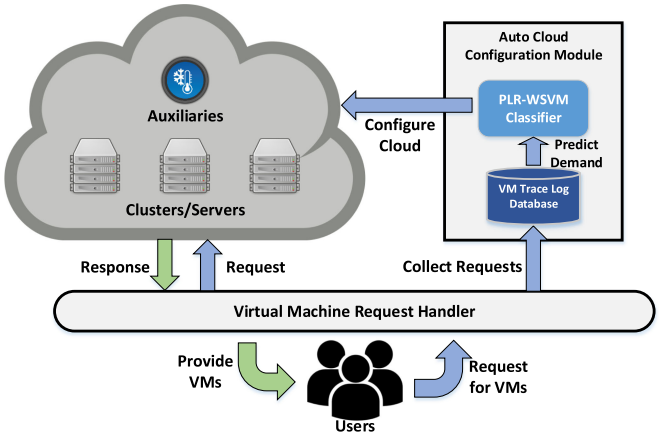


Fig. 3. The system framework.

resource demand in computing systems. These methods decrease the predictive costs compared with the linear regression [20], [21]. However, these approaches have the weak performance when the demands are changing abruptly.

3 Cloud Capacity Prediction Framework

To address the problem of preparing the cloud capacity in practice, in this paper, we propose PLR-WSVM, a framework predicting the changing trend of the cloud capacity provisioning in the following period. Instead of predicting the specific amount of demands in the future, PLR-WSVM is capable of providing more accurate prediction of the prospective changing trend (i.e., gradual, abrupt increasing, or abrupt decreasing). Fig. 3 shows the interaction among the *users*, *cloud resource*, *VM request handler*, and *auto cloud configuration module*. The *VM request handler* aims to receive the requests from the *users* and apply *cloud resource* for VM resource while sending the collected requests data to *auto cloud configuration module*. The *auto cloud configuration module* will predict the prospective cloud resource demands according to the collected requests and dynamically configure *cloud resource* based on the prediction. In this section, we aim to show the structure of *auto cloud configuration module* which plays a crucial role in the prediction of cloud resource demands.

3.1 Overview of Structure

Fig. 4 illustrates the schematic workflow of *auto cloud configuration module*: Step 1, extract the statistical information (e.g., the number of requests and the requesting company) from the *virtual machine trace log database* as the features for the prediction; Step 2, predict the cloud capacity demands in the upcoming period using WSVM modeled by the extracted statistical information; Step 3, output the prediction and provide the recommendation for the preparing cloud capacity; Step 4, insert the current request into the database and compare the current situation with the prediction; Step 5, re-segment the time series (i.e., the changing trend of request capacity) employing PLR and update the *labeled request changing trend dataset* using the new data; Step 6, update the classifier according to the predefined weights of each changing trend (i.e., gradual, abrupt increasing, and abrupt decreasing) if the prediction has a significant error.

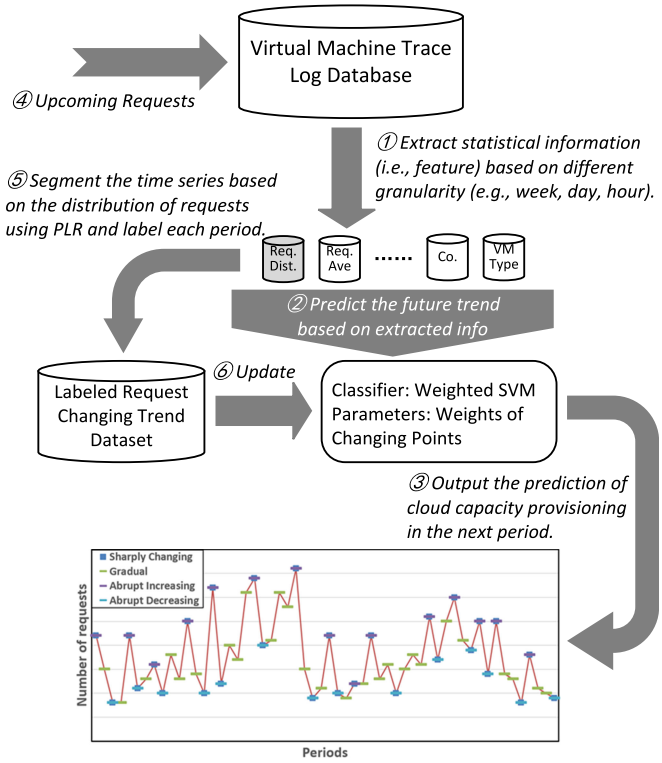


Fig. 4. The schematic structure of auto cloud configuration module.

3.2 Piecewise Linear Representation

In order to cast the prediction of cloud capacity provisioning as a classification problem, it is necessary to convert the changing cloud capacity demand (i.e., continuous values) into specific labels. In this paper, we assume that the changing trend of cloud capacity provisioning consists of gradual fluctuation, abrupt increasing change, and abrupt decreasing change. However, it is time-consuming and inefficient to manually identify these real-time data. To this end, the selection of time series segmentation strategies is imperative to address the problem of automatically identifying the changing trend in this scenario.

The virtual machine trace log database is comprised of a large amount of request records, where any statistical information (i.e., the summarization of requests) and label (i.e., the timestamp of abrupt changing point) are directly included. After accumulating the number of requests in each period (e.g., week, day, or hour), an ordered time series $T = \{t_1, t_2, \ldots x_n\}$ is given where $t_i (1 \leq i \leq n)$ presents the number of requests in the ith period. The goal of time series segmentation is to split T into $T_{seg} = \{S_1\{x_1, x_2, \ldots x_i\}, S_2\{t_i, t_{i+1}, \ldots t_j\}, S_3\{t_j, t_{j+1}, \ldots t_k\}, \ldots S_M\{t_l, t_{l+1}, \ldots t_m\}\}$ where $S_I (1 \leq I \leq M)$ represents the segmentation that belongs to T, and t_i, t_j, and t_l demonstrate abrupt increasing or decreasing points, respectively. Traditional time series segmentations are generally grouped into three categories [22]:

- *Sliding Window*: The segmentation begins within the predefined window size and grows until the bound error is met. The following segmentation is starting from the ending point of the previous one. Generally, we compare the variance in the segmentation to the predefined threshold to determine whether the segmentation should continue growing or not.

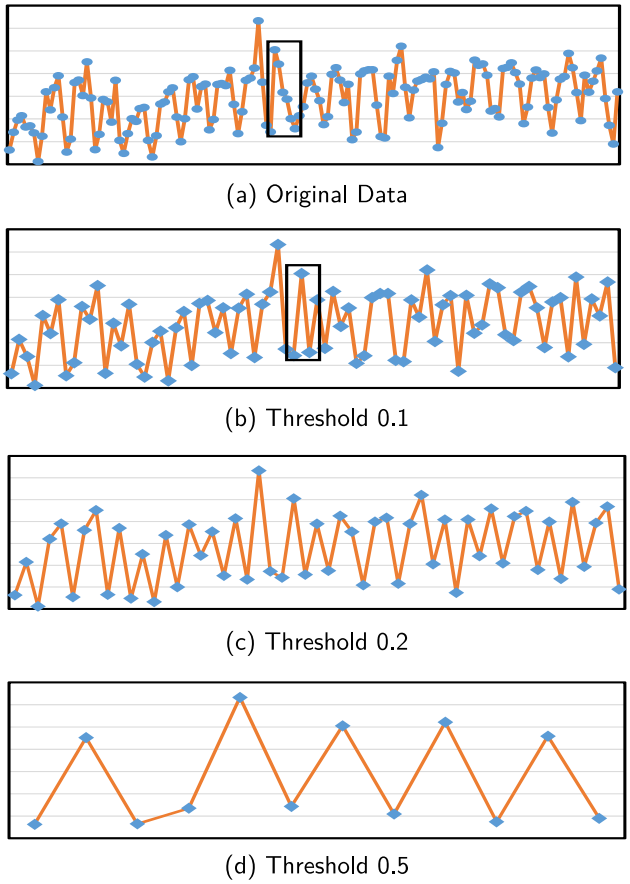


Fig. 5. The possible abrupt changing points identified by PLR using different thresholds

- *Top-Down*: The time series is segmented recursively based on the bound error until each segment meet the demand.
- *Bottom-Up*: The time series is segmented into the finest segments, then the segments keep merging until the bound error is met.

In this paper, we adopt PLR (Piecewise Linear Representation), a top-down time series segmentation strategy, to automatically segment T and identify the abrupt changing points. PLR is a geometrical algorithm that splits the segmentation at the point which has the greatest distance from the line segment connecting the beginning and ending points of current segmentation. The time series will be partitioned recursively until the distance from each point to the line is less than the predefined threshold (more details can be found in our previous work [10]). Therefore, the predefined threshold will directly determine the degree and density of identified changing points. In our experiments, the number of requests in each period is normalized according to the maximum one, and the thresholds are selected based on the normalized time series. Fig. 5 shows the possible abrupt changing points identified by PLR using different thresholds.

As observed in Fig. 5, with the increasing value of threshold (i.e., the degree of changing trend becomes higher), more and more gradual points are filtered and only the abrupt changing ones are retained. Compared to the traditional detection strategy which directly calculates the changing ratio or amount of requests in the current and previous periods, PLR is capable of identifying the changing trend making use of the whole segmentation. For example, the black rectangle in Fig. 5a shows a time series of changing cloud capacity provisioning where the ordered points are 505, 442, 317, 287, 201, and 157, respectively. The traditional strategy is insensitive toward this changing trend since there are many transition states between the beginning point (i.e., 505) and the ending one (i.e., 157). However, as a top-down strategy, PLR can identify the abrupt changing trend considering not only the adjacent periods but also the whole segmentation (see the black rectangle in Fig. 5b).

3.3 Weighted Support Vector Machine

3.3.1 Weighted Three-Class Classification

Since the significance of each changing trend is different and the abrupt changing periods are more imperative than those gradual ones, in this paper, we leverage Weighted Support Vector Machine (WSVM) to fit this classification model. The typical SVM is to generate a classification hyperplane that is capable of separating the samples into two classes within the maximum margin [10], [23]. In terms of the three-class classification in the problem, the one-vs-rest strategy is used in the SVM model, and the demonstration of SVM is

$$\begin{aligned} &\min_{w,b,\xi_i} \frac{1}{2} \parallel w \parallel^2 + C \sum_{i=1}^{l} \xi_i, \\ &s.t. \quad y_i(\langle w, \phi(\vec{x}_i)\rangle \vec{x}_i + b) \geq 1 - \xi_i, \\ &\xi_i \geq 0, \quad i = 1, 2, \ldots l, \end{aligned} \tag{1}$$

where $\{(\vec{x}_i, y_i) | \vec{x}_i \in R^n, y_i \in \{1, -1\}\}$ presents the features and labels of the ith training sample in which label 1 means the current training class while -1 means other classes, w is the vector that presents the hyper-plane in the feature space, b is a bias value of the hyper-plane, l is the amount of training samples, ξ_i is the slack variable that describes the maximum distance from the sample $\vec{x}_i$ to the functional margin, ϕ is the kernel function, $\langle w, \phi(\vec{x}_i)\rangle$ means the inner product of w and $\phi(\vec{x}_i)$, and C denotes the penalty coefficient that balance the trade-off between the accuracy and the maximum margin. Compared to the typical SVM, WSVM changes the constraint condition of each Lagrange multiplier α_i in the dual formulation:

$$\begin{aligned} &\min_{\alpha} \frac{1}{2} \sum_{i=1}^{l} \sum_{i=1}^{l} \alpha_i \alpha_j y_i y_j \langle \vec{x}_i, \vec{x}_j \rangle - \sum_{i=1}^{l} \alpha_i \\ &s.t. \quad \sum_{i=1}^{l} y_i \alpha_i = 0 \\ &0 \leq \alpha_i \leq C\mu_i, \quad i = 1, 2, \ldots l, \end{aligned} \tag{2}$$

where μ_i represents the weight of each training sample $\vec{x}_i$. If the weight of $\vec{x}_i$ is equal to 1, the constraint condition of α_i is equivalent to the one in the typical SVM (i.e., $0 \leq \alpha_i \leq C$).

In this paper, we aim to train a WSVM model to classify the changing trend of customer demands based on the current VM resources and the related statistical information (i.e., features). Due to the difference between the overestimation and the under-estimation of customer demands in practical applications, we increase the weight of samples which have sharply changed demands with respect to previous periods. Particularly, we consider that the sharply

increasing trend is more important than the decreasing one, because the VM providers often maintain current resource assignments for the sharply decreasing demands. Therefore, the loss of each sample is:

$$l_i = \begin{cases} 0, & |\xi_i| \geq 1 \\ \mu_i \xi_i, & |\xi_i| < 1, \end{cases} \tag{3}$$

where $\mu_i = \alpha$ if the sample $(\vec{x_i}, y_i)$ is a sharply increasing period, $\mu_i = \beta$ if the sample $(\vec{x_i}, y_i)$ is a sharply decreasing period, and $\mu_i = 1$ is a gradual period. Here, α and β can be defined based on the practical requirement of VM providers. Thus the loss function of WSVM is $L_{WSVM} = \sum_{i=1}^{l} l_i$.

3.3.2 Incremental Learning

In terms of the irregularly changing trend of cloud capacity demand, it is imperative to constantly update the model utilizing the upcoming requests for the accurate prediction. However, the cost of instantly re-training model is large that cannot meet the demand in such rapid changing scenario and it is necessary to update the model to fit the volatile demands in the future. To this end, we propose an incremental learning strategy to address this problem within the WSVM.

The main idea of the iteration process in classical SVMs is to heuristically select and update some pairs of Lagrange multipliers α [24], [25]. During the selection process of Lagrange multipliers, each sample is checked and its multiplier will be selected if the sample violates the KKT conditions, then the second multiplier is chosen based on the approximation of the step size:

$$\begin{aligned} E_{(i,j)} &= |E_i - E_j| \\ &= |\mu_i \cdot (\langle \vec{w}, \vec{x_i} \rangle + b - y_i) - \mu_j \cdot (\langle \vec{w}, \vec{x_j} \rangle + b - y_j)|, \end{aligned} \tag{4}$$

where E_i and E_j are the training error on the ith and jth samples respectively, and the weights μ_i and μ_j are utilized to magnify the training error in corresponding classes. Therefore, with the coming requests in the following period, the new samples (i.e., the following period) will be appended to the training set, and the corresponding features (i.e., the statistical information) will be checked whether the sample is against the KKT conditions or not. The new sample will be selected to optimize the corresponding Lagrange multiplier as the first choice, and the second sample will be searched based on Equation (4) leveraging the error set $\vec{E}$ that WSVM keeps. In addition, in order to decrease the effect of antiquated data, the model will be automatically reconstructed when the size of training set has reached a predefined threshold. Algorithm 1 describes the process of the incremental learning strategy in details.

4 Experimental Design and Evaluation

In this section, we evaluate the efficacy of proposed PLR-WSVM on the prediction of resource demand in cloud computing. In Section 4.1, we briefly introduce the experimental setup including the dataset of real virtual machine trace log and baseline algorithms. In Section 4.2, we show the comparison between regression-based and classification-based strategies and discuss the evaluation and advantages of our proposed approach.

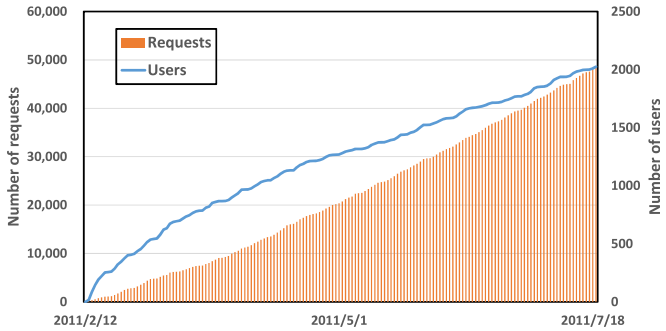


Fig. 6. The increasing number of total cloud service request and users over time

Algorithm 1. PLR-WSVM Applying the Incremental Learning

Input:
T: cloud capacity provisioning time series;
$\vec{x_i}$: the statistical information of the period t_i;
μ_i: the weight of the ith period;
Per: the predefined duration of training period;
PLR_t: the threshold of PLR;
n: the window size of training set;
$ratio_t$: the threshold describes the ratio of new samples in the training set;

Output:
$Pred$: the prediction of following period;

1: Normalize the time series T according to $\tilde{t_i} = \frac{t_i - t_{min}}{t_{max} - t_{min}}$;
2: Apply PLR to segment the normalized T based on $T_{seg} = PLR(T)$ and obtain the labels $Y = \{y_1, y_2, \ldots, y_{|T_{seg}|}\}$ where y_i means the changing trend of the ith period with respect to the $(i-1)$th one;
3: Train WSVM model M using $(x_{\vec{i-1}}, y_i)$ with corresponding weight μ_i, where $x_{\vec{i-1}}$ means the extracted statistical features of the $(i-1)$th period;
4: **while** the system is working **do**
5: **while** the count of time has met Per since last update **do**
6: Normalize the sample t_k based on previous t_{max} and t_{min};
7: Append and label t_k based on the last segmentation S_M of T_{seg}: y_k;
8: Update M leveraging $(x_{\vec{k-1}}, y_k, \mu_k)$ considering the KKT conditions;
9: Based on the updated M, the $Pred$ (i.e., the changing trend y_{k+1} of the upcoming period) can be obtained applying $\vec{x_k}$;
10: **end while**
11: **if** the ratio of training set mets $ratio_t$ **then**
12: Prepare the new time series $T = \{t_{k-n}, t_{k-n+1}, \ldots t_k\}$;
13: Reconstruct M using the new T;
14: **end if**
15: **end while**

4.1 Experiments Setup

4.1.1 Dataset and Preprocessing

In this paper, we evaluate PLR-WSVM using the real virtual machine trace log of IBM Smart Cloud Enterprise. The dataset includes 48,368 records generated by 2,024 users who requested for the virtual machine services during 5 months in 2011. Fig. 6 presents the increasing trend of users and requests over time. Each record in the dataset has 21 fields

TABLE 1
The Description of Statistical Features

No.	Feature	Description
1	Current Demands	The number of requests during current period (i.e., hour).
2	Average Demands	The average number of requests in past seven days.
3	Variance Demands	The variance of requests received in past seven days.
4	Historic Demands	The number of requests during the same time period in last week.
5	VM Type Distribution	The distribution of requested virtual machine types during current period.
6	Company Distribution	The distribution of companies which are requesting for services during current period.

describing the request, such as the requested VM type and the start/end time of current service. To make full use of the raw trace log, we employ a 2-step data processing to format the data for training the model:

1) *Selecting the granularity of time*: The trace log recorded each request by seconds, and each record contains the start/end time of the request. In the dataset, the longest lifetime of a request is 155 days, and the major requests (i.e., 72.6 percent) finish their missions within a day. Requests are being received every moment, and it is time-consuming and unnecessary to frequently adjust the cloud capacity provisioning every hour in practice. Therefore, it is reasonable and appropriate to predict the cloud capacity demand by days.
2) *Filtering the information*: The selection of feature and the statistics of requests both are imperative for modeling. The statistical information is sensitive to the time series aggregation granularity. With different granularity, the aggregated data can provide various meaningful information (see Fig. 1). In addition, there are 21 attributes of each record. Based on our previous research [1], [10], we extracted 6 statistical features from the trace log and show these features in Table 1.

4.1.2 Baseline Strategies

In this paper, we propose a solution to cast the cloud capacity provisioning prediction as a classification problem. Instead of providing the specific predictions of cloud resource demands from traditional regression approaches, our proposed method can predict the trend (i.e., abrupt increasing, abrupt decreasing, or gradual changing) more effectively. Thus, to evaluate our proposed PLR-WSVM, we design the experiments to (1) compare PLR to traditional time series segmentation strategies (i.e., sliding window and bottom-up) and (2) compare WSVM to several state-of-the-art classifiers (e.g., SGDClassifier and ETs); (3) evaluate selected regression-based algorithms which are robust to the provisioning prediction; (4) compare our proposed classification-based method to the effective regression-based approaches. The detailed descriptions of baseline strategies are presented in Table 2.

4.2 Result and Discussion

In this section, we aim to evaluate the effectiveness of PLR-WSVM and answer the following questions:

- Compare to other time series segmentation methods, why PLR is competitive in segmenting the time series of cloud capacity provisioning?
- Compare to the classification-based strategies, what is the drawback of regression-based methods?
- The robustness and efficacy of PLR-WSVM in the prediction of cloud capacity provisioning.

4.2.1 Time Series Segmentation

In order to evaluate the effectiveness of Piecewise Linear Representation (PLR), we compare PLR to another two traditional time series segmentations (i.e., sliding window and bottom-up). The main idea of sliding window is to grow a

TABLE 2
The Description of Baseline Strategies

Category	Method Name	Description
Time Series Segmentation	Sliding Window	A segment is grown until it exceeds some error bound [22].
	Bottom-up	Finest possible segments are merged based on some criteria [22].
Classification	SGDClassifier	Stochastic Gradient Descent Classification [26]
	KNeighborsClassifier	k-Nearest-Neighbors-based Classification [27]
	GaussianNB	Gaussian Naive Bayes
	GBC	Gradient Boosting Classification [28]
	ETs	Extremely Randomized Trees [29]
Regression	BayesianRidge	Bayesian Ridge Regression [30]
	ElasticNet	Linear Regression Model [31]
	GBR	Gradient Boosting Regression [28]
	KernelRidge	Kernel Ridge Regression [32]
	Lars	Least Angle Regression Model [33]
	MovingAverage	Naive Predictor [10]
	NNReg	Nearest Neighbour Regression
	PAR	Passive Aggressive Regression [34]
	SGDRegressor	Stochastic Gradient Descent Regression [26]
	SVMReg	SVM Regression

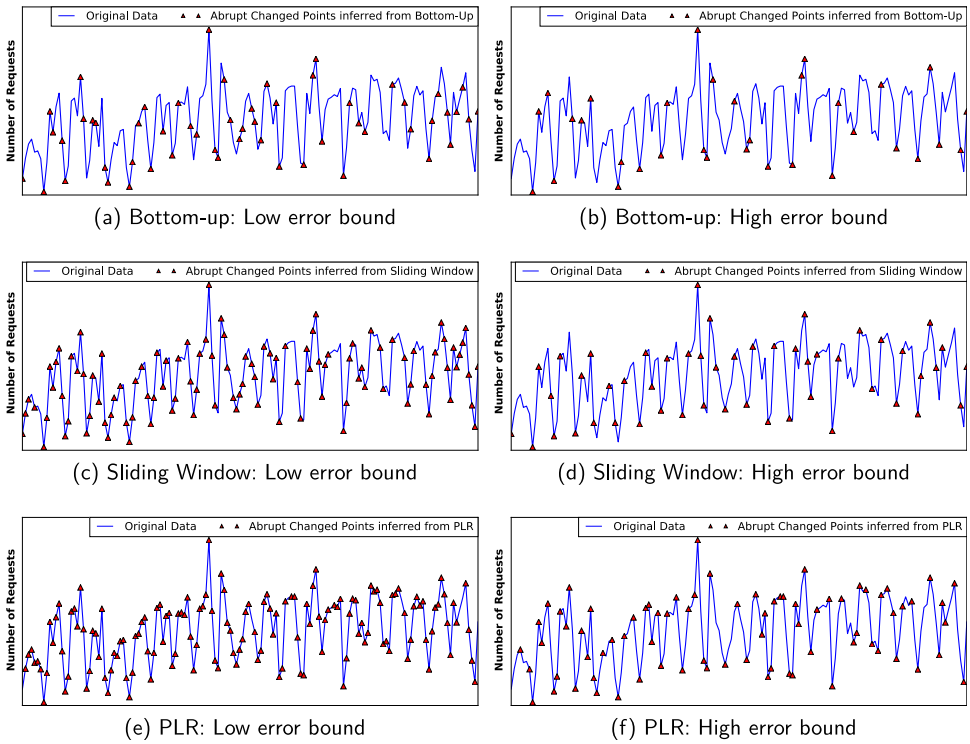


Fig. 7. The comparison of time series segmentation algorithms.

segment until the predefined error bound (i.e., the variance) is met. The bottom-up strategy is to split the time series into segments where each unit consists of two timestamps, then the adjacent segments are merged until the predefined error bound is met [22]. Fig. 7 presents the comparison among three time series segmentation methods using different error bounds where the red triangles demonstrate the abrupt changing points. As observed in Fig. 7, PLR is capable of nearly identifying all the peaks and troughs (i.e., abrupt increasing and decreasing points) using the high error bound. The bottom-up method cannot effectively find abrupt changing points out, while the sliding window strategy has many misidentifications although it outperforms the bottom-up method. In addition, according to the selection of error bound, the density and degree of changing points are different. With the lower error bound, the time series segmentation methods label more points which are changing gradually. On the contrary, the method, adopting the higher error bound, will only find out the points with great fluctuation. In terms of the adaptability of the error bound, PLR outperforms other baseline strategies, which is capable of catching almost all the key points (i.e., abrupt changing points). In other words, employing PLR, users can customize the density and degree to label the changing points what they need.

4.2.2 Classification-Based Algorithms

After adopting the selected time series segmentation strategy, we can leverage the labeled temporal data to evaluate the effectiveness of the weighted SVM (WSVM). In this paper, we compare our proposed method to several selected classification-based methods in predicting the changing trend of cloud capacity demands. Table 3 presents the performance of the prediction using the state-of-the-art classifiers based on the labeled data adopting PLR. In this experiment, the error bound of PLR is 0.02, and the weights of each type of changing points (i.e., gradual, abrupt increasing, or abrupt decreasing) are 1.00, 1.90, and 1.70, respectively.

As observed in Table 3, WSVM well outperforms other baseline strategies, especially in predicting the abrupt changing points. Although some approaches (i.e., GaussianNB)

TABLE 3
The Comparison between WSVM and other Baseline Strategies in Predicting Abrupt Changing Points

Approach	F1-score			Recall			Precision			Accuracy
	GRAD[1]	INCR	DECR	GRAD	INCR	DECR	GRAD	INCR	DECR	
SGDClassifier	0.481	0.391	0.304	0.564	0.455	0.207	0.419	0.342	0.571	0.405
KNeighborsClassifier	**0.574**	0.459	0.440	0.527	0.564	0.379	0.630	0.388	0.524	0.488
GaussianNB	0.485	0.095	0.060	**0.909**	0.055	0.034	0.331	0.375	0.222	0.327
GBC	0.516	0.481	0.561	0.436	0.564	0.552	0.632	0.419	0.571	0.518
ETs	0.509	0.417	0.547	0.509	0.455	0.500	0.509	0.385	**0.604**	0.488
WSVM-original	0.494	0.511	**0.590**	0.364	0.618	**0.621**	0.769	**0.436**	0.562	0.536
WSVM-incremental	0.494	**0.538**	0.579	0.345	**0.709**	0.569	**0.864**	0.433	0.589	**0.542**

GRAD represents the performance of predicting gradual points while INCR and DECR represent the prediction of increasing points and decreasing points, respectively.

have the good performance in predicting specific types of changing points, the imbalanced performance shows the drawbacks of predicting the abrupt changing points that we are more concerned with. Nevertheless, our proposed method provides a good trade-off among the gradual, abrupt increasing, and abrupt decreasing changing points. In addition, compared with the original WSVM which re-trains the model when the new requests are coming, our proposed incremental WSVM has the similar performance. However, 0.542, the average accuracy of the incremental WSVM, not only represents the effectiveness of WSVM predicting the prospective changing trend in a three-class classification problem, but also implicitly means the efficiency of our incremental learning strategy. With the incremental strategy, the model will be re-trained only if the current prediction is wrong or the model is not updated for a long time. Therefore, the average accuracy of the incremental WSVM means that our proposed method saves nearly 60 percent of time consumption in re-training the model. In the following experiments, without any statement, all the WSVMs represent the incremental version.

4.2.3 Regression-Based Algorithms

Before the comparison between classification-based and regression-based strategies, it is imperative to understand the performance of the state-of-the-art regression algorithms in the scenario of predicting the cloud capacity demands. As shown in Table 2, we evaluate ten effective regression approaches and Fig. 8 presents the prediction of the selected methods fitting the original capacity changing time series in three different periods. From top to bottom, the time series are fitted by selected algorithms ordered by Table 2. From left to right, the selected algorithms fit three types of time series ordered by the density of abrupt changing points.

As shown in Fig. 8, the major approaches are capable of predicting the cloud resource provisioning well when the time series are gradual, while only the minor ones can roughly describe the changing trend when the time series become more unstable. With the increasing density and instability of the time series, the regression-based methods make more fitting errors. Although the regression-based methods can provide the rough description of changing trend, the accurate identification of each abrupt changing point is the prior demand that the regression approaches cannot meet in this scenario.

4.2.4 Classification-Based versus Regression-Based

In order to comprehensively understand the advantages of PLR-WSVM, we compare our proposed strategy to the selected regression-based methods in this section. In terms of the different types of outputs in classification-based and regression-based approaches (i.e., discrete and continuous variables), we adopt two experiments to evaluate the effectiveness of selected approaches: (1) compare the consistency (i.e., increase or decrease) between the original data and the prediction of regression, and (2) transform continuous regression values into discrete labels and directly compare the prediction. Table 4 shows the performance of the selected regression-based algorithms predicting the changing trend. As observed in Table 4, the accuracy of each regression-based approach approximates 33.3 percent which is the random guesses of the three-class classification problem. In addition, the major approaches are biased toward the prediction of abrupt increasing and decreasing points, where the performance of predicting the gradual points is poor. Compared to these regression-based approaches, our proposed PLR-WSVM has the better performance, especially in the prediction of abrupt increasing points.

For the second comparison, in order to transform the resulting regressions into the labels, we compare the changing (i.e., increasing or decreasing) ratio of the current and previous period with the predefined threshold to determine whether the current capacity of requests is abruptly changed or not. For example, with the predefined threshold 1, the current period will be considered as an abrupt increasing point only if the request capacity of the current period is double of the capacity of the previous one. Fig. 9 presents the comparison between PLR-WSVM and each regression-based approaches with different thresholds.

As shown in Fig. 9, with the increasing threshold, the performance of predicting gradual points is greatly improved. However, the prediction of abrupt changing points becomes worse, which is against our original intention. This observation presents the drawback of regression-based approaches predicting the changing trend in this scenario. Different with these methods, our proposed PLR-WSVM is capable of customizing the changing degree using PLR and the importance of each changing trend adopting the predefined weights.

4.2.5 Effect of Different Weights

In previous experiments, we compare the classification-based approaches with the regression-based ones and evaluate the effectiveness of our proposed PLR-WSVM. In fact, the definition of the weight (i.e., μ_i) to each trend (i.e., gradual, increase, and decrease) can affect the performance of our proposed method. Generally, the larger weights on the sharply increasing samples will improve the accuracy of prediction the abrupt increasing trend while the performance of classifying other kinds of trends (e.g., gradual and decrease) will be declined, and vice versa. Therefore, in order to help users understand the effect of different weights to our proposed approach, Fig. 10 illustrates the performance of PLR-WSVM under different setups of weights. In this experiment, we predefine each samples using $\mu = 1$ as the default weight. For the sharply increasing and decreasing samples, we utilize $\mu_{INCR} = 1 + \alpha$ and $\mu_{DECR} = 1 + \beta$ respectively.

As shown in Fig. 10, the performance of PLR-WSVM changes with the adjustment of weights. Note that, the prediction of gradual trends is stable no matter how we adjust the weight of sharply increasing samples. This means our proposed method can improve the performance of predicting the sharply increasing trends which are more important than the decreasing ones while maintaining the accurate prediction of gradual trends.

5 Conclusion

Due to the time-variant and volatile characteristics, it is a big challenge to predict the prospective resource demands in cloud services. The typical regression-based techniques

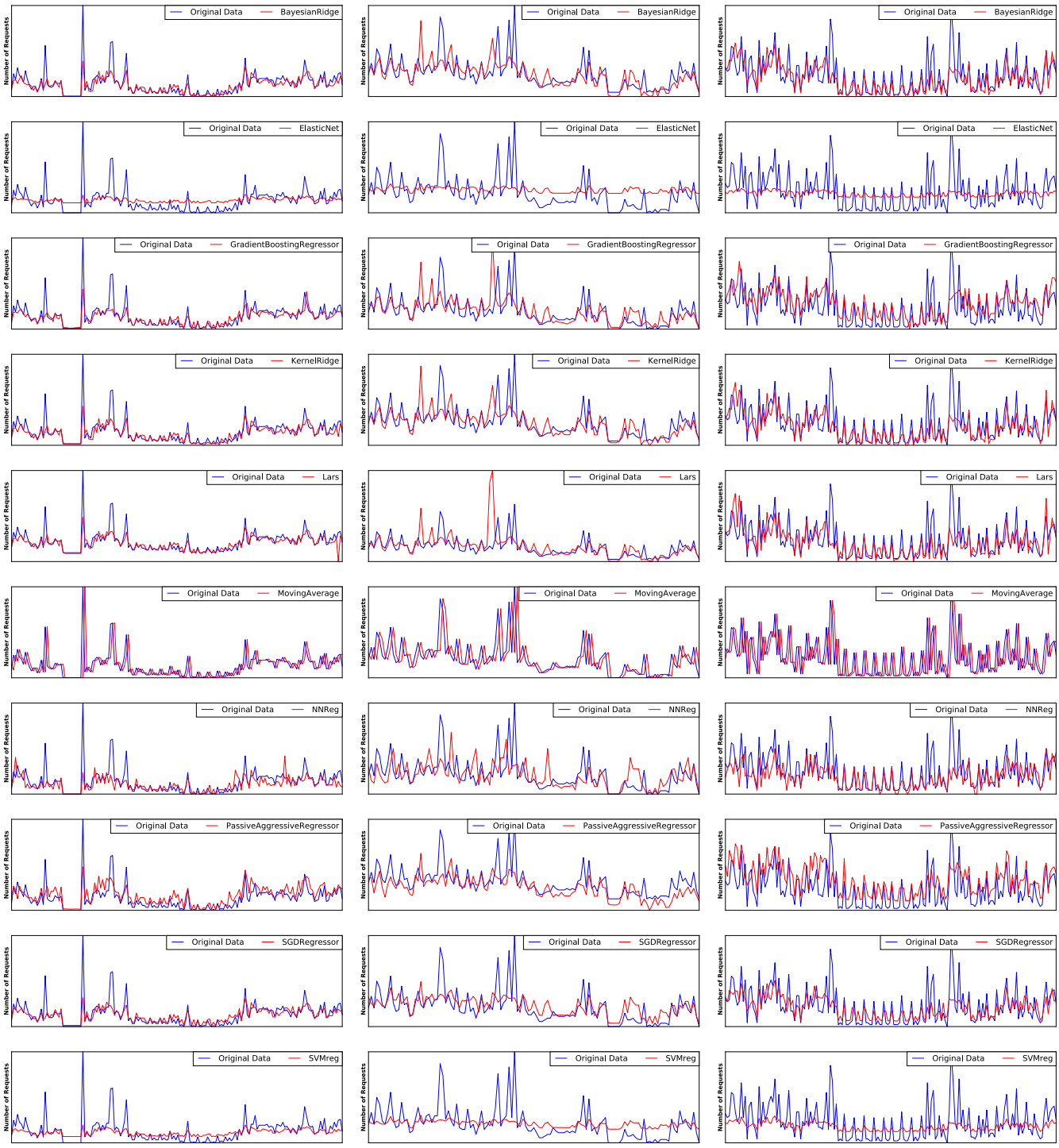


Fig. 8. Regression-based prediction results of three types of time series.

are capable of providing the prediction within the specific amount of cloud resource demands, however, the regression strategies are not robust in predicting the abrupt changing demands. In practice, the cost of error predicting abrupt changing demands is much larger than that predicting gradual changing demands, since more available cloud resources are not fully utilized or more customers cannot be satisfied instantly. In addition, it is unnecessary to predict the demands within specific values, because the prediction is always different from the actual demands and can be considered as the reference.

To this end, we construct an integrated framework that transforms the cloud resource planning problem into a three-class classification problem by employing PLR-WSVM to identify the gradual, abrupt increasing, and abrupt decreasing changing demands. In particular, we utilize PLR to automatically label the time series of original cloud resource demands, and train WSVM using the extracted statistical information of each period in the times series. Different from traditional regression-based techniques, our proposed method is capable of customizing the degree of abrupt changing demands (i.e., the predefined

TABLE 4
The Performance of Regression-Based Algorithms Predicting the changing Trend

Approach	Accuracy	Increasing	Decreasing	Gradual
BayesianRidge	0.305	0.338	0.392	0.000
ElasticNet	0.383	0.277	0.419	**0.536**
GBR	0.341	0.369	0.446	0.000
KernelRidge	0.323	0.400	0.378	0.000
Lars	0.317	0.369	0.378	0.036
MovingAverage	0.335	0.369	0.419	0.036
NNReg	0.407	0.400	0.419	0.393
PAR	0.323	0.369	0.405	0.000
SGDRegressor	0.317	0.354	0.405	0.000
SVMReg	0.323	0.369	0.405	0.000
PLR-WSVM	**0.542**	**0.709**	**0.569**	0.345

threshold in PLR) and the significance of different types of changing trend (i.e., the weight of each sample in WSVM) in order to minimize the total provisioning costs. In addition, for the efficiency of the framework deployed in practical applications, we propose an incremental learning strategy to satisfy the demand of real-time prediction with instantly upcoming requests. We evaluate our proposed framework utilizing the trace data of IBM Smart Cloud Enterprise. The experimental results show the effectiveness of PLR-WSVM. Compared to the baselines and the-state-of-art approaches,

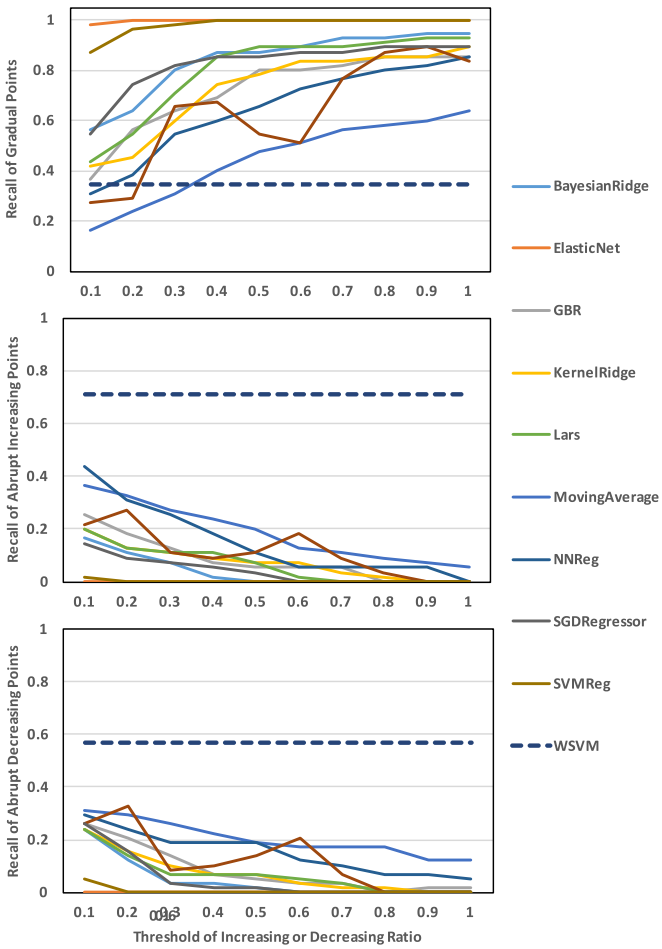
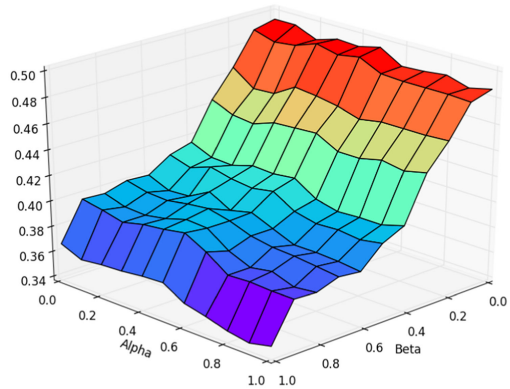
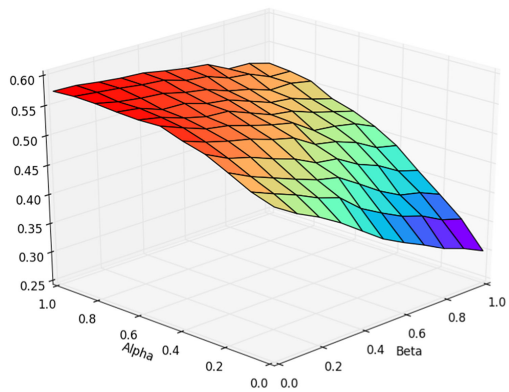


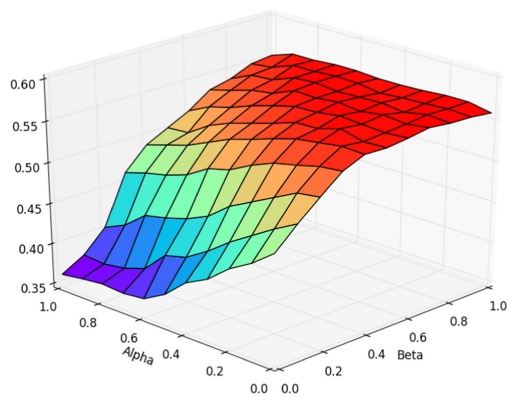
Fig. 9. The comparison between PLR-WSVM and the regression-based approaches using different threshold



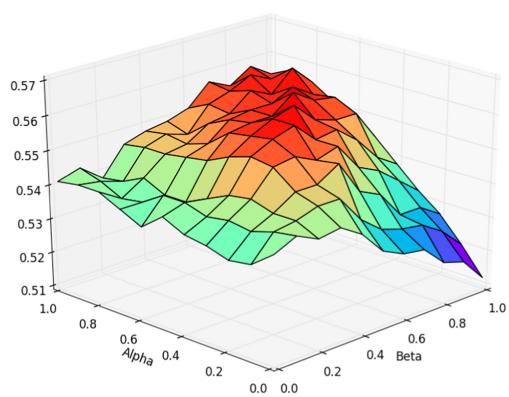
(a) F1 of gradual trend



(b) F1 of sharply increasing trend



(c) F1 of sharply decreasing trend



(d) Accuracy

Fig. 10. The performance of PLR-WSVM under different setups of weights.

our proposed framework achieves more robust and accurate performance, especially in the prediction of abrupt changing demands.

There are several ideas to extend our work in the future. First, the time series segmentation strategy still has some limitations such as the selection of the threshold and the obscure relation between the threshold and the degree of cloud changing demands. The adaptive and explainable threshold selecting technique would address this problem. Second, the features used in our framework is the simple statistical information, and more meaningful information (e.g., the relation between requests) may be useful for the cloud capacity planning. Third, the regression-based strategy is able to provide the prediction within specific values while the classification-based one has better performance of identifying abrupt changing demands. Can we integrate the corresponding advantages of these two representative approaches to provide the prediction of cloud resource demands? Can we integrate the corresponding advantages of these two representative approaches to provide the prediction of cloud resource demands, such as building a multi-task learning framework [35]?

Acknowledgments

This work is partially supported by the Jiangsu Key Laboratory of Big Data Security & Intelligent Processing NJUPT Grant BDSIP1803, the Jiangsu Provincial Natural Science Foundation of China under Grant BK20171447, the Jiangsu Provincial University Natural Science Research of China under Grant 17KJB520024, the National Natural Science Foundation of China under Grant 61503313 and No. 91646116, the Natural Science Foundation of Fujian Province (China) under Grant 2017J01118, the Fundamental Research Funds for the Central Universities of China (No. 30916015104), the National key research and development program: key projects of international scientific and technological innovation cooperation between governments (No. 2016YFE0108000); CERNET next generation Internet technology innovation project (NGII20160122); The Project of ZTE Cooperation Research(2016ZTE04 11), Jiangsu province key research and development program: Social development project (BE2017739), and Jiangsu province key research and development program: Industry outlook and common key technology projects (BE2017100).

References

[1] Y. Jiang, C.-S. Perng, T. Li, and R. N. Chang, "Cloud analytics for capacity planning and instant VM provisioning," *IEEE Trans. Netw. Service Manag.*, vol. 10, no. 3, pp. 312–325, Sep. 2013.

[2] R. Buyya, A. Beloglazov, and J. Abawajy, "Energy-efficient management of data center resources for cloud computing: A vision, architectural elements, and open challenges," vol. 12 no. 4, pp. 6–17, 2010.

[3] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. Buyya, "Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Softw. Practice Experience*, vol. 41, no. 1, pp. 23–50, 2011.

[4] S. Chaisiri, B.-S. Lee, and D. Niyato, "Optimization of resource provisioning cost in cloud computing," *IEEE Trans. Serv. Comput.*, vol. 5, no. 2, pp. 164–177, Apr.-Jun. 2012.

[5] H. A. Lagar-Cavilla, et al., "Snowflock: Rapid virtual machine cloning for cloud computing," in *Proc. 4th ACM Eur. Conf. Comput. Syst.*, 2009, pp. 1–12.

[6] D. Samant and U. Bellur, "Handling boot storms in virtualized data centers-a survey," *ACM Comput. Surveys*, vol. 49, no. 1, 2016, Art. no. 16.

[7] Y. Jiang, C.-S. Perng, T. Li, and R. Chang, "Asap: A self-adaptive prediction system for instant cloud resource demand provisioning," in *Proc. IEEE 11th Int. Conf. Data Mining*, 2011, pp. 1104–1109.

[8] Y. Jiang, C.-S. Perng, T. Li, and R. Chang, "Self-adaptive cloud capacity planning," in *Proc. IEEE 9th Int. Conf. Serv. Comput.*, 2012, pp. 73–80.

[9] C. Zeng, et al., "Fiu-miner: A fast, integrated, and user-friendly system for data mining in distributed environment," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 1506–1509.

[10] Q. Zhou, B. Xia, Y. Jiang, Q. Li, and T. Li, "A classification-based demand trend prediction model in cloud computing," in *Proc. Int. Conf. Web Inf. Syst. Eng.*, 2015, pp. 442–457.

[11] T. M. Kroeger and D. D. Long, "The case for efficient file access pattern modeling," in *Proc. 7th Workshop Hot Topics Operating Syst.*, 1999, pp. 14–19.

[12] T. M. Kroeger and D. D. Long, "Design and implementation of a predictive file prefetching algorithm," in *Proc. USENIX Annu. Tech. Conf. Gen. Track*, 2001, pp. 105–118.

[13] Z. Gong, X. Gu, and J. Wilkes, "Press: Predictive elastic resource scaling for cloud systems," in *Proc. Int. Conf. Netw. Service Manag.*, 2010, pp. 9–16.

[14] A. Gandhi, M. Harchol-Balter, R. Raghunathan, and M. A. Kozuch, "Autoscale: Dynamic, robust capacity management for multi-tier data centers," *ACM Trans. Comput. Syst.*, vol. 30, no. 4, 2012, Art. no. 14.

[15] Z. Shen, S. Subbiah, X. Gu, and J. Wilkes, "Cloudscale: Elastic resource scaling for multi-tenant cloud systems," in *Proc. 2nd ACM Symp. Cloud Comput.*, 2011, Art. no. 5.

[16] J. D. Hamilton, *Time Series Analysis*. Princeton, NJ, USA: Princeton Univ. Press, 1994, vol. 2.

[17] T. G. Dietterich, "Machine learning for sequential data: A review," in *Proc. Joint IAPR Int. Workshops Statistical Techn. Pattern Recognit. Structural Syntactic Pattern Recognit.*, 2002, pp. 15–30.

[18] P. Whitle, *Hypothesis Testing in Time Series Analysis*. Stockholm, Sweden: Almqvist & Wiksells, 1951, vol. 4.

[19] R. F. Engle, "Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation," *Econometrica, J. Econometric Soc.*, vol. 50, pp. 987–1007, 1982.

[20] W. Iqbal, M. N. Dailey, and D. Carrera, "Black-box approach to capacity identification for multi-tier applications hosted on virtualized platforms," in *Proc. IEEE Int. Conf. Cloud Service Comput.*, 2011, pp. 111–117.

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

[22] E. Keogh, S. Chu, D. Hart, and M. Pazzani, "An online algorithm for segmenting time series," in *Proc. IEEE Int. Conf. Data Mining*, 2001, pp. 289–296.

[23] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[24] J. C. Platt, "12 fast training of support vector machines using sequential minimal optimization," *Advances in Kernel Methods*, Cambridge, MA, USA: MIT Press, pp. 185–208, 1999.

[25] B. Xia, Z. Ni, T. Li, Q. Li, and Q. Zhou, "Vrer: Context-based venue recommendation using embedded space ranking SVM in location-based social network," *Expert Syst. Appl.*, vol. 83, pp. 18–29, 2017.

[26] Y. Tsuruoka, J. Tsujii, and S. Ananiadou, "Stochastic gradient descent training for l1-regularized log-linear models with cumulative penalty," in *Proc. Joint Conf. 47th Annu. Meet. ACL 4th Int. Joint Conf. Natural Language Process.*, 2009, pp. 477–485.

[27] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," in *Proc. 47th Annu. IEEE Symp. Foundations Comput. Sci.*, 2006, pp. 459–468.

[28] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statistics*, vol. 29, pp. 1189–1232, 2001.

[29] B. Xia, H. Zhang, Q. Li, and T. Li, "Pets: A stable and accurate predictor of protein-protein interacting sites based on extremely-randomized trees," *IEEE Trans. Nanobiosci.*, vol. 14, no. 8, pp. 882–893, Dec. 2015.

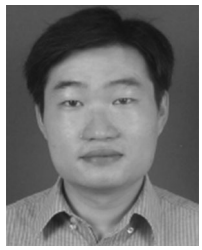
[30] D. J. MacKay, "Bayesian interpolation," *Neural Comput.*, vol. 4, no. 3, pp. 415–447, 1992.

[31] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Roy. Statistical Soc., Series B*, vol. 67, no. 2, pp. 301–320, 2005.
[32] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT press, 2012.
[33] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, "Least angle regression," *Annals Statistics*, vol. 32, no. 2, pp. 407–499, 2004.
[34] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *J. Mach. Learn. Res.*, vol. 7, no. Mar, pp. 551–585, 2006.
[35] W. Xue, W. Zhou, T. Li, and Q. Wang, "MTNA: A neural multi-task model for aspect category classification and aspect term extraction on restaurant reviews," in *Proc. Eighth Int. Joint Conf. Natural Language Process.*, 2017, pp. 151–156. [Online]. Available: http://aclweb.org/anthology/I17–2026

Bin Xia received the PhD degree in computer science from the Nanjing University of Science and Technology, in 2018. Currently, he is current an assistant professor in the School of Computer Science and Technology, Nanjing University of Posts and Telecommunications (NJUPT). His research interests include recommender system, data mining, and deep learning.

Tao Li received the PhD degree in computer science from the Department of Computer Science, University of Rochester, Rochester, New York, in 2004. He was a professor in the School of Computing and Information Sciences, Florida International University, Miami, Florida. He was also a professor in the School of Computer Science & Technology, Nanjing University of Posts and Telecommunications (NJUPT). His research interests included data mining, computing system management, information retrieval, and machine learning. He received the US National Science Foundation (NSF) CAREER Award and multiple IBM Faculty Research Awards. Sadly, he passed away in 2017.

Qifeng Zhou received the BS and MS degrees in dynamical set & automatization from WuHan University, in 1999 and 2002, respectively, and the PhD degree in control theory & control engineering from Xiamen University, in 2007. In 2002, she joined the Department of Automation of Xiamen University. Her present research interests include mainly in the machine learning, intelligence system, SVM, kernel learning, and digital image processing.



Qianmu Li received the BS and PhD degrees in computer science from the Nanjing University of Science and Technology, in 2001 and 2005, respectively. In 2012, he acted as an academic visitor with the Florida International University. He is currently an full professor in the School of Computer Science and Engineering, Nanjing University of Science and Technology. His current interests include data mining and information security. He is a member of the ACM and the CCF.

Hong Zhang received the BS and MS degrees in computer science from the Nanjing University of Science and Technology (NUST), China. He is currently an full professor in the School of Computer Science and Engineering, Nanjing University of Science and Technology. His current interests include confidence software technology and information security.