Price forecasting for spot instances in Cloud computing

Zhicheng Cai[a,*], Xiaoping Li[b], Rubén Ruiz[c], Qianmu Li[a]

[a]*School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, 210094, China*
[b]*School of Computer Science and Engineering, Southeast University, Nanjing, China*
[c]*Instituto Tecnológico de Informática, Acc. B. Universitat Politècnica de València, València, Spain.*

Abstract

Big data applications usually need to rent a large amount of virtual machines from cloud computing providers. As a result of the polices employed by Cloud providers, the prices of the resources have a stochastic behavior. Recently, Spot prices fluctuate greatly or have multiple regimes. Choosing virtual machines according to trends of prices is helpful to decrease the resource rental cost. Existing price predicting methods are unable to accurately predict prices in these environments. Therefore, a dynamic-ARIMA and two markov regime-switching autoregressive model based forecasting methods have been developed in this paper. Experimental results show that the proposals are better than the existing MonthAR for most scenarios.

Keywords: Cloud computing, Spot price, Forecast, Markov regime-switching, Scheduling

[*]Corresponding author.
Email addresses: `caizhicheng@njust.edu.cn` (Zhicheng Cai), `xpli@seu.edu.cn` (Xiaoping Li), `rruiz@eio.upv.es` (Rubén Ruiz), `qianmu@njust.edu.cn` (Qianmu Li)

1. Introduction

From the perspective of big data applications, cloud users require precise price prediction in order to save on rental. Usually, these applications consume a large quantity of computation resources. Cloud computing offers access to hundreds or even thousands of Virtual Machines (VM) for speeding up the processing of these applications [1]. At the same time, executing big data applications on cloud computing platforms saves the cost of establishing and maintaining private data centers. Cloud resource providers provision different pricing models. The commonly used models are fixed price and stochastic price models. For example, Amazon EC2 provisions on-demand VM instances with a fixed price model and spot VM instances with a stochastic model. Generally resources with stochastic price models provision cheaper prices than those with fixed pricing models. Since reserved and on-demand VM instances are fixedly priced, only spot VM instances are considered for price prediction. Spot prices are stochastically set as a result of auctioning spot VM instances according to real time user demands. Spot VM instances of different VM types in different physical regions have different stochastic spot prices. A VM instance is out-of-bid if the spot price is higher than that of the current bid. These characteristics make spot prices fluctuate. Good price forecast is helpful for choosing appropriate VM types, selecting right renting periods and setting optimal bids to save on rental costs.

In auction based public Clouds, stochastically arrived user demands and unpredictable user bids determine final Spot prices which make Spot prices prediction complex [2, 3, 4]. Recent trends, such as great fluctuations and switching regimes (different statistical means and variances in different time periods), make it more difficult. The probability density functions (PDF) of inter-price times (the length of intervals between changes of prices) established in 2010 by Javadi et al. [4] show that peaks are around two hours, i.e., prices during one hour are usually the same in 2010 (stable). However, recent PDFs of inter-price times show that spot prices change more frequently and greatly. For example, the PDF of inter-price times of spot instances (during period from 28-04-2016 to 28-07-2016 and period from 03-04-2017 to 15-05-2017) demonstrates that 41.6 % of inter-price times are smaller than one hour although the price changes smaller than 5 % of the average price have been ignored. As a whole, spot prices fluctuate greater than before. Spot prices of some VM types even exhibit switching regimes. Figure 1 shows spot prices of the Amazon EC2 VM types "m4.2xlarge-us-east-1b-linux-unix" and "m4.2xlarge-us-east-1d-linux-unix". Prices of different time periods have different statistical characteristics such as mean values and variances which indicate that spot prices switch among several hidden regimes. Ben-Yehuda et al. [3, 5] studied trace files of Amazon EC2 and tried to discover how the Amazon prices its unused EC2 capacities. It is possible that spot prices of Amazon EC2 are limited by a dynamic bottom price (determined by an autoresression model) which ignores the bids lower than the bottom price. High spot prices may reflect market changes but most low prices are usually indicative of dynamic bottom prices, i.e., the two factors indicate that spot prices have two or more different regimes. The existing of different regimes means Spot prices are nonlinear.

In existing scheduling algorithms, different types of probability models have been used to help Cloud users recognize changes of Spot prices such as one or multiple step probability matrix of transition from one price to another [6, 7, 8], the probability density function of Spot prices [9], the probability of an out-of-bid event within a time interval [10, 11, 12], the probability of Spot instances staying available over time given a starting price and a bid [13], Q-learning based action selecting rules [14], etc. Usually static probability models are used to describe the transition probabilities among prices or probability density functions of failures as a whole whereas the correlation of multiple sequential prices is not considered which means that the corresponding methods cannot predict trends of sequential prices.

Autoregression based methods consider the correlation of multiple sequential prices which can predict trends of spot prices [2]. However, great Spot price fluctuations lead to many unstable Spot price time series which decrease the performance of existing autoregression-based prediction methods designed for linear and stable time series [15]. Autoregressive integrated moving average model (ARIMA) is an extension of autoregression decreasing the impact of unstable trends on predictions by differencing [15]. Single, double and triple exponential smoothing can also be used for modeling unstable time series considering trends and seasonality [16]. Predicted values of single exponential smoothing and double exponential smoothing compose a straight line respectively, therefore, SES and DES are not suitable for long-term prediction. Triple exponential smoothing models both trends and seasonality. Exponential smoothing methods only find a unique combination of parameters trying to fit all data. For example, there is only one smoothed seasonality pattern for all data in triple exponential smoothing. At the same time, traditional autoregression and ARIMA methods assume that the time series is linear and there is only a single regime for which a uniform model is built. However, many Spot prices are nonlinear and there are different regimes with (or without) different seasonality patterns. It is hard to find a uniform autoregression, ARIMA or exponential smoothing model suitable for all switching regimes and accurately forecast prices of different regimes. Therefore, building models for different regimes respectively and choosing appropriate regimes for forecast are crucial for an accurate prediction. Nonlinear models are usually used for describing nonlinear time series such as the threshold autoregression (TAR) [17] and the Markov regime-switching autoregressive model (MRS-AR) [18]. TAR extended from autoregression builds different linear autoregression models for different regimes and the switching among regimes depends on transition variable values. It is very complex to define an appropriate transition variable [19]. For example, Spot prices with the same threshold variable values may belong to different regimes when we use Spot prices (or lagged prices) as transition variables directly. MRS-AR is a generalized version of TAR in which the regime switching is much more flexible [20]. In MRS-AR, a Markov stochastic process is used to model switching of regimes where regimes are considered as states of the Markov stochastic process. The probability transition matrix describes the transition among regimes rather than defined transition variables in TAR. The Markov process part is used to describe the switching among regimes and the AR part is enclosed to model the trend of each regime. Therefore, MRS-AR is used to predict Spot prices in this paper.

For spot prices with switching regimes, it is crucial to determine the number of regimes and build different models for different regimes. In this

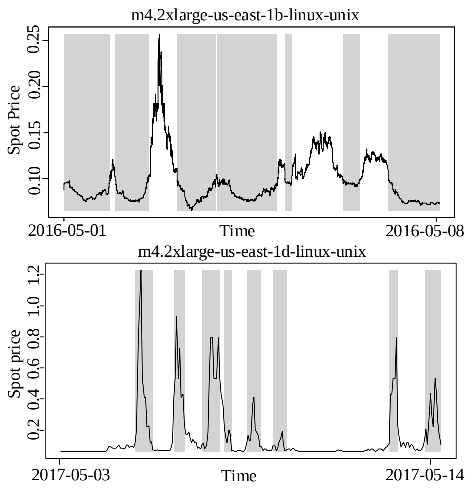


Figure 1: The spot prices in dollars of Amazon EC2 virtual machine type "m4.2xlarge-us-east-1b-linux-unix" of period from 01-05-2016 to 08-05-2016 and "m4.2xlarge-us-east-1d-linux-unix" of period from 03-05-2017 to 14-05-2017.

paper, the DBSCAN (density-based spatial clustering of applications with noise) clustering algorithm is adopted to determine the number of regimes. Then, MRS-AR with different autoregression models for different regimes are established. Choosing the right regimes is crucial for accurate forecast which means misspecification of the regime can lead to substantial losses in forecast accuracy [21]. In the literature, the absolute probability distribution over regimes of each prediction step, obtained from the conditional probability distribution and transition matrixes among regimes (see details in Section 5.1.3), is usually used to forecast. However, the absolute probability distribution of non-seasonal Markov chain converge close to equilibrium distributions very fast which cannot be used to forecast the switching of regimes appropriately for long-term prediction. For example, the regime with the largest probability is considered as the forecasted regime only for one-step forecast [19]. The expected mean determined by the conditional probability distribution [21] is used to predict without specifying regimes. In this paper, two methods are proposed to choose appropriate regimes for future times where transition probabilities of regimes and ARMA are combined. For predicting short-term prices, we assume that short-term prices belong to the same regime (lasting rule) with the last spot price which results in the first prediction method MRS-AR-L. Because regimes switch stochastically for long forecast periods, we need to predict the switching of regimes. For each regime, the duration of each regime occurrence compose a duration time series. Base on this duration time series, an autoregression and moving average model is proposed to predict future durations of each regime (switching rule) which results in another prediction method MRS-AR-SW. Using all historical data to build a uniform model is usually not helpful to predict local trends [20]. Therefore, only data of a recent time window is used to predict and different models are built for different time windows dynamically. The MRS-AR-L and MRS-AR-SW with dynamic models are called DMRS-AR-L and DMRS-AR-SW separately. Differencing is another way to make a time series stable which improves the prediction accuracy of traditional autoregression based methods. Therefore, in this paper, a dynamic autoregressive integrated moving average model (dynamic-ARIMA) based prediction method is proposed for comparison with MRS-AR. The main contributions of this paper are as follows:

(1) New characteristics of spot prices are considered such as great fluctuations and switching regimes.

(2) Two markov regime-switching autoregressive dynamic model based

forecasting methods DMRS-AR-L and DMRS-AR-SW are proposed for spot prices with switching regimes.

(3) A dynamic-ARIMA is developed to forecast spot prices which differences prices to decrease fluctuations.

(4) The short-term and long-term prediction accuracies have been improved by the proposals and guides to choose appropriate forecasting methods are given.

The rest of this paper is organized as follows. Section 2 gives an overview of the related work. The spot price prediction problem is described in Section 3 and some preliminaries are given in Section 4. Section 5 describes the proposed forecasting methods. Experimental results are shown in Section 6 and Section 7 concludes this research, pointing out future research directions.

2. Related works

In the literature, many algorithms have been designed to save the rental cost of resources with fixed prices for bag of tasks [22, 23], MapReduce tasks [24], workflows [25, 26, 27] and bag-of-task based workflows [28]. Spot instances with stochastic prices can be used to decrease the resource rental cost further. Related works about spot instances can be divided into two types according to the provider and user perspectives. From the perspective of cloud computing providers, auction strategies and resource management methods have been developed. Zhang et al. [29] and Vanmechelen et al. [30] studied how to set spot price and allocate limited capacities to different VM types or users. A bid price adjusting method was developed by Sadashiv et al. [31] allowing cloud users to modify bids after an initial bid has been set. Sadashiv et al. [31] assumed that bids can be adjusted at any time for avoiding out-of-bid events. However, the bids for rented spot instances of Amazon EC2 cannot be adjusted after being submitted. For minimizing resource rental cost, makespan or other objectives of Cloud users, algorithms for scheduling web-request tasks [32], parallel tasks [33] and workflows [34] to Spot instances have been developed. In these task scheduling methods on Spot instances, methods for choosing appropriate VM types, selecting proper time intervals and setting optimal bids are crucial to decrease rental cost. Many scheduling algorithms choose the spot instance with the cheapest current price to minimize cost. Li et al. [35] choosed Spot instances for workflows according to the minimum monetary cost determined by current Spot prices. A genetic algorithm was proposed by Vintila et al. [36] to estimate the execution time and budget of bag of tasks based on the current spot prices. Poola et al. [37, 38] developed an intelligent bidding strategy for workflow scheduling which considers current spot prices, on-demand prices and so on. However, choosing Spot instance types according to current prices ignored the trends of spot prices. In related works, statistical probability models and time series models based price prediction methods are commonly used to recognize price changes and make appropriate resource renting decisions.

In the literature, different kinds of probability models have been applied to recognize price changes. Zheng et al. [9] predicted the probability density function (PDF) of spot prices by simulating Cloud provider behaviors. One step transition probability of each price pair was used by Tang et al. [6]

to make optimal bids by dealing with Markov Decision Problems. Zafer et al. [8] builded a Markov process to simulate Spot prices consisting of a probability p to stay unchange and $1 - p$ to select a new random price. The multi-step transition probability from one price to another, the kernel of a semi-Markovoian chain, was used to make optimal bids by Song et al. [7]. Based on one-step transition probabilities between prices, the probability of Spot instances staying available (without out-of-bid events) over time given a starting price and a bid is obtained by Chohan et al. [13]. Yi et al. [10, 11] generated the probability density function (PDF) of a failure (out-of-bid events) within a time interval given a starting price and a bid according to historical prices directly. Based on the PDF of a failure, the practical task execution times are estimated. Similarly, Jangjaimon et al. [12] estimated the practical task execution time considering the delay of obtaining new resources after revocation. A probability density function of practical task execution times on different VM types with different bids was generated by Andrzejak et al. [39] using Monte-Carlo simulation which is an implicit way to study changes of Spot prices. A Q-learning method was developed by Abundo et al. [14] which studies action selecting rules based on historical price changes. These probability models are static and usually consider the relationship of two prices without considering trends of multiple sequential spot prices changing along with time.

Time series model based spot price prediction methods are commonly used to recognize patterns of time series and predict trends. Using predicted prices to select VM types, time intervals and set bids is the key to significantly decrease rental costs greatly. A month seasonal autoregressive (MonthAR) model was developed by Singh et al. [2] to predict spot prices. The predicted price is the regression of past 24 hour prices and each price of the same hour of past three months. However, recent Spot prices fluctuate greatly producing unstable time series decreasing the accuracy of linear MonthAR. ARIMA has been use to predict unstable time series in many fields by differencing the original data [15]. Single, double and triple exponential smoothing are also suitable for describing unstable time series [16]. Single exponential smoothing (SES) considers the last smoothed value as predicted value. In double exponential smoothing (DES), the last predicted trend is combined with the last smoothed value to forecast. The predicted values of SES and DES are on a horizontal line and an oblique straight line respectively. Therefore, SES and DES cannot be used for long-term prediction on time series with changing trends and means. Triple exponential smoothing (TES) models the seasonality with smoothed seasonal trends. However, Spot prices of many VM types are nonlinear processes with switching regimes. A unique autoregression, ARIMA or triple exponential smoothing model cannot describe time series with multiple regimes well. The threshold autoregression (TAR) [17] is a popular model for nonlinear time series which builds one linear autoregression model for each regime. The switching among regimes in TAR depends on transition variable values and it is very complex to define appropriate transition variables [19]. Hamilton et al. [18] proposed a markov-switching autoregression model for time series with switching AR models (MRS-AR) which is an extension of TAR with much flexible regime transition strategies. In MRS-AR, the regime transition is defined by a Markov process rather than transition variables. MRS-AR has been used to model many different time series with changing regimes such as exchange rates [21, 40] and financial time series [20]. In MRS-AR, for each prediction step, the absolute probability distribution over regimes can be used to forecast regimes which is generated from the last conditional probability distribution over regimes multiplying transition probabilities among regimes [21]. However, the absolute probability distribution of non-seasonal Markov chain converges close to equilibrium distributions very fast. Therefore, the absolute probability distribution is only suitable for predicting short-term regimes rather than long-term regimes. For example, Chen et al. [19] used the regime with the largest absolute probability to produce one-step forecast. Expectation based prediction is another way to produce forecast values. Yuan et al. [21] used the expected mean to forecast without specifying regimes which is determined by the probability distribution of regimes and the mean of each regime. However, specifying appropriate regimes is benefit to improve the prediction accuracy. Therefore, in this paper, two methods are proposed to predict regimes which use ARMA to model historical durations of regimes combined with the probability transition matrix among regimes.

To summarize, statistical probability models have been widely used to recognize changes of Spot prices without considering trends of sequential prices. Existing time series based models such as autoregression, ARIMA, SES, DES and TES with a single model cannot describe nonlinear Spot prices with multiple regimes well. Therefore, in this paper, a dynamic-ARIMA is proposed which difference prices to deal with great fluctuations and two markov-switching autoregression model based prediction methods are proposed which build separate models for different regimes.

3. Problem description

As stated, there are mainly two kinds of pricing models in cloud computing platforms [2]: (1) Fixed pricing models: prices of resources are fixed, e.g., On-demand VM instances and Reserved VM instances of Amazon EC2; (2) Variable pricing models: prices of resources change stochastically according to real-time market demands, e.g., prices of spot VM instances (spot prices) of Amazon EC2. Spot VM instances of Amazon EC2 are sold to users by auctions. When a user rents a spot instance, the user gives a bid price. Only when the bid price is higher than the spot price, the user has the possibility to get the VM instance. After the user gets the VM, the user needs to pay according to the spot price. If the spot price is higher than the bid price, the cloud computing provider will withdraw the VM (out-of-bid event) and the last interval will not be charged. When out-of-bid events occur, recovery from previous checkpoints will consume additional time and cost. Spot instances are usually cheaper than On-demand VM instances with the same configurations. However, spot instances are unreliable because of out-of-bid events resulting from the stochastic prices. Choosing appropriate VM types, selecting right renting periods and elaborating bids based on forecasted spot prices (predicted trends) is helpful for minimizing resource rental cost.

In this paper, we aim to develop several prediction algorithms to forecast spot prices accurately. For a spot price time series, using all past data to forecast future prices is time consuming. Therefore, only data of a fixed length of time (a window) is usually used, e.g., the prices of past 480 hours. As time goes, the time window moves forward. For each time window, the time series of spot prices is represented by $\{Y_t\}_{t=1}^{T}$, where t indicates the index of time units (e.g., the index of hours, minutes or seconds) and T is the length of the time series of the current time window. The objective of proposed forecasting methods is to minimize the forecasting errors. To evaluate the performance of predictions the popular Mean Absolute Percentage Error (MAPE) [41] is used as defined in the following.

$$MAPE_n = \frac{100}{n} \sum_{l=1}^{n} |\frac{\widehat{Y}_T(l) - A_T(l)}{A_T(l)}| \tag{1}$$

where $A_T(l)$ is the actual value, $\widehat{Y}_T(l)$ is the forecasted value of prediction step l and n is the length of forecast period. Note that *MAPE* has been criticized in the past, most notably in [42]. However, it is still the most easy to understand and most widely used error measure in the literature.

4. Preliminaries

4.1. Autoregressive Integrated Moving Average Model

Autoregressive-Moving Average (ARMA) is a popular model for describing a (weakly) stationary stochastic process. ARMA consists of two parts, an autoregressive (AR) part and a moving average (MA) part. The AR part involves regressing of past values and the MA part is a linear combination of past error terms. The model of ARMA(p, q) is as follows.

$$Y_t = \sum_{i=1}^{p} \phi_i Y_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t \tag{2}$$

where $\varepsilon_t \sim i.i.d.N(0, \sigma^2)$ is the error item of time t, p is the order of autoregression part, ϕ_i is the autoregression parameter of Y_{t-i}, q is the order of moving average part and θ_j is the moving average parameter of

ε_{t-j}. ARMA is usually suitable for stable time series and great fluctuations of spot prices will decrease the prediction accuracy of ARMA. ARIMA is an extension of ARMA by applying an initial differencing step to reduce the non-stationarity which improves the prediction accuracy on spot prices with great fluctuations. The model of ARIMA$(p, 1, q)$ is as follows.

$$Y_t - Y_{t-1} = \theta_0 + \sum_{i=1}^{p} \phi_i (Y_{t-i} - Y_{t-i-1}) + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t \tag{3}$$

where $\varepsilon_t \sim i.i.d.N(0, \sigma^2)$.

4.2. Markov regime-switching AR model

The linear based ARIMA uses a single model to describe time series. However, spot prices of some VM types have switching regimes, i.e., different periods have different regimes (different statistical characteristics). The spot prices parade among these regimes. Therefore, different time series models are needed to describe different regimes. For example, two different autoregressive (AR) models are required for modeling two different regimes. For the first regime, the AR model might be

$$Y_t = c_1 + \sum_{i=1}^{p} \phi_{i,1} Y_{t-i} + \varepsilon_{t,1} \tag{4}$$

where $\varepsilon_{t,1} \sim i.i.d.N(0, \sigma_1^2)$, c_1 is the intercept value and $\phi_{i,1}$ is the autoregressive parameter of Y_{t-i}. For the second regime, the AR model might be

$$Y_t = c_2 + \sum_{i=1}^{p} \phi_{i,2} Y_{t-i} + \varepsilon_{t,2} \tag{5}$$

where $\varepsilon_{t,2} \sim i.i.d.N(0, \sigma_2^2)$, c_2 is the intercept value and $\phi_{i,2}$ is the autoregressive parameter of Y_{t-i}. In other words, different regimes have different intercept values and autoregressive parameters. Each regime can be defined as a state of a Markov process. The spot price parades among these states. Assuming that there are k states for a give time series and each state is modeled by an AR model, a Markov regime-switching AR model can be obtained as follows

$$Y_t = c_{s_t} + \sum_{i=1}^{p} \phi_{i,s_t} Y_{t-i} + \varepsilon_{t,s_t} \tag{6}$$

where $\varepsilon_{t,s_t} \sim i.i.d.N(0, \sigma_{s_t}^2)$, $s_t \in R$ and $R = \{1, ..., k\}$. s_t is a discrete stochastic variable, which can be described by a Markov process with fixed state transition probabilities among states.

5. Proposed prediction methods

Autoregression based methods are prediction models which can forecast trends of prices [2]. However, two recent characteristics of spot prices (great fluctuations and switching regimes) decrease the accuracy. In this paper, two types of prediction methods are proposed to forecast spot prices considering these two characteristics. At first, two Markov regime-switching AR model based spot price predicting methods are developed which build different AR models for different regimes to improve prediction accuracy. Since differencing is a promising method to decrease the fluctuation of spot prices, a dynamic ARIMA predicting method is proposed which uses differencing to stabilize the time series.

5.1. Markov regime-switching AR based prediction methods

The original spot price time series consists of prices of each second (after padding). Using prices of seconds to predict long-term prices will need extremely complex models. Therefore, we first sample the original spot prices by taking the maximum values for each hour. For building the MRS-AR model, the number of regimes should be determined first [18]. However, the number (and characteristics) of regimes changes as the time window moves (as time goes). Therefore, static MRS-AR with fixed parameters (such as a fixed number of regimes, fixed parameters for each AR part) is not suitable for predicting spot prices accurately. Therefore, different MRS-AR models are built for different time windows dynamically. After the MRS-AR model is obtained, spot prices are predicted based on different regime selection rules. This dynamic MRS-AR model based prediction framework is called DMRS-AR which consists of four steps as shown in Algorithm 1. In this paper, two regime selection rules (the lasting and the switching rules) are proposed which generates two prediction methods DMRS-AR-L and DMRS-AR-SW for short-term and long-term prediction separately. Details of these algorithms are shown in following sections.

5.1.1. Maximum value based sampling

Spot instances are charged in hours according to the initial spot price of each instance hour. If the spot price exceeds the bid in the middle of an instance hour, the spot instance is interrupted and the last partial instance hour is not charged. In other words, users usually care much about the maximum spot price in the next hour to maintain the bid price above the spot price. Using second-based data to predict prices of future hours, days or even weeks is time consuming. Therefore, spot prices are usually predicted in an hourly basis [2]. Prices are sampled by averaging prices of the same hour in Season-AR [2]. However, spot prices fluctuate nowadays much more than

```
Algorithm 1: DMRS-AR framework
Input: original prices of the current time window $\{Y_t\}_{t=1}^{T'}$, the number of prediction steps $F$
begin
    $\{Y_t\}_{t=1}^{T}$ ←Sample spot prices $\{Y_t\}_{t=1}^{T'}$ by taking the maximum price for each hour;
    Determine the number of regimes for the current time window by a clustering algorithm ;
    Establish the MRS-AR model based on $\{Y_t\}_{t=1}^{T}$ and the number of regimes;
    Predict spot prices according to the current MRS-AR and different regime selection rules;
    return Forecasted spot prices
```

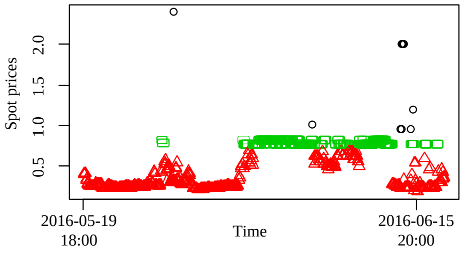
before according to the PDF of inter-price times. Average-based sampling will decrease the accuracy of predicting future maximum prices. Therefore, in this paper, the original spot prices $\{Y_t\}_{t=1}^{T'}$ are sampled by taking the maximum price for each hour which leads to a new time series $\{Y_t\}_{t=1}^{T}$. The maximum-value based sampling is helpful to predict the maximum price of future hours to avoid out-bid-events.

5.1.2. Clustering of spot prices

Determining the number of regimes is the basis to build MRS-AR. Spot prices can be divided into different regimes according to different mean values, variances and so on. Clustering algorithms are promising methods to distinguish regimes with different mean values. Many clustering algorithms need a predefined number of clusters and cannot recognize abnormal points. The density-based spatial clustering of applications with noise (DBSCAN) clustering algorithm [43] is able to determine the number of clusters itself according to the density of prices and recognize abnormal points. Therefore, the DBSCAN is chosen where "eps" (the maximum radius of the neighborhood) and "minPts" (the minimum number of points required to form a dense region) are two important parameters. Figure 2 shows the clustering results of DBSCAN for the spot prices of Amazon EC2 VM type "m4.4xlarge-us-east-1e". Spot prices are clustered into two types with some noises. Let x be the number of clusters according to the result of DBSCAN. The number of regimes of DMRS-AR is initialized by $k = x$. However, we found that spot prices usually have a main cluster (e.g., the cluster represented by triangles in Figure 2) which still have different fluctuations (different variances) in different time periods. Since building separate models for different regimes is helpful for improving the prediction accuracy, the main cluster is divided into two sub-clusters and k is updated by $k = x + 1$. For prices in Figure 2, the number of regimes of DMRS-AR is 3.

5.1.3. Establishing the MRS-AR model

The parameter set $\Theta_M = \{c_r, \phi_{i,r}, \sigma_r, P_{r,d}\}$ ($r = 1, 2, ...k$, $d = 1, 2, ...k$ and $i = 1, 2, ...p$) of MRS-AR is determined based on the given number of regimes where c_r is the mean value of the regime r, $\phi_{i,r}$ is the i-th AR coefficient of the regime r, σ_r is the standard deviation of the regime r and $P_{r,d}$ is the transition probability from regime r to d. MRS-AR is established by finding the parameter values which maximize the likelihood of given observations



(a)

Figure 2: Clustering results of DBSCAN for spot prices of Amazon EC2 virtual machine type ”m4.4xlarge-us-east-1e”

Table 1: The matrix of state transition probabilities

	Regime 1	**Regime 2**
Regime 1	0.93192626	0.070535
Regime 2	0.06807374	0.929465

($\{Y_t\}_{t=1}^{T}$) [15]. Expectation Maximization (EM) is an iterative method to find the maximum likelihood estimation of MRS-AR model parameters [15, 44]. During each iteration of EM, the conditional probability density of $s_t = r$ for a given observation Y_t (labeled by $P(s_t = r|Y_t, \Theta_M)$) is obtained based on the intermediate parameters Θ_M. $P(s_t = r|Y_t, \Theta_M)$ represents the probability density of s_t belonging to the state (regime) r for the given Y_t and Θ_M. The set of the conditional probabilities $P(s_t = r|Y_t, \Theta_M)$ $(r = 1, 2, ...k)$ forms a $(k \times 1)$ vector, labeled by $\xi_{t|t}$. In the last iteration, we get the final $\xi_{t|t}$ $(t = 1, 2, ..., T)$ and the final parameters Θ_M. Θ_M consists of AR parameters c_r and $\phi_{i,r}$ $(i = 1, 2, ...p)$ of each regime r and the matrix of state transition probabilities as shown in Table 1.

5.1.4. Prediction

The MRS-AR model divides spot prices into separate regimes which have different AR models. Selecting the right regime for each forecasted price is the basis to improve the prediction accuracy. In this paper, two methods have been proposed to choose regimes: (1) Choose the last regime as future regimes (called lasting rule) which results in a predicting method called DMRS-AR-L, (2) Use an autoregression and moving average model to predict the switching of regimes (called switching rule) which results in a spot price predicting method referred to as DMRS-AR-SW.

In the lasting rule, we assume that future spot prices have the same regime with the last spot price of the current time window. At first, the regime of the last spot price is determined based on $\xi_{t|t}$ $(t = 1, 2, ..., T)$. Let $P(s_T = z|Y_T, \Theta_M)$ be the maximum value of vector $\xi_{T|T}$ $(z = arg\max_{r=1,2,...,k}\{P(s_T = r|Y_T, \Theta_M)\})$, which means that the last spot price Y_T has the highest probability of belonging to regime z. According to the lasting rule, predicted spot prices are assumed to be belonging to the same regime z with Y_T. AR parameters (c_z and $\phi_{i,z}$,$i = 1, 2, ...p$) of regime z are used to predict spot prices as follows.

$$\widehat{Y}_T(l) = c_z + \sum_{i=1}^{p} \phi_{i,z}\widehat{Y}_T(l-i) \tag{7}$$

where $\widehat{Y}_T(w)$ is the $w-$th step predicted value when $w > 0$, $\widehat{Y}_T(w) = Y_{T+w}$ when $w \leq 0$. $\widehat{Y}_T(1)$, $\widehat{Y}_T(2)$, ..., $\widehat{Y}_T(F)$ are predicted one by one.

In the switching rule, an AR model is used to predict the regimes of future prices. As mentioned above, the regime of each Y_t can be determined based on the maximum element of $\xi_{t|t}$ $(t = 1, 2, ..., T)$. Each occurrence of a regime usually lasts for a time period which is called the duration of the regime.

When regimes switch as time goes, the durations of each regime will construct a time series (If one regime only lasts an hour, it is considered noise). For the example in Figure 3, there are two switching regimes. The periods of regime 1 have a gray background. The durations of regime 1 construct a time series $D_1 = (16, 14, 16, 54, 9, 6, 3, 2, 45, 10, 7)$ and the durations of regime 2 construct a time series $D_2 = (7, 5, 11, 11, 18, 13, 10, 33, 11, 7, 12)$. An ARMA model is proposed to predict future durations based on past durations for each regime. The orders of autoregression and moving average parts are all set to be 5 according to the autocorrelation function (ACF) and Partial autocorrelation function (PACF) of prices [15]. For each regime r, the predicted durations compose a queue Q_r . For example, the predicted duration queues are $Q_1 = (15, 16, 1, 38, 14, 8, 18)$ and $Q_2 = (6, 9, 12, 13, 19, 17, 18)$ for the regime 1 and 2 in Figure 3 separately. When predicting 1 to $F-$th step future prices, we assume that prices switch among different regimes and each regime extends over the predicted duration every time. In different regimes, different AR parameters are used to predict spot prices. At the beginning, we assume that the last price Y_t belongs to regime z which has lasted for E_z hours until the end of the current time window. The first element v of Q_z is ejected and regime z will still last for $\max\{v - E_z, 0\}$ hours. In the period of regime z, AR parameters of regime z are used to predict prices according to equation (7). When regime z finishes, the regime m with the maximum transition probability ($m = \arg\max_{d=R-\{z\}}\{P_{z,d}\}$) is chosen as the next regime. Then, the first element v of Q_m is ejected and regime m will continue for time v. In the period of regime m, AR parameters of regime m are used to predict prices. When regime m finishes,

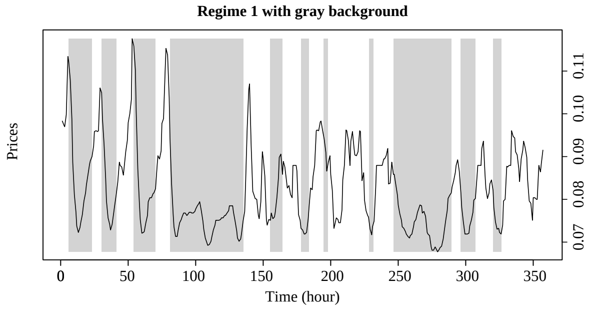


Figure 3: Time series of the duration of each regime

another regime will be chosen and the above process iterates until F prices are predicted. For the example in Figure 3, the last price belongs to regime 2 which has lasted for 30 hours until the end of the time window. However, the first predicted duration of regime 2 is only 6 hours (the first element ejected from Q_2) which is shorter than 30 hours. Therefore, regime 2 finishes and switches to regime 1 which will last for 15 hours (the first element ejected from Q_1). After the first 15 prices are predicted, the regime switches from 1 to 2 which will extend over 9 hours (the first element ejected from current Q_2). The process iterates until all prices are predicted.

5.1.5. Description of DMRS-AR-L and DMRS-AR-SW

The formal description of DMRS-AR-L is shown in Algorithm 2. At first, the original spot prices $\{Y_t\}_{t=1}^{T'}$ are sampled. Then, the clustering algorithm DBSCAN is used to determine the number of regimes represented by k. Next, MRS-AR model with $k+1$ regimes is built. Let $z \leftarrow \arg\max_{r=1,2,\ldots,k}\{P(s_T = r|Y_T, \Theta_M)\}$ be the regime of the last spot price. At last, AR parameters of regime z are used to predict F future spot prices iteratively according to equation (7).

Algorithm 2: DMRS-AR-L

```
Input: ${Y_t}_{t=1}^{T'}$, the number of prediction steps $F$
begin
    Initialize $l \leftarrow 1$;
    ${Y_t}_{t=1}^{T}$ $\leftarrow$Sample spot prices ${Y_t}_{t=1}^{T'}$ by taking the maximum price
      for each hour;
    Use DBSCAN to cluster ${Y_t}_{t=1}^{T}$ and get the cluster number $x$;
    $\Theta_M \leftarrow$ Establish MRS-AR model with $k = x + 1$ regimes;
    $z \leftarrow \arg\max_{r=1,2,...,k}\{P(s_T = r|Y_T, \Theta_M)\}$;
    while $l \leq F$ do
        Predict $\widehat{Y}_T(l)$ using the AR model of regime $z$ according to
          equation (7);
        $l \leftarrow l + 1$;
    return $\widehat{Y}_T(1), \widehat{Y}_T(2), ..., \widehat{Y}_T(F)$
```

Algorithm 3 is the formal description of DMRS-AR-SW. Similar with DMRS-AR-L, the original spot prices are sampled and DBSCAN is used to determine the number of regimes based on which MRS-AR model is established. Then, the regime of each price Y_t is assumed to be $\arg\max_{r=1,2,...,k}\{P(s_t = r|Y_t, \Theta_M)\}$ and the durations of each regime r construct a queue D_r. For each regime r, ARMA$(5, 5)$ is used to predict future durations making up a queue Q_r based on D_r. Next, the regime $z = \arg\max_{r=1,2,...,k}\{P(s_T = r|Y_T, \Theta_M)\}$ of the last spot price is selected as the first predicted regime which has lasted for E_z hours to the end of the current time window (T). v is the first element of Q_z and the current regime z will still last for $\max\{v - E_z, 0\}$ hours. u is the index of predicted price of the current regime. If $u \leq v$, the AR parameters of current regime z is used to predict price $\widehat{Y}_T(l)$. Otherwise, the current regime switches to $m = \arg\max_{d=R-\{z\}}\{P_{z,d}\}$ and the predicted duration v is updated to be equal to the first element of Q_m. Finally, $\widehat{Y}_T(1), \widehat{Y}_T(2), ..., \widehat{Y}_T(F)$ are predicted one by one using different regimes.

5.2. Dynamic-ARIMA prediction method (D-ARIMA)

Because differencing can improve the performance of ARMA models on fluctuating time series, ARIMA (ARMA with differencing) is also used to predict spot prices. Similar with the DMRS-AR framework, spot prices are also sampled first. In traditional ARIMA-based prediction methods, a uniform ARIMA model is established for different time windows based on which all prices are predicted. However, spot prices of different time windows have different statistical characteristics and establishing models for prices of different time windows separately is helpful to improve the prediction accuracy. Therefore, in this paper, different ARIMA models are established for each time window dynamically. The Maximum Likelihood Estimation (MLE) is one of the most important methods to estimate the parameters of ARIMA which finds the parameter values that maximize the likelihood of given observations [15]. For each time window, spot prices are predicted based on the established model as follows.

$$\begin{aligned}\widehat{Y}_T(l) - \widehat{Y}_T(l-1) = \theta_0 + \sum_{i=1}^{p} \phi_i(\widehat{Y}_T(l-i) - \widehat{Y}_T(l-i-1)) \\ + \sum_{j=1}^{q} \theta_j E[\varepsilon_{T-j+l}|Y_1, Y_2, ..., Y_T]\end{aligned} \tag{8}$$

where $\widehat{Y}_T(w)$ is the $w-$th step predicted value when $w > 0$, $\widehat{Y}_T(w) = Y_{T+w}$ when $w \leq 0$ and

$$E[\varepsilon_{T+m}|Y_1, Y_2, ..., Y_T] = \begin{cases} 0,\ m > 0 \\ \varepsilon_{T+m},\ m \leq 0 \end{cases}$$

The proposed dynamic-ARIMA (D-ARIMA) is formally described in Algorithm 4. The original spot prices are sampled first. In step 4, the ARIMA model is established by MLE based on sampled $\{Y_t\}_{t=1}^{T}$ and the parameter set of ARIMA $\Theta_A = \{\theta_0, (\phi_1, ..., \phi_p), (\theta_1, ..., \theta_q)\}$ is obtained. In step 5, F prices are predicted iteratively according to equation (8).

6. Performance Evaluation

In this paper, the performance of DMRS-AR-L, DMRS-AR-SW and D-ARIMA are evaluated on Amazon EC2 realistic spot prices. The proposals have been implemented in R and Java. The codes of the proposals are available at the website Github [45].

6.1. Spot instances

Amazon EC2 have many regions and the "us-east" region is one of the largest and popular regions. The proposals are evaluated on spot prices of three months from 28-04-2016 to 28-07-2016 obtained through the Amazon EC2 command line interface (CLI) from the "us-east" region. Spot instances of Amazon EC2 can be divided into three main types: computation-intensive, io-intensive and memory-intensive. Each main type consists of various VM types with diverse configurations (different operation systems, CPU speeds, memory sizes and IO speeds). In this paper, spot prices of 100 VM types are tested in total.

6.2. Compared algorithms and settings

As stated, only the MonthAR proposed by Singh et al. [2] considers the same spot price forecasting in the literature. The MonthAR is an AR model based prediction method which assumes that spot prices have month-based seasonal trends. However, we found that recent spot prices of many VM types do not have significant seasonal trend or the week-based seasonal trend is more significant than the month-based seasonal trend. The autocorrelation function (ACF) describes the similarity between observations as a function of the time lag between them which is a method to check the seasonality [15]. The ACF of Amazon EC2 VM type ”c4.2xlarge-us-east-1b-linux-unix” is shown in Figure 4 (the time lag is in hours), which shows significant week-based trend (higher ACF at the 168-th hour). Therefore, the MonthAR [2] is modified by taking week-based seasonality for a fair comparison, which forms a new method Week-AR. The proposals are compared with both MonthAR and WeekAR. The proposals are also compared with exponential smoothing methods: SES, DES and TES with week seasonality (called WeekES). In MonthAR, WeekAR, SES, DES and WeekES, the parameters of different time windows are determined dynamically too.

For a fair comparison, the autoregressive orders of D-ARIMA, DMRS-AR-L and DMRS-AR-SW are set to be 24 which is identical to the order of Season-AR [2]. The differencing order of the D-ARIMA takes 1 since the first difference of spot prices has been stable for most time periods. Because Season-AR has no moving average part, the order of the moving average part of D-ARIMA is also set to be 0. For the DBSCAN clustering algorithm, values of parameters ”eps” and ”minPts” are needed [43]. ”minPts”

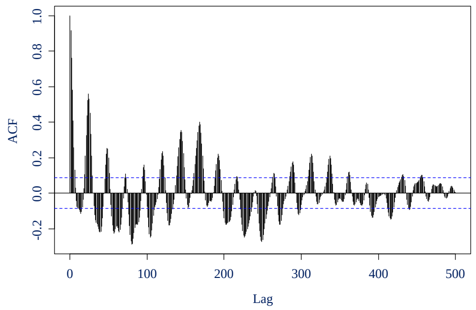


Figure 4: ACF of Amazon EC2 VM type "c4.2xlarge-us-east-1b-linux-unix"

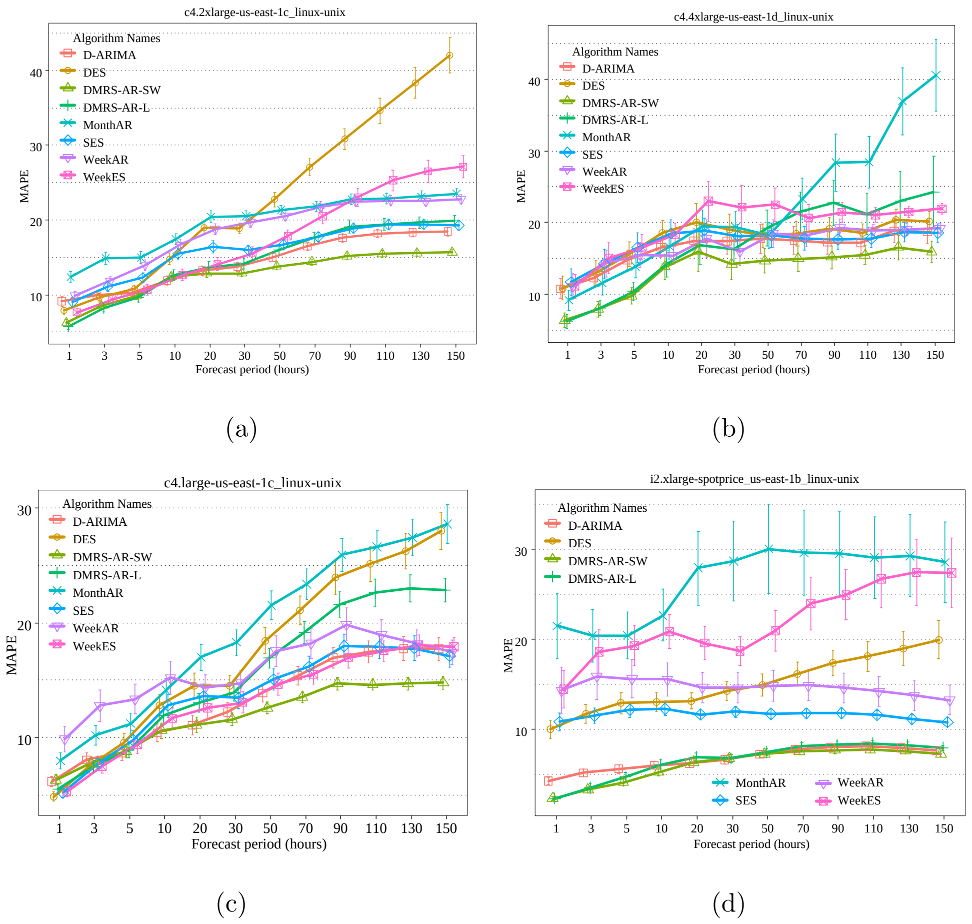
determines how many points are needed to construct a cluster. Because each regime regress on past 24 prices, the number of prices of each regime should be larger than 24. Therefore, "minPts" is equal to 24. Let d be the distance of a point p to its 4-th nearest neighbor (4-dist value). When sorting the points in descending order of their 4-dist values, the graph of the 4-dist values (called sorted 4-dist graph) gives some hints on how to determine the "eps" [43]. Usually, the 4-dist value of the first valley (threshold point) in the sorted 4-dist graph is adopted as the value of "eps" [43]. However, the valley is hard to be identified by computers automatically. According to the sorted 4-dist graph of spot prices, we found that the 4-dist value of the threshold point is approximately equal to one-eighth of the average price. Therefore, "eps" is set to be $\sum_{t=1}^{T} Y_t/(8 \times T)$ in this paper.

The length of time windows used to train the models has a great impact on the forecasting accuracy. We have evaluated the proposals with different window lengths taking values from $\{160, 320, 480, 640\}$ hours. Table 2 shows the *MAPE*s and computation times (seconds) of DMRS-AR-L with different window lengths on VM type "c4.2xlarge" and other results are not given here because of space limitation. Experimental results show that the *MAPE* decreases first and then increases as the length of windows increases. In other words, increasing the length of training windows is beneficial to improve the prediction accuracy while too long training windows decrease the accuracy on the contrary. At the same time, longer training windows make the proposals consume longer computation times. Therefore, 480 hours is chosen as the length of time windows which has the best performance and appropriate computation times. During the experiment, the window moves forward and $F = 168$ prices (one price for each hour) are predicted for each time window. When the forecast period n increases from 1 to F, a set $\{MAPE_n\}_{n=1}^{F}$ is obtained which consists of average prediction errors of different forecasting periods. For example, $MAPE_1$ represents the prediction error of the next hour, $MAPE_{24}$ is the average prediction error of the next 24 hours (next day) and $MAPE_{168}$ denotes the average prediction error of the next 168 hours (next week).

6.3. Experimental results

Spot prices of one hundred VM types have different statistical characteristics, such as means, trends, seasonality and linearity. According to these characteristics, VM types are categorized into five classes. Performances of forecast algorithms are evaluated on five classes of Spot prices. Because of the space limitation, the complete experimental results are available at the website Github [46].

The first class of Spot prices contains two types of linear processes and the switching among different processes are gentle. This class mainly consists of Spot prices of c4.2xlarge-1b, c4.2xlarge-1c, c4.2xlarge-1d, c4.4xlarge-1d, c4.8xlarge-1d, c4.8xlarge-1e, c4.xlarge-1b, c4.large-1b, c4.large-1c, c4.large-



(a)

(b)

(c)

(d)

Figure 5: Means plot of the Mean Absolute Percentage Error (%) with 95% confidence intervals on the first class of prices.

1d, c4.large-1, i2.xlarge-1b, i2.xlarge-1e, m4.xlarge-1b and m4.xlarge-1d linux-unix VM types. Figure 5 shows the experimental results on the first class of Spot prices which illustrate that DMRS-AR-SW has the smallest *MAPE* than the other algorithms for most cases. DMRS-AR-L has similar or smaller *MAPE* with DMRS-AR-SW when forecast period is smaller than about 24 hours, however, the accuracy of DMRS-AR-L decreases when the forecast period increases. D-ARIMA is a little worse than DMRS-AR-SW and better than other algorithms. These results are consistent with our expectations. The reason is that DMRS-AR algorithms describe two types of linear processes with different regimes respectively while D-ARIMA, WeekAR, WeekES and other prediction algorithms use a single model to describe the two different linear processes. DMRS-AR algorithms fits time series of each regime better than D-ARIMA and Season-AR. DMRS-AR-L uses the latest regime to predict spot prices which improves the prediction accuracy of short-term spot prices. When the forecast period increases, the regime changes which decreases the performance of DMRS-AR-L. However, DMRS-AR-SW tries to predict the switching of regimes and uses models of different regimes to predict accordingly which is helpful to improve the prediction accuracies of long forecast periods. Figure 6 shows an example of predicted prices of different forecast algorithms on the first Spot price class. The Spot price example have two regimes and DMRS-AR algorithms modeled them respectively. Figure 6 (a) illustrates that the trend of Spot prices is followed well by changing regimes appropriately. However, DMRS-AR-L use the last regime (the last regime has a greater variance than that of another regime) to predict prices as shown in Figure 6 (b). D-ARIMA buids a uniform ARIMA model with a biased mean value to describe two different regimes. WeekAR and MonthAR assume that Spot prices have week or month based seasonality. When the seasonality is not significant, week or month based seasonality may lead to unstable autoregreesion models as shown in Figure 6 (d) and (h). In Figure 6 (e) and (g), SES takes the last smoothed value as predicted prices without trends and DES assumes that the last trend of prices is continued. SES and DES cannot follow the changing of prices. For the WeekES in Figure 6 (f), the week based seasonality is well predicted, however, the predicted mean has a great deviation with the practice mean. The reason is that predicted prices are calculated based on the last smoothed value. When the last smoothed value is greatly different with the smoothed value of the same time at last week, there will be great deviation for all the predicted values.

The second class of prices is composed of a main type of linear processes and the remaining prices are non-linear such as Spot prices of c4.xlarge-1c, c4.xlarge-1d, m4.large-1d, m4.xlarge-1c and i2.xlarge-1c linux-unix VM types. *MAPE*s on this class of prices are shown in Figure 7 which illustrate that WeekAR, D-ARIMA and DMRS-AR-SW are better than other algorithms. However, these algorithms are only a little better than the SES with predicted prices on a horizontal line. The reason is as follows. Figure 8 shows an example of forecasted prices on the second class Spot prices. The prices contain linear increasing processes and non-linear decrease processes. Different linear increasing processes have similar autoregression correlations, which can be well described by an AR model of Markov-AR (the residual standard error is 0.0002547287). Since prices of the main cluster with similar means is only modeled by two regimes in DMRS-AR, the remaining different decreasing processes are described by a uniform linear regime. However, different decrease processes have different decreasing speeds and each decreasing process contains a sharp decrease which means that error items of different decreasing process have different probability distributions (the decreasing processes are non-linear). Therefore, the nonlinear decreasing processes are not well described by the AR model of DMRS-AR (The residual standard error is 0.002911596 ten times of 0.0002547287). Figure 8 (a) illustrates that the increasing processes are well predicted by DMRS-AR-SW, however, predicted decreasing speeds are lower than practice decreasing speeds in the decreasing processes leading to great violations for the long-term prediction. The short-term prediction of DMRS-AR-L is accurate as shown in Figure 8 (b). The increasing regime has an linear unstable AR model, therefore, the long-term prediction is unstable or even explosive. According to other subfigures of Figure 8, D-ARIMA, WeekAR and MonthAR usually converge on the means very quickly and D-ARIMA converges faster than WeekAR. D-ARIMA and WeekAR have a good prediction accuracy for the short-term prediction while MonthAR cannot follow the trends of prices correctly because of the absence of month seasonality for practice prices. Predicted prices of SES are on a horizontal line and WeekES copies the previous seasonality directly leading to great deviation when the seasonality is not significant.

The third class of Spot prices consists of more than two types of linear processes with similar means and different variance. For example, Spot prices of linux/unix type: c4.4xlarge-1c, c4.4xlarge-1e, c4.8xlarge-1c, c4.8xlarge-1d, m4.large-1b, m4.large-1e, and so on, belong to this class. Figure 9 shows *MAPE*s of this class of prices, which denotes that DMRS-AR-SW gets similar performance with SES and most of other algorithms are poorer than SES. The reason is that D-ARIMA, WeekAR, MonthAR and DMRS-AR with one or two regimes cannot describe more than two types of linear processes well. For the example in Figure 10, DMRS-AR-SW cannot follow the changes among more than two linear processes. Most of algorithms usually converge to means, leading to similar *MAPE* with SES.

The forth class of Spot prices have long linear processes with significantly different means. This class includes linux/unix VM types: c4.4xlarge-1b, i2.4xlarge-1e, i2.8xlarge-1e, c4.8xlarge-1b, etc. *MAPE*s of this class are shown in Figure 11 indicating that DMRS-AR-L and SES get the best prediction accuracy. DMRS-AR-L predicts prices using the last regime and each regime usually last a long time (longer than 10 hours), therefore, DMRS-AR-L get the best performance especially for short-term prediction. The transition among different regimes are not fixed, therefore, DMRS-AR-SW cannot predict the switching among regimes accurately on this class of prices as shown in Figure 12 (a). Figure 12 (b) illustrates that D-ARIMA builds a uniform model which converges to the mean of the whole time series. Spot prices of the last class contain many different short linear processes with different means. The prediction of switching regimes on the fifth class is complex than on the forth class as shown in Figure 13. All the proposed algorithms cannot get better performance.

Experimental results are also analyzed by the analysis of variance (ANOVA) method [47]. The three main hypotheses (normality, homoscedasticity and independence of the residuals) are checked. Numerical tests are usually very strict. For example, numerical tests will normally reject the hypothesis that the data comes from a normal distribution. Therefore, graphical tests are commonly used in practice. In this paper, normal QQ plots of residuals, residual plots vs. each factor level and dispersion plots of residuals over run numbers are used to test the three main hypotheses respectively.

For example, Figure 14 shows the normal QQ plots of residuals for the $MAPE$ of DMRS-AR-L on "c4.2xlarge-us-east-1c-linux-unix". According to these graphs, most points of QQ plots are near the straight line, different algorithms have similar variances and the residuals over run numbers are like white noises. Therefore, the three main hypotheses is acceptable. For each forecast period, ANOVA is performed to proof whether there are significant differences among different forecast algorithms. For the example on "c4.2xlarge-us-east-1c-linux-unix" with the forecast period equals to 90 hours, $p < 2e - 16$ means that there are significant difference among forecast algorithms. Then, Tukey multiple comparisons of means are used to recognize the difference between each pair of forecast algorithms. For the above example, differences of means with 95% family-wise Tukey confidence levels are shown in Figure 15 which illustrates that DMRS-AR-SW is significantly better than D-ARIMA, SES and so on. Details of three main hypotheses check, ANOVA results and Tukey multiple comparisons can be found at the website Github [46], which indicates similar results with means plots with 95% confidence intervals.

To summarize, DMRS-AR-L and DMRS-AR-SW predict prices accurately when the forecast period is shorter than about 24 hours for the first four classes. For example, the *MAPE* is smaller than about 10% when the forecast period is shorter than 5 hours and smaller than about 15% when the forecast period increases to 10 hours for most cases. As the forecast period increases, DMRS-AR-SW gets the best performance on the first class of prices. However, for other four classes of prices, more than two linear processes are modeled by a single linear regime or the switching of regimes cannot be predicted accurately. Therefore, DMRS-AR-SW cannot get significant better performance than SES with horizontal predicted prices. Although WeekAR get the lowest *MAPE* on the second class of prices, all algorithms cannot follow the changes well. D-ARIMA and WeekAR use a single model to describe different linear processes making the short-term prediction performance poorer than DMRS-AR-L. D-ARIMA and WeekAR usually converge too fast on the mean of total time series, therefore, they are not suitable for long-term prediction too. There is no significant month-seasonality for most Spot prices which produce great prediction deviations of MonthAR. D-ARIMA and Season-AR have an average computation time of 10 seconds which are much faster than DMRS-AR algorithms. The average computation times of DMRS-AR-SW and DMRS-AR-L are within 2 minutes which can still fulfill the time requirement compared with hour-based forecast periods. The above experimental results can be used to guide cloud users to choose appropriate spot price prediction methods taking account of the chosen VM types and application spans.

7. Conclusions and future works

In this paper, we found that spot prices usually have switching regimes and traditional ARMA models are not suitable their forecasting. Two Markov regime-switching autoregressive model based prediction methods, DMRS-AR-L and DMRS-AR-SW, have been proposed. They are compared with several different forecast methods. Experimental results show that DMRS-AR-L gets the best performance when the forecast period is shorter than about 24 hours for most cases. On the contrary, when the forecast period increases, DMRS-AR-SW gets the best performance for prices with less than three types of linear processes. The D-ARIMA and WeekAR usually converge too fast on means. Therefore, DMRS-AR-L is useful to guide the VM provisioning when the cloud application span is short (short-term provisioning). When the cloud application span increases (e.g., several days), we need to predict spot prices of next several days by DMRS-AR-SW (long-term VM provisioning).

According to the experimental results, all the compared algorithms cannot predict the long-term prices accurately for the second, third and fifth classes of prices. The reason is that there are more than two types of linear processes with similar means which cannot be recognized by clustering methods or the regime switching pattern is hard to be obtained. Therefore, designing much more appropriate forecast algorithms for the second, third or fifth classes of prices is desirable. After spot prices have been predicted by the proposals, designing algorithms to select appropriate VM types, rent the right intervals and set optimal bids are also promising future works.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No.61602243 and No.61572127), the Natural Science Foundation of Jiangsu Province (Grant No.BK20160846), Jiangsu Key Laboratory of Image and Video Understanding for Social Safety (Nanjing University of Science and Technology, Grant No. 30916014107) and the Fundamental Research Funds for the Central University (Grant No.30916015104). Rubén Ruiz is partially supported by the Spanish Ministry of Economy and

Competitiveness, under the project ”SCHEYARD” (No.DPI2015-65895-R) financed by FEDER funds.

References

[1] E. Dadashov, U. Cetintemel, T. Kraska, Putting analytics on the spot: Or how to lower the cost for analytics, IEEE Internet Computing 18 (18) (2014) 70–73.

[2] V. K. Singh, K. Dutta, Dynamic price prediction for Amazon spot instances, in: Hawaii International Conference on System Sciences, 2015, pp. 1513–1520.

[3] O. Agmon Ben-Yehuda, M. Ben-Yehuda, A. Schuster, D. Tsafrir, Deconstructing Amazon EC2 spot instance pricing, in: IEEE International Conference on Cloud Computing Technology and Science Proceedings, 2011, pp. 304–311.

[4] B. Javadi, R. K. Thulasiramy, R. Buyya, Statistical modeling of spot instance prices in public cloud environments, in: IEEE International Conference on Utility and Cloud Computing, 2011, pp. 219–228.

[5] O. Agmon Ben-Yehuda, M. Ben-Yehuda, A. Schuster, D. Tsafrir, Deconstructing Amazon EC2 spot instance pricing, ACM Transactions on Economics & Computation 1 (3) (2011) 304–311.

[6] S. Tang, J. Yuan, X. Y. Li, Towards optimal bidding strategy for Amazon EC2 cloud spot instance, in: IEEE International Conference on Cloud Computing, 2012, pp. 91–98.

[7] Y. Song, M. Zafer, K. W. Lee, Optimal bidding in spot instance market, in: IEEE INFOCOM, 2012, pp. 190–198.

[8] M. Zafer, Y. Song, K. W. Lee, Optimal bids for spot VMs in a cloud for deadline constrained jobs, in: IEEE Sixth International Conference on Cloud Computing, 2012, pp. 75–82.

[9] L. Zheng, C. Joe-Wong, C. W. Tan, M. Chiang, X. Wang, How to bid the cloud, ACM Sigcomm Computer Communication Review 45 (5) (2015) 71–84.

[10] S. Yi, A. Andrzejak, D. Kondo, Monetary cost-aware checkpointing and migration on Amazon cloud spot instances, IEEE Transactions on Services Computing 5 (4) (2012) 512–524.

[11] S. Yi, D. Kondo, A. Andrzejak, Reducing costs of spot instances via checkpointing in the amazon elastic compute cloud, in: IEEE International Conference on Cloud Computing, 2010, pp. 236–243.

[12] I. Jangjaimon, N. F. Tzeng, Effective cost reduction for elastic clouds under spot instance pricing through adaptive checkpointing, IEEE Transactions on Computers 64 (2) (2015) 396–409.

[13] N. Chohan, C. Castillo, M. Spreitzer, M. Steinder, A. Tantawi, C. Krintz, See spot run: using spot instances for mapreduce workflows, in: Usenix Conference on Hot Topics in Cloud Computing, 2010, pp. 1–7.

[14] M. Abundo, V. D. Valerio, V. Cardellini, F. L. Presti, QoS-aware bidding strategies for vm spot instances: A reinforcement learning approach

applied to periodic long running jobs, in: Ifip/IEEE International Symposium on Integrated Network Management, 2015, pp. 53–61.

[15] J. D. Hamilton, Time Series Analysis, 1994, Princeton University Press.

[16] C. C. Holt, Forecasting seasonals and trends by exponentially weighted moving averages, International Journal of Forecasting 20 (1) (2004) 5–10.

[17] H. Tong, K. S. Lim, Threshold autoregression, limit cycles and cyclical data, Journal of the Royal Statistical Society 42 (3) (1980) 245–292.

[18] J. D. Hamilton, A new approach to the economic analysis of nonstationary time series and the business cycle., Econometrica 57 (2) (1989) 357–84.

[19] R. Chen, A. Langnau, Turning points detection of business cycles: A model comparison, Social Science Electronic Publishing.

[20] M. Fornaciari, C. Grillenzoni, Evaluation of on-line trading systems: Markov- switching vs time-varying parameter models, Decision Support Systems 93 (2016) 51–61.

[21] C. Yuan, Forecasting exchange rates: The multi-state markov-switching model with smoothing, International Review of Economics and Finance 20 (2) (2011) 342–362.

[22] R. V. D. Bossche, K. Vanmechelen, J. Broeckhove, Online cost-efficient scheduling of deadline-constrained workloads on hybrid clouds, Future Generation Computer Systems 29 (4) (2013) 973–985.

[23] X. Zuo, G. Zhang, W. Tan, Self-adaptive learning pso-based deadline constrained task scheduling for hybrid iaas cloud, IEEE Transactions on Automation Science & Engineering 11 (2) (2014) 564–573.

[24] Y. Wang, W. Shi, Budget-driven scheduling algorithms for batches of mapreduce jobs in heterogeneous clouds, IEEE Transactions on Cloud Computing 2 (3) (2014) 306–319.

[25] E. K. Byun, Y. S. Kee, J. S. Kim, S. Maeng, Cost optimized provisioning of elastic resources for application workflows, Future Generation Computer Systems 27 (8) (2011) 1011–1026.

[26] S. Abrishami, M. Naghibzadeh, D. H. J. Epema, Deadline-constrained workflow scheduling algorithms for infrastructure as a service clouds, Future Generation Computer Systems 29 (1) (2013) 158–169.

[27] M. Malawski, G. Juve, E. Deelman, J. Nabrzyski, Algorithms for cost- and deadline-constrained provisioning for scientific workflow ensembles in iaas clouds, Future Generation Computer Systems 48 (1) (2015) 1–18.

[28] Z. Cai, X. Li, R. Ruiz, Q. Li, A delay-based dynamic scheduling algorithm for bag-of-task workflows with stochastic task execution times in clouds, Future Generation Computer Systems 71 (2017) 57–72.

[29] Q. Zhang, Q. Zhu, R. Boutaba, Dynamic resource allocation for spot markets in cloud computing environments, in: IEEE International Conference on Utility & Cloud Computing, 2011, pp. 178–185.

[30] K. Vanmechelen, W. Depoorter, J. Broeckhove, Combining futures and

spot markets: A hybrid market approach to economic grid resource management, Journal of Grid Computing 9 (1) (2011) 81–94.

[31] N. Sadashiv, S. M. D. Kumar, R. S. Goudar, Hybrid spot instance based resource provisioning strategy in dynamic cloud environment, in: International Conference on High Performance Computing and Applications, 2014, pp. 1 – 6.

[32] C. Qu, R. N. Calheiros, R. Buyya, A reliable and cost-efficient auto-scaling system for web applications using heterogeneous spot instances, Journal of Network and Computer Applications 65 (2016) 167–180.

[33] H. Huang, L. Wang, B. C. Tak, L. Wang, Cap3: A cloud auto-provisioning framework for parallel processing using on-demand and spot instances, in: IEEE Sixth International Conference on Cloud Computing, 2013, pp. 228–235.

[34] D. Jung, J. B. Lim, H. Yu, A workflow scheduling technique to consider task processing rate in spot instance-based cloud, Lecture Notes in Electrical Engineering 301 (2014) 483–494.

[35] J. Li, S. Su, X. Cheng, M. Song, L. Ma, J. Wang, Cost-efficient coordinated scheduling for leasing cloud resources on hybrid workloads, Parallel Computing 44 (C) (2015) 1–17.

[36] A. Vintila, A. M. Oprescu, T. Kielmann, Fast (re-) configuration of mixed on-demand and spot instance pools for high-throughput computing, in: ORMaCloud, 2013, pp. 25–32.

[37] D. Poola, K. Ramamohanarao, R. Buyya, Fault-tolerant workflow scheduling using spot instances on clouds, Procedia Computer Science 29 (2014) 523–533.

[38] D. Poola, K. Ramamohanarao, R. Buyya, Enhancing reliability of workflow execution using task replication and spot instances, ACM Transactions on Autonomous & Adaptive Systems 10 (4) (2016) 1–21.

[39] A. Andrzejak, D. Kondo, S. Yi, Decision model for cloud computing under sla constraints, in: IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2010, pp. 257–266.

[40] H. Mostafaei, M. Safaei, Point forecast markov switching model for US Dollar/Euro Exchange Rate, Sains Malaysiana 41 (4) (2012) 481–488.

[41] J. S. Armstrong, F. Collopy, Error measures for generalizing about forecasting methods: Empirical comparisons, International Journal of Forecasting 8 (1) (1992) 69–80.

[42] R. J. Hyndman, A. B. Koehler, Another look at measures of forecast accuracy, International Journal of Forecasting 22 (4) (2006) 679–688.

[43] M. Ester, H. P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise, in: International Conference on Knowledge Discovery and Data Mining, 1996, pp. 226–231.

[44] A. P. Dempster, N. M. Laird, D. B. Rubin, Maximum likelihood from

incomplete data via the em algorithm, Journal of the Royal Statistical Society 39 (1) (1977) 1–38.

[45] Z. Cai, X. Li, R. Ruiz, Q. Li, Source codes of D-ARIMA, Season-AR, DMRS-AR-SW and DMRS-AR-L algorithms, `https://github.com/czcnjust/ElasticSim/wiki/Spotpriceprediction-code`, accessed, 2017.5.11.

[46] Z. Cai, X. Li, R. Ruiz, Q. Li, Experimental results of D-ARIMA, Season-AR, DMRS-AR-SW and DMRS-AR-L algorithms on spot prices, `https://github.com/czcnjust/ElasticSim/blob/master/SpotpriceresultsAttach.pdf`, accessed, 2017.5.11.

[47] T. Bartz-Beielstein, M. Chiarandini, L. Paquete, M. Preuss, Experimental methods for the analysis of optimization algorithms, Springer, 2010.

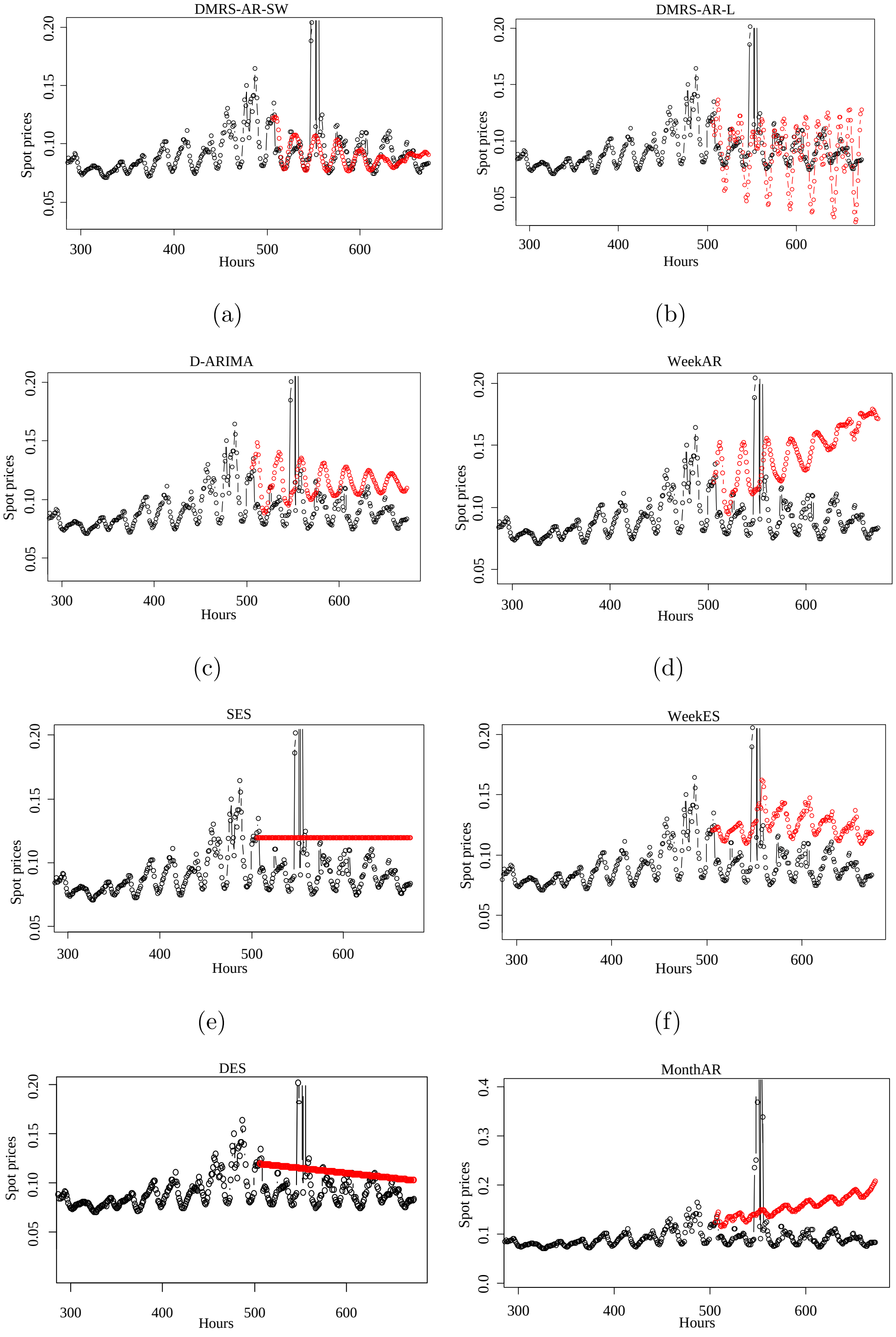
Algorithm 3: DMRS-AR-SW

```
Input: ${Y_t}_{t=1}^{T'}$, the number of prediction steps $F$
begin
    Initialize $l \leftarrow 1$;
    $\{Y_t\}_{t=1}^{T} \leftarrow$ Sample spot prices $\{Y_t\}_{t=1}^{T'}$ by taking the maximum
      price for each hour;
    Use DBSCAN to cluster $\{Y_t\}_{t=1}^{T}$ and get the cluster number $x$;
    $\Theta_M \leftarrow$ Establish the MRS-AR model with $k = x + 1$ regimes;
    The regime of each price $Y_t$ is assumed to be
      $\arg\max_{r=1,2,...,k}\{P(s_t = r|Y_t, \Theta_M)\}$;
    Construct duration queue $D_r$ for each regime $r$;
    Use ARMA$(5, 5)$ to predict future duration queue $Q_r$ for each
      regime $r$ based on $D_r$;
    Initialize the current regime $z$ to be the regime of the last spot
      price;
    $v \leftarrow$ Eject$(Q_z)$;
    Update $v \leftarrow \max\{v - E_z, 0\}$ and $u \leftarrow 0$;
    while $l \leq F$ do
        $u \leftarrow u + 1$;
        if $u \leq v$ then
            Predict $\widehat{Y}_T(l)$ using the AR model of regime $z$ according to
              equation (7);
        else
            Switch the current regime to $m = \arg\max_{d=R-\{z\}}\{P_{z,d}\}$;
            Update $z \leftarrow m$, $u \leftarrow 0$;
            $v \leftarrow$ Eject$(Q_z)$;
        $l \leftarrow l + 1$;
    return $\widehat{Y}_T(1), \widehat{Y}_T(2), ..., \widehat{Y}_T(F)$
```

```
Algorithm 4: D-ARIMA
Input: $\{Y_t\}_{t=1}^{T'}$, the number of prediction steps $F$
begin
    Initialize $l \leftarrow 1$;
    $\{Y_t\}_{t=1}^{T} \leftarrow$Sample spot prices $\{Y_t\}_{t=1}^{T'}$ by taking the maximum price
      for each hour;
    Establish ARIMA based on $\{Y_t\}_{t=1}^{T}$ and get parameter set
      $\Theta_A = \{\theta_0, (\phi_1, ..., \phi_p), (\theta_1, ..., \theta_q)\}$;
    while $l \leq F$ do
        Predict $\widehat{Y}_T(l)$ based on $\Theta_A$ according to equation (8);
        $l \leftarrow l + 1$;
    return $\widehat{Y}_T(1), \widehat{Y}_T(2), ..., \widehat{Y}_T(F)$
```

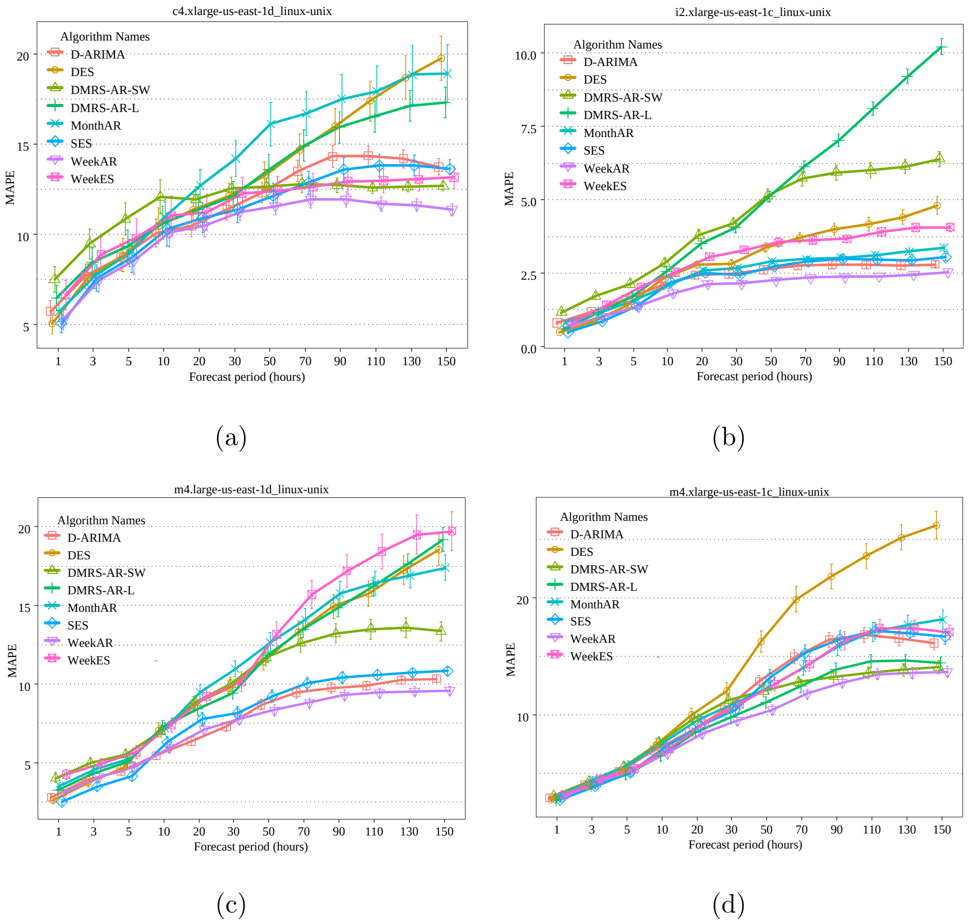
Table 2: MAPE of different time window lengths

window length	**computation time**	**MAPE of 12 hours**	**MAPE of 24 hours**
160h	20s	5.46%	6.59%
320h	51s	5.72%	5.44%
480h	73s	5.3 %	5.40%
640h	85s	7.6 %	6.0%



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 6: An example of predicted prices of different forecast algorithms on the first Spot price class.



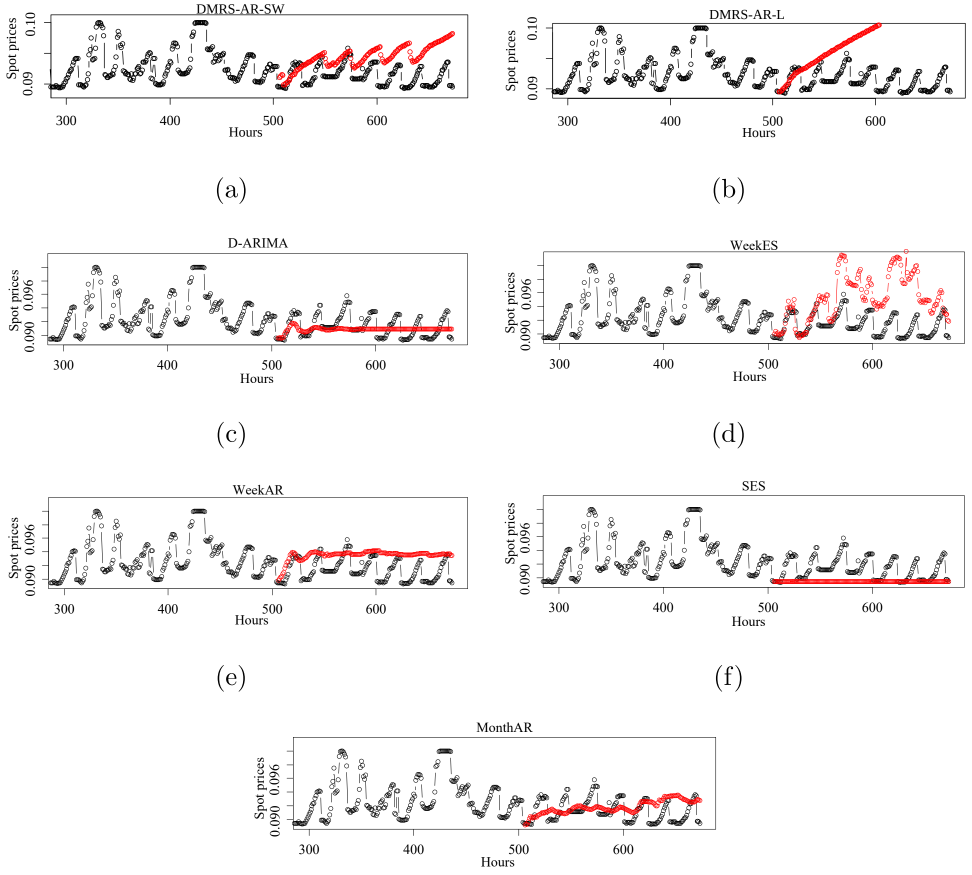
(a)

(b)

(c)

(d)

Figure 7: Means plot of the Mean Absolute Percentage Error (%) with 95% confidence intervals on the second class of prices.



(a)

(b)

(c)

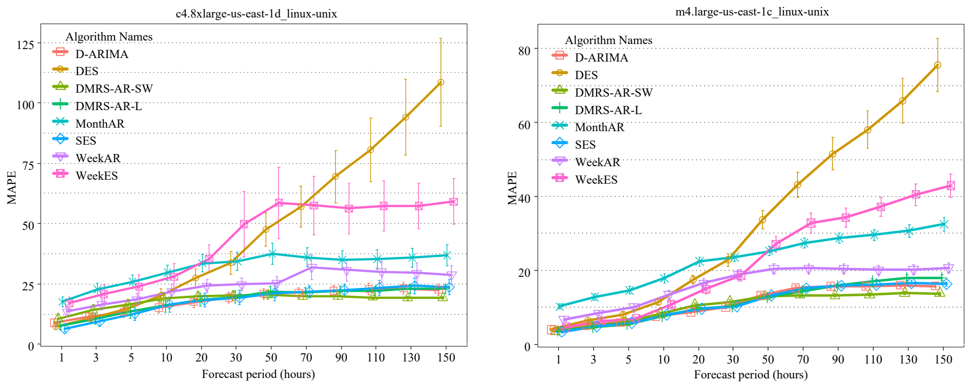
(d)

(e)

(f)

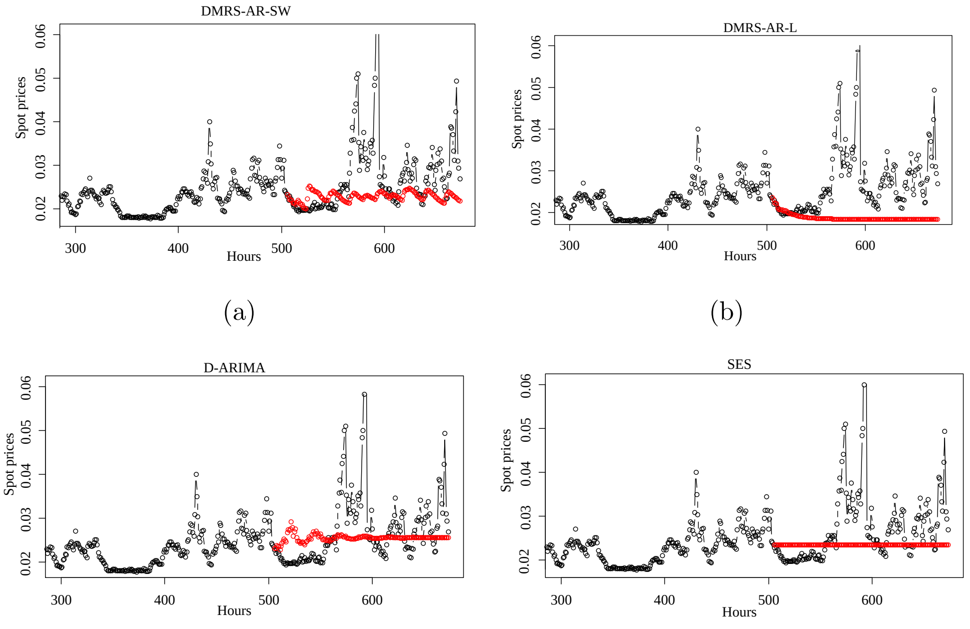
(g)

Figure 8: Predicted prices of different forecast algorithms on the second class of prices.



(a) (b)

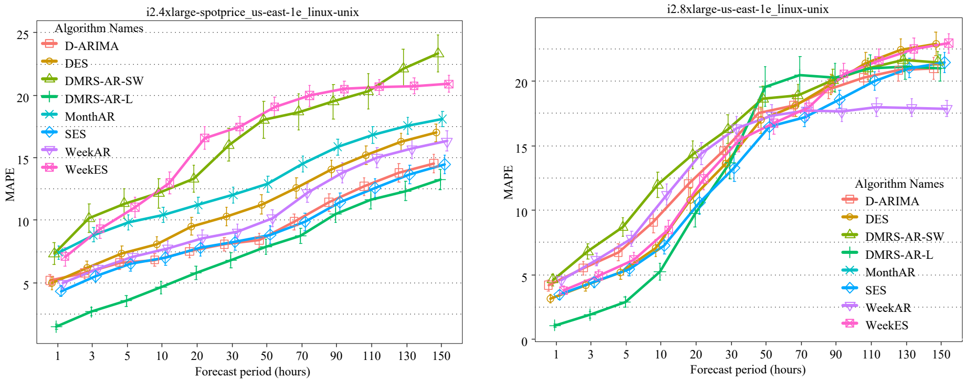
Figure 9: Means plot of the Mean Absolute Percentage Error (%) with 95% confidence intervals on the third class of prices.



(a) (b)

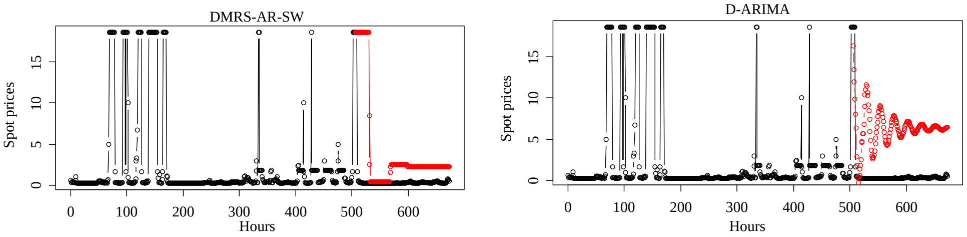
(c) (d)

Figure 10: Predicted prices of different forecast algorithms on the third class of prices.



(a) (b)

Figure 11: Means plot of the Mean Absolute Percentage Error (%) with 95% confidence intervals on the forth class of prices.



(a) (b)

Figure 12: Predicted prices of different forecast algorithms on the forth class of prices.

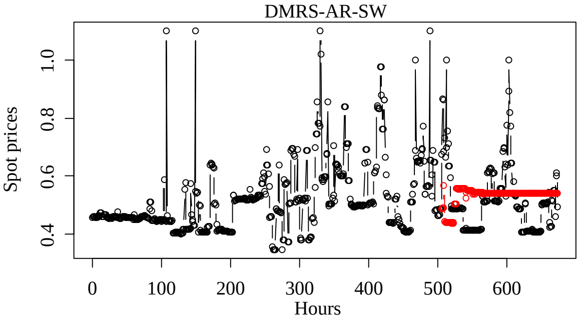


Figure 13: Predicted prices of different forecast algorithms on the fifth class of prices.

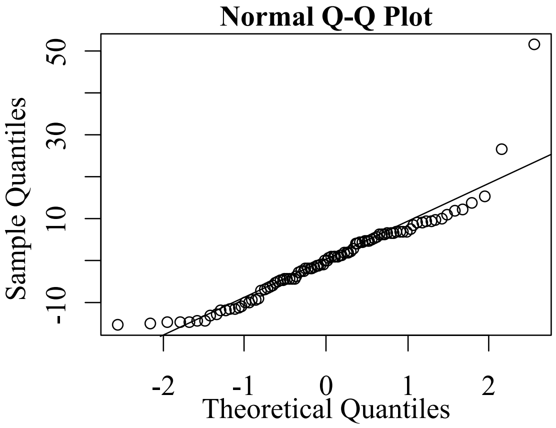


Figure 14: Normal QQ plots of residuals for the $MAPE$ of DMRS-AR-L on "c4.2xlarge-us-east-1c-linux-unix" with the forecast period equals to 90 hours.

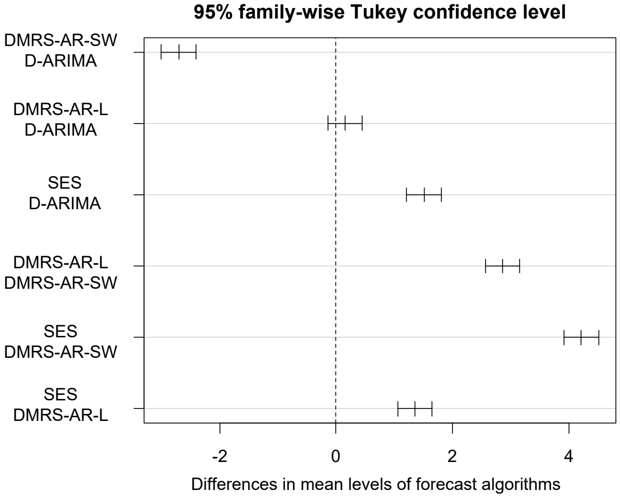


Figure 15: Differences of means with 95% family-wise Tukey confidence levels on "c4.2xlarge-us-east-1c-linux-unix" with forecast period of 90 hours.