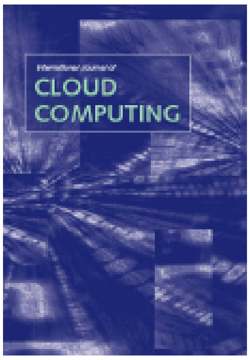
INDERSCIENCE PUBLISHERS
Linking academia, business and industry through research



International Journal of Cloud Computing

ISSN online: 2043-9997 - ISSN print: 2043-9989
https://www.inderscience.com/ijcc

Amazon EC2 spot price prediction using LSTM time series prediction model

Khandelwal Veena, Khandelwal Shantanu

DOI: 10.1504/IJCC.2024.10049932

Article History:

Received:	18 January 2022
Accepted:	04 April 2022
Published online:	26 January 2024

Amazon EC2 spot price prediction using LSTM time series prediction model

Khandelwal Veena*

Manipal University,
Jaipur, India
Email: vn.khandelwal@gmail.com
*Corresponding author

Khandelwal Shantanu

KPMG Services Pte. Ltd, Singapore
Email: shantanukhandelwal@protonmail.com

Abstract: Amazon EC2 spot instances provide access to unused Amazon EC2 capacity at high discounts relative to on-demand and reserved prices. Spot prices fluctuate based on the demand and supply of available unused capacity of EC2. When users request spot instances, they specify the maximum spot price they are willing to pay. Optimum maximum spot price estimation is crucial to control costs and have uninterrupted access to spot instances. We analyse spot price fluctuations for any seasonal or residual component and present a stacked LSTM-based prediction model based on the deep learning RNN model. In order to analyse Amazon spot pricing, we use time-smoothed spot prices at frequency of one hour. Our experiments with the new Amazon EC2 spot pricing model show that the LSTM model predicts future spot prices with different lead times with very low RMSE values.

Keywords: Amazon EC2; compute instances; new spot pricing model; spot price prediction; long short-term memory; LSTM.

Reference to this paper should be made as follows: Veena, K. and Shantanu, K. (2024) ‘Amazon EC2 spot price prediction using LSTM time series prediction model’, *Int. J. Cloud Computing*, Vol. 13, No. 1, pp.62–79.

Biographical notes: Khandelwal Veena is an Assistant Professor at the SRMIST, Delhi NCR, Ghaziabad Campus. She received her MTech in Computer Science from Rajasthan Technical University in 2014. She received her PhD in Computer Science from the Rajasthan Technical University in 2019. Her current research interests include cloud computing, cost optimisation, data availability and data security in cloud computing.

Khandelwal Shantanu is a Manager at the KPMG Services Pte. Ltd, Singapore. He has a Master’s in Cyber Security and Incident Response. He specialises in security assessments, ATM penetration testing, vulnerability assessment and penetration testing across web applications, mobile applications and infrastructure domains. He has worked with many international banks and financial institutions through his career.

1 Introduction and motivation

Amazon web services provides several kinds of cloud services to its customers. These services are made available to the customers at different pricing plans. Cloud service providers invest in hardware and software resources to offer these resources to their customers. IaaS is one of the several cloud services offered by Amazon. Since the cloud resources are requested in an on-demand fashion, cloud service providers might resort to over-estimating their customer's peek demands to fulfill the SLA requirements, which leads to very low utilisation of their resources as they stay idle during off-peak periods. Rather than wasting spare compute capacity, Amazon offers this spare compute capacity to the customers at steep discounts. Customers use spot instances for reducing costs and get faster results without any long-term commitments. As spot instances are prone to interruption, they are suitable for executing batch jobs, fault-tolerant jobs, and for jobs that do not have any deadline constraint. In November 2017, Amazon launched its new pricing policy. The new spot pricing model simplified the spot instance purchase experience. With the new model, prices are more predictable (Pary, 2018). Prices are now no more decided by the bid prices but are set by the market demand and supply. Amazon marketing people run the show. Users can set the maximum price they are willing to pay. Spot instances continue to run until the cost of the spot instances exceeds the maximum price set by the user. The default maximum price is on-demand price. Spot price fluctuations are market-driven due to which spot instances have low reliability and a high risk of instances being reclaimed by Amazon EC2. Due to these challenges in spot instance acquisition, cost reduction benefits cannot attract ample customers.

For customers with strict cost constraints, determining the maximum spot price in a spot pool can significantly control execution cost and prevent instance termination. We study the new spot pricing policy intending to control and reduce execution costs of using spot instances. If spot prices can be accurately predicted in advance, it will help users set the maximum price they are willing to pay and select availability zone and region, for instance, selection.

The paper analyses spot history to study any seasonal trends and variations in spot prices after the introduction of new spot pricing policy. We also predict spot prices for lead times one-hour, two-hours, four-hours and one-day. Spot price prediction would assist the user to set the maximum price of spot instances depending on the length of time for which spot instances are required. This paper proposes a time series-based spot price prediction model using a sliding window that captures past spot prices in any AWS region and availability zone. Our proposed model will help many cloud users to understand spot pricing, set the maximum spot price value and control execution costs. Spot price prediction will help cloud clients predict how long their spot instances would be able to run. In summary, we have made the following contributions:

1. According to Amazon new spot pricing policy, spot prices are nearly stationary. We articulate that spot prices are not stationary in most of the regions and availability zones. We show that there is a minimal seasonal and residual component. The majority of spot price change is due to nonlinear trends in spot prices.
2. To the best of our knowledge, this is the first attempt to analyse spot history after the new spot pricing policy in November 2017.

3 We perform re-sampling on spot history time-series data into the hourly frequency to analyse data and draw additional insights.

4 Machine learning methods are not very good at predicting time series data. We propose a deep learning approach based on the celebrated long short-term memory (LSTM) networks, which are unique kinds of recurrent neural networks (RNNs), to predict future spot prices using sequences.

5 We exhibit that our spot price prediction model can predict future spot prices with sufficient accuracy.

The rest of this paper is listed as follows. Section 2 introduces some related works. Section 3 provides methodology and analysis of the spot instance pricing history data and describes the price data distribution. Section 4 describes the model and algorithm that we used to predict the price. Section 5 presents our experimental environment and our measurement methodology and discusses the results of our experiments. Section 6 concludes the paper.

2 Literature review

Several previous works focus on Amazon EC2 spot price modelling and prediction. Lucas-Simarro et al. (2015) use simple moving average for predicting next hour spot price. Reverse engineering is performed by Ben-Yehuda et al. (2013) to predict spot price. Empirical distributions models for spot price prediction are proposed by Andrzejak et al. (2010), Mazzucco and Dumas (2011), Javadi et al. (2013) and Zhao et al. (2012). Markovian models for spot price prediction is proposed by Chohan et al. (2010). Singh and Dutta (2015) take global (monthly and daily) and seasonality trends into account and assume month and hour as important predictor variables. Hassan and Hammad (2020) propose a framework to mitigate spot instance reclaiming risk and achieve cost reduction by monitoring Amazon EC2 markets and hopping instances between these markets. Instance migration involves an extra cost of saving the state of the job and migrating to another market.

Several works focus on using machine learning and ensemble methods for solving prediction problems of various applications. Khandelwal et al. (2020, 2018) analyse spot price history from April 2015 to March 2016 and predict spot prices up to one week ahead using random forests. Authors also propose a bidding strategy based on spot price prediction using random forests to set bid time and bid price depending on the job length. Shorter the job length, the smaller the bid price. Baughman et al. (2018) predict spot price using the LSTM approach. Authors use spot history data from 3 September 2016 and 10 September 2016. With the LSTM approach, authors predict spot price with root mean square error (RMSE) 0.423. Portella et al. (2018) analyse the Amazon EC2 spot pricing model by using the time-smoothed moving averages method. The authors study spot history data from September to November 2016. To increase availability up to 90%, authors set the bid price up to 30% of the on-demand price.

Alourani and Kshemkalyani (2020) propose providing reliability to spot instances using docker containers for checkpointing and restoring container images, thereby eliminating the need for fault-tolerant mechanisms. Chhetri et al. (2017) decompose the observed spot prices from 2 July 2016 and 19 October 2016, into a seasonal, trend, and residual component using seasonal and trend decomposition using loss (STL) and find a strong seasonal component in spot prices. The authors find the seasonal naive method to be most robust for spot price prediction. Chittora and Gupta (2020) use a two layers stacked LSTM model for spot price prediction purposes. The authors do not specify the period of the dataset. With their approach RMSE value varies from 0.04 to 0.20.

The new spot pricing model has also been studied by several authors. Al-Theiabat et al. (2018) propose LSTM approach for spot price prediction. LSTM approach is compared with the ARIMA model, using different accuracy measures commonly used in TSA. Authors use spot history data from December 2017 to March 2018. Authors sample data at a four-hour frequency. Kong et al. (2021) use the k-adaptive mean square error (k-AMSE) approach to evaluate the volatility of price data. The authors present a spot price prediction model based on the gated recurrent unit (GRU) network. GRU networks solve the problem of vanishing gradient but have slow convergence and low learning efficiency. The spot price history data used is from 28 August 2018 to 24 November 2018. GRU networks maintain a longer-term information dependence.

We experimented with different amounts of history for predicting the next 24 hours spot price. Our experiments show that when time sequences are used for a higher duration, the prediction error increases. Most of the works that predict spot price are based on spot history before November 2017. Only a few works use data after the new spot pricing model to predict the next hour price. The new model does not require the user to bid for spot instances but to set the maximum price he is willing to pay. Secondly, this work performs preprocessing such as re-sampling and normalisation to spot history data for better prediction, which lowers RMSE to a great extent. The model proposed predicts spot prices at the lead times ranging from 1 hour to 24 hours. During our analysis of recent spot history traces from June to September 2021, we find negligible seasonal and residual components, thereby eliminating the need for decomposing the observed spot prices into these components and run time series forecasting for each component separately.

3 LSTM model structure

3.1 Artificial neural network

Artificial neural networks (ANNs) are computing systems inspired by the human nervous system. They have a massively interconnected and parallel processing architecture that can solve problems through machine learning neurons. ANNs can identify complex nonlinear relationships between inputs and outputs without inputting direct knowledge of the physical processes. Neural networks used in deep learning consist of different layers connected and work on the structure and functions of a human brain. It learns from vast volumes of data and uses complex algorithms to train a neural network. An ANN consists of an input layer, one or more hidden layers and an output layer.

Several types of ANN exist, and each type is used to solve a particular class of problems. Each ANN involves a different network architecture, a different number of hidden layers and a different number of neurons in each layer. In general, feedforward neural networks are suitable for general regression and classification problems. Convolutional neural networks can be used for image recognition. RNNs are used for speech recognition, image captioning, and input translation into other languages. Text mining and sentiment analysis can be performed using RNN for natural language processing.

In a feedforward network, the information flows only in a forward direction from the input nodes, through the hidden layers and then to the output nodes. There are no cycles or loops in the network. Such networks do not require memorising the past output. Based on gradients of the loss function, the error propagates backwards, and the weights are updated to optimise the loss function. The gradients become smaller during backpropagation. Learning in the earlier layers is very slow and results in high training time problems. Apart from this, there are other several issues in feedforward networks. Feed-forward networks cannot handle sequential data properly as they consider only the current input and cannot memorise previous inputs. Approximations done with linear models are inadequate for time series problems due to the dynamics of nonlinear behaviour.

3.2 The solution to feed-forward networks – RNNs

RNNs can memorise what is going on in the hidden layers. The hidden layers as they produce data, feed into the next one, so the hidden layers might have an output that goes off to the next layer or output Y, but that output also goes back into the next prediction coming in. This allows RNN to handle sequential data. It considers the current input and also the previously received inputs. Any time series problem like predicting the stock prices in a particular month can be solved using RNN. The architecture of a RNN consists of an input layer, one or more hidden layers and an output layer. RNN‘s consist of repeating modules that serve as a memory to store important information from the previous step. RNNs also include a feedback loop that allows them to accept a sequence of inputs. RNNs work on the principle of saving output of a layer and feeding this back to the input in order to predict the output of the layer. While training an RNN, the slope can be either too small or very large, making the training very difficult. When the slope is too small, the problem is called vanishing gradient. When the slope tends to grow exponentially instead of decaying, this problem is called exploding gradient. Gradient problems have several issues such as long training times, poor performance and lousy accuracy. A solution to the gradient problem is required because it might sometimes be difficult for the error to backpropagate to the beginning of the sequence to predict the output.

LSTM networks are a solution to the vanishing gradient problem. They can clip the gradient as it comes out and also expand it. We can increase the size of the memory network to handle more information. LSTMs are a special kind of RNNs capable of learning long-term dependencies. Remembering information for long periods is their default behaviour. All RNNs have the form of a chain of repeating modules of a neural network. In standard RNN, this repeating module will have a straightforward structure, such as a single tanh layer. LSTMs also have a chain-like structure, but the repeating module has a different structure. Instead of having a single neural network layer, four interacting layers are communicating in a very special way.

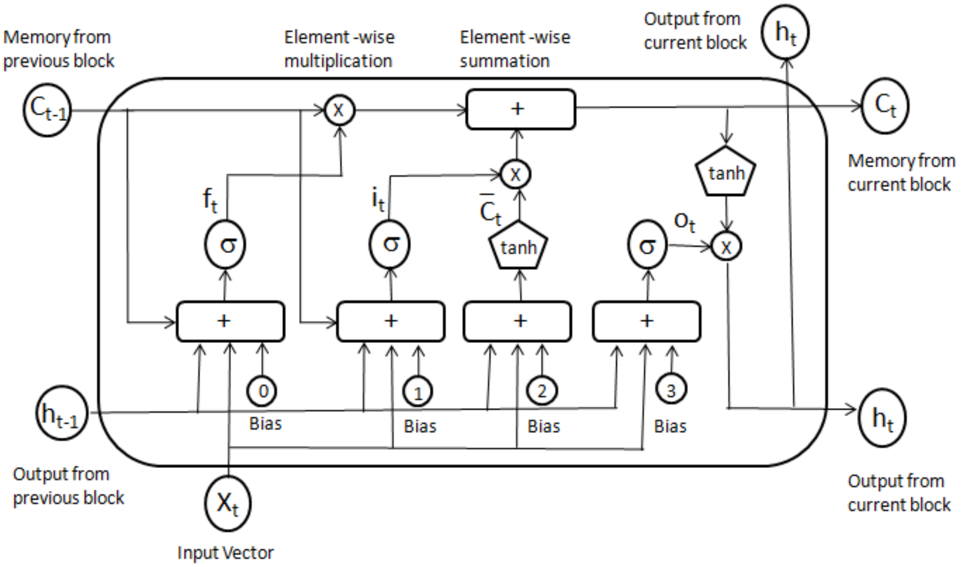
Three step process of LSTMs (Colah, 2021) is shown in Figure 1.

The first step in the LSTM model is to decide which information is to be omitted from the cell in that particular time step. The sigmoid function decides it. It looks at the previous state (h_{t-1}) and the current input x_t for each number in the cell state C_{t-1}, and computes the function

$$f_t = \sigma(W_f[h_(t-1), x_t] + b_f), \tag{1}$$

where f_t is a forget gate and is a vector with values ranging from 0 to 1, corresponding to each number in the cell state, C_{t-1}. W_f and b_f are the weight matrices and bias, respectively, of the forget gate.

Figure 1 Structure of LSTM network with input, output and forget gate


Memory from previous block
Element -wise multiplication
Element -wise summation
Output from current block
h_t
C_{t-1}
C_t
Memory from current block
f_t
i_t
$\bar{C}_t$
O_t
σ
tanh
X
+
0
1
2
3
Bias
h_{t-1}
Output from previous block
X_t
Input Vector
Output from current block

The second step decides how much should this unit add to the current state. In the second layer, there are two parts. One is the sigmoid function and the other is the tanh. In the sigmoid function, it decides which values to let through (0 or 1). tanh function gives weightage to the values which are passed, deciding their level of importance (–1 to 1). The product of the two values is added to the old memory C_{t-1}.

$$i_t = \sigma(W_i[h_{(t-1)}, x_t] + b_i) \tag{2}$$

$$N_t = \tanh(W_n[h_{(t-1)}, x_t]) + b_n \tag{3}$$

$$C_t = C_{t-1}f_t + N_t i_t \tag{4}$$

C_{t-1} and C_t represent cell states at time $t-1$ and t. W and b represent weight and bias of the cell state.

The third step is to decide the output. First, a sigmoid layer decides what parts of the cell state make it to the output. Then, the cell state is put through tanh to push the values between –1 and 1 and multiply it by the output of the sigmoid gate.

$$o_t = \sigma(W_o[h_{(t-1)}, x_t] + b_o) \tag{5}$$

$$h_t = o_t * \tanh(C_t), \tag{6}$$

where o_t = output gate. W_o and b_o are the weight matrices and bias, respectively, of the output gate. It allows the passed in information to impact the output in the current time step.

4 Methodology

4.1 Study area and data

Amazon global cloud infrastructure consists of 25 regions with 81 availability zones. Each region comprises several availability zones. This study focuses on analysing spot pricing trends, predicting future spot prices and assessing the model‘s ability in spot price forecasting. Therefore, region and availability zone of input, amount of data, and correlation of the data series are considered. Spot price history for c4.8xlarge instance type of Linux/UNIX operating system is retrieved from Amazon EC2 for four regions and their availability zones from 17 June 2021 to 17 August 2021 for training, validation and testing purpose. Spot history for 18th August 2021 is used for 24 hour future predictions. The LSTM model is constructed to forecast the future spot prices with one-hour, two-hours, four-hours and one-day of lead time. Regions and instance types under study are listed in Table 1. Spot history data collected in the study includes spot prices at different times during the study interval.

Table 1 Amazon EC2 regions, availability zones, instance types and other parameters for the study

Region name	*Region*	*Availability zones*	*Other parameters*
US East (Ohio)	us-east-2	a, b, c	Period of study: 90 days
US East (North Virginia)	us-east-1	a, b, c, d, e, f	Time of study: 17 June 2021 to 17 August 2021, time of future forecast: 18 August 2021
US West (N. California)	us-west-1	a, c	Operating system: Linux/UNIX
US West (Oregon)	us-west-2	a, b, c	Instance type: c4.8xlarge, r4.8xlarge

4.2 Model evaluation criteria

To evaluate the performance of time series forecasting models, root mean squared error (RMSE) is a good estimator for measuring the standard deviation sigma of a typical observed value from the model’s prediction. RMSE value zero indicates perfect prediction skills, or ‘no error’ in effect. The LSTM model produces reliable results when the RMSE values are small.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}} \tag{7}$$

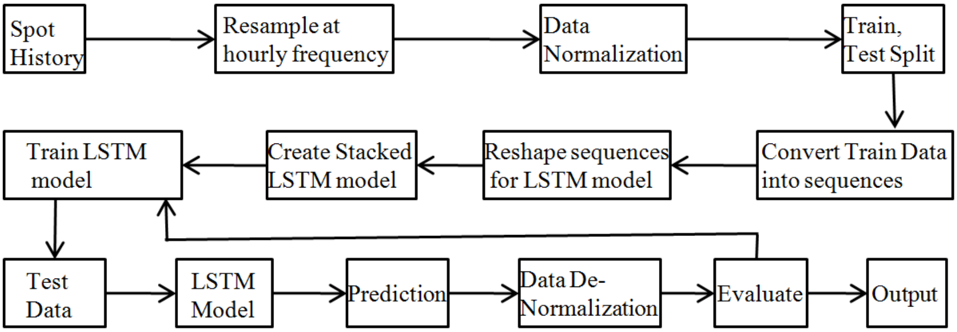
Mean absolute percentage error (MAPE): using MAPE, accuracy is estimated in terms of the differences in the actual versus predicted spot prices.

$$MAPE = \left(\frac{1}{n} \sum_{i=1}^{n} \frac{abs(\hat{y}_i - y_i)}{y_i} \right) * 100 \tag{8}$$

4.3 Workflow

Spot price history contains pricing data whenever spot price changes. The pricing data is not at regular intervals. In some regions, there are less spot price fluctuations so the length of the data series is less. While in some other regions there are more price fluctuations, so the size of the training set is different for different regions and availability zones. In order to make prediction easier and manageable, we perform resampling on spot history time-series data into the hourly frequency to analyse data and draw additional insights from data. Amazon EC2 considers previous hour spot price for charging spot instances per hour until new spot price is declared. We resample spot prices on hourly basis accordingly to fill in the missing values. Python programming language is used for implementation purpose. Open source Python software libraries, such as. Numpy, Pandas, Matplotlib, Keras, Sklearn are imported. Further, data is normalised using MinMax Scaler from sklearn library. We analyse input data characteristics because of their effect on model performance. These characteristics include the quantity of input data, and the correlation of the measured data series. Input data type includes hourly spot price in several regions and their availability zones. We consider spot prices within different regions and their availability zones. The total available data is subdivided into three non-overlapping sets for the purposes of training, validation, and testing. The workflow used in spot price prediction is shown in Figure 2.

Figure 2 Workflow of spot price prediction using LSTM



4.4 Statistical anaysis

In November 2017, Amazon simplified its spot pricing policy. Accordingly, the price of the spot instance became stable to a large extent. We study spot history after the new spot pricing policy. The volatility of spot pricing is measured using the following metrics.

- Autocorrelation: Autocorrelation function is used to identify correlation between values of the same variable in successive time periods. It is a mathematical tool for finding repeating patterns, between spot prices in successive time periods. Given measurements $Y_1, Y_2, \ldots, Y_N$ at time $X_1, X_2, \ldots, X_N$ and lag k, the autocorrelation function is defined as

$$\eta_k = \frac{\sum_{i=1}^{N-k}(Y_i - \overline{Y})(Y_{i+k} - \overline{Y})}{\sum_{i=1}^{N}(Y_i - \overline{Y})^2} \tag{9}$$

Figure 3 Autocorrelation in spot prices observed for c4.8xlarge instance types in us-east-1 region (see online version for colours)

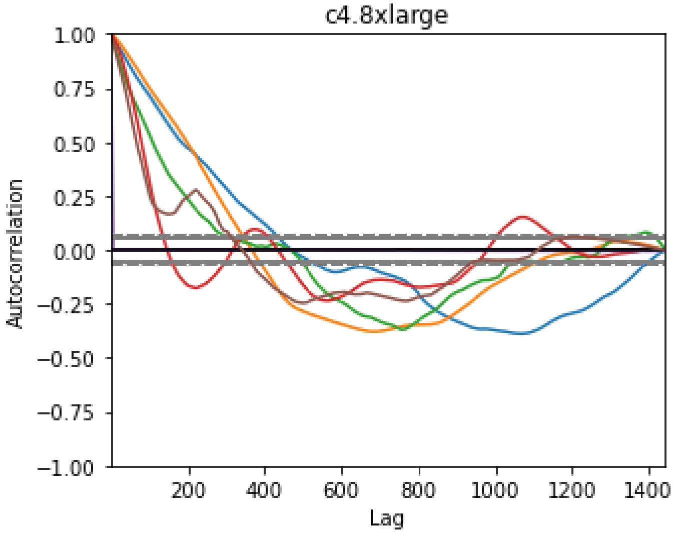
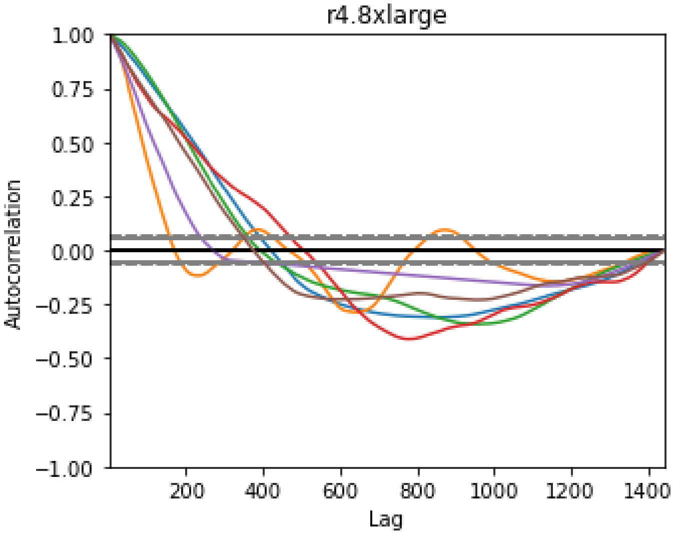


Figure 4 Autocorrelation in spot prices observed for r4.8xlarge instance types in us-east-1 region (see online version for colours)



- Autocorrelation analysis: To uncover hidden patterns in spot history data and help us select the correct forecasting method, we perform autocorrelation analysis to help identify seasonality and trend in our spot history data. Figures 3 and 4 show autocorrelation graphs of instance type c4.8xlarge in region us-east-1 and availability zones 1a, 1b, 1c, 1d, 1e, 1f and instance type r4.8xlarge in region us-east-1 and availability zones 1a, 1b, 1c, 1d, 1e, 1f. We see a gradual decay in data which says that data is not stationary.

Table 2 ADFTest and p-value of spot prices in different regions and their availability zones

SN	*Instance*	*ADF test (test if difference at the alpha = 0.05 significance level)*	*p-value*
1	us-east-2a c4.8xlarge	0.9548 True	0.6467
2	us-east-2b c4.8xlarge	0.64625 True	0.2213
3	us-east-2c c4.8xlarge	0.4626 True	0.0091
4	us-east-1a c4.8xlarge	0.4670 True	0.9751
5	us-east-1b c4.8xlarge	0.9211 True	0.3549
6	us-east-1c c4.8xlarge	0.9631 True	0.2707
7	us-east-1d c4.8xlarge	0.7404 True	0.1475
8	us-east-1e c4.8xlarge	0.0000 False	3.22E-28
9	us-east-1f c4.8xlarge	0.3424 True	0.2732
10	us-west 1a c4.8xlarge	0.4861 True	0.156
11	us-west 1c c4.8xlarge	0.3635 True	0.6775
12	us-west 2a c4.8xlarge	0.7582 True	0.8013
13	us-west 2b c4.8xlarge	0.4987 True	0.7390
14	us-west 2c c4.8xlarge	0.9853 True	0.6967
15	us-east-2a r4.8xlarge	0.9900 True	1.0000
16	us-east-2b r4.8xlarge	0.0100 False	4.42E-07
17	us-east-2c r4.8xlarge	0.0000 False	0.0000
18	us-east-1a r4.8xlarge	0.9300 True	0.9966
19	us-east-1b r4.8xlarge	0.5895 True	0.2680
20	us-east-1c r4.8xlarge	0.9031 True	0.8089
21	us-east-1d r4.8xlarge	0.5833 True	0.4333
22	us-east-1e r4.8xlarge	0.9900 True	0.9989
23	us-east-1f r4.8xlarge	0.9900 True	0.9974
24	us-west 1a r4.8xlarge	0.4531 True	0.7526
25	us-west 1c r4.8xlarge	0.4460 True	0.0490
26	us-west 2a r4.8xlarge	0.9072 True	0.4851
27	us-west 2b r4.8xlarge	0.7016 True	0.2621
28	us-west 2c r4.8xlarge	0.8610 True	0.9770

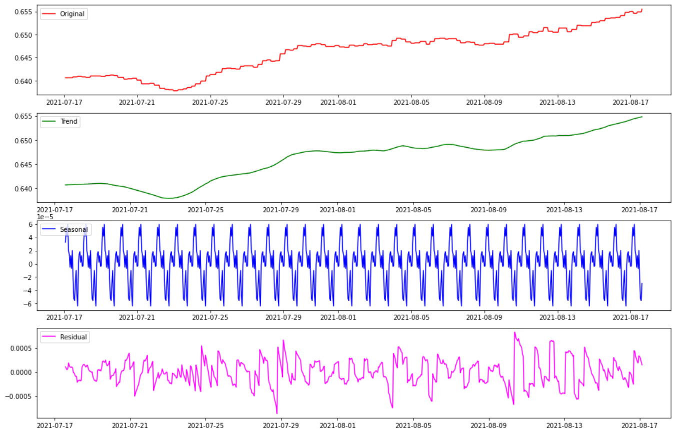
To confirm mathematically, whether the spot prices are stationary or not, we perform stationarity test – augmented Dickey Fuller test (ADFTest) to more quantitatively determine whether we need to difference our data in order to make it stationary at the alpha = 0.05 significance level. We also compute p-value using statsmodels python library to test whether spot prices in any availability zones are stationary or not. If p-value is above 0.05, the data are not stationary. Only for two evaluations out of 28 evaluations the test returns false and their p-values are less than 5%. The results of the test are shown in Table 2. From the results, it is concluded that the Amazon EC2 spot history data after the new spot pricing policy is not stationary in most of the cases.

- Standard deviation: Standard deviation is used to primarily measure volatility of spot prices. It depicts the average amount by which the spot price of an instance type in a given region has differed from its mean value over a given period of time.

4.5 Trend analysis

Figure 5 shows seasonal additive decomposition of spot prices using moving averages for c4.8xlarge instance type of region us-east-1a for a period of one month. The figure clearly shows that seasonality trend and residual values are very low. Majority of spot price change is due to nonlinear trend in spot prices.

Figure 5 Trend, seasonality and residual components in spot pricing (see online version for colours)



We also draw boxplots to study explanatory data analysis of monthly, weekday and hourly trends in spot prices, Figure 6 shows these trends for instance type c4.8xlarge in region us-east-1a. Boxplots suggests that there is a significant change in spot prices on monthly basis since the medians in all the monthly boxplots lie at different locations. In weekday boxplots the minimum score is almost same in most of the cases but there is a significant difference in the maximum score. The lower and upper quartile values are different in some cases. The median is towards the minimum score side which shows that majority of the spot prices are higher than the median price value. The minimum and maximum score in the hourly boxplots do not show any significant variation. This suggests that prices during a day do not vary largely. If the maximum predicted price for a day is used to bid for spot instances, then the jobs of 24 hours execution length can be executed without spot instance termination.

Figure 6 Boxplots showing monthly, weekday and hourly trends in spot pricing (see online version for colours)

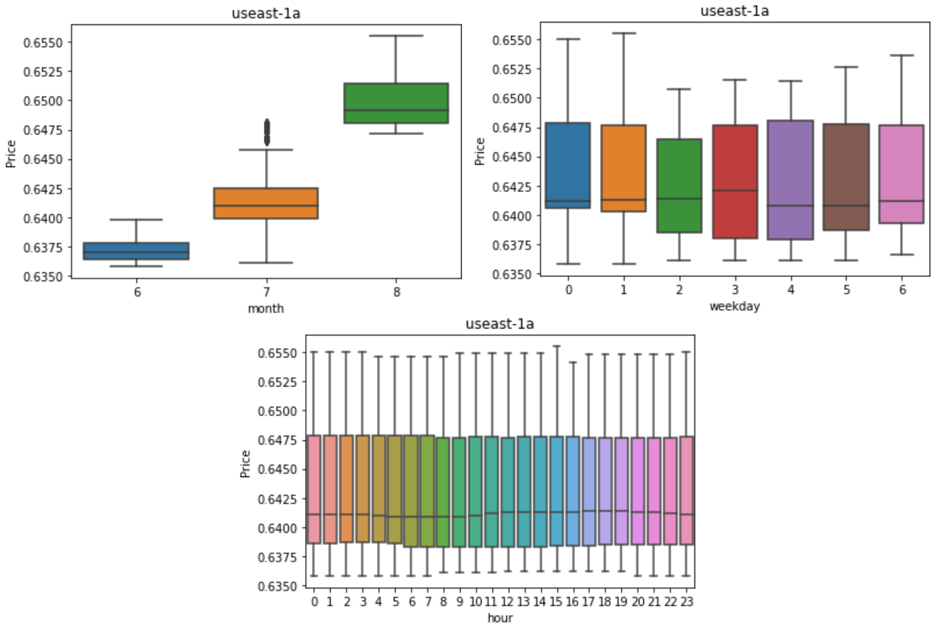
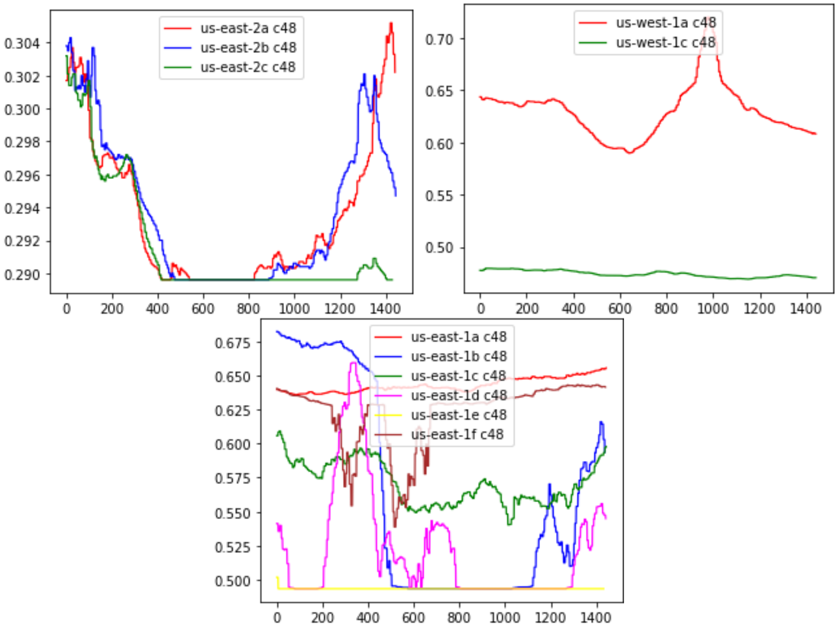


Figure 7 Spot prices in different regions and availability zones for c4.8xlarge instance type (see online version for colours)



Figures 7, 8 and 9 show the spot price plots of different regions and availability zones. In very few regions-availability zones, spot prices show less fluctuations while in most of the regions there are more fluctuations.

Figure 8 Spot prices in different regions and availability zones for r4.8xlarge instance type (see online version for colours)

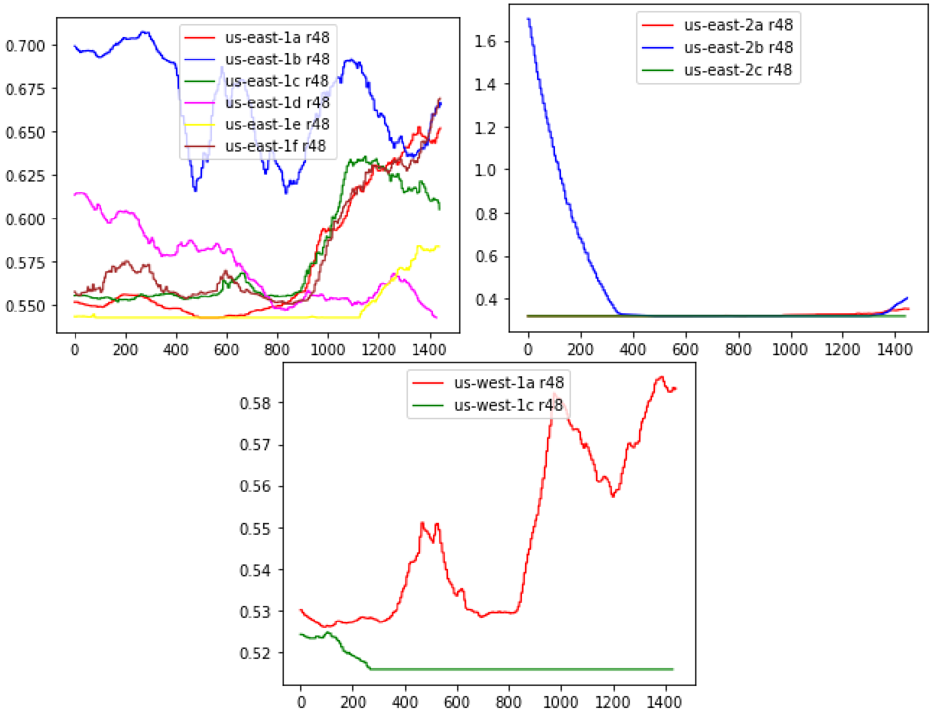
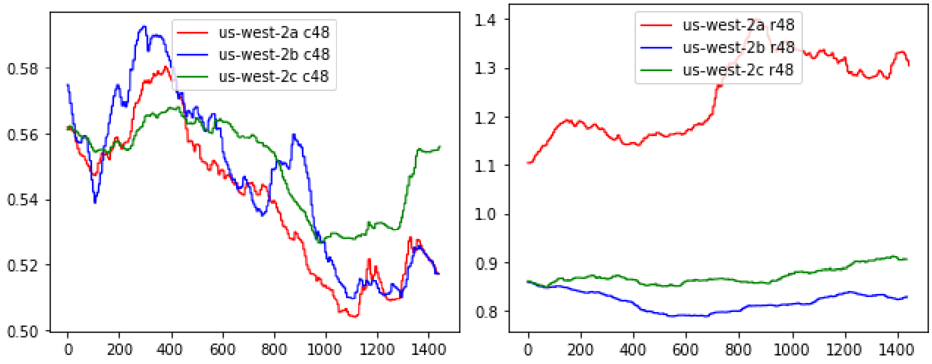


Figure 9 Spot prices in different regions and availability zones for c4.8xlarge and r4.8xlarge (see online version for colours)



This necessitates spot price prediction for ensuring availability to execute fault tolerant jobs. The spot price plots when drawn with respect to each other may seem stationary but stationarity test suggest that spot prices are not stationary.

4.6 Visualisation for parameters tuning

Visualisation of the performance of LSTM model is performed to make an informed decision on the hyperparameters values such as number of layers in the network, the number of nodes per layer, number of epochs. We evaluate the underfitting or overfitting by visualising the training loss vs. validation loss and training accuracy vs. validation accuracy over a number of epochs to determine if the model has been sufficiently trained. Based on the plots, the hyperparameters are adjusted. For this purpose, the spot price dataset is split across three sets: train, validation, and test. The train data is used to train the model while the validation model is used to test the fitness of the model.

5 Spot price prediction

5.1 One-hour, two-hours, four-hours ahead spot price predictions

LSTM model is used for spot price prediction for lead time one-hour, two-hours, four-hours and one-day for instance types c4.8xlarge and r4.8xlarge in four different regions and their availability zones. After fine tuning the parameters for minimal validation and testing accuracy, the parameters used are: epochs = 40, seq-size = 24, single layer LSTM 32 units, dense layers 32 units, dense layer 1 unit, optimiser = 'adam', metrics = 'accuracy'. One-hour, two-hours, and four-hours ahead spot price predictions for instance types c4.8xlarge are shown in Figure 10. One-hour, two-hours, and four-hours ahead spot price predictions for instance type r4.8xlarge are shown in Figure 11.

Table 3 RMSE values on training and test set for two-hours and four-hours ahead spot price prediction

Instance type	*RMSE train – 2 hours*	*RMSE test – 2 hours*	*RMSE train – 4 hours*	*RMSE test – 4 hours*
us-east-2a c4.8xlarge	0.00024634	0.00048523	0.00037906	0.00119532
us-east-1a c4.8xlarge	0.00027227	0.00070686	0.00048493	0.00091509
us-west-1a c4.8xlarge	0.00177761	0.00517093	0.00292832	0.00450977
us-west-2a c4.8xlarge	0.00193741	0.00384210	0.00247454	0.00553192
us-east-2a r4.8xlarge	0.00014030	0.00123007	0.00019764	0.00316505
us-east-1a r4.8xlarge	0.00067995	0.01185842	0.00109505	0.02122362
us-west-1a r4.8xlarge	0.00092926	0.00825723	0.00124579	0.01154611
us-west-2a r4.8xlarge	0.00677813	0.00830856	0.00823757	0.01583930

Tables 3 and 4 show the RMSE and MAPE values on training and testing dataset for different lead times for instance type c4.8xlarge and r4.8xlarge.

5.2 One-day ahead spot price predictions

Using the LSTM model we predict one day ahead spot prices also with slight change in the hyperparameters. LSTM parameters for one-day ahead future predictions after tuning are epochs = 40, seq-size = 48, single layer LSTM 32 units, dense layers 32 units, dense layer 1 unit, optimiser = 'adam', metrics = 'accuracy'. Table shows the RMSE values for one-day ahead future predictions for instance type c4.8xlarge and r4.8xlarge. RMSE values obtained for one-day ahead future predictions for r4.8xlarge are much higher than c4.8xlarge RMSE values. This shows that LSTM models are good predictors for some instance types only when used for long time future predictions.

Figure 10 One-hour, two-hours, and four-hours ahead spot price predictions for c4.8xlarge instance (see online version for colours)

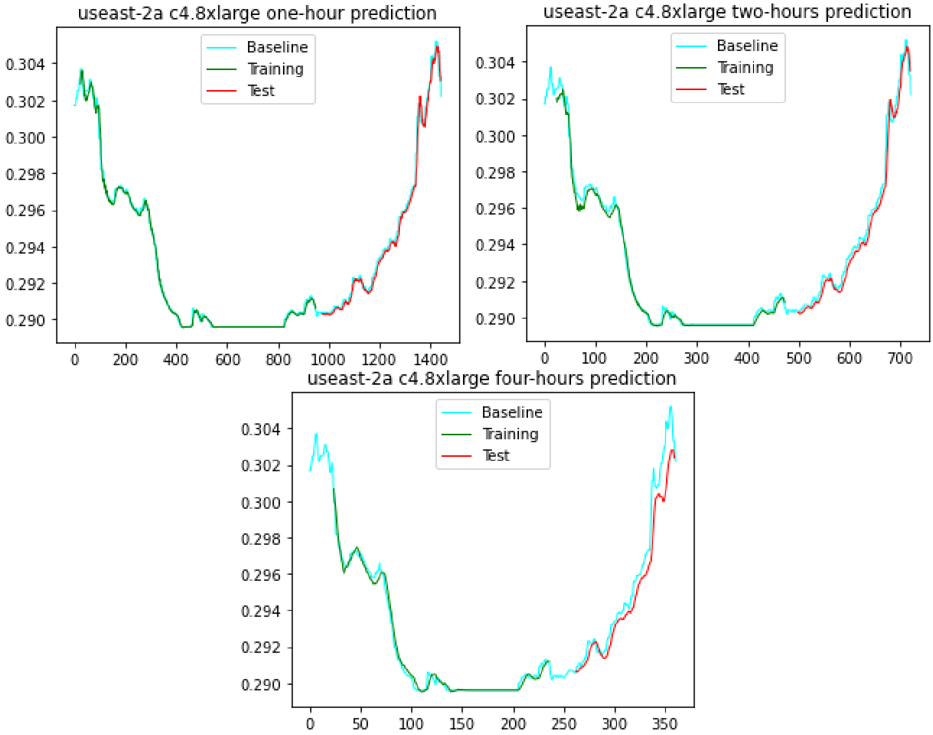


Table 4 MAPE values on training and test set for two-hours and four-hours ahead spot price prediction

Instance type	*MAPE train – 2 hours*	*MAPE test – 2 hours*	*MAPE train – 4 hours*	*MAPE test – 4 hours*
us-east-2a c4.8xlarge	0.00055965	0.00076075	0.00055559	0.00229325
us-east-1a c4.8xlarge	0.00036839	0.00111701	0.00041848	0.00168028
us-west-1a c4.8xlarge	0.00140327	0.00241974	0.00194658	0.00252283
us-west-2a c4.8xlarge	0.00150351	0.00278093	0.00251506	0.00510838
us-east-2a r4.8xlarge	0.00017346	0.01215850	0.00019694	0.01647193
us-east-1a r4.8xlarge	0.00075540	0.00246415	0.00063108	0.02200244
us-west-1a r4.8xlarge	0.00067464	0.00100317	0.00113461	0.00113461
us-west-2a r4.8xlarge	0.00201416	0.00207624	0.00380677	0.00516745

We also predict two-hour ahead spot prices using SARIMAX model. With SARIMAX model training RMSE errors = 0.02 and testing RMSE errors = 0.01. Since SARIMAX model includes seasonal effects and we have seen that there are no or little seasonal effect in new spot pricing model, this model predicts spot prices with higher error rate. Therefore, we conclude that LSTM model predicts future spot prices with much higher accuracy.

Figure 11 One-hour, two-hours, and four-hours ahead spot price predictions for r4.8xlarge instance (see online version for colours)

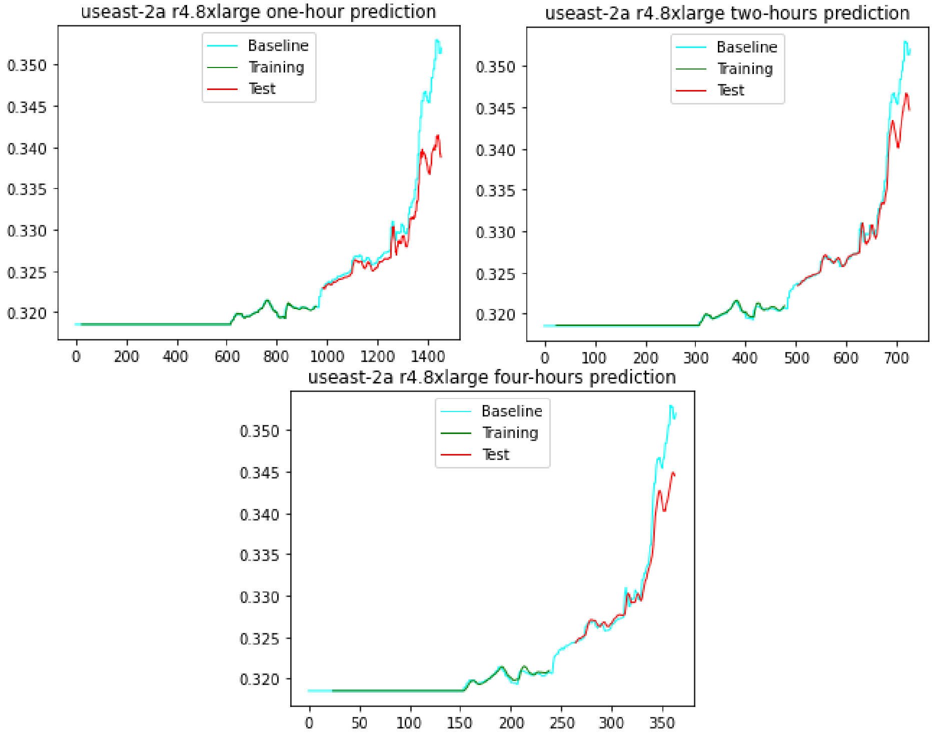


Table 5 RMSE values on training and test set for one-day ahead spot price prediction

SN	*Instance type*	*RMSE lead time – 48 hours*
1	us-east-2a c4.8xlarge	0.00438072
2	us-east-1a c4.8xlarge	0.00727534
3	us-west-1a c4.8xlarge	0.00093549
4	us-west-2a c4.8xlarge	0.00172467
5	us-east-2a r4.8xlarge	0.08763642
6	us-east-1a r4.8xlarge	0.05947822
7	us-west-1a r4.8xlarge	0.02308352
8	us-west-2a r4.8xlarge	0.08570071

6 Conclusions

This paper proposes an effective approach to Amazon EC2 spot price forecasting based on the LSTM neural network model. The model is assessed to forecast Amazon EC2 spot prices. We show that there are minimal seasonal and residual components in spot pricing. Contrary to considering all the trend components separately, the developed model uses the historical spot price data made available by AWS CLI. The data under study comprises compute optimised instance c4.8xlarge and memory optimised instance r4.8xlarge in four different regions and their associated availability zones. The LSTM model learns autocorrelation and long-term dependencies between sequential time steps of 24 hours for one-hour, two-hours, and four-hours ahead spot price prediction, and sequential time steps of 48 hours for one-day ahead spot price prediction. When higher time steps are used for forecasting, RMSE error increases. The superior performance of LSTM was demonstrated at multiple instance types, regions and availability zones.

Since LSTM models are in general data-driven models, they are efficient in resolving sequential data problems efficiently. Specifically, LSTM (or ANN)-based models only provide highly accurate forecasts upto 24 hours for some instance types only. Therefore, these models should be combined with some machine learning models to obtain better performance with long-term forecasts.

References

Alourani, A. and Kshemkalyani, A.D. (2020) 'Provisioning spot instances without employing fault-tolerance mechanisms', in *2020 19th International Symposium on Parallel and Distributed Computing (ISPDC)*, IEEE, July [online] https://doi.org/10.1109%2Fispdc51135.2020.00026.

Al-Theiabat, H., Al-Ayyoub, M., Alsmirat, M. and Aldwair, M. (2018) 'A deep learning approach for Amazon EC2 spot price prediction', in *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*, IEEE, October [online] https://doi.org/10.1109%2Faiccsa.2018.8612783.

Andrzejak, A., Kondo, D. and Yi, S. (2010) 'Decision model for cloud computing under SLA constraints', in *2010 IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems*, IEEE, 17–19 August, pp.257–266.

Baughman, M., Haas, C., Wolski, R., Foster, I. and Chard, K. (2018) 'Predicting Amazon spot prices with LSTM networks', in *Proceedings of the 9th Workshop on Scientific Cloud Computing*, ACM, June [online] https://doi.org/10.1145%2F3217880.3217881.

Ben-Yehuda, O.A., Ben-Yehuda, M., Schuster, A. and Tsafrir, D. (2013) 'Deconstructing Amazon EC2 spot instance pricing', *ACM Transactions on Economics and Computation*, Vol. 1, No. 3, p.16.

Chhetri, M.B., Lumpe, M., Vo, Q.B. and Kowalczyk, R. (2017) 'On forecasting Amazon EC2 spot prices using time-series decomposition with hybrid look-backs', in *2017 IEEE International Conference on Edge Computing (EDGE)*, June [online] https://doi.org/10.1109%2Fieee.edge.2017.29.

Chittora, V. and Gupta, C.P. (2020) 'Dynamic spot price forecasting using stacked LSTM networks', in *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, IEEE, December [online] https://doi.org/10.1109%2Ficiss49785.2020.9315988.

Chohan, N., Castillo, C., Spreitzer, M., Steinder, M., Tantawi, A.N. and Krintz, C. (2010) 'See spot run using spot instances for mapreduce workflows', *HotCloud*, 22–25 June, Vol. 10, pp.7–7.

Colah (2021) *Understanding LSTM Networks*, September [online] https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (accessed 15 September 2021).

Hassan, A.J. and Hammad, M. (2020) 'An approach to reduce cloud spot instances cost', *International Journal of Computing and Digital Systems*, September, Vol. 9, No. 5, pp.813–823 [online] https://doi.org/10.12785%2Fijcds%2F090504.

Javadi, B., Thulasiram, R.K. and Buyya, R. (2013) 'Characterizing spot price dynamics in public cloud environments', *Future Generation Computer Systems*, Vol. 29, No. 4, pp.988–999.

Khandelwal, V., Gupta, C.P. and Chaturvedi, A.K. (2018) 'Perceptive bidding strategy for Amazon EC2 spot instance market', *Multiagent and Grid Systems*, April, Vol. 14, No. 1, pp.83–102 [online] https://doi.org/10.3233%2Fmgs-180282.

Khandelwal, V., Chaturvedi, A.K. and Gupta, C.P. (2020) 'Amazon EC2 spot price prediction using regression random forests', *IEEE Transactions on Cloud Computing*, January, Vol. 8, No. 1, pp.59–72 [online] https://doi.org/10.1109%2Ftcc.2017.2780159.

Kong, D., Liu, S. and Pan, L. (2021) 'Amazon spot instance price prediction with GRU network', in *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, IEEE, May [online] https://doi.org/10.1109%2Fcscwd49262.2021.9437881.

Lucas-Simarro, J.L., Moreno-Vozmediano, R., Montero, R.S. and Llorente, I.M. (2015) 'Cost optimization of virtual infrastructures in dynamic multi-cloud scenarios', *Concurrency and Computation: Practice and Experience*, Vol. 27, No. 9, pp. 2260–2277.

Mazzucco, M. and Dumas, M. (2011) 'Achieving performance and availability guarantees with spot instances', in *2011 IEEE 13th International Conference on High Performance Computing and Communications (HPCC)*, IEEE, 2–4 September, pp.296–303.

Pary, R. (2018) *New Amazon EC2 Spot Pricing Model*, March [online] https://aws.amazon.com/blogs/compute/new-amazon-ec2-spot-pricing/ (accessed 15 September 2021).

Portella, G., Rodrigues, G.N., Nakano, E. and Melo, A.C. (2018) 'Statistical analysis of Amazon EC2 cloud pricing models', *Concurrency and Computation: Practice and Experience*, March, Vol. 31, No. 18 [online] https://doi.org/10.1002%2Fcpe.4451.

Singh, V.K. and Dutta, K. (2015) 'Dynamic price prediction for Amazon spot instances', in *2015 48th Hawaii International Conference on System Sciences (HICSS)*, IEEE, 5–8 January, pp.1513–1520.

Zhao, H., Pan, M., Liu, X., Li, X. and Fang, Y. (2012) 'Optimal resource rental planning for elastic applications in cloud market', in *Parallel & Distributed Processing Symposium (IPDPS), 2012 IEEE 26th International*, IEEE, 21–25 May, pp.808–819.