International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 20 (2017) pp. 10276-10283

Forecasting Price of Amazon Spot Instances using Neural Networks

Sonam Agarwal
Department of Computer Science & Engineering,
Motilal Nehru National Institute of Technology Allahabad, Allahabad - 211004, India.

Ashish Kumar Mishra
Department of Computer Science & Engineering,
Motilal Nehru National Institute of Technology Allahabad, Allahabad - 211004, India.
Orcid Id: 0000-0002-7532-5585

Dharmendra K. Yadav
Department of Computer Science & Engineering,
Motilal Nehru National Institute of Technology Allahabad, Allahabad - 211004, India.

Abstract

Spot price or variable price resources are the recent advancement in cloud computing business models. The spot pricing mechanism follows auction based cloud model in which the price of spot instances changes with time. If the termination of instance is initiated by user before completion of an hour, then user is bound to pay for the complete hour. However, if the instance termination is initiated by Amazon then the user doesn't have to pay for the partial hour. The termination of instance is done by cloud provider without any prior notification to the user when the bid price of user gets down to the current spot price. This poses a severe limitation to the applications where the time of availability is very important. Thus, it becomes very important for the users to predict spot prices before placing their bids. This article presents a method for predicting spot prices using techniques of artificial neural network. It also discusses the proposed methodology in detail and shows the experimental evaluation and results on various instances of Amazon Elastic Compute Cloud (EC2). For most of the instances, percentage error varies from 1% to 8.6% using the recurrent neural network technique. The proposed technique reduces the percentage of average error significantly in-comparison to other techniques.

Keywords: AWS, Neural Networks, Price Prediction, Spot Instances.

INTRODUCTION

Cloud computing provides pooled resources of computing accessible over Internet. Computing resources comprise of both hardware and software. Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), and Network as a Service (NaaS) are the different model of cloud services There are following models for the deployment of cloud: public cloud, private cloud, community cloud, and hybrid cloud. Cloud enables use of resources based on pay as you go model. Amazon is one of the largest service provider of cloud computing resources. The three types of instances that are offered by Amazon in around 16 regions across the globe are Reserved Instances, On-demand Instances and Spot Instances

The distinguishing feature of spot instance is its dynamic pricing. The prices of spot instances vary dynamically with time based on demand and supply of cloud resources in the datacenters across the globe. Customers place bids to obtain spot instances using an online auction platform. The auction platform determines the market clearance price also known as spot price and if the users bid above the aforementioned price they obtain the instances. Cloud vendors provide current and archived spot price data so as to assist their customers in bidding process. Amazon Web Services (AWS) provide a web based API access to the users for bidding spot instances. The spot instance bid request consists of the following parameters:

1) Instance type
2) Number of instances
3) Availability zone
4) Bid amount of user i.e. price/instance/per hour.

Amazon provides public cloud computing services in 16 regions [1] (6 in Asia, 1 in South America, 3 in the European Union and 6 in the North America). Each region contains more than one availability zones. These zones denote physical server locations. Several virtual machine instance types are provided at each availability zone. The types of machines differ by their objective and can be categorized as:

1) General machines (with fixed and variable computing power)
2) Compute-optimized machines
3) Memory-optimized machines
4) Storage-optimized machines

5) Graphics Processing Unit (GPU) mode machines

A. The spot instance pricing technique

The Amazon EC2 spot instance pricing mechanism is very complicated due to its all-time fluctuation in prices and the possibility of sudden termination of instances. The termination can be initiated either by the user or cloud provider. The steps followed in the spot instance billing price mechanism is as follows:

1) User bids for a spot instance in a given availability zone for a selected machine type.
2) The spot instance is created only if bid price becomes greater than the current spot price. If the bid price becomes equal to the current spot price, cloud provider can either start or not to start, depending on its policy.
3) On setup, the cost of getting an instance is set as a spot price.
4) In spite of continuous variation in the spot prices with the variation in the demand and supply, pricing algorithm works on hourly basis. So, any of the two, either cloud provider or user can interrupt an hour of computation.
5) If the instance is interrupted by the users without completion of an hour, the user has to pay for the complete hour.
6) If the interruption is due to cloud provider without completion of an hour, the partial hour utilized by user will be free.
7) If an hour is completed without interruption, the user has to pay for this hour and a new price is decided as the spot price at the beginning of new hour.
8) Here an important point of consideration is that the bid price cannot be changed after the creation of instance.

Two important conclusions can be inferred from the above points. First, users get benefited sometime because they get the computation services free of cost in case of the cloud provider terminates the instance. Second, at the termination of each instance by cloud provider leads to the starting of new instance in the future. Initiation of a new instance need some booting time which is not counted in the effective execution time of the simulation.

B. Price forecasting methods

Commonly price forecasting mechanism can be categorized into two types namely time series approach and simulation approach [2]. Time series approach primarily depends on the history of past market prices, while simulation approach requires exact modeling of system components with their cost related information. Simulation approach can be costly because of involvement of enormous computations for huge amount of data. Artificial Neural Network (ANN) is one of the most popular method for forecasting prices based on time series approach. Now researchers have come with the hybrid models that overcome the short comings of the individual models. Study shows that various mechanisms have been developed for predicting the prices. The use of Neural Network for predicting prices can be taken as an improvement to many price forecasting techniques. Neural networks resemble to the neurons of brain that are extremely organized, well connected simple processing units and perform a particular task as the brain does. ANN models are classified on the basis of number of hidden layers, learning algorithm, type of learning function etc.

LITERATURE REVIEW

Spot price prediction is a key research area; as more accurate prediction is directly related to the more profit for the users. Accurate price prediction not only can help users to save the computation time for their task, but also can lead to significant cost reduction for the resource usage. Therefore, in recent years significant efforts have been put into developing models which can predict future trends.

There are various techniques that have been developed in the past for prediction of spot prices. The authors of [3] had presented a paper which uses gradient descent algorithm to estimate the weighted coefficients for previous month data. Gradient descent algorithm is primarily exploited for solving a system of linear equations. This is achieved by reformulating the linear equations into a quadratic minimization problem. The error is calculated by taking the difference between actual value of spot price and the predicted value from the model.

A multilayer perceptron, a class of feedforward artificial neural network is being used in the article [4]. The obtained experimental results show that neural networks are well suited for such kind of prediction and could be very useful for users bidding on spot instances.

In [5], an algorithm to optimize cost and time of running simulations on the public computational clusters using spot pricing mechanism is being proposed by authors. This strategy increases the simulation time notably. The fruitful time can be achieved at the price of minimum reference price level and it is approximately 33% of the simulation time. Spot instances are not guaranteed for compute resources and can be terminated at any point of time. Due to this abrupt termination, the computation results may get lost. Thus, the abrupt termination of spot instances makes them unreliable for the users. To increase the reliability of spot instances, some fault tolerant techniques need to be utilized. In [6], authors have discussed various checkpointing and work migration techniques, which can be used to achieve the goal of minimizing monetary costs while maximizing reliability.

In the current implementation, the prices of spot instances increase with the increase in the demand of reserved and on demand instances. This results into the sudden revocation of spot instances. One possible solution can be the setting of high bids and so increase in total cost of execution. However, observing past prices of spot instances, it can be said that this technique also does not give the assurance of uninterrupted running of spot instances.

There is trade-off between cost and total computation time in the use of Elastic spot instances (ESIs) [7]. To complete a particular task within a time limit, users can preserve the running instance for some more time even with increase in bid price. The limit of increase in price can be fixed by the user, if the increase in price crosses the limit, users can decide for the planned termination of instance. A set of bidding mechanisms is being proposed for minimizing resource provisioning cost and unpredictability [8]. The problem is being formulated as a Constrained Markov Decision Process (CMDP) to find an optimal bidding mechanism. Linear programming based on this model is used by authors to get an optimal randomized bidding scheme. Using price traces of live instances and models of workload, various adaptive checkpointing schemes have been compared in terms of job completion time and monetary costs.

In [9], authors have described various pricing models of cloud computing. They have presented various traits of cloud pricing models. The three main parameters on which a user will evaluate a cloud provider are pricing approach, quality of service, and utilization period. The pricing approach could be fixed price approach without any consideration of size or quantity, constant price with unit rate or determined quantity of purchase with rate of unit price. What is provided to customers by the service provider is laid down in the quality of service document. The duration of time in which user has the full right to utilize the services of provider depending on the service level agreements between customer and service provider.

PROPOSED METHOD

The proposed method uses recurrent neural network (RNN) technique to predict spot prices. The input of RNN is current input as well as just previous input. RNN [10] is a type of artificial neural network in which directed cycles are used to connect units. It results into creation of network internal states for showing dynamic temporal behaviors. Dissimilar to feedforward, RNNs use their internal memory to process random sequences of inputs. In the proposed algorithm, the use of Long Short Term Memory (LSTM) networks is being used for spot price prediction.

LSTM [11] is a special kind of RNN architecture proposed by authors in the year 1997. LSTMs help in preserving errors to be back propagated through layers and time. They permit recurrent nets that keep on learning for many time (over 1000) steps by sustaining constant error. This opens a channel for associating cause and effect from the remote locations. Their work is extremely good over a various kinds of problems, and are now being used extensively.

Long-term dependency problem can be avoided with the use of LSTMs. The default behavior of LSTMs is to memorize information for long duration of time, leaving the thing which is hard to learn for them, Persons don’t start thinking every time from very beginning about any event. For example, in the case of reading some articles by humans, each word is interpreted by them depending on the understanding of previous words. Due to perseverance in human thoughts, they can’t leave everything and start thinking afresh.

Persistence of information was not supported by neural network earlier. It was one of the main drawbacks of the neural networks. An example can be the classification of events happening at each point in a movie. The way traditional networks can use the understanding of previous events for the later one in the movie, was not obvious. This issue is handled by RNNs. These networks have loops to allow persistence of information which enables RNNs to predict future patterns depending on the history of data.

The first step for predicting the spot prices is to get the spot price history data provided by AWS. The archived spot price history data is retrieved by first creating an account with AWS and then using AWS Command Line Interface (CLI) for data retrieval. After data retrieval, the proposed method has been implemented in Python. Algorithm 1 has been used for prediction which is basically the algorithm of time series prediction using neural networks.

A. Requirements for Implementation

Neural networks’ concepts can be implemented using Python language. Users need to have the latest version of Scikit-learn installed. It can be installed either through pip or conda manager. The following libraries need to be installed before implementing LSTM network in Python.

1) **Pandas:** Pandas library [12] is used for loading the dataset. Pandas is a software library written for the Python programming language for data manipulation and analysis.
2) **Matplotlib:** Matplotlib [13] is used to plot the input dataset and output. It is a library of Python programming language, used for plotting graphs. NumPy is its numerical mathematical extension. Object-oriented API is provided by it for plots embedding into applications using GUI toolkits e.g. wxPython, Tkinter, ‘Qt, or GTK+.
3) **Keras:** Keras [14] provides high-level building blocks to develop models of in-depth learning. It is a library of model-level. It depends on a well-optimized tensor manipulation library. This library serves as Keras

backend engine. Currently, two backend implementations of Keras, are available:

a. TensrorFlow
b. Theano

The TensrorFlow backend is the latest backend engine for Keras. It is being used as a backend engine for the proposed implementation.

4) **NumPy:** NumPy [15] is the primary package for scientific computing through Python.

Algorithm 1 for Spot Price Prediction

Dataset: M ={m0, m1 ,m2,...,ml}

Input 1 (TrainingSet): T = {T0, T1 ,T2,...,Tj}

Input 2 (TestSet): S = {Sj, Sj+1,Sj+2,...,Sl }

Output: P ={Pj, Pj+1,Pj+2,...,Pl}

Step1: Create a random seed to ensure that results can be reproduced.

Step2: Normalize the dataset using minMaxScaler.

Step3: Convert an array of values into a dataset matrix:

procedure
  CREATEDATASET (set, factor=1)
  for k ← 0...k <= (l − factor − 1) **do**
    dX[k] ← set[k : (k + factor), 0]
    dY [k] ← set[(k + factor), 0]
    return dX, dY
  end for
end procedure

Step 4: Preparing the training and test datasets for modeling:

TX, TY = createDataset(T, factor)

SX, SY = createDataset(S, factor)

Step 5: Create LSTM network and fit the model using trainee set T.

Step 6: Do predictions both for training dataset and test dataset and calculate percentage error.

$$E = (\frac{100}{N}) \times \sum_{k=j}^{l} |\frac{P_k - SY_k}{SY_k}| \quad (1)$$

B. Input dataset

The input data set for the program code is the CSV file containing the spot prices for 90 days. This CSV file is created by converting the json file obtained by executing AWS CLI command on the console. There are various steps involved in getting the spot price history data from AWS. The steps are as follows:

1) Create an account on AWS by providing all the necessary details including credit card details.
2) When the account is activated by AWS, go to Identity and Access Management Console (IAM).
3) In IAM console, go to the “User" column on the left side to create a new user.
4) While creating a new user, some policies need to be attached to the user. It is necessary to make the user as an admin otherwise there will be access issues in retrieving the data. To ensure this, a policy named “Administrator Access” needs to be attached to the user.
5) Once the user is created, AWS provides every user with a unique access key id and a secret access key. Users have to save them for future reference.

Once the configuration settings have been done, the next step is installing AWS CLI for getting the spot price history data. The first requirement for using AWS CLI is installing python. The AWS services are managed by a tool named as AWS CLI [16]. This tool can be used for downloading, configuring and controlling various AWS services through command line and scripts can be used to automate them. AWS CLI provides some simple commands for transferring file to and from Amazon Simple Storage Service (S3). The sample command that needs to be run for fetching the spot price history data into a file in json format is given below:

aws ec2 describe-spot-price-history –instance-types m1.large –start-time 2017-03-08T07:08:09 –end-time 2017-04-09T08:09:10

C. Instances used for Testing

There are more than 64 types of spot instances offered by Amazon in 16 regions (6 in Asia, 1 in South America, 3 in the European Union and 6 in the North America). Each region contains more than one availability zones which represent physical server locations. Table 1 shows the detail of instances used for prediction.

Table 1: Instances used for Prediction

Instance ID	Instance Type	Availability Zone
I1	c3.2xlarge	us-west-1
I2	m1.large	ap-southeast-1
I3	m2.2xlarge	ap-southeast-2
I4	m1.small	ap-southeast-1
I5	c1.xlarge	us-west-2
I6	m2.2xlarge	ap-southeast-1
I7	c4.large	us-west-1

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 20 (2017) pp. 10276-10283

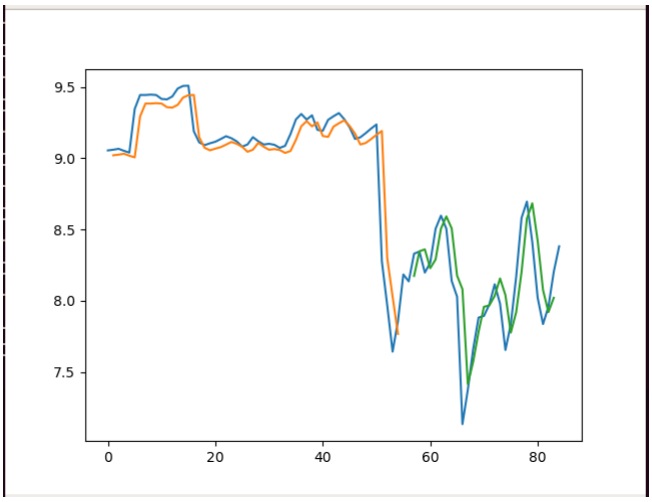

Figure 1: Prediction outcomes for **c3.2xlarge** instance, region **us-west-1**

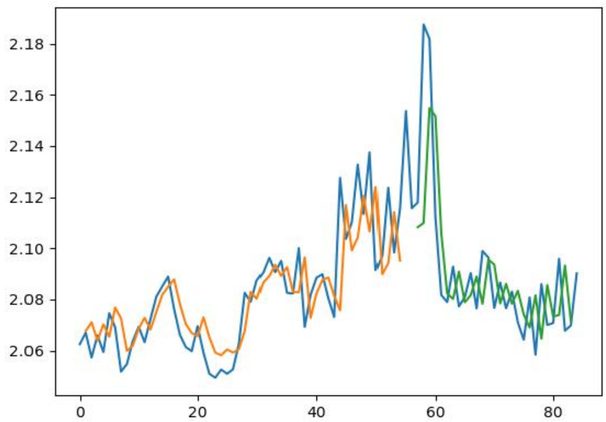

Figure 2: Prediction outcomes for **m1.large** instance, region **ap-southeast-1**

EXPERIMENTAL OUTCOMES AND DISCUSSION

The system is implemented in Python. In the proposed method, the dataset has been broken into the training dataset and test dataset. The training dataset contains 67% of the data that is utilized for training the model, leaving the rest 33% of the data for testing the model. The algorithm has been tested on various instances of four different availability zones. The error percentage for various instances is mentioned in table 2.

The graphs of the prediction are shown in Figures 1-7. X-axis and y-axis of the graph represent the days and the spot price (in cents) respectively. The legend for the graphs shown in figures are:

1) Blue: Shows the original spot price values.
2) Orange: The training dataset.
3) Green: The test dataset values i.e. the predicted spot price.

A. Comparison with other techniques

The proposed method uses recurrent neural network technique and gives an error percentage of 5.23% for 6 instances. For most of the instances the percentage error varies from 1% to 8.6% using the recurrent neural network technique. However, the error percentage using the gradient descent algorithm presented by Singh et al. in paper [3] is 9.4%.

Table 2: Percentage error for Instances used for Prediction

Instance ID	Instance Type	Availability Zone	Percentage Error
I1	c3.2xlarge	us-west-1	4.72%
I2	m1.large	ap-southeast-1	1.15%
I3	m2.2xlarge	ap-southeast-2	57.30%
I4	m1.small	ap-southeast-1	0.20%
I5	c1.xlarge	us-west-2	0.16%
I6	m2.2xlarge	ap-southeast-1	16.54%
I7	c4.large	us-west-1	8.61%

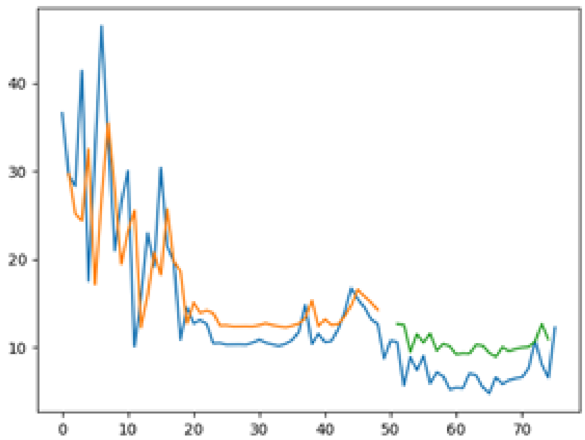

Figure 3: Prediction outcomes for **m2.2xlarge** instance, region **ap-southeast-2**

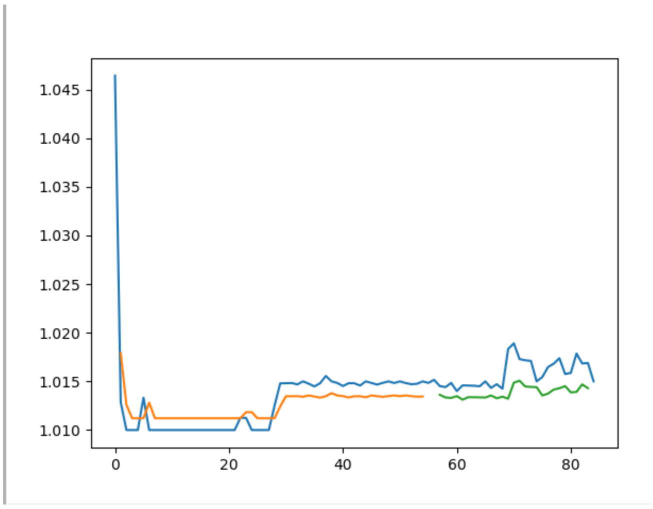

Figure 4: Prediction outcomes for **m1.small** instance, region **ap-southeast-1**

Turchenko et al. presented a paper [17] on "Spot price prediction for cloud computing using neural networks" which used feedforward neural network technique for predicting spot prices. The average error percentage for m1.linux instances is 3.3%. Whereas in the proposed case it is 1.15%.

The reason for very high percentage error for m2.2xlarge in the region ap-southeast-2 is due to the fact that at some point of time the spot price becomes 50 times more than the previous timestamps. Thus, the proposed method generates high percentage error if there is a sudden shoot up in the spot price by 50 times or more.

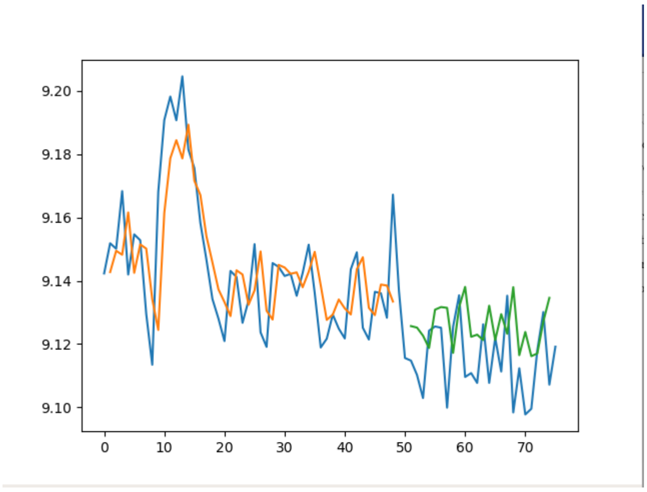

Figure 5: Prediction outcomes for **c1.xlarge** instance, region **us-west-2**

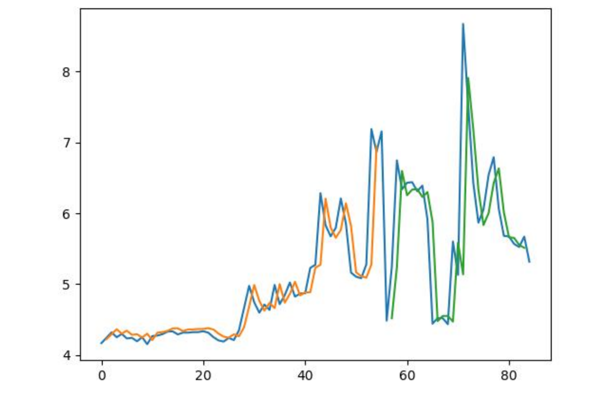

Figure 6: Prediction outcomes for **m2.2xlarge** instance, region **ap-southeast-1**

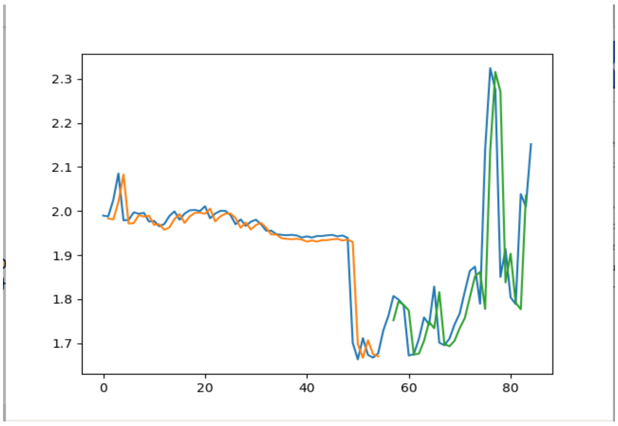

Figure 7: Prediction outcomes for **c4.large** instance, region **us-west-1**

CONCLUSION

A method is being proposed for estimating spot prices using the past 90 days spot price history data provided by Amazon. Although many neural network techniques have been implemented for spot price prediction but the proposed method uses recurrent neural network using LSTM which gives a good accuracy on prediction results. This gives an error percentage of 5.23% for 6 instances. For most of the instances the percentage error varies from 1% to 8.6% using the recurrent neural network technique.

The proposed approach used ANNs to approximate complex nonlinear functions. They can resolve indeterminate relation between input and output variables. Due to loops, these networks allow persistence of information. This makes RNNs suitable for predicting future trends based on the previous history of data. They can also implement multiple training algorithms. On the other hand, the disadvantage is that the network is not elastic for the small data and it is over-fitting for large data.

FUTURE WORK

In the proposed method, history data of 90 days is used for spot price prediction. To eliminate the timestamp factor in the algorithm, the mean price for all the timestamps of a particular day has been taken in the computation. This is done because the timestamps of a particular day, for which Amazon provides the spot price history data, were not separated uniformly. So in future, one can modify the algorithm to take into account the timestamp factor as well for better precision and accuracy.

REFERENCES

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

[2] N. Pandey and K. G. Upadhyay, “Different price forecasting techniques and their application in deregulated electricity market: A comprehensive study”, In 2016 International Conference on Emerging Trends in Electrical Electronics Sustainable Energy Systems (ICETEESES), pages 1–4, March 2016.

[3] Vivek Kumar Singh and Kaushik Dutta, “Dynamic price prediction for amazon spot instances”, In Proceedings of the 2015 48th Hawaii International Conference on System Sciences, HICSS ’15, pages 1513– 1520, Washington, DC, USA, 2015. IEEE Computer Society.

[4] Volodymyr Turchenko, Vladyslav Shults, Iryna Turchenko, Richard M Wallace, Mehdi Sheikhalishahi, Jose Luis Vazquez-Poletti, and Lucio Grandinetti, “Spot price prediction for cloud computingusing neural networks” International Journal of Computing, 12(4):348–359, 2014.

[5] Bogumil Kaminski and Przemyslaw Szufel, “On optimization of simulation execution on amazon EC2 spot market”, Simulation Modelling Practice and Theory, 58:172–187, 2015.

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

[7] W. Dawoud, I. Takouna, and C. Meinel, “Increasing spot instances reliability using dynamic scalability”, In 2012 IEEE Fifth International Conference on Cloud Computing, pages 959–961, June 2012.

[8] ShaoJie Tang, Jing Yuan, and Xiang-Yang Li, “ Towards optimal bidding strategy for amazon ec2 cloud spot instance”, In Proceedings of the 2012 IEEE Fifth International Conference on Cloud Computing, CLOUD ’12, pages 91–98, Washington, DC, USA, 2012. IEEE Computer Society.

[9] May Al-Roomi, Shaikha Al-Ebrahim, Sabika Buqrais, and Imtiaz Ahmad, “Cloud computing pricing models: A survey”, International Journal of Grid and Distributed Computing, 6(5):93–106.

[10] Recurrent neural network. https://en.wikipedia.org/wiki /Recurrent_neural _network, Accessed on July 30, 2017.

[11] Long short-term memory. https://en.wikipedia.org/wik i/Long_shortterm_memory, Accessed on July 10, 2017.

[12] pandas (software). https://en.wikipedia.org/wiki/ Pandas_(software), Accessed on June 25, 2017.

[13] matplotlib. https://en.wikipedia.org/wiki/Matplotlib, Accessed on May 15, 2017.

[14] Keras: The Python Deep Learning library. https://keras.io/, Accessed on April 30, 2017.

[15] NumPy. https://en.wikipedia.org/wiki/NumPy, Accessed on December 26, 2016.

[16] AWS Command Line Interface. https://aws.amazon.com/cli/, Accessed on January 17, 2017.

[17] Volodymyr Turchenko, Vladyslav Shults, Iryna Turchenko, Richard M Wallace, Mehdi Sheikhalishahi, Jose Luis Vazquez-Poletti, and Lucio Grandinetti, “Spot price prediction for cloud computing using neural networks”, International Journal of Computing, 12(4):348-359, 2014.

[18] Artan Mazrekaj, Isak Shabani, and Besmir Sejdiu, “Pricing schemes in cloud computing: An overview”, International Journal of Advanced Computer Science and Applications, 7(2):80–86, 2016.

[19] M. K. Mohan Murthy, H. A. Sanjay, and J. P. Ashwini, “Pricing models and pricing schemes of iaas providers: A comparison study”, In Proceedings of the International Conference on Advances in Computing, Communications and Informatics, ICACCI ’12, pages 143–147, New York, NY, USA, 2012. ACM.

[20] C. Vecchiola, S. Pandey, and R. Buyya, “High-performance cloud computing: A view of scientific applications”, In 2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks, pages 4– 16, Dec 2009.

[21] M. Mattess, C. Vecchiola, and R. Buyya, “Managing peak loads by leasing cloud infrastructure services from a spot market”, In 2010 IEEE 12th International Conference on High Performance Computing and Communications (HPCC), pages 180–188, Sept 2010.

[22] A. Iosup, S. Ostermann, M. N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema, “Performance analysis of cloud computing services for many-tasks scientific computing”, IEEE Transactions on Parallel and Distributed Systems, 22(6):931–945, June 2011.

[23] Bahman Javadi, Ruppa K. Thulasiram, and Rajkumar Buyya, “Characterizing spot price dynamics in public cloud environments”, Future Gener. Comput. Syst., 29(4):988–999, June 2013.

[24] Simon Ostermann and Radu Prodan, “Impact of variable priced cloud resources on scientific workflow scheduling”, In Proceedings of the 18th International Conference on Parallel Processing, Euro-Par’12, pages 350–362, Berlin, Heidelberg, 2012. Springer-Verlag.

[25] S. Chaisiri, R. Kaewpuang, B. S. Lee, and D. Niyato, “Cost minimization for provisioning virtual servers in amazon elastic compute cloud”, In 2011 IEEE 19th Annual International Symposium on Modelling, Analysis, and Simulation of Computer and Telecommunication Systems, pages 85–95, July 2011.

[26] Y. El-Khamra, H. Kim, S. Jha, and M. Parashar, “Exploring the performance fluctuations of hpc workloads on clouds”, In 2010 IEEE Second International Conference on Cloud Computing Technology and Science, pages 383–387, Nov 2010.

[27] B. Javadi, R. K. Thulasiramy, and R. Buyya, “Statistical modeling of spot instance prices in public cloud environments”, In 2011 Fourth IEEE International Conference on Utility and Cloud Computing, pages 219– 228, Dec 2011.

[28] Samuel V Angiuoli, James R White, Malcolm Matalka, Owen White, and W Florian Fricke, “Resources and costs for microbial sequence analysis evaluated using virtual machines and cloud computing”, PloS one, 6(10):e26624, 2011.

[29] M. Mihailescu and Y. M. Teo., “Dynamic resource pricing on federated clouds”, In 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, pages 513–517, May 2010.

[30] Orna Agmon Ben-Yehuda, Muli Ben-Yehuda, Assaf Schuster, and Dan Tsafrir, “Deconstructing amazon ec2 spot instance pricing”, ACM Trans. Econ. Comput., 1(3):16:1–16:20, September 2013.

[31] V. Kushwaha and Y. Simmhan, “Cloudy with a spot of opportunity: Analysis of spot-priced vms for practical job scheduling”, In 2014 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), pages 1–8, Oct 2014.

[32] S. Yi, D. Kondo, and A. Andrzejak, “Reducing costs of spot instances via checkpointing in the amazon elastic compute cloud”, In 2010 IEEE 3rd International Conference on Cloud Computing, pages 236–243, July 2010.

[33] http://en.wikipedia.org/wiki/Hardware-assisted_virtual-ization, Accessed on June 13, 2017.