International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 08 | August - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

Estimating Cloud Spot Instances for Elastic Cloud Environments

Pooja Kushwaha[1], Prof. Ashish Tiwari[2]

Abstract: Cloud services are used globally in different kinds of applications. In order to understand the price variation for spot instance prices the prediction algorithms are required. In this context the proposed work introduces based on data driven technique price prediction model. The data mining technique provides us the ability to learn with the historical data trends and identify or recognize the similar trends of the data. This ability makes it essential for various new generation applications. The proposed method is a supervised learning model and implemented using the SVR (support vector regression) model. The proposed method works in three phases. In first phase, the data collection and preprocessing is adopted. In next phase, the sliding window protocol is executed for grouping of similar data patterns. Finally, the training and testing of SVR model is performed for predicting the price of spot instances. The performance of the proposed system is evaluated in terms of the prediction accuracy.

Keywords: Cloud Computing, Elastic Cloud, Spot Instances, Regression Analysis.

I. INTRODUCTION

Spot instance estimation is critically important for elastic cloud application. Elastic Cloud Applications refer to applications that are built and deployed on cloud infrastructure using elastic scaling capabilities. Elasticity is the ability of an application or system to automatically scale its resources up or down based on demand [1]-[2].

When building applications on the cloud, developers can leverage the elasticity of the underlying infrastructure to handle varying workloads efficiently [3]. This elasticity is particularly useful for applications that experience fluctuating traffic or demand patterns. Instead of provisioning fixed resources, elastic applications can dynamically allocate and deallocate resources as needed, ensuring optimal performance and cost-efficiency [4]. Elastic cloud applications often make use of cloud-native technologies and services, such as containers and server less computing, to facilitate scalability and manage resources effectively. Here are a few key concepts related to elastic cloud applications: Auto Scaling: This feature allows cloud platforms to automatically adjust the number of resources (e.g., virtual machines, containers) allocated to an application based on predefined rules or metrics like CPU usage, network traffic, or queue length [5]-[6]. As demand increases, additional resources are provisioned, and as demand decreases, excess resources are terminated. Load Balancing: Elastic applications often utilize load balancers to distribute incoming network traffic across multiple instances or nodes of the application. Load balancers help optimize resource usage and ensure high availability and scalability by directing traffic to the most suitable resources [7]. Serverless Computing: Serverless architectures enable developers to focus on writing application code without having to manage the underlying infrastructure. With serverless platforms, developers can deploy functions or microservices that automatically scale up or down based on demand, executing only when triggered and paying only for actual usage [8].

Containerization: Containers, such as Docker containers, provide a lightweight and portable runtime environment for applications. They encapsulate the application and its dependencies, making it easier to

deploy and scale across different environments. Container orchestration platforms like Kubernetes can automate the management and scaling of containerized applications [9]. Thus, elastic cloud applications offer scalability, flexibility, and cost-efficiency by leveraging the elastic nature of cloud infrastructure. They enable organizations to handle varying workloads effectively and adapt to changing demands without overprovisioning or under provisioning resources [10].

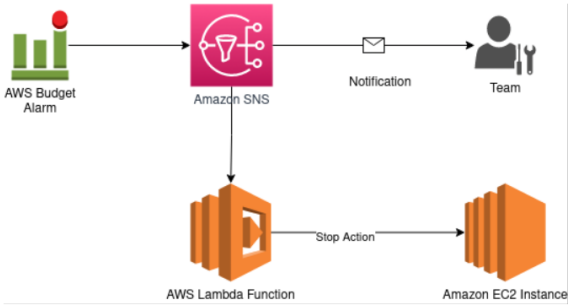

Fig.1 Illustration of Amazon Spot Instances

Benefits of using spot instances include cost savings, as the prices are often significantly lower than on-demand instances, and the ability to access additional compute capacity during times of surplus availability [11]. However, it's important to note that spot instances are not suitable for all types of workloads. Applications that require continuous availability, have strict time constraints, or cannot tolerate interruptions may be better served by using on-demand or reserved instances that offer fixed pricing and guaranteed availability [12].

Each cloud provider has its own implementation and terminology for spot instances, so it's recommended to refer to the specific documentation of your chosen cloud provider for more detailed information and guidelines on using spot instances effectively [13].

II. ESTIMATING CLOUD SPOT INSTANCES

Several approaches have been developed based on data models for estimating cloud spot instances. Predicting cloud spot instances can be approached using data-driven models that analyze historical spot instance pricing data and other relevant features. Here's an outline of a potential approach for building such models [14]-[15]:

Data Collection: Gather historical spot instance pricing data from the cloud provider's APIs or other available sources. The data should include information about spot prices, timestamp, instance type, region, and any other relevant features that might impact pricing.

Feature Engineering: Extract useful features from the collected data that could potentially influence spot instance prices. These features may include time of day, day of the week, seasonality patterns, instance type, region, availability zone, network traffic, and any other relevant factors. Additionally, you can consider incorporating external data sources, such as market trends, news, or economic indicators that may impact spot prices.

Data Preprocessing: Cleanse and preprocess the data to handle missing values, outliers, and any inconsistencies. Perform normalization or scaling if necessary to ensure all features are on a comparable scale.

Model Selection: Choose a suitable data-driven model for predicting spot instance prices. Several options include [16]-[17]:

Regression Models: Linear regression, polynomial regression, or other regression techniques can be used to model the relationship between the features and spot instance prices.

Time Series Models: Spot instance pricing data often exhibit temporal dependencies and patterns. Time series models such as ARIMA (AutoRegressive Integrated Moving Average), SARIMA (Seasonal ARIMA), or more advanced models like LSTM (Long Short-Term Memory) can be employed to capture these patterns.

Machine Learning Models: Gradient Boosting, Random Forest, or Support Vector Regression (SVR) can be utilized to build predictive models by training on historical data.

Deep Learning Models: If you have a large amount of data, neural network architectures like Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), or Transformer models can be employed to capture complex relationships and patterns.

Model Training and Evaluation: Split the collected data into training and testing sets. Train the chosen model on the training data and evaluate its performance on the testing data. Common evaluation metrics include mean absolute error (MAE), root mean square error (RMSE), or mean squared error (MSE).

Model Tuning and Optimization: Fine-tune the model parameters and hyperparameters to improve its performance. Techniques like cross-validation or grid search can be employed to find the optimal set of parameters.

Prediction and Deployment: Once the model is trained and evaluated, it can be used for predicting future spot instance prices based on new input data. Deploy the model in a production environment where it can receive new data and generate predictions as needed.

It's important to note that predicting spot instance prices accurately can be challenging due to the dynamic and volatile nature of the spot market. Factors such as supply and demand, instance availability, and changes in the cloud provider's pricing algorithms can all impact spot prices. Therefore, continuous monitoring and retraining of the model may be necessary to maintain its accuracy and effectiveness over time.

III. PROPOSED MODEL FOR ESTIMATING CLOUD SPOT INSTANCES

The proposed model uses the SVR model for estimating cloud spot instances. The support vector regression (SVR) model is a modified version of the support vector machine (SVM) with a modification in the objective or loss function. The advantages of support vector machines are [18]

1) Effective in high dimensional spaces.
2) Still effective in cases where number of dimensions is greater than the number of samples.
3) Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
4) Versatile: different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation (see Scores and probabilities, below).The classification using the support vector machine (SVM) is depicted in figure 2.

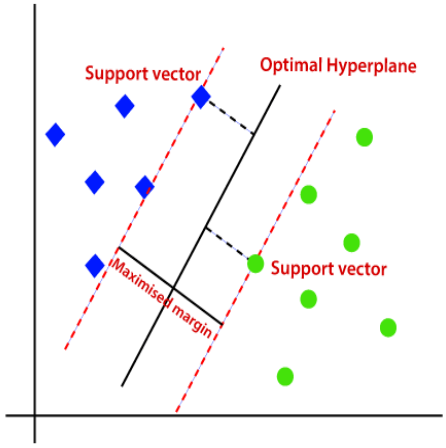

Fig. 2 The SVM model [8]

The support vector regression can be designed as a least squares optimization (LS optimization) as:

for (i=1 : n)

{

Update weights and bias

and

$$\boldsymbol{Minimize}\{\frac{e_1^2+e_2^2+\cdots\ldots\ldots e_n^2}{n}\} \quad \textbf{(1)}$$

}

The SVR can be used for both linearly separable and non-linearly separable data.

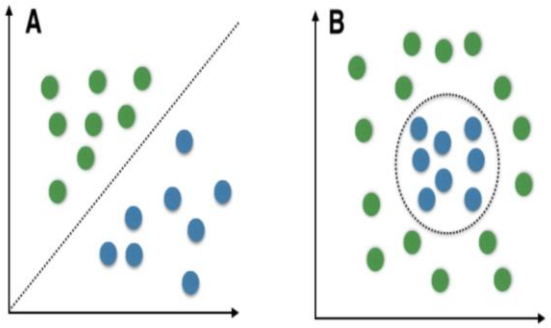

Fig.3 The linearly and non-linearly separable data

The least squares minimization approach is the fastest and most stable approach to convergence. The iterative update of the support vectors keeps changing the bias and weights to minimize the least squares objective function. The Kernalized SVR is modelled as:

To analyze non-linear data sets, linear SVR model is not applicable. Hence, Kernelized SVR is needed. The Kernel is typically a non-linear function. The Radial Basis Function Kernel (RBF) is the similarity between two points in the transformed feature space. Mathematically the SVR-RBF is defined as [19] :

$$\boldsymbol{K(X, X') = e^{-\gamma|X-X'|^2}} \quad \textbf{(2)}$$

$$\boldsymbol{\gamma = \frac{1}{2\sigma}} \quad \textbf{(3)}$$

Here,

$\boldsymbol{\gamma}$ is called the free parameter of RBF

$\boldsymbol{\sigma}$ is called the feature factor

K represents the RBF Kernel

$\boldsymbol{X\ and\ X'}$ are the samples in an input feature space

$\boldsymbol{|X - X'|}$ is termed as the Euclidean Distance

The loss function is computed as:

The support vector regression can be designed as a least squares optimization (LS optimization) as:

Minimize $\{\frac{1}{n}\Sigma(predicted\ value - actual\ value)^2\}$

Or

Minimize$\{\frac{1}{n}\Sigma(error)^2\}$

The performance metrics of the machine learning based classifier is generally done based on [2].

The parameters which can be used to evaluate the performance of the ANN design for time series models is given by:

1) Mean Absolute Error (MAE)
2) Mean Absolute Percentage Error (MAPE) and
3) Mean square error (MSE)

The above mentioned errors are mathematically expressed as:

$$\boldsymbol{MAE} = \frac{1}{N}\sum_{t=1}^{N} |V_t - \widehat{V}_t| \quad \textbf{(4)}$$

Or

$$\boldsymbol{MAE} = \frac{1}{N}\sum_{t=1}^{N} |e_t| \quad \textbf{(5)}$$

$$MAPE = \frac{100}{N}\sum_{t=1}^{N}\frac{|V_t - \hat{V}_t|}{V_t} \quad (6)$$

The mean square error (MSE) is given by:

$$MSE = \frac{1}{N}\sum_{t=1}^{N} e_t^2 \quad (7)$$

Here,

N is the number of predicted samples
V is the predicted value
$\hat{V}_t$ is the actual value
e is the error value

IV. SIMULATION RESULTS

The simulation results have been performed on MATLAB for the ease of mathematical analysis.

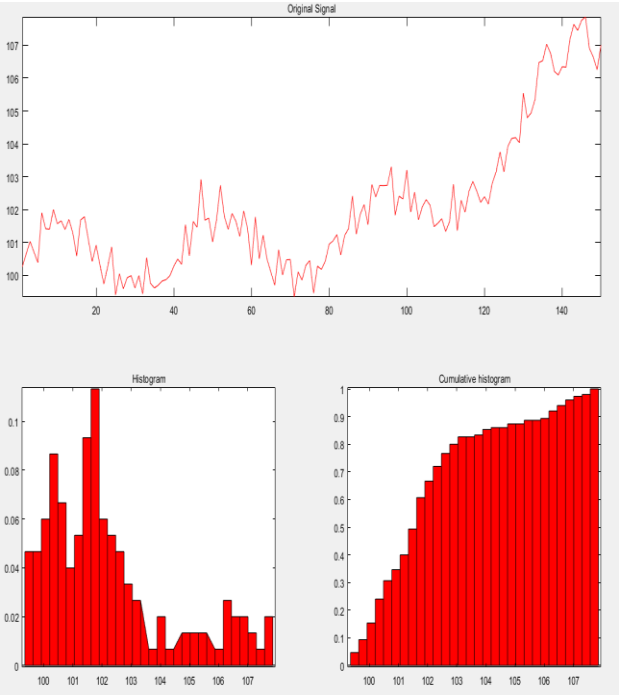

Fig. 4 Original Histogram

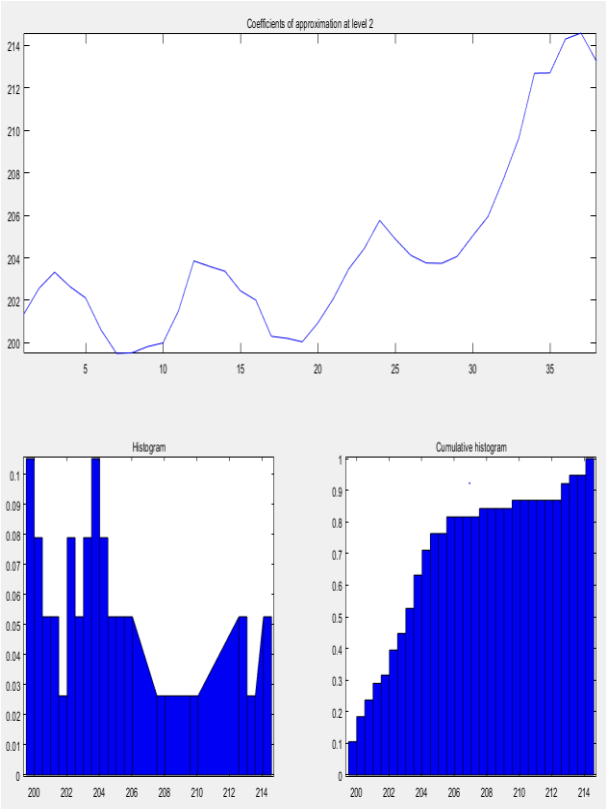

Fig. 5 Histogram of Approximates

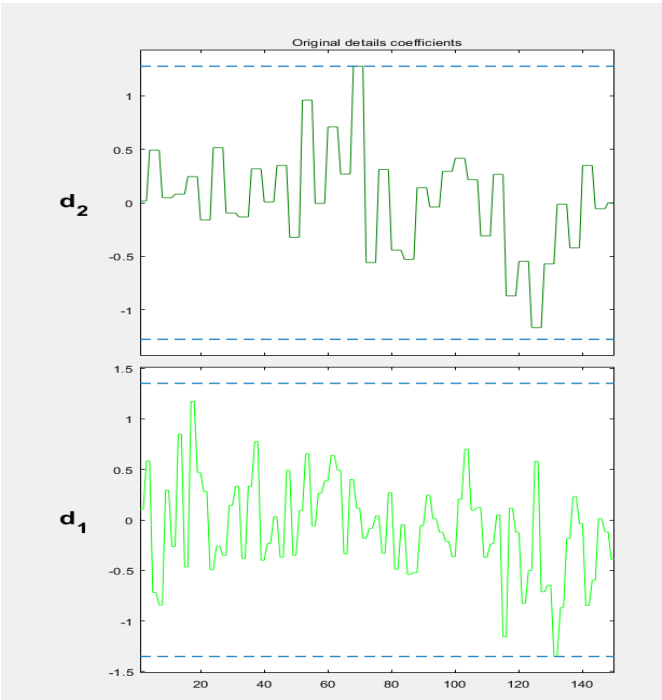

Fig. 6 Histogram of details

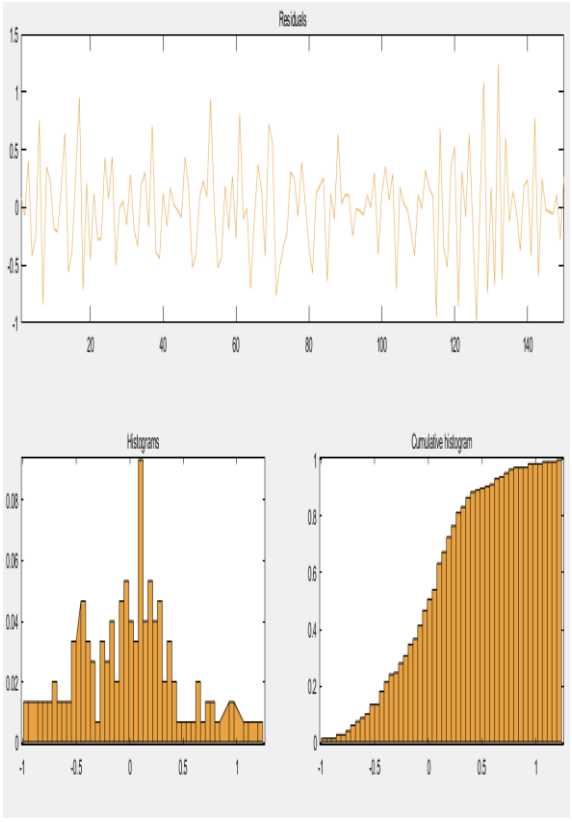

Fig. 7 Residual Histograms

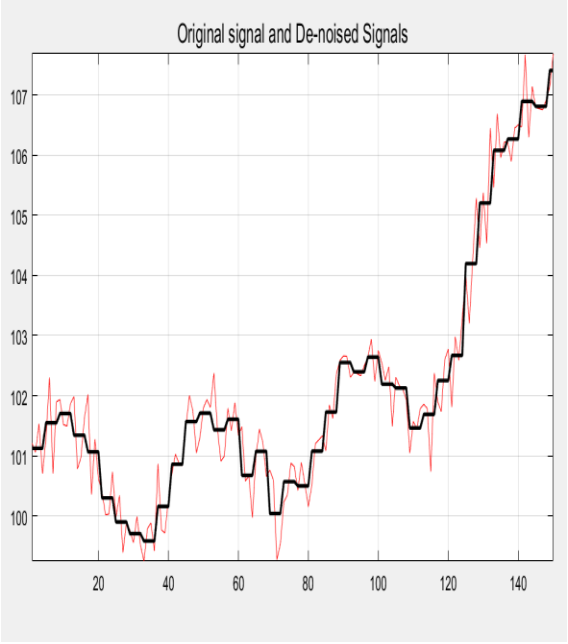

Fig. 8 Data post filtration

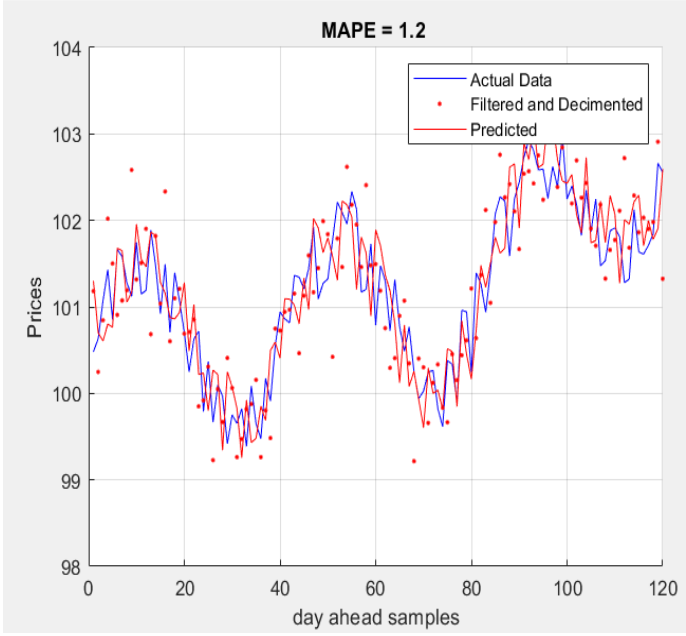

Fig. 9 Estimations for spot instances

The histogram analysis of the data allows in understanding the effect of the wavelet based filtering and eventual impact on the spot instance estimation. The mean absolute percentage error for the proposed work is around 1% which renders an accuracy of around 99% which is quite high for volatile spot instance estimation.

Conclusion:

As cloud applications especially elastic cloud keep growing, estimating cloud spot instances is exceedingly important. Understanding the many elements that affect cost, putting good bidding techniques into practice, and preventing interruptions are all necessary for estimating cloud spot instances. Users may minimize disruptions and optimize costs while maximizing the benefits of spot instances by utilizing historical pricing analysis, real-time monitoring, and intelligent bidding techniques. For budget-conscious organizations looking for cost savings without sacrificing

workload performance and scalability, cloud spot instances present an appealing solution. The proposed work presents an SVR based approach for estimating cloud spot instances which attains an error percentage of 1% approximately. Thus the proposed model is shown to be effective in estimating cloud spot instances with high accuracy.

References:

1. SS Nezamdoust, MA Pourmina, F Razzazim, " Optimal prediction of cloud spot instance price utilizing deep learning" The Journal of Supercomputing, Springer 2023, vol.79, pp.–7647.
2. S. Lee, J. Hwang and K. Lee, "SpotLake: Diverse Spot Instance Dataset Archive Service," 2022 IEEE International Symposium on Workload Characterization (IISWC), Austin, TX, USA, 2022, pp. 242-255.
3. D. Katayama, K. Kasai and T. Koita, "Migration Destination Selection Algorithm for Spot Instances using SPS," 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan, 2022, pp. 6690-6692
4. A. C. Zhou, J. Lao, Z. Ke, Y. Wang and R. Mao, "FarSpot: Optimizing Monetary Cost for HPC Applications in the Cloud Spot Market," in IEEE Transactions on Parallel and Distributed Systems, 2021, vol. 33, no. 11, pp. 2955-2967
5. G. J. Portella, E. Nakano, G. N. Rodrigues, A. Boukerche and A. C. M. A. Melo, "A Novel Statistical and Neural Network Combined Approach for the Cloud Spot Market," in IEEE Transactions on Cloud Computing, 2023 vol. 11, no. 1, pp. 278-290.
6. 6. Liu D, Cai Z, Lu Y (2019) Spot price prediction based dynamic resource scheduling for web applications. In: 2019 Seventh International Conference on Advanced Cloud and Big Data (CBD). IEEE, pp 78–83 7.
7. Varshney P, Simmhan Y (2019) AutoBot: Resilient and cost-effective scheduling of a bag of tasks on spot VMs. IEEE Trans Parallel Distrib Syst 30(7):1512-1527
8. Sharma P, Lee S, Guo T, Irwin D, Shenoy P (2017) Managing risk in a derivative IaaS cloud. IEEE Trans Parallel Distrib Syst 29(8):1750-1765
9. Mishra AK, Yadav DK (2017) Analysis and prediction of Amazon EC2 spot instance prices. Int J Appl Eng Res 12(21):11205– 11212
10. Teylo L, Arantes L, Sens P, Drummond LM (2019) A bag-of-tasks scheduler tolerant to temporal failures in clouds. In: 2019 31st International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD). IEEE, pp. 144–151
11. Khandelwal V, Chaturvedi AK, Gupta CP (2020) Amazon EC2 spot price prediction using regression random forests. IEEE Trans Cloud Comput 8(1):59–72
12. Khashei M, Bijari M (2011) A novel hybridization of artificial neural networks and ARIMA models for time series forecasting. Appl Soft Comput 11(2):2664-2675.
13. Liu Y, Wang Z, Zheng B (2019) Application of regularized GRU-LSTM model in stock price prediction. In: 2019 IEEE 5th International Conference on Computer and Communications (ICCC). IEEE, pp 1886-1890
14. Abbasimehr H, Shabani M, Yousefi M (2020) An optimized model using LSTM network for demand forecasting. Comput Ind Eng 143(106435):1-13.
15. DAI G, MA C, XU X (2019) Short-term traffic flow prediction method for urban road sections based on space-time analysis and GRU. IEEE Access 7(1):143025-143035
16. Song J, Tang S, Xiao J, Wu F, Zhang Z (2016) LSTM-in-LSTM for generating long descriptions of images. Comp Visual Media 2(4):379–388.
17. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15(1):1929-1958.
18. Wang Z, Zhu R, Zheng M, Jia X, Wang R, Li T (2019) A regularized LSTM network for short-term traffic flow prediction, In: 2019 6th International Conference on Information Science and Control Engineering (ICISCE). IEEE, pp 100-105

19. Singh VK, Dutta K (2015) Dynamic price prediction for Amazon spot instances, In: 2015 48th Hawaii International Conference on System Sciences (HICSS). IEEE, pp 1513–1520.

20. F. J. Baldan, S. Ramirez-Gallego, C. Bergmeir, F. Herrera and J. M. Benitez, "A Forecasting Methodology for Workload Forecasting in Cloud Systems," in IEEE Transactions on Cloud Computing, 2018, vol. 6, no. 4, pp. 929-941