Spot-On: Predictive Modeling and Comparative Analysis of AWS EC2 Spot Instance Pricing

Sanket Hanumant Veer, Manoj Tumkur Shivashankar

School of Computing, University of Georgia,
Athens, GA, USA

sankethanumant.veer@uga.edu

manoj.tumkurshivashankar@uga.edu

***Abstract*— Spot instances, a cost-effective and flexible computing option offered by cloud providers such as Amazon Web Services (AWS), Microsoft Azure, etc., enable users to bid on idle and surplus computing resources. However, predicting the fluctuating spot instance prices is crucial for optimizing resource utilization. This project addresses the challenge of implementing and comparing multiple prediction models for Amazon EC2 spot price instances, considering historical data before and after the 2018. The primary objectives include spot price prediction, identification of the most effective model, and measurement of the predictability of Amazon spot-price data. By evaluating the accuracy and performance metrics of these models, the study provides scientific analysis and reasoning to enhance our understanding of their efficiency or inefficiency in predicting spot prices. The insights derived from this research contribute to the broader conversation on resource optimization in cloud computing environments, offering valuable guidance for users seeking to leverage spot instances effectively.**

Key words: AWS, GRU, LSTM, Random Forest, Spot instances, XGBoost,

I. Introduction

Nowadays, the increase in the amount of data generated at every second has posed a lot of problems in managing the same[20]. Performing computational tasks on enormous data demands lot high-capacity computational resources. This has made the user to migrate towards the cloud which offers different Flavours of resources and help in pacing up the execution. Spot Instances offered by Amazon EC2 are the computational units which offers different Virtual machine, Operating systems, CPU, memory, and networking capacity. Users enjoys the freedom of selecting the instances which they desire and can utilize these resources for their computational tasks.

The prices of the spot instances vary dynamically based on the demand and supply in the market [21]. When the demand is less the prices will be lowered and when the demand more the prices will be expensive. These spot instances are made available to the users on auction(bidding) mechanism. Users can buy the desired instances for some amount of cost and use them whenever the price of that specific instance is less than their auctioned value. Users will be given a notification soon after the instance price goes over the bid price and their connection for that instance will be terminated. It will be only available when the price of the instance goes below the auctioned price.

The termination of the spot instance availability due the demand in market is termed as out-of-bid event.

Out-of-bid events has lot of caused problems on the processes running on the instances. Sometimes users must restart the entire process from the initial stage because of the loss of computational resources. And this happens frequently and randomly as the prices of the spot instance varies in unexpected intervals depending on demand in market for instances.

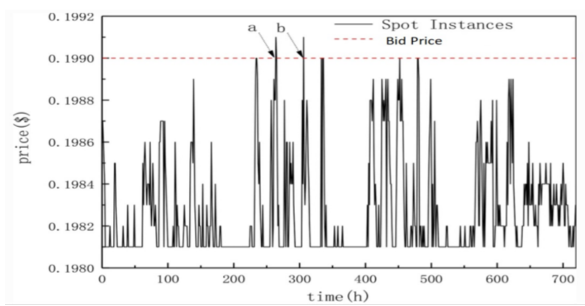

Fig.1 [Ref. 4]: Spot price in us-east-2a c4.xlarge instance (Oct, 27 to Nov, 26 2017)

In Fig.1, an example of a bidding process for a specific spot instance VM (i.e., c4.xlarge) in US-east-2a region is displayed. Here, vertical axis denotes the price of the spot instance and horizontal axis represents the time frame. Dashed red line is the bidding price set by the user and solid black line denotes the spot instance price at that instance of time. The points (a) and (b) are the situations where the user will lose the spot instance since the spot instance price overshoots the bidding price.

It is very challenging to find the desired spot instance within the auctioned price due to volatile nature of the market where the prices of the spot instances change instantaneously in unexpected manner. This will cause lot interruptions and out-of-bid events. In order to solve these issues, our projects aim in implementing few predicting machine learning models using the previous datasets so that it will give information to the users about the prices of the spot instance and their availability when users intend to use them. The prediction of the spot instance price helps in strategic planning in reducing the cost spent on the instances and helps in selecting the best or desired instances for less amount of money. It also enhances the reliability of the tasks running on the instances as users will be aware of the price variations of the instance which they will be using at that instant.

This project addresses the challenge of predicting Amazon EC2 spot prices both before and after the 2018 pricing mechanism change by implementing and comparing multiple prediction models based on historical data. The primary objectives include developing diverse machine learning and statistical models, identifying the most accurate model, and measuring the predictability of Amazon spot-price data through quantitative performance metrics. The evaluation will consider the models' adaptability to the pricing mechanism change and assess their strengths and limitations in capturing spot price fluctuations. The project aims to provide a scientific analysis and reasoning for the efficiency or inefficiency of these models, contributing valuable insights for cloud users and providers in optimizing resource allocation and cost-effectiveness. The documentation will comprehensively detail the modeling process and present findings through clear reports with visualizations and statistical summaries.

II. Literature Review

The landscape of predicting Amazon EC2 spot prices is characterized by a diverse array of approaches and models, spanning various techniques from machine learning, time series analysis, and game theory[1]. Predominantly, research in this domain has been prolific before 2018, showcasing a range of methodologies such as time series analysis, linear regression, decision trees, and more complex models like the combination of Autoregressive Integrated Moving Average (ARIMA) and Support Vector Regression (SVR)[3]. Notably, the latest pricing mechanisms have spurred a new wave of research, with a limited yet emerging number of models published in 2021. These recent models exhibit a keen interest in leveraging advanced techniques, including the fusion of convolutional neural networks (CNNs) and long short-term memory (LSTM) networks [7].

The paper [2], centres around the application of time series analysis for predicting Amazon EC2 spot instance prices. The authors specifically emphasize the critical aspect of short-term prediction accuracy, highlighting the importance of precise forecasting in a dynamic and fluctuating cloud computing environment. By employing time series analysis, the paper aims to capture and comprehend the temporal patterns inherent in EC2 spot instance pricing, offering insights into the short-term dynamics of the market[2]. This research provides a valuable contribution to the understanding of time-dependent trends in EC2 spot pricing, addressing the challenges associated with real-time decision-making and resource allocation in cloud computing scenarios.

In the realm of predicting Amazon EC2 spot instance prices, various machine learning techniques have been explored to capture the intricate pricing dynamics [1]. A noteworthy paper in this domain is "CloudSpotter: An Autonomic System for Efficient Spot Instance Management in the Cloud,"[1] delves into the application of linear regression to model spot instance prices. The study employs linear regression as a foundational technique to understand and predict the linear relationships within spot pricing, offering valuable insights into the cost implications for cloud users. Moreover, decision trees and random forests have also been investigated in the literature, showcasing their efficacy in handling the complexities of cloud pricing [1]. These techniques have been employed in various studies, illustrating their potential for capturing non-linear patterns and enhancing the accuracy of predictions in the context of Amazon EC2 spot instance pricing.

In contribution of [3] the authors introduced an innovative approach to enhance the prediction accuracy of Amazon EC2 spot instance prices. The authors advocate for a hybrid time series model that intricately combines the autoregressive integrated moving average (ARIMA) and support vector regression (SVR) approaches. By integrating these two distinct methodologies, the model aims to leverage the strengths of each, capturing both linear and non-linear patterns inherent in EC2 spot pricing dynamics. This hybridization signifies a nuanced understanding of the complexities associated with cloud computing pricing, providing a comprehensive framework for more accurate and robust predictions[3].

In their contribution titled "A Hybrid Deep Learning Approach for Amazon EC2 Spot Price Prediction,"[7]. The proposed approach embraces a hybrid deep learning strategy, skilfully integrating Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. By combining these two powerful neural network architectures, the authors aim to leverage the spatial and temporal dependencies inherent in EC2 spot price data, thereby enhancing the predictive accuracy of their model[7]. The research contributes to the evolving landscape of predictive modelling for Amazon EC2 spot prices, offering a hybridized deep learning solution poised to tackle the intricacies of this dynamic and volatile domain.

III. Methodology

In this section, we present the detailed methodology involved in development and implementation of the approach which is further divided into 5 sub-sections. These subsections will provide detailed information about the Approach Design, Data Selection and Filtration, machine learning models that were used in this project like Random Forest [8], LSTM [9], XGBoost [10] and GRU [11].

A. Approach Design

The approach design of the project is presented in Fig. 2 in simplified format.

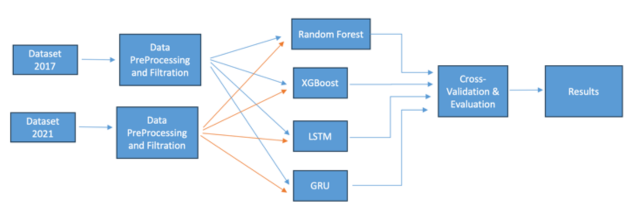

Fig. 2: Approach Design

AWS launched a new pricing model in November 2017 to simplify spot purchasing experience for the users [12]. The pricing model before this change and after this change led to a significant variation in the price range of the spot instances. To analyse this change, we used two datasets. One from 2017 January to October i.e., before the mechanism change and another from 2021 i.e., after the pricing model change.

The approach design is mainly classified into 5 tasks:

a) Data Selection
b) Data Pre-processing and filtration
c) Building predicting model and training
d) Cross-Validation and Evaluation
e) Results Analysis

As mentioned, Amazon EC2 price datasets of years 2017 and 2021 are selected to train and test the machine learning models.

B. Data Pre-processing and filtration

For the data pre-processing and filtration phase of this project, the focus is on refining the dataset to enhance the efficiency of the prediction models. Initially, the dataset, specific to the US-East-1 region, was extracted, comprising essential columns such as Date-Time, Price, Region, Operating System, and Virtual Machine. To streamline the analysis, the project narrows down its scope by selecting Linux/Unix as the Operating System and m2_4xlarge as the Virtual Machine. This strategic filtering process ensures a more targeted examination of relevant data for spot price prediction. Furthermore, to enable effective model training and testing, the Date-Time information is encoded into separate columns representing year, month, date, hour, minute, and second. This temporal breakdown aims to capture the temporal patterns inherent in spot prices. The Price column is identified as the target attribute, while the remaining columns (Region, Operating System, Virtual Machine, Date-Time breakdown) are considered as features. This meticulous data pre-processing step sets the stage for a more focused and meaningful analysis. Notably, the dataset statistics reveal a significant variation between 2017 and 2021, with a shift in size, unique values, and mean spot prices, underlining the importance of adapting the prediction models to evolving data trends over time.

C. Building predicting model and training

In this study, a comprehensive approach is adopted to build predictive models with high accuracy for forecasting Amazon EC2 spot prices. The chosen models encompass a diverse set of techniques, each renowned for its efficacy in capturing distinct patterns within the data.

1) Random Forest: The Random Forest [8] model is implemented to harness the power of ensemble learning. This technique combines the outputs of multiple decision trees to produce a single, robust prediction. The decision trees collectively contribute to the model's ability to handle complex relationships and interactions within the Amazon EC2 spot price dataset. Parameters such as the number of trees, tree depth, and feature subsets are carefully tuned to optimize the model's performance.

2) XGBoost: XGBoost [9], an optimized distributed gradient boosting library, is employed for its efficiency and flexibility. This model is designed to handle large datasets and complex relationships, making it particularly suitable for the dynamic and intricate nature of Amazon EC2 spot prices. Hyperparameter tuning is conducted to enhance the model's adaptability, ensuring it captures subtle variations and trends in spot prices effectively.

3) *LSTM(Long-Term-Short-Memory):* The Long Short-Term Memory (LSTM) [10] network, a type of recurrent neural network with specialized memory cells, is integrated into the predictive modelling framework. LSTMs are well-suited for capturing long-term dependencies in sequential data, and in the context of spot price prediction, they are expected to excel in recognizing patterns influenced by historical pricing trends. Similar to other models, hyperparameter optimization is conducted to fine-tune the LSTM architecture for optimal performance.

4) *GRU (Gated Recurrent Unit):* Gated Recurrent Units (GRUs) [11], a gating mechanism in recurrent neural networks (RNNs), are employed to capture temporal dependencies in the time-series data of spot prices. The GRU architecture is particularly suitable for handling sequences with long-range dependencies, making it an ideal candidate for time-series prediction. These models excel in capturing sequential dependencies in time-series data, allowing them to discern intricate patterns in the historical spot price information. The hyperparameters, such as the number of units and learning rates, are fine-tuned to optimize the performance of GRU.

All the above four models are built and trained for both datasets for the selected region i.e., US-East-1 region a, b, c, and d.

D. Cross-Validation and Evaluation

In this study, a robust cross-validation and evaluation methodology is employed to rigorously assess the performance of various predictive models. Cross-validation is pivotal in ensuring the reliability and generalization of the models by systematically partitioning the dataset into training and testing sets across multiple iterations. This process minimizes the risk of overfitting and provides a more accurate representation of the models' predictive capabilities on unseen data. The chosen evaluation metrics, including Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE), offer a comprehensive view of the models' accuracy, precision, and overall predictive performance. MSE quantifies the average squared differences between predicted and actual values, while MAPE provides insights into the relative accuracy of the models by calculating the percentage difference between predicted and actual values. RMSE, on the other hand, assesses the overall precision by measuring the square root of the average squared differences. The combination of these metrics ensures a multifaceted evaluation, enabling the identification of the model or models that best align with the dynamic nature of AWS EC2 spot instance pricing. This meticulous cross-validation and evaluation process is fundamental to deriving meaningful conclusions and informing decision-making in the context of optimizing resource allocation and cost-effectiveness in cloud computing environments.

E. Results Analysis

In the results analysis phase of this project is a comprehensive examination of the evaluation metrics for all four predictive models (i.e., Random Forest, XGBoost, LSTM and GRU) is conducted. Metrics such as Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE) are scrutinized to gauge the accuracy and precision of each model in predicting AWS EC2 spot instance prices. The comparison encompasses a detailed exploration of how well the models align with the actual spot instance values, providing valuable insights into their reliability across various scenarios. Additionally, a visual analysis is performed by comparing the predicted spot instance prices against the actual values for each model, offering a tangible representation of their predictive capabilities. This thorough examination aims to not only identify the model that consistently outperforms others in terms of predictive accuracy but also to elucidate potential strengths and weaknesses in handling the dynamic nature of spot pricing. The result analysis serves a crucial part in measuring the predictability of the given model for the given two datasets,

IV. Experimentation and Results

A. Experimentation Setup

The experimentation focuses on the datasets from years 2017 and 2021 taken from [13], capturing diverse data trends before and after the significant pricing mechanism change in 2018. The analysis is specifically confined to the US-East-1 region, encompassing sub-regions a, b, c, and d to ensure a comprehensive representation of the AWS EC2 spot instance pricing dynamics.

The dataset is further refined by narrowing down the Operating System (OS) to Linux/UNIX and the Virtual Machine (VM) type to m2_4xlarge. This strategic filtration ensures a targeted examination of spot instance prices influenced by specific OS and VM configurations. The experimental platform selected for the study is Google Colabotary, providing a cloud-based environment conducive to collaborative Python-based research. Google Colabotary offers a versatile and scalable platform, allowing seamless integration of various machine learning libraries and frameworks necessary for the implementation and evaluation of the predictive models.

The programming language of choice for this experimentation is Python, owing to its widespread adoption in the field of machine learning and data analysis. Python provides a rich ecosystem of libraries, including but not limited to scikit-learn[14], TensorFlow[15], and PyTorch[16], which are essential for building and evaluating diverse predictive models. Scikit-learn, a Python library, provides a range of supervised and unsupervised learning methods. It relies on familiar technologies such as NumPy[17], pandas[18], and Matplotlib[19] that you may already be familiar with. A common evaluation metric chosen is Mean Absolute Percentage Error (MAPE) to compare the results obtained from various models.

B. Experimentation using Random Forest

Sklearn[14] is a Python library for machine learning and data analysis. It provides various tools and methods to fit, train, and predict models on data. The model RandomForestRegressor is implemented using this library. Choosing appropriate hyperparameters, the model is trained on 2017 and 2021 data separately. Fig. 4 (a) shows the prediction results for 2017 data for m2_4xlarge VM at US-east-1a region. The blue line denotes the predicted value and orange line denotes the actual value for the testing data. MAPE value for data is 0.17 whereas Fig. 4 (b) shows the prediction results for 2021 data for m2_4xlarge VM at US-east-1a region. MAPE value for data is 0.065

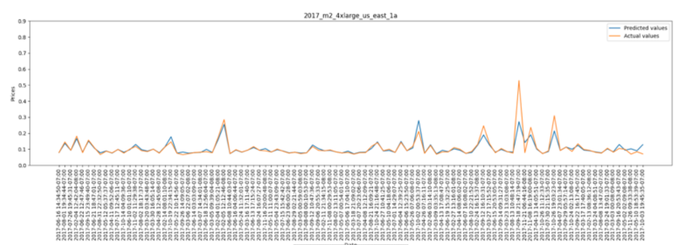

Fig. 4(a): Prediction results for 2017 data for m2_4xlarge VM at US-east-1a region using Random Forest

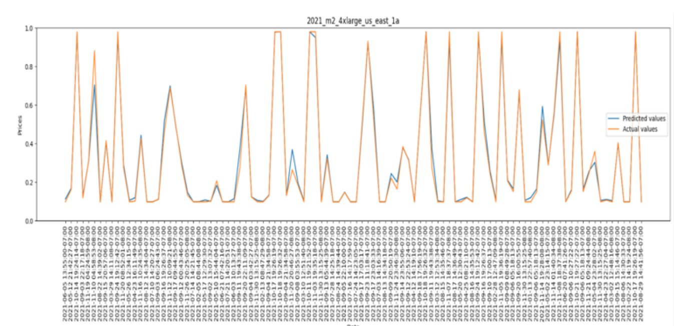

Fig. 4(b): Prediction results for 2021 data for m2_4xlarge VM at US-east-1a region using Random Forest

C. Experimentation using XGBoost

The model XGBoost is extracted from the xgboost library. XGBoost [5] is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. Fig. 5 (a) shows the prediction results for 2017 data for m2_4xlarge VM at US-east-1a region. The blue line denotes the predicted value and orange line denotes the actual value for the testing data. MAPE value for data is 0.11 whereas Fig. 5 (b) shows the prediction results for 2021 data for m2_4xlarge VM at US-east-1a region. MAPE value for data is 0.025

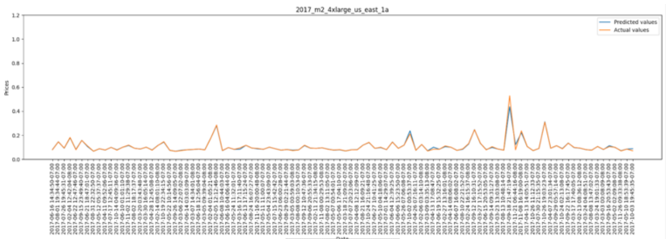

Fig. 5(a): Prediction results for 2017 data for m2_4xlarge VM at US-east-1a region using XGBoost.

Fig. 5(b): Prediction results for 2021 data for m2_4xlarge VM at US-east-1a region using XGBoost.

D. Experimentation using LSTM

The LSTM model is imported using Keras library. Keras[6] is an open-source library that provides a Python interface for artificial neural networks. Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools for working with image and text data to simplify programming in deep neural network area. Choosing appropriate hyperparameters, the model is trained on 2017 and 2021 data separately. Fig. 6 (a) shows the prediction results for 2017 data for m2_4xlarge VM at US-east-1a region. The blue line denotes the predicted value and orange line denotes the actual value for the testing data. MAPE value for data is 0.26 whereas Fig. 6 (b) shows the prediction results for 2021 data for m2_4xlarge VM at US-east-1a region. MAPE value for data is 0.19

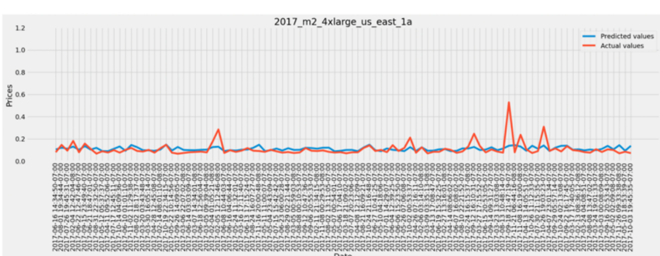

Fig. 6(a): Prediction results for 2017 data for m2_4xlarge VM at US-east-1a region using LSTM

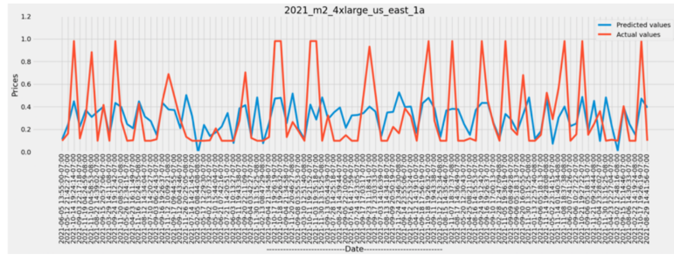

Fig 6(b): Prediction results for 2021 data for m2_4xlarge VM at US-east-1a region using LSTM

E. Experimentation using GRU

The GRU model is also imported from Keras library. Choosing appropriate hyperparameters, the model is trained on 2017 and 2021 data separately. Fig. 7 (a) shows the prediction results for 2017 data for m2_4xlarge VM at US-east-1c region. The blue line denotes the predicted value and orange line denotes the actual value for the testing data. MAPE value for data is 0.32 whereas Fig. 7 (b) shows the prediction results for 2021 data for m2_4xlarge VM at US-east-1c region. MAPE value for data is 0.31

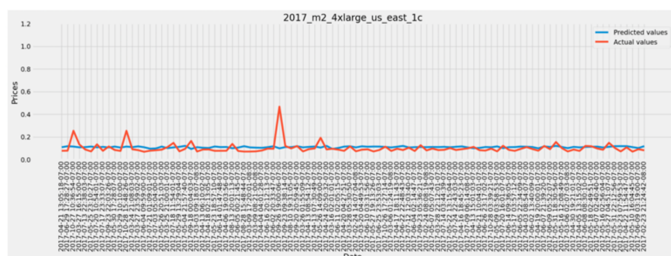

Fig 7(a): Prediction results for 2017 data for m2_4xlarge VM at US-east-1c region using GRU

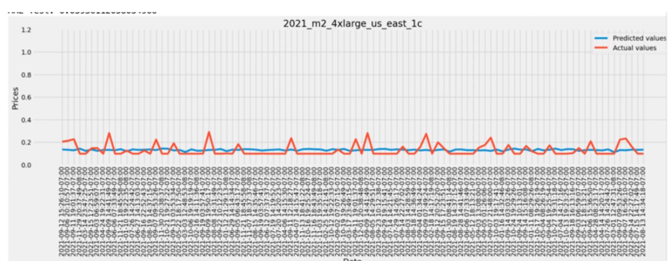

Fig 7(a): Prediction results for 2021 data for m2_4xlarge VM at US-east-1c region using GRU

F. Overall Prediction Analysis

The final comparison of the predicted price Vs actual price by all four models is presented in Fig. 8(a) and (b) where the actual spot instance price for that time stamp is represented using solid blue color line, prediction by XGBoost is given by orange color, prediction by Random Forest, LSTM and GRU is denoted by green, red and purple color solid lines respectively.

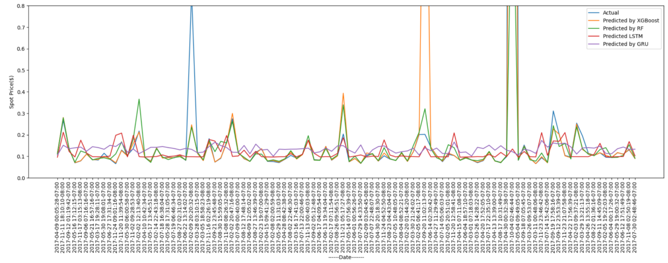

Fig. 8 (a): Prediction Analysis for the 2017 m2-4xlarge-us-east 1b dataset.

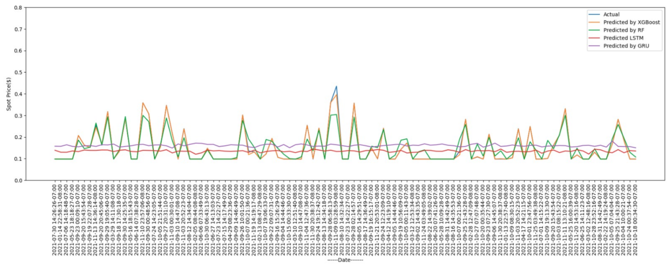

Fig. 8 (b): Prediction Analysis for the 2021 m2-4xlarge-us-east 1b dataset.

The MAPE value comparison table for all models with all the regions is present in Fig 9.

2017 Results table (MAPE values)

	Region a	Region b	Region c	Region d
RF	0.17	0.21	0.29	0.11
XGBoost	0.11	0.18	0.24	0.09
LSTM	0.26	0.21	0.22	0.24
GRU	0.27	0.29	0.31	0.32

2021 Results table (MAPE values)

	Region a	Region b	Region c	Region d
RF	0.065	0.029	0.009	0.0007
XGBoost	0.025	0.004	0.001	0.0003
LSTM	0.19	0.25	0.11	0.09
GRU	1.30	0.34	0.32	0.13

Fig. 9: MAPE value comparison for the dataset 2017 and 2021 performing predictions using the models Random Forest, XGBoost, LSTM and GRU

V. Discussion

This section highlights crucial insights from the study on predicting AWS EC2 spot instance pricing. Notably, regional variations in spot prices are recognized, emphasizing the importance of incorporating geographical parameters for accurate predictions. Data preprocessing steps, including collection, merging, and filtering, are essential for refining datasets, optimizing model training, and improving overall predictability. Ensemble learning models, specifically XGBoost and Random Forest, exhibit robust performance in capturing complex pricing patterns. Furthermore, the discussion underscores that predictability is intricately linked to dataset characteristics, such as the number of unique values, the influence of outliers, and the range of price values. These findings contribute valuable perspectives on effective predictive modeling for dynamic cloud computing environments, informing resource allocation decisions, and optimizing cost-effectiveness.

VI. Conclusion

In conclusion, the findings from this study underscore notable distinctions in the predictability of Amazon EC2 spot instance prices between the years 2017 and 2021. The analysis reveals that the 2021 dataset exhibits higher predictability compared to its 2017 counterpart, indicative of potential shifts in pricing dynamics or improved model adaptability to more recent patterns. Furthermore, the study highlights the efficacy of Ensemble Learning models, particularly XGBoost and Random Forest, in achieving superior predictive performance. These models demonstrate robustness in capturing intricate patterns within the spot instance pricing data, showcasing their effectiveness in handling the complexities introduced by the dynamic nature of Amazon EC2 spot pricing. As cloud computing environments evolve, the preference for Ensemble Learning techniques becomes evident, offering promising avenues for enhancing the accuracy and reliability of spot price predictions. The insights gained from this research contribute valuable knowledge to the field of cloud resource optimization and aid stakeholders in making informed decisions regarding AWS EC2 spot instance utilization.

Acknowledgment

The authors would like to thank Prof. Dr. Inn Kee Kim and School of Computing at the University of Georgia, Athens, GA, USA for providing the opportunity to conduct this study and to make available the required resources in order to complete the project.

References

[1] L. Wang, M. Korupolu, and S. R. Ufukte, "CloudSpotter: An Autonomic System for Efficient Spot Instance Management in the Cloud," 2012 IEEE 8th International Conference on Autonomic Computing (ICAC).

[2] Iman Keivanloo, Fabiano C. Botelho, and Alan Schmitt, "Time Series Analysis for Amazon EC2 Spot Instance Pricing," IEEE 26th International Symposium on Software Reliability Engineering, 2015.

[3] Bingyang Wei, Cong Xu, Zongpeng Li, and Francis C.M. Lau, "A Hybrid Time Series Model for Amazon EC2 Spot Instance Price Prediction," 2017 IEEE International Conference on Cloud Computing.

[4] Liu, W., Wang, P., Meng, Y. et al. Cloud spot instance price prediction using kNN regression. Hum. Cent. Comput. Inf. Sci. 10, 34 (2020). https://doi.org/10.1186/s13673-020-00239-5

[5] https://xgboost.readthedocs.io/en/stable/

[6] https://en.wikipedia.org/wiki/Keras

[7] Vinay Kumar, Alok Ranjan, and Pradeep K. Atrey, "A Hybrid Deep Learning Approach for Amazon EC2 Spot Price Prediction," 2019 IEEE International Conference on Big Data.

[8] Ho, T. K. (1995). Random decision forests. In Proceedings of 3rd international conference on document analysis and recognition (Vol. 1, pp. 278–282).

[9] Hochreiter, S. & Schmidhuber, J"urgen, 1997. Long short-term memory. Neural computation, 9(8), pp.1735–1780

[10] Chen, T. & Guestrin, C., 2016. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '16. New York, NY, USA: ACM, pp. 785–794. Available at: http://doi.acm.org/10.1145/2939672.2939785.

[11] R. Dey and F. M. Salem, "Gate-variants of Gated Recurrent Unit (GRU) neural networks," 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, USA, 2017, pp. 1597-1600, doi: 10.1109/MWSCAS.2017.8053243.

[12] https://aws.amazon.com/blogs/aws-cloud-financial-management/ad-everything-you-need-to-know-about-aws-billing-conductors-new-pricing-model/

[13] https://www.isi.edu/~calvin/data/ec2-spot-price/

[14] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011

[15] Abadi, Mart'in et al., 2016. Tensorflow: A system for large-scale machine learning. In 12th $USENIX$ Symposium on Operating Systems Design and Implementation ($OSDI$ 16). pp. 265–283.

[16] Paszke, A. et al., 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Advances in Neural Information Processing Systems 32. Curran Associates, Inc., pp. 8024–8035. Available at: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[17] Harris, C.R. et al., 2020. Array programming with NumPy. Nature, 585, pp.357–362.

[18] McKinney, W. & others, 2010. Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference. pp. 51–56.

[19] Hunter, J.D., 2007. Matplotlib: A 2D graphics environment. Computing in science & engineering, 9(3), pp.90–95.

[20] https://hbr.org/2012/10/big-data-the-management-revolution

[21] https://aws.amazon.com/ec2/spot/pricing/#:~:text=With%20Spot%20Instances%2C%20you%20pay,the%20Spot%20Instance%20history%20page.