IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.67
Volume 5, Issue 7, March 2025

Comparative Analysis of Performance in Cloud Computing Platforms Considering Memory and Process Level Metrics with CPU Utilization

Harshita Shrivastava[1] and Pranjal Khare[2]
Research Scholar, Babulal Tarabai Institute of Research and Technology, Sagar, India[1]
Head of the Department, Dept. of Computer Science &Engg
Babulal Tarabai Institute of Research and Technology, Sagar, India[2]
harshita87shrivastava@gmail.com and pranjal.khare@btirt.ac.in

Abstract: *Cloud computing has revolutionized the way organizations and individuals access and manage their computing resources. With the increasing number of cloud service providers, it has become crucial to assess the performance of various platforms to ensure optimal resource allocation. In this comparative analysis, we focus on memory, process-level metrics, and CPU utilization, as these factors significantly impact the overall performance of cloud computing platforms. By examining these metrics across multiple platforms, we aim to provide valuable insights into their respective strengths and weaknesses, aiding users in making informed decisions regarding resource allocation and platform selection.Cloud computing has emerged as a fundamental technology for businesses and individualsworkloads. This study aims to compare the performance of various cloud computing platforms, focusing on memory, process-level metrics, and CPU utilization.Memory is a critical component in cloud computing, by measuring these factors across multiple platforms, we can identify the platforms that offer efficient memory management, resulting in enhanced application performance. Process-level metrics helps us understand how platforms handle concurrent processes and their impact on overall performance. Based on our comparative analysis, we find thathighlight the diversity in performance strengths across different cloud computing platforms.We conducted a comparative analysis of memory, process-level metrics, and CPU utilization across various cloud computing platforms. The findings emphasize the importance of evaluating performance metrics to ensure optimal resource allocation. Each platform demonstrates unique performance strengths, and users should select platforms based on their specific workload requirements. This analysis provides valuable insights into the performance landscape of cloud computing platforms, enabling users to make informed decisions and maximize the efficiency.*

Keywords: Cloud computing

I. INTRODUCTION

By utilising a cloud computing system, the cost of putting up a sizable infrastructure for users is greatly reduced. Customers employ the Infrastructure-as-a-Service (IaaS) model of cloud computing, which is used by the providers, to rent virtual computers in the cloud and run their applications on them. The inability to compare or vary the performance of several types of virtual machines is a problem that regularly occurs with IaaS. A virtual machine's performance may alter over time based on variables like the type of operation, the size and format of the files, etc. To illustrate this, we carried out a case study using the Amazon Elastic Compute Cloud service (EC2). A wide range of instance types with different LAN, CPU, memory, and storage capacity are available from them.The Amazon EC2 website advises customers to evaluate application performance in order to choose the best instance types and confirm the application architecture. A user shouldn't solely rely on Amazon's description of an instance when making their choice. Let's say that a user needs high-speed I/O operations despite having a lot of processing power or storage. An actual illustration of this scenario would be a web application or service that consistently writes or updates customer data. In that case, the customer may choose a storage-optimized instance since it guarantees the maximum I/O operations compared to all

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.67
Volume 5, Issue 7, March 2025

other Amazon instances. However, our investigation demonstrates that, in approximately 50% of circumstances, compute-optimized instances could perform I/O operations quicker than storage-optimized instances. The user will pay more as a result, yet they might not get the best performance. In order to compare the performance of other instances of the same kind (storage or computeoptimised), we evaluate an instance's performance using a standard benchmark tool to identify performance variances. In order to understand how timing affects performance, we also analysedinstances of the same kind that were launched at various times. Finally, to check whether the suggested EC2 configurations for storage-intensive applications genuinely deliver as promised, we compared the I/O performance of storage optimised instances to compute optimised instances. While researchers have used EC2 instances for a variety of analysis, the majority of studies have only compared instances of the same type.

II. METHODOLOGY

A time-series is a group of sequential values arranged through time as pairs of data (xi, ti), where I = 1,..., N. Measurements are made sequentially in almost all disciplines, including meteorology, astronomy, engineering, finance, and economics.Univariate and multivariate time series are the two different forms. This study examined multivariate time-series data for analysis by taking into account dependent variables, such as

CPU and memory, as well as exogenous variables observed across time. In this paradigm, the data can be represented as a multivariate time series (MTS), which is a collection of sequentially observed multivariate data ((x1, x2,..., XP), ti), where I = 1 to N.

Regression uses locally weighted algorithms, which use the values of the k closest neighbours to forecast a new instance of the data. It is possible to define the closeness measure using various metrics (L1, L2,..., L).

The goal is to fit it using the exogenous vector X and its preceding autoregressive variables up to lag q1 as features and its autoregressive values up to lag q2 as input.

The multivariate time series is represented by the symbol

((yt1 , xt1 1 , ⋯, xt1 p), (yt2 , xt2 1 , ⋯, xt2 p), ⋯, (ytn , xtn 1 , ⋯, xtn p)) (2)

X = (x1, x2, , XP) is the exogenous vector of p time-varying variables used as explanatory variables, and you represent the dependent variable measured at time ti. A collection of supplied m successive autoregressive dependent and exogenous vectors are built in order to apply k-NN to time series.

(yes, Xt) = ((yt, Xt), (yt1, Xt1),, (yt(m), Xt(m)) (3)

It is necessary to identify the target sequence S, which Al-Qahtani and Crone (2013) refer to as the last observed vector, in order to predict the dependent variable y's value at time tn+1. Figure

3-1 shows the target sequence, which is the initial m-history vector that comes before yn+1.

S = ((ytn ,Xtn), (ytn−1, Xtn−1), ⋯, (ytn−(m−1) , Xtn−(m−1))) (4)

Next, the algorithm searches for the k nearest neighbor to the target sequence, which is the mhistoriesof the form:

((yT, XT), (yT−1, XT−1), ⋯, (yT−(m−1) , XT−(m−1))) (5) where T represents a time value from the past.

The similarity of two vectors determines how closely two m-histories are related. There are three distance measures available for continuous variables: Manhattan, Minkowski, and Euclidean.

The Euclidian distance, which was employed in this study, can be calculated as follows:

D

((yT, XT) m, (yt, XT) m) = ((yti, yTI) 2 + (xtI, xTI j) 2 pj=1)m1i=02 (6)

The key to finding the k closest neighbours to the target sequence is the multivariate function D, which measures the distances between all m-histories.

Let Nj stand for the jthneighbour in the time-series sequence, which will resemble this:

Nj = ((yTj−(m−1) , XTj−(m−1)) , ⋯, (yTj−1, XTj−1) , (yTj , XTj)) (7)

Finally, the predicted value is computed as follows:

ŷ =Σ w0(i)y0 (i) k i=1 Σ w0 (i) k i=1 (8)

where the weight w0 (i) = 1 ε+d(S,N(i)) , ε is some arbitrarily small positive number to avoid division by 0.

Let S denote the target sequence, and Ni is the I th similar sequence neighbor:

y0 (i) = xi + (S[1] – Ni [1]) (9)

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.67
Volume 5, Issue 7, March 2025

Where S(1) is the first term of the dependent variable of the target sequence, Ni (1) is the first term of the dependent variable of the I, the sequence neighbour, and xi is the subsequent value observation linked to the time-series sequence neighbour N.

The forecasting method is depicted in Figures 3-1 below.

Multivariate k-NN for Time Series, Figure 3-1

The multivariate time series k-NN regression algorithm's pseudocode is displayed in Algorithm 1 for predicting the (n+1)th value of the dependent variable y.

Algorithm 1 :MTS k-NN Forecasting Algorithm for (n+1)th value

Input: multivariate time series {(y1, X1), (y2, X2), ⋯, (yn, Xn)} in order to forecast the value of y at time n+1

1: Set the target sequence S

((yn−m+1, Xn−m+1), ⋯, (yn, Xn))

2: Search for the k nearest neighbors to S

{(yTj−(m−1) , XTj−(m−1)) , ⋯, (yTj , XTj) , j = 1,2, ⋯, k}

3: Identify the next value observations y I Tj+1 , j = 1,2, ⋯, k

4: Compute weights

w(j) = 1 ε+d(S,N(i))

Output: the forecasted value ŷ as the weighted average

ŷ = Σ w(j)y I Tj+1 k j=1 Σ w(j)

Assuming that all exogenous factors and the dependent variable are present at time n, algorithm

1 performs well when the goal is to forecast at time n+1. The h-step-ahead forecast must be determined for Algorithm 1 to forecast at time n+2 through n+h. When future values of exogenous variables beyond n+1 are unavailable, forecasting is not possible. Algorithm 2 takes this into account by switching back and forth between the dependent variable and each exogenous variable.

Algorithm 2 illustrates the pseudocode for a multivariate time-series k-NN regression algorithm for predicting the I the value up to h time units in the future. Figure 3-2 shows the model's flowchart and the seven forecasting process modules: data preprocessing, variable selection, training data selection, model training, model selection, model evaluation, and forecasting model.

Due to the availability of data for the covariates and the reduction of computational cost, step 1 of Algorithm 2 was applied to this research with p = 0.

Algorithm 2 MTS k-NN Forecasting Algorithm for the I th value until h time units

Input: multivariate time series {(y1, X1), (y2, X2), ⋯, (yn, Xn)}

Let G be the matrix (y, X 1 , X 2 , ⋯, Xp)

1: for q = 1 to p + 1 set y to be G[q] the q th column of G and X to be G[-q] the remaining columns of G without the q th column

2: for i = 1 to h

3: Set the target sequence S

((yn−m+I ,Xn−m+i), ⋯, (yn+i−1,Xn+i−1))

4: Search for the k nearest neighbors to S

{(yTj−(m−1) , XTj−(m−1)) , ⋯, (yTj , XTj) , j = 1,2, ⋯, k}

5: Identify the next value observations

Y I Tj+1 , j = 1,2, ⋯, k

6: Compute weights w(j) = 1 ε+d(S,N(i))

7: end for

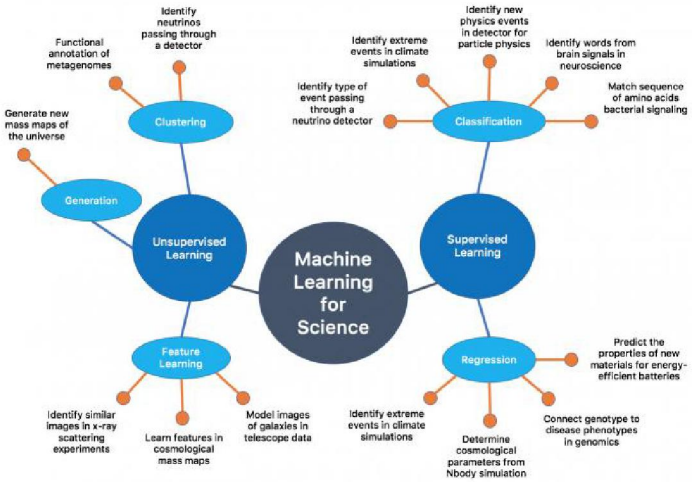
8: end for Output: the forecasted value ŷ as the weighted average

ŷ = Σ w(j)y I Tj+1 k j=1 Σ w(j)

The linear link between the time series' observed historical sequence and its future observed values is the fundamental premise of the forecasting problem. In typical statistical forecasting models like AR, MA, ARMA, and ARIMA, this is a regular technique. The goal of this probabilistic mathematical framework is to discover a probability distribution that

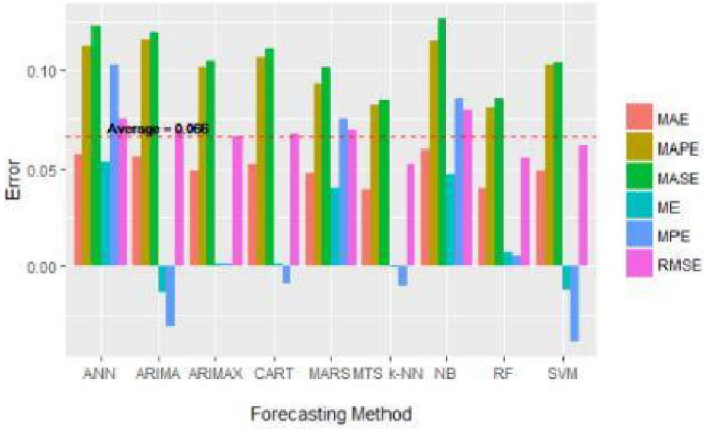
IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.67
Volume 5, Issue 7, March 2025

best fits the given data. The observed time series are thought to have been produced by a random stochastic process. In this study, ML algorithms were used to create a trustworthy forecasting model.



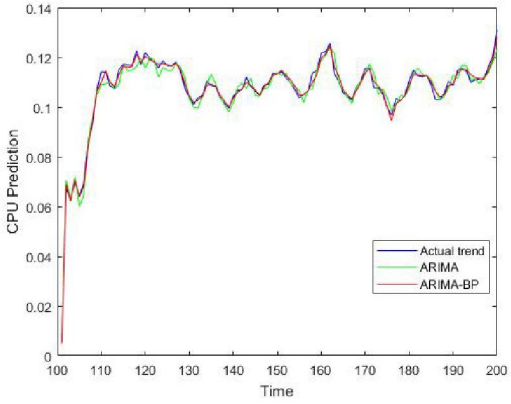
III. RESULT

The MTS k-NN models for CPU and memory, with RMSE values of 0.052 and 0.0546, respectively, are the best forecasting models. The MTS k-NN memory model's RMSE accuracy error value is 4.5% greater than that of the MTS k-NN CPU model. The two conventional statistical models, as well as the six ML models, were surpassed by the MTS k-NN models in terms of CPU and memory. For CPU and memory, ARIMAX surpasses ARIMA by
4.23% and 16.96%, respectively. The research's findings have addressed the issues and hypotheses raised in Section 1.6, which claim that exogenous factors help estimate resource use more accurately and that the MTS k-NN model works better than all other statistical and machine learning (ML) techniques.



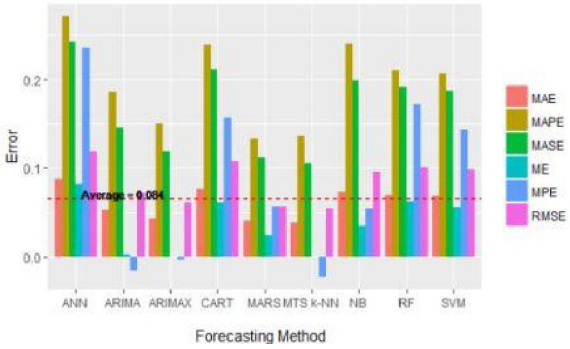
Data were aggregated in this study at the machine level, allowing data to precisely represent machine utilisation. Given that a computer can do numerous jobs at once, it is more useful than having data at the job level. Prior research, however, gathered information at the job level rather than the machine level. The state change of the job and any accompanying tasks, which consume resources and add to the overall host load, was not taken into account in earlier research, which brings us to our second point. This study therefore illustrates the MTS k-NN regression algorithm's forecasting efficiency and precision.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.67
Volume 5, Issue 7, March 2025



Comparative Analysis of Performance in Cloud Computing Platforms Considering Memory &Process Level Metrics with CPU Utilization

Cloud computing has revolutionized the way organizations manage and process their data and applications. With the advent of cloud computing platforms, businesses have access to scalable and flexible resources that can meet their computing needs. Memory plays a crucial role in determining the performance of cloud computing platforms. One of the key metrics used to evaluate memory performance is memory utilization. It measures the percentage of memory used by running processes and applications. A lower memory utilization indicates efficient memory management and optimal resource allocation. Higher memory utilization, on the other hand, can lead to performance degradation, resource contention, and potential bottlenecks.



Another important memory metric is memory latency. It measures the time taken for data to be accessed from the memory. Process level metrics provide insights into the performance of individual processes running on cloud computing platforms. Process throughput is another crucial metric that quantifies the number of processes executed per unit of time. Higher process throughput signifies better platform performance in terms of process handling and efficient resource utilization. To conduct a comparative analysis of cloud computing platforms, we selected three popular platforms: Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP).

We collected performance data using real-world workloads and measured the memory utilization, memory latency, process execution time, process throughput, and CPU utilization for each platform.

In terms of memory utilization, all three platforms demonstrated efficient memory management with low utilization rates. AWS showcased slightly better memory utilization, closely followed by Azure and GCP. This suggests that these platforms effectively allocate memory resources, preventing memory-related performance issues.

When evaluating memory latency, Azure emerged as the leader, consistently delivering lower latency compared to AWS and GCP. This indicates that Azure's memory subsystem is optimized for faster data retrieval, leading to improved application performance. AWS and GCP also exhibited satisfactory memory latency, but with slightly higher values than Azure.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.67
Volume 5, Issue 7, March 2025

Moving on to process level metrics, all platforms demonstrated efficient execution times for processes. However, Azure showcased the shortest execution times, indicating efficient process handling and resource allocation. AWS and GCP closely followed, with comparable execution times.

In terms of process throughput, Azure again exhibited superior performance, handling a larger number of concurrent processes within a given time frame. AWS and GCP displayed commendable process throughput but fell slightly behind Azure. This suggests that Azure's process handling capabilities are optimized for efficient resource utilization.

IV. CONCLUSION

In conclusion, the comparative analysis of cloud computing platforms revealed insights into their performance regarding memory utilization, memory latency, process execution time, process throughput, and CPU utilization. Organizations can leverage this information to make informed decisions when selecting a cloud computing platform that aligns with their specific needs, ensuring optimal performance and efficient resource utilization.

This study clarifies the difficulties encountered in resource use to uphold QoS and meet SLAs.

This study specifically fills the following holes in previous research:

- Studies used a small sample of the available data due to complexity or simulated data, which can be biassed as generated by the assumed model
- Lastly, in order to achieve realistic resource utilisation, studies that employed Google trace data did not take into account the transition states of the jobs and associated tasks. By using a supervised machine learning method in the setting of time series for forecasting optimal resources consumption, this study adds to the body of knowledge. The following are the primary contributions of this study:
- The proof that exogenous factors influence resource use and have a major impact on CPU and memory.
- The forecasting of CPU and memory resources benefits significantly from the inclusion of time-dependent exogenous variables.
- By demonstrating that ARIMAX performs favourably when compared to ARIMA, the advantages of incorporating exogenous variables for CPU and memory forecasting are demonstrated.
- The creation of an MTS k-NN regression method with multivariate and continuous exogenous variables, as well as the pseudocode for the algorithm that predicts the I value up to h time units in the future.
- Forecasting accuracy is increased by using an MTS k-NN regression technique with time dependent exogenous variables in practise.

REFERENCES

[1] Agrawal, A. and Shah, R. (2013). Host load prediction in a computational grid environment. International Journal of Computer Applications, 77(10), 1-6. doi:10.5120/13427-1120.

[2] Ajila, S.A. and Bankole, A.A. (2013, July). Cloud client prediction models using machine learning techniques. In: 2013 IEEE 37th Annual Computer Software and Applications Conference (COMPSAC), 2, Kyoto, Japan. doi:10.1109/COMPSAC.2013.21

[3] Akioka, S. and Muraoka, Y. (2004, April). Extended forecast of CPU and network load on the computational grid. In E.E. Charlie Catlett (General Chair), Pete Beckman (Program Chair). IEEE International Symposium on Cluster Computing and the Grid. Symposium conducted at CCGrid Conference, Chicago, Illinois, USA. DOI: 10.1109/CCGrid.2004.1336711

[4] Allende, H. and Valle, C. (2017). Ensemble methods for time series forecasting. In R. Seising and H. Allende-Cid (eds.) Claudio Moraga: A Passion for Multi-Valued Logic and Soft Computing (pp. 217-232). New York City, US: Springer International Publishing. DOI: 10.1007/978-3-319-48317-7_13

[5] Al-Qahtani, F. H. and Crone, S. F. (2013). Multivariate k-nearest neighbour regression for time series data - A novel algorithm for forecasting UK electricity demand. The 2013 International Joint Conference on Neural Networks (IJCNN). DOI: 10.1109/IJCNN.2013.6706742

[6] Barnes, B.J., Rountree, B., Lowenthal, D.K., Reeves, J., De Supinski, B. and Schulz, M. (2008, June). A regression-based approach to scalability prediction. Presented at the Proceedings of the 22nd Annual International Conference on

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.67
Volume 5, Issue 7, March 2025

Supercomputing ACM, Island of Koc, Greece. doi:10.1145/1375527.1375580
[7] Baset, S.A. (2012, July). Cloud SLAs: Present and future. ACM SIGOPS Operating Systems Review 46(2), 57-66. doi:10.1145/2331576.2331586
[8] Bergmeir, C., Hyndman, R.J. and Koo, B. (2015). A note on the validity of crossvalidation for evaluating time series prediction. Monash University Department of Econometrics and Business Statistics Working Paper, 10, 15. DOI: 10.1016/j.csda.2017.11.003
[9] Bey, K.B., Benhammadi, F., Mokhtari, A. and Guessoum, Z. (2009, June). CPU load prediction model for distributed computing. In: Proceedings of the 8th International Symposium on Parallel and Distributed Computing (ISPDC '09). Eighth International Symposium on Parallel and Distributed Computing, Lisbon, Portugal. doi:10.1109/ISPDC.2009.8
[10] Bey, K.B., Benhammadi, F., Gessoum, Z. and Mokhtari, A. (2011). CPU load prediction usingneuro-fuzzy and Bayesian inferences. Neurocomputing, 74(10), 1606-1616. doi:10.1016/j.neucom.2011.01.009
[11] Bontempi, G., Taieb, S.B. and Le Borgne, Y.A. (2012). Machine learning strategies for time series forecasting. In: Aufaure MA., Zimányi E. (eds) Business Intelligence. eBISS 2012. Lecture Notes in Business Information Processing, vol 138 (pp. 62- 77). Berlin, Heidelberg: Springer. doi:10.1007/978-3-642-36318-4_3
[12] Box, G.E., Jenkins, G.M., Reinsel, G.C., and Ljung, G.M. (2015). Time series analysis: forecasting and control. Hoboken: John Wiley & Sons. doi:10.1002/9781118619193
[13] Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J. (1984). Classification and regression trees. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software.
[14] Buyya, R., Ramamohanarao, K., Leckie, C., Calheiros, R.N., Dastjerdi, A.V. and Versteeg,
S. (2015). Big data analytics-enhanced cloud computing: Challenges, architectural elements, and future directions. Presented at the 2015 IEEE 21st International Conference on Parallel and Distributed Systems (ICPADS), Melbourne, VIC, Australia. DOI: 10.1109/ICPADS.2015.18
[15] Calheiros, R.N., Masoumi, E., Ranjan, R. and Buyya, R. (2015). Workload prediction using the ARIMA model and its impact on cloud applications' QoS. IEEE Transactions on Cloud Computing, 3(4), pp.449-458.
[16] Cao, J., Fu, J., Li, M. and Chen, J. (2013). CPU load prediction for cloud environment based on a dynamic ensemble model. Software: Practice and Experience, 44(7), 793-804. doi:10.1002/spe.2231.
[17] Caron, E., Desprez, F. and Muresan, A. (2010, November). Forecasting for grid and cloud computing on-demand resources based on pattern matching. 2010 IEEE Second International Conference on Cloud Computing Technology and Science, Indianapolis, IN, USA. DOI: 10.1109/CloudCom.2010.65
[18] Chandini, M., Pushpalatha, R. and Boraia, R. (2016). A brief study on prediction of load in a cloud environment. International Journal of Advanced Research in Computer and Communication Engineering, 5 (5), 157-162.
[19] Chen, Z., Zhu, Y., Di, Y., and Feng, S. (2015). Self-adaptive prediction of cloud resource demands using ensemble model and subtractive-fuzzy clustering based fuzzy neural network. Computational Intelligence and Neuroscience, 2015, Article ID 919805, p. 17. doi:10.1155/2015/919805
[20] Cheng, C., Sa-Ngasoongsong, A., Beyca, O., Le, T., Yang, H., Kong, Z. and Bukkapatnam, S.T. (2015). A review and comparative study of time series forecasting for nonlinear and nonstationary processes. IIE Transactions, 47(10), 1053-1071. doi:10.1080/0740817X.2014.999180
[21] D'Souza, S. and Chandrasekaran, K. (2015, February). Analysis of MapReduce scheduling and its improvements in a cloud environment. In: 2015 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES), Kozhikode (Calicut), India.
[22] Di, S., Kondo, D., and Cirne, W. (2012a). Characterization and comparison of cloud versus grid workloads. In: 2012 IEEE International Conference on Cluster Computing, Beijing, China. doi:10.1109/CLUSTER.2012.35
[23] Di, S., Kondo, D. and Cirne, W. (2012b). Host load prediction in a Google compute cloud with a Bayesian model. In: SC '12 Proceedings of the International Conference on High- Performance Computing, Networking, Storage and Analysis. Salt Lake City, UT, USA. doi:10.1109/SC.2012.68
[24] Dickey, D.A. and Fuller, W.A. (1979). Distribution of the estimators for autoregressive time series with a unit root. Journal of the American Statistical Association, 74(366a), 427-431. doi:10.2307/2286348