Resource Utilization Prediction in Cloud Computing using Hybrid Model

Anupama K C[1], Shivakumar B R[2], Nagaraja R[3]
Department of Information Science and Engineering
Bangalore Institute of Technology, Bangalore, Karnataka, India

***Abstract*—In cloud environment, maximum utilization of resource is possible with good resource management strategies. Workload prediction plays a vital role in estimating the actual resource required for successful execution of an application on cloud. Most of the existing works concentrated on predicting workloads which either showed clear seasonality/trend or for irregular workload patterns. This paper presents a new perspective in forecasting both seasonal and non-seasonal workloads. To accomplish this, a hybrid prediction model which is a combination of statistical and machine learning technique is proposed. Suppose the seasonality exists in the workload pattern, Seasonal Auto Regressive Integrated Moving Average (SARIMA) model is applied for prediction. For non-seasonal workloads Long Short-Term Memory networks (LSTM) or AutoRegressive Integrated Moving Average (ARIMA) model is used based on the results of normality test. This paper presents a prediction model which forecasts the actual resource required for diverse time intervals of daily, hourly and minutes utilization. The experimental results confirm that accuracy of the prediction of LSTM model outperformed ARIMA for irregular workload patterns. The SARIMA model accurately forecasts the resource usage for forthcoming days. This work actually helps the cloud service provider (CSP) to analyze the workload and predict accordingly to avoid over or under provisioning of the cloud resources.**

Keywords*—Workload prediction; SARIMA; LSTM; ARIMA; cloud service provider

I. Introduction

The Cloud computing is a utility computing model which is convenient to access the pool of computing resources such as physical machines, servers, applications, computing, storage, networks and various other services. The cloud computing model provides majorly three services such as software, platform and infrastructure as a service based on pay per usage model [1]. The elasticity feature of cloud computing enables the users to dynamically change the resource request periodically based on the demand. Due to fluctuating demands, the cloud manager must be able to leverage resources by provisioning and de-provisioning the resources to meet the current request. Insufficient provisioning of the resources causes Service Level Agreements (SLA) violation, poor Quality of Service (QoS), performance degradation which in turn causes customer dissatisfaction. On the contrary overprovisioning leads to wastage of resources which increases the cost and energy. For the seamless working of the system, judicious study of the dynamic and accurate resource provisioning is essential.

Workload prediction is one of the most critical and important aspect in managing cloud infrastructure in a flawless way. Every application requires resources to complete its execution and these resources come in the virtual form. The Cloud Data Centers (CDC) comprises of various resources like CPU, Memory, bandwidth, software, etc. which are allocated to the users on demand to complete their task execution. As per the previous works, it is well noted that resources provisioned to execute an application is always greater than actual resources required to complete it [2]. The reason for over provisioning of resources is to avoid SLA violations and to achieve QoS satisfaction. In most of the cases, the resources are being wasted in the process of allocation.

Accurate prediction can be used to decide the appropriate amount of resource to fulfill the demands. There is a need to employ a reliable and precise prediction model to achieve accurate estimation of the future workload. Usually in CDC, user's task arrives in an irregular pattern with heterogeneous resource requirement. This situation poses a major challenge to predict the precise workload [3]. Researchers have designed various workload prediction models and resource usage forecasting models, primarily concentrated on predicting CPU and memory utilization [4-7]. Various research works have used only statistical methods to predict workload and they are unable to predict accurate results for large and heterogeneous data [8]. Several research works have been carried out to address prediction of high dimensional and greatly varying cloud workloads using machine learning models. It is observed that, they were able to achieve promising prediction results. However, the statistical models are able to proactively predict temporal workloads in a controlled mode. Therefore, it is understood that combining both statistical and machine learning techniques when applied on heterogeneous data would result in better prediction accuracy [9]. Nevertheless, moderately a smaller number of research works has been carried out in the area of resource prediction at task level [10]. By predicting the resource utilization at task level aids in characterization of tasks, majorly impacts the process of task allocation, VM creation and capacity planning [11].

The main objective of this work is to accurately predict the CPU and memory utilization for different time intervals benefiting the cloud management for proper utilization of the available resources. The proposed Hybrid prediction model uses both statistical and machine learning approaches to achieve better quality prediction results with accuracy. This paper proposes a workload prediction model which is aimed to

predict the actual resource consumption of Central Processing Unit (CPU) and memory against provisioned resources. The pre-processed historical data is used to train the proposed prediction model. The predicted results are further used by the task classifiers to classify the tasks according to the resource utilization types which in-turn aids in resource management. Throughout this paper, prediction and forecasting has been used interchangeably. Remainder of the paper is structured as: Section 2 presents the overview of the existing works related to prediction using machine learning, statistical and hybrid methods in terms of resource utilization and accuracy. The Section 3 broadly explains proposed architecture and working principles of the prediction model. Section 4 presents the analysis of the workload trace, experimental setup and the obtained results. Finally, Section 5 concludes the work and mentions the future scope of the work.

II. Related Works

This section summarizes different prediction methods based on machine learning, statistical and hybrid approaches.

A. Machine Learning Methods

Machine learning and deep learning methods works on large and multivariate time series forecasting problems. Learning models outperforms when they are applied to complex and highly nonlinear data. They also make the most accurate long-term predictions. Chen et al. [1] applied L-PAW (deep Learning based Prediction Algorithm for cloud Workloads) utilized top-sparse encoder and gated recurrent unit block into Recurrent Neural Network (RNN) to achieve accurate prediction for high-dimensional workloads. Applying various evaluation metrics enhances in understanding the quality of the model. Yu et al. [6] developed a learning approach based on clustering to predict long-term workloads. In their work, K-medoid algorithm was used for clustering and multi-layer perceptron neural network for learning the patterns of workload and compared with non-clustering-based learning approach. Combining different prediction approaches can further improve accuracy of the results obtained. Furthermore, Qiu et al. [12] designed a deep learning prediction model for VM workload prediction using Deep Belief Network (DBN) with multiple-layered Restricted Boltzmann Machines (RBMs) and a regression layer. The authors have monitored only CPU usage and also the performance of the model is appreciable. Many more works have been proposed based on recurrent neural networks of deep learning. Wang et al. [13] and Guo et al. [14] proposed a prediction method using LSTM. The former proposed a model called LSTMtsw to predict the future resource request trend of users and forecast CPU and memory resources and results proved that LSTM outperforms BPNN model. Whereas, the later worked on VM workload prediction using N-LSTM or the novel LSTM and compared the results with other LSTM variants in terms of prediction accuracy but the training and testing time required was high in their proposed model. Nguyen et al. [15] designed and implemented a new approach to predict workload by stacking Recurrent Neural Networks and Autoencoder on different datasets to compare prediction accuracy. Better prediction accuracy results may be possible if LSTM and autoencoder combination was used. It is understood that, machine learning methods outperforms statistical method in terms of forecasting horizons and accuracy.

B. Statistical Methods

Statistical forecasting uses historical data to predict the future demands and these methods have been successfully used for short-term predictions. Calheiros et al. [8] presented cloud workload prediction for SaaS providers which was based on the ARIMA model to achieve the accuracy in resource utilization. Even though results showed an accuracy around 91%, there is a possibility to work on achieving better quality of service. Shyam et al. [16] proposed Bayesian model for accurate prediction of long-term and short-term resource requirements mainly considering CPU and memory intensive applications. Appreciable work has been done on predicting the resource utility. The understanding of the nature of workload can be further enriched if high-level metrics are used in prediction. Nashold et al. [17] forecasted CPU utilization in clusters using SARIMA and LSTM for both long term and short term tasks. The study concentrated only on predicting highest CPU utilization for the upcoming intervals of time but handling of memory utilization, which is as important as CPU is not considered in the work. Adhikari et al. [18] have proposed their work on time series models, salient features, related issues and importance of forecasting in various practical domain. Finally, it is observed that combining different prediction approaches can improve the forecasting accuracy. Parmezan et al. [19] has done an extensive study on evaluating statistical and machine learning model for time series prediction that deal with univariate data. They have experimented and compared the results using 55 real and 40 synthetic time series datasets. The work narrow downs the prediction by dealing with only univariate data, multivariate data are not considered in their work. From the overall study, it is found that statistical methods limits the understanding of forecasting horizons and accuracy.

C. Hybrid Prediction

The following researchers have attained better workload prediction accuracy by combining statistical, machine learning and mathematical models. Bi Jing et al. [3] proposed an approach that combines ARIMA and Haar wavelet to predict the number of tasks arriving at the successive time interval in the data center and results proved that hybrid approaches results in better prediction accuracy. Computing resource like CPU, memory etc., which plays a vital role in resource allocation are not addressed in the work. Janardhanan et al., [20] used LSTM and ARIMA models for forecasting CPU workloads in Google's cluster data. From the results it is found that LSTM has 20% less forecasting error when compared to ARIMA model and performed with better consistency. Experiment is implemented on a single machine and forecasted CPU utilization for that machine. Furthermore, Cetinski et al., [5] proposed an Advanced Model for Efficient Workload Prediction in the Cloud (AME-WPC) combining statistical and the machine-learning techniques to improve the accuracy of the workload prediction over time. The proposed approach proved to be efficient in terms of managing resource and minimizing the operational costs. But forecasting of specific cloud resource would aid in scaling of resource automatically and in the process of scheduling.

Islam et al., [9] presented an Error Correction Neural Network (ECNN) and Linear Regression learning algorithms for adaptive resource provisioning in cloud to forecast future resource demands especially for e-commerce applications. Predicting resource usage for different time intervals may be hourly, daily and monthly will provide a good insight for dynamically scaling the resource in cloud environment. Furthermore, Ullah et al. [21] designed a prediction model which takes real-time resource utilization and feeds these values to ARIMA and Autoregressive Neural Networks (AR-NN). The model is used to predict CPU utilization for the forthcoming four hundred minutes and it is observed that other physical resources are excluded in the study which are essential for predicting resource usage in IaaS cloud. From all these works, it is clear that combining both statistical and machine learning techniques when applied on heterogeneous data would result in better prediction accuracy.

III. System Model

The proposed work forecasts the resource required for incoming tasks and predicting resource for the next given time period. This work considers CPU and Memory utilization, as they play an important role in minimizing cost and energy in CDC. This section presents an overview of the proposed prediction model as shown in the Fig. 1. Initially, the proposed models are trained and tested with the historical workload traces. The historical workloads trace comprises of the properties of executed tasks from the CDC. The historical data is fed to the pre-processing module for training and testing. The pre-processed module cleans the trace data and test the normality of the datasets. Afterwards, depending on the test results, the hybrid prediction module chooses the appropriate models for forecasting the CPU and memory usage. Secondly, depending on the test results, the hybrid prediction module chooses the appropriate models for forecasting the CPU and memory usage. Finally, the forecasted results are fed to Evaluation model, where the accuracy of the prediction is measured using statistical metrics.

Once the model is trained and tested with better accuracy, then the model is ready for the prediction of resources for the new incoming applications. Every user's applications are executed in the cloud platform using internet. Each application is divided as tasks for example, T_1, T_2, T_3…T_n, etc. as shown in Fig. 1. The incoming tasks are fed to pre-processing module. The incoming job or task attribute values will undergo normality and seasonality test to choose the appropriate prediction model. The hybrid prediction model produces CPU and memory requirements to execute given task or application without the violation of SLA.

A. Pre-processing

The main goal of this process is to transform the obtained historical workload traces into a proper format as required by the proposed model. Since the available dataset/traces may contain noisy data, it is necessary to eliminate such before applying to any models. The cleaned data undergoes seasonality and normality test in the preprocessing module which checks normality for correlated data.

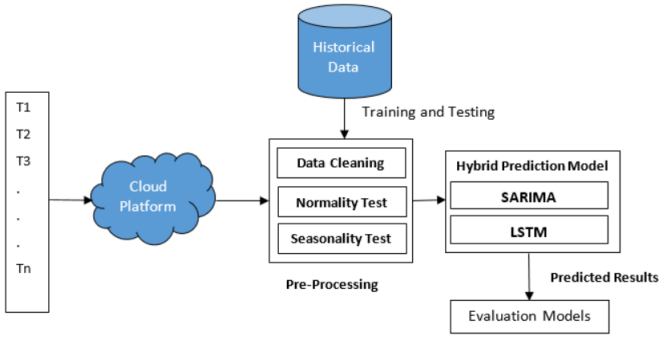

Fig. 1. Hybrid Resource Utilization Prediction Model.

Data Cleaning: The missing values, outlier values which are present in the trace data are cleaned by dropping such tuples from the dataset. The valuable information is retained and organized for further process. All the attributes do not contribute to the prediction process. Hence only those attributes which are very essential are extracted and stored separately. For example, in this experiment the attributes such as provisioned CPU, CPU used, provisioned memory, the actual memory used etc., are considered for CPU and memory utilization prediction. Thus, data cleaning process enhances the prediction accuracy.

Normality Test: The cleaned data has to undergo the normality test to determine, how likely the underlying dataset is normally distributed. Jarque-Bera test is applied to calculate kurtosis and skewness of the dataset [21]. If the test results show normal distribution then, statistical based prediction model is applicable else machine learning model is applied.

Seasonality Test: The simplest way to test and analyse for seasonality component such as daily, weekly or monthly in time series data by plotting the dataset. If the graph shows repeating spikes, that indicates there exists seasonality in the data. Hence for the data that exhibits seasonal patterns, SARIMA model is used for resource forecasting. On the other hand, non-seasonal time series data, ARIMA model is applied.

B. Hybrid Prediction Model

After exploring many works on prediction methods, it is understood that combining machine learning and statistical methods have proved to predict with better accuracy. Therefore, following two methods which are popularly known for time series prediction are used, namely:

- Seasonal ARIMA, which is an extension of ARIMA used in analyzing time series with seasonality.
- LSTM - a variant of RNN, which is capable of solving the problem of long-term dependency.

Since the work focuses on data containing trends and seasonality, SARIMA model was chosen which supports in analyzing the seasonal characteristics of the time series data. When it comes to forecasting data with complexity and non-linearity, LSTM method is applied, which is strong enough in identifying the complex pattern and structure in the given data. The following section discusses each of the method in detail.

1) SARIMA Model: Experiment uses SARIMA model which is similar to ARIMA with the exception that it takes seasonality into account. Seasonality is a regular pattern that appears in time series data, where changes are repeated over S time periods. Here S indicates the number of time periods till the pattern repeats again. For example, consider any monthly data appearing with high seasonality in a particular month and low seasonality in other months of the year. In such case, S=12 (months per year) and S=4 (for quarterly) is the span of the periodic seasonal behavior. Here, the time period of the day and month is considered.

In SARIMA both seasonal and non-seasonal factors are incorporated and denoted as SARIMA (p,d,q) × (P,D,Q)s. The lowercase notations denotes the non-seasonal component of the model, where p=non-seasonal AR (Auto Regression) parameter, d=non-seasonal differencing parameter, q= non-seasonal MA (Moving Average) parameter. The uppercase notations denotes the seasonal component of the model, where P=seasonal AR (Auto Regression) parameter, D=seasonal differencing states how many differencing orders to apply to make the time series stationary, Q= seasonal MA parameter and s= time span of repeating pattern [20]. The general form of the SARIMA model is given by (1):

$$\Phi_P(B^s)\, \phi(B)\, \nabla_s^D \nabla^d x_t = \Theta_Q(B^s)\, \theta(B) w_t \quad (1)$$

Where, $\{w_t\}$ is the Gaussian white noise process. s is the period of the time series. The AR and MA components are represented by polynomials $\phi(B)$ and $\theta(B)$ of orders p and q. The seasonal AR and MA components are represented by $\Theta_P(B_S)$ and $\Theta_Q(B_S)$, and their orders are P and Q. Ordinary and seasonal difference components are indicated as ∇^d and ∇_s^D. (B) is the backshift operator and the expressions are presented from (2) - (7):

$$\text{AR: } \phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \ldots\ldots - \phi_p B^p \quad (2)$$

$$\text{Seasonal AR: } \Phi_P(B^s) = 1 - \Phi_1 B^s - \Phi_2 B^{2s} - \ldots\ldots - \Phi_p B^{Ps} \quad (3)$$

$$\text{MA: } \theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \ldots\ldots + \theta_q B^q \quad (4)$$

$$\text{Seasonal MA: } \Theta_Q(B^s) = 1 + \Theta_1 B^s + \Theta_2 B^{2s} + \ldots.. + \Theta_Q B^{Qs} \quad (5)$$

$$\nabla^d = (1-B)^d \quad (6)$$

$$\nabla_s^D = (1-B^s)^D \quad (7)$$

The entire approach of SARIMA model is summarized as follows:

- Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) are used to identify the seasonal and non-seasonal orders of SARIMA model.
- Coefficients of the model are estimated.
- Verify for the tolerance and predicting the future workload based on historical data.

2) LSTM Model: Recurrent neural networks are more popular and effective methods compared to traditional methods in machine learning. Along with stationary input and output patterns RNN also deals with sequences of arbitrary length. Since RNN suffer from short-term memory, LSTM was generated as the solution to vanishing gradient & gradient explosion problems. This paper focuses on predicting the CPU and memory utilization requested by the users and experimented using LSTM [9], [20],[22]. In most of the recent works, LSTM has proved with better prediction accuracy when compared to other machine learning techniques especially for time series data.

LSTM operation is based on the mechanism of RNN. LSTM model is capable of capturing important features and remembering those information for a long interval of time. The Memory cell is a special feature of LSTM model which is an intermediate type of storage as in the Fig. 2. Memory cell is also called as gated cell as it is the one which decides about ignoring or preserving the memory information. Gated cells consist of sigmoid layer which outputs the numbers between zero and one. Value zero indicates to preserve information and one indicates to remove the information. The decisions are based on the weight values assigned during the training process. Hence the model learns about preserving what it needs and deleting the irrelevant information. Overall the LSTM model has three layers or gates: the forget gate, input gate and finally the output gate. The weights and the biases for the model are represented as : (W_f, W_i, W_g, W_o) and (b_f, b_i, b_g, b_o).

Forget gate**:** Initial step in LSTM model is to decide which information needs to be removed from the memory. The forget gate or the sigmoid function looks at the values h_t-1 and x_t to make this decision. The output f_t is a number between 0 and 1 indicating removing or preserving the information respectively. The output of this gate is calculated as in (8):

$$f_t = \sigma (W_f * [h_{t-1}, x_t] + b_f) \quad (8)$$

Where, b_f is a constant and it is called the bias value.

Input gate: This gate controls the flow of information by adding or deleting new information into the LSTM memory. The input gate has two parts: tanh and sigmoid layer respectively. The tanh layer creates g_t, a vector of new values that could be added to LSTM memory. The sigmoid layer i_t decides which values to be updated. Outputs of these two layers are computed as (9) and (10):

$$i_t = \sigma (W_i * [h_{t-1}, x_t] + b_i) \quad (9)$$

$$gt = \tanh(Wg * [ht-1, xt] + bg) \quad (10)$$

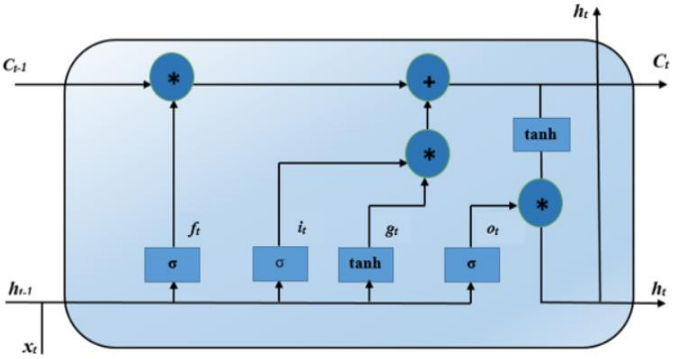

Fig. 2. Internal Structure of LSTM Block.

Combining these two layers provides an update to the LSTM memory. Updating the old value C_{t-1} into the new or current value C_t is computed by multiplying the old value by f_t the forget layer, and adding $i_t * g_t$. The mathematical equation is represented as (11):

$$C_t = f_t * C_{t-1} + i_t * g_t \quad (11)$$

Output gate: This primarily uses sigmoid layer to decide which part of LSTM memory will contribute to the final output. Later, a non-linear tanh function is performed to map values between -1 and 1. Lastly, the result of tanh is multiplied by the output of sigmoid layer. The output is calculated using the following (12) and (13):

$$o_t = \sigma (W_o * [h_{t-1}, x_t] + b_o) \quad (12)$$

$$h_t = o_t * \tanh(C_t) \quad (13)$$

Where o_t is the output value and h_t is the representation of the output and as value between -1 and 1.

C. Evaluation Criteria

To measure the accuracy of the work, two metrics such as Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) are used to evaluate the errors of forecasting. MAE is a measurement metric where the absolute error is the absolute value of the difference between the forecasted value and the actual value and is calculated using (14).

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left(y_i^p - y_i^a\right) \quad (14)$$

MAPE is used in forecasting accuracy of a prediction model. Lesser MAPE value indicates better prediction accuracy in terms of percentage. It is defined as in (15).

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{y_i^p - y_i^a}{y_i^a}\right) * 100\% \quad (15)$$

In the above formula, predicted value is y_i^p, the actual value is y_i^a and N is the number of the predicted values in the dataset.

IV. Experimental Evaluation

This section explains the dataset and experimental results of the prediction model. The experiment is performed significantly on business-critical workload traces collected from Bitbrains cloud [23]. Bitbrains cloud is a mid-size datacenter majorly hosting business-critical workloads and mainly specialized in business computation and managed hosting for enterprises. The Bitbrains faststorage trace contains information of 1250 VMs connected to fast Storage Area Network devices. The dataset is available for 30 days duration with the sample rate of 5 minutes and organized as one file per VM. Each file contains seven performance metrics: the provisioned CPU capacity, the CPU utilization, the provisioned memory capacity, the actual memory utilized, the network I/O throughput, disk I/O throughput and the number of cores provisioned.

SARIMA: Bitbrains cloud data was used to check whether the dataset has seasonality component in it. Experiment was conducted by plotting 30 days data. The description about the dataset is provided in the workload trace analysis part. After plotting the time series data, it is identified that there exists strong seasonality component in the dataset as in Fig. 3. It is observed that each data point looks similar to the data points of every other day. This interpretation leads to conclude that there is regularity in the patterns with respect to CPU and memory usage.

It is well known fact that ARIMA does not support seasonal component or it can be modelled for the non-seasonal time series data. ARIMA expects to remove (or reduce) seasonality by computing differences and the method is called differencing. Since there exists seasonality in the dataset SARIMA model is applied to predict the CPU and memory utilization for the upcoming days.

The first step is to identify the order of ARIMA model: AR (p), MA(q) and I(d) and it is done by plotting ACF and PACF at different lag lengths. The results of ACF and PACF functions clearly indicates the seasonal MA component and also to identify the maximum order of AR parameter estimation is done using Akaike's Information Criterion (AIC) and Bayesian Information Criterion (BIC) to select the best SARIMA model from other alternatives. Different parameter combinations was tried and ultimately choose the best model with lower AIC and BIC score. After estimating the parameters, the model was validated by testing against the actual Bitbrains dataset.

The forecasted CPU and memory values are tend to be close to the actual points. Fig. 4 and Fig. 5 shows actual and predicted graph of CPU and memory utilization, where black line indicates actual and red line indicates the predicted results.

The results are compared with the dataset of previous 29 days data. After selecting the best model, the forecasting of CPU and memory usage for the next three days are predicted. Various accuracy metrics like MAE and MAPE are used to test the predicted results.

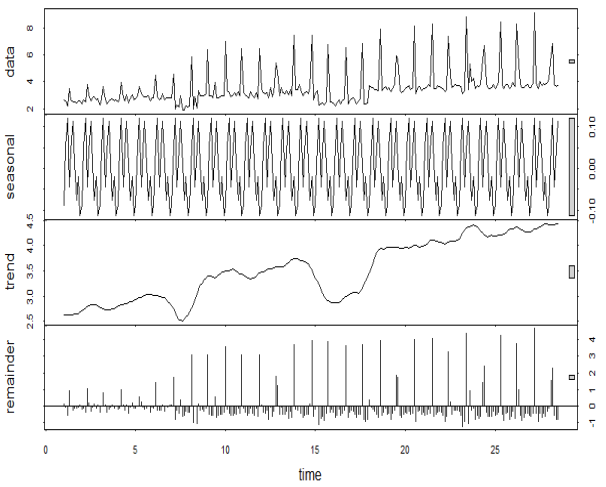

Fig. 3. Time Series Plot Representing Seasonal and Trend Components.

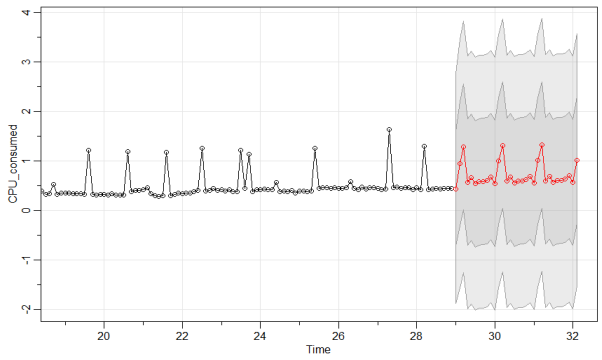

Fig. 4. CPU Utilization Prediction using SARIMA.

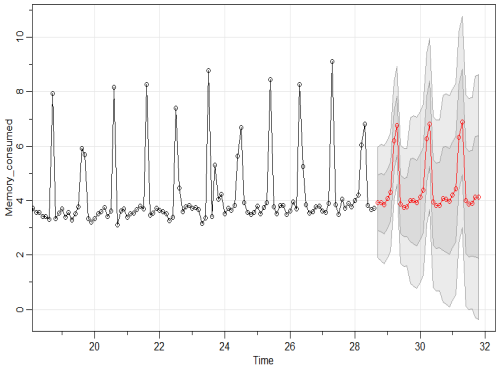

Fig. 5. Memory Utilization Prediction using SARIMA.

When experiment was compared with the predicted values of memory, it is observed that more than 10% of better prediction accuracy was found in case of CPU prediction. Since there is more fluctuation in the consumption of memory throughout the dataset, the values obtained after calculating MAE and MAPE are 6.08 and 1.52 respectively, whereas, in case of CPU usage prediction, the values of MAE were 5.83 and 0.49 for MAPE. Finally, it is found that the accuracy can be improved with the parameter estimation for the model. SARIMA model predicts the seasonality that occur over a period of time. Thus, SARIMA model is used to forecast the future resource consumption for the upcoming days and for months. This apparently helps the cloud service provider to analyze the trend and can avoid overprovisioning or under-provisioning of the resources.

LSTM: The implementation of LSTM model to evaluate the prediction accuracy of CPU and memory usage was efficiently tested on the Bitbrains dataset. The 30 samples of 5 minutes are average filtered to obtain a sample of 150 minutes. The CPU and memory forecasting are computed for both hourly and for minutes usage. Firstly, the experiment was conducted for predicting the CPU and memory consumption per hour interval. The .csv file consisting of CPU utilized and the actual memory utilized with respect to time was loaded into the working environment. Experiment was performed by varying number of training epochs and batch size. Epochs is the number of times the data is fed into the network and batch size allows to segment the data so that the network can process as small parts. It is found after the experiment that more the number of hidden layer size of LSTM, lesser was the accuracy. Thus, by modifying the weights and layers in the LSTM model, CPU and memory utilization prediction was accomplished. Fig. 6 represents the graph showing the actual and predicted values of CPU utilized in percentage vs Time in hours.

Similarly, the actual and predicted memory utilization in the units of 100MB with respect to time in hours is presented in Fig. 7. To illustrate the prediction accuracy, the LSTM model was compared with ARIMA model. LSTM model performed with better in terms of accuracy compared to ARIMA model. The predicted values of MAE and MAPE are shown in Table I.

Primarily the experiment focused on predicting CPU and memory considering per hour interval. Next focus was on predicting the same for every five minutes interval. The forecasted CPU and memory usage for every five minutes interval and for total of 24 hours is as shown in Fig. 8 and Fig. 9.

As observed in the Fig. 8 and 9, there is a high spike during 24th hour of the day. This infers that there is highest amount of CPU and memory is consumed at that particular time interval. Even then accuracy of the LSTM model performed extremely well when compared with ARIMA model. The calculated values of MAE and MAPE are shown in the Table I. MAE-Hr, MAPE-Hr and MAE-Min, MAPE-Min are the scores obtained for hourly and minutes CPU and memory usage prediction.

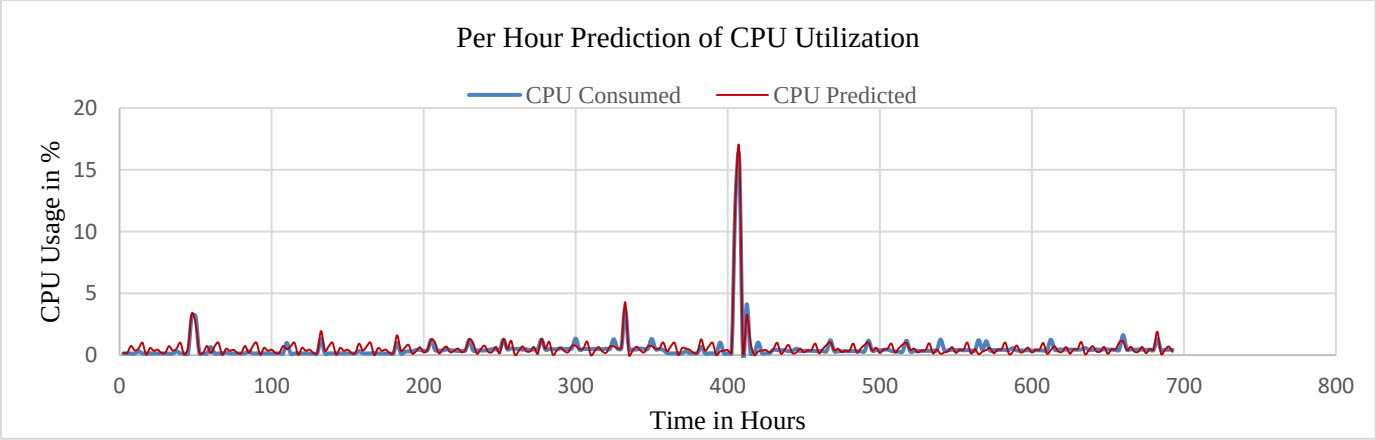

Fig. 6. Prediction of CPU utilization with LSTM Model - Time in Hours.

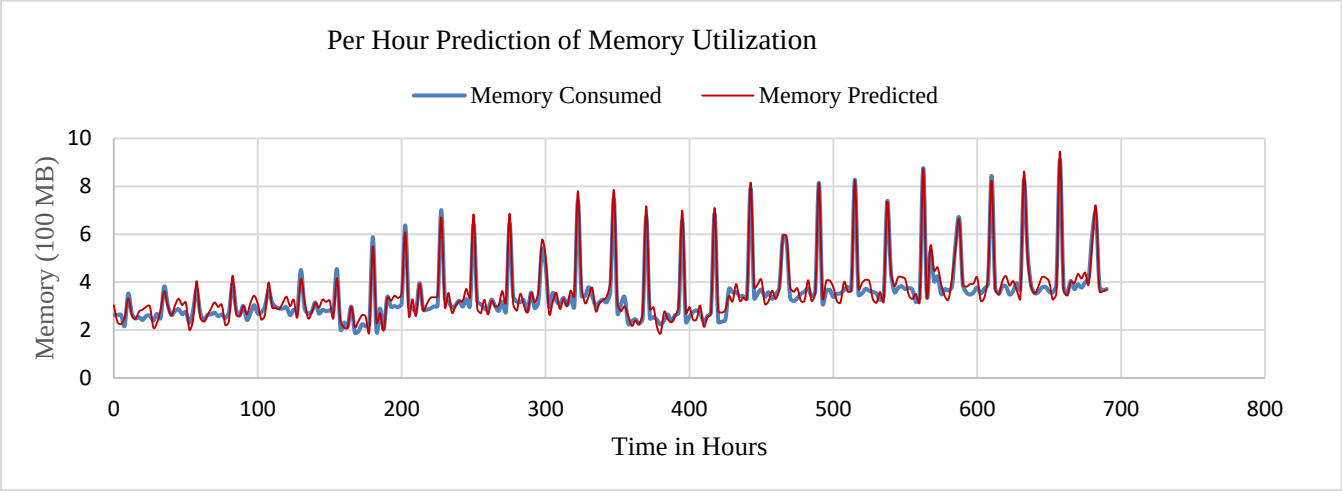

Fig. 7. Prediction of Memory utilization with LSTM Model - Time in Hours.

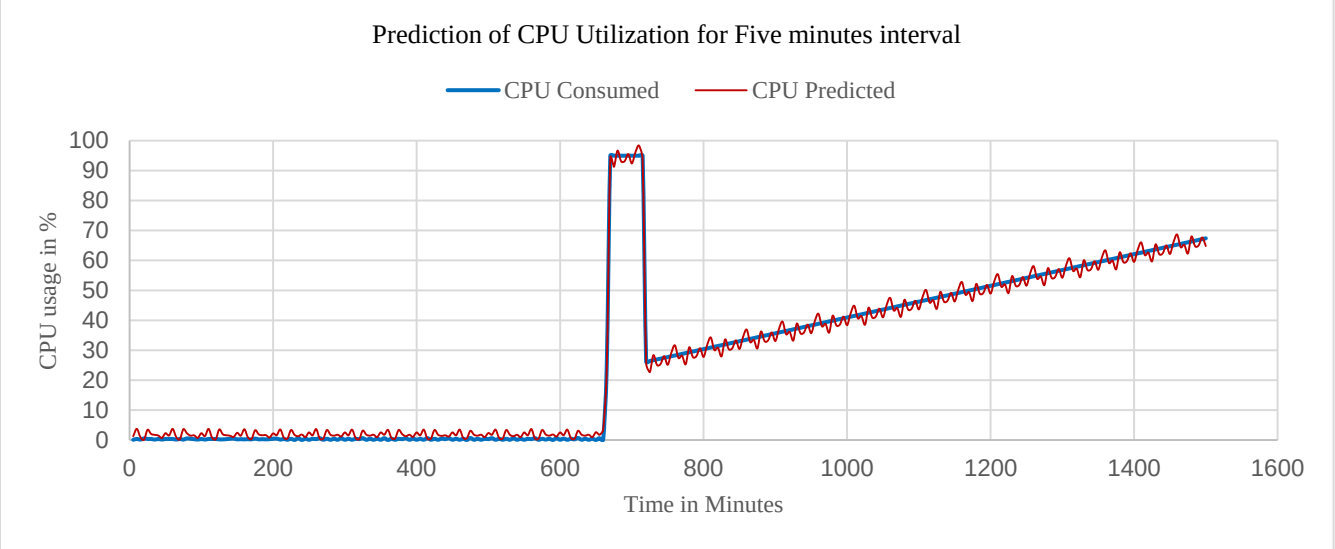

Fig. 8. Prediction of CPU utilization with LSTM Model - Time in Minutes.

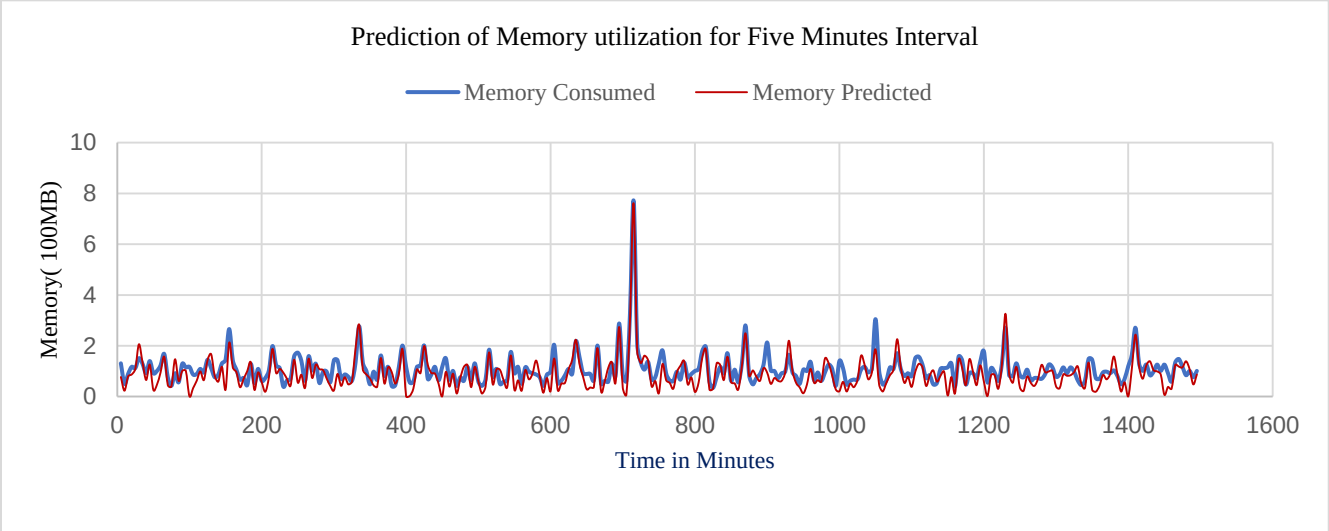

Fig. 9. Prediction of Memory utilization with LSTM Model - Time in Minutes.

TABLE I. PREDICTION ACCURACY COMPARISON - LSTM AND ARIMA MODELS

Model	MAE-Hour	MAPR-Hour	MAE-Minutes	MAPE-Minutes
LSTM-CPU	**5.082**	**0.485**	**8.529**	**0.695**
LSTM-Memory	**6.3835**	**0.642**	**9.071**	**0.802**
ARIMA-CPU	**10.86**	**1.298**	**15.253**	**2.961**
ARIMA-Memory	**13.506**	**1.912**	**19.182**	**3.287**

As seen in Table I, it is noticed that the prediction error was less during hourly forecasting of CPU and memory than forecasting for minutes. Results show that the LSTM model performs outstandingly well when compared to ARIMA model with reduced accuracy. ARIMA model performed better during hourly forecasting when compared to forecasting for minutes using the same model. Thus by overall observation it is found that LSTM models can selectively forget and retain the most relevant information as it flows through various layers. LSTM networks are the excellent model for forecasting workloads as it uses less computational resources when compared to RNN and more accurate forecasting results when compared to statistical models like ARIMA.

V. CONCLUSION

Accurate prediction of resource utilization is necessary for better resource management. This paper presented a prediction model for forecasting the resources like CPU and memory utilization. The model focuses on predicting both seasonality and random workload patterns. SARIMA model was able to predict the seasonality that occur over a period of three days for memory and CPU usage with a MAPE score of 1.52 and 0.49 respectively. Experiment was conducted using fastStorage, real trace data of Bitbrains data center. The results of the proposed method show that LSTM network has better prediction accuracy than ARIMA model. The model attained a MAPE score difference of 0.157 for hourly and 0.107 for minutes prediction of CPU and memory utilization. Thus the proposed workload prediction model is capable of predicting both seasonal and irregular workload patterns which aids in minimizing resource wastage. In future, study focuses on developing an approach particularly task level resource usage prediction.

REFERENCES

[1] Chen, Zheyi, Jia Hu, Geyong Min, Albert Y. Zomaya, and Tarek El-Ghazawi. "Towards accurate prediction for high-dimensional and highly-variable cloud workloads with deep learning." IEEE Transactions on Parallel and Distributed Systems 31, no. 4 (2019): 923-934.

[2] Liu, Jinwei, Haiying Shen, and Liuhua Chen. "CORP: Cooperative opportunistic resource provisioning for short-lived jobs in cloud systems." In 2016 IEEE International Conference on Cluster Computing (CLUSTER), pp. 90-99. IEEE, 2016.

[3] Bi, Jing, Libo Zhang, Haitao Yuan, and MengChu Zhou. "Hybrid task prediction based on wavelet decomposition and ARIMA model in cloud data center." In 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), pp. 1-6. IEEE, 2018.

[4] Amiri, Maryam, and Leyli Mohammad-Khanli. "Survey on prediction models of applications for resources provisioning in cloud." Journal of Network and Computer Applications 82 (2017): pp. 93-113.

[5] Cetinski, Katja, and Matjaz B. Juric. "AME-WPC: Advanced model for efficient workload prediction in the cloud." Journal of Network and Computer Applications 55 (2015): pp. 191-201.

[6] Yu, Yongjia, Vasu Jindal, Farokh Bastani, Fang Li, and I-Ling Yen. "Improving the smartness of cloud management via machine learning based workload prediction." IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), vol. 2, pp. 38-44. IEEE, 2018.

[7] Kaur, Gurleen, Anju Bala, and Inderveer Chana. "An intelligent regressive ensemble approach for predicting resource usage in cloud computing." Journal of Parallel and Distributed Computing 123 (2019): pp. 1-12.

[8] Calheiros, Rodrigo N., Enayat Masoumi, Rajiv Ranjan, and Rajkumar Buyya. "Workload prediction using ARIMA model and its impact on cloud applications' QoS." IEEE transactions on cloud computing 3, no. 4 (2014): pp. 449-458.

[9] Islam, Sadeka, Jacky Keung, Kevin Lee, and Anna Liu. "Empirical prediction models for adaptive resource provisioning in the cloud." Future Generation Computer Systems 28, no. 1 (2012): pp. 155-162.

[10] Borkowski, Michael, Stefan Schulte, and Christoph Hochreiner. "Predicting cloud resource utilization." In Proceedings of the 9th International Conference on Utility and Cloud Computing, pp. 37-42. 2016.

[11] Anupama, K. C., R. Nagaraja, and M. Jaiganesh. "A Perspective view of Resource-based Capacity planning in Cloud computing." 1st International Conference on Advances in Information Technology (ICAIT), pp. 358-363. IEEE, 2019.

[12] Qiu, Feng, Bin Zhang, and Jun Guo. "A deep learning approach for VM workload prediction in the cloud." 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), pp. 319-324. IEEE, 2016.

[13] Wang, Hengjian, John Pannereselvam, Lu Liu, Yao Lu, Xiaojun Zhai, and Haider Ali. "Cloud workload analytics for real-time prediction of user request patterns." IEEE 20th International Conference on High Performance Computing and Communications; pp. 1677-1684. IEEE, 2018.

[14] Guo, Wei, et al. "Short-Term Load Forecasting of Virtual Machines Based on Improved Neural Network." IEEE Access 7 (2019): pp. 121037-121045.

[15] Nguyen, Hoang Minh, Sungpil Woo, Janggwan Im, Taejoon Jun, and Daeyoung Kim. "A workload prediction approach using models stacking based on recurrent neural network and autoencoder." IEEE 18th International Conference on High Performance Computing and Communications pp. 929-936. IEEE, 2016.

[16] Shyam, Gopal Kirshna, and Sunilkumar S. Manvi. "Virtual resource prediction in cloud environment: a Bayesian approach." Journal of Network and Computer Applications 65 (2016): pp. 144-154.

[17] Nashold, Langston, and Rayan Krishnan. "Using LSTM and SARIMA Models to Forecast Cluster CPU Usage." arXiv preprint arXiv:2007.08092, 2020.

[18] Adhikari, Ratnadip, and Ramesh K. Agrawal. "An introductory study on time series modeling and forecasting." arXiv preprint arXiv:1302.6613, 2013.

[19] Parmezan, Antonio Rafael Sabino, Vinicius MA Souza, and Gustavo EAPA Batista. "Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model." Information sciences 484 (2019): pp. 302-337.

[20] Janardhanan, Deepak, and Enda Barrett. "CPU workload forecasting of machines in data centers using LSTM recurrent neural networks and ARIMA models." In 2017 12th International Conference for Internet Technology and Secured Transactions (ICITST), pp. 55-60. IEEE, 2017.

[21] Zia Ullah, Qazi, Shahzad Hassan, and Gul Muhammad Khan. "Adaptive resource utilization prediction system for infrastructure as a service cloud." Computational intelligence and neuroscience 2017.

[22] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): pp. 1735-1780.

[23] Iosup, Alexandru, Hui Li, Mathieu Jan, Shanny Anoep, Catalin Dumitrescu, Lex Wolters, and Dick HJ Epema. "The grid workloads archive." Future Generation Computer Systems 24, no. 7 (2008): pp. 672-686.Available: http://gwa.ewi.tudelft.nl/datasets/gwa-t-12-bitbrains.