Wireless Personal Communications (2021) 116:1949–1969
https://doi.org/10.1007/s11277-020-07773-6

Performance Assessment of Time Series Forecasting Models for Cloud Datacenter Networks' Workload Prediction

Jitendra Kumar[1] · Ashutosh Kumar Singh[2]

Published online: 3 September 2020

Abstract

The resource scaling has been influential in enabling the cloud service providers to provision the resources on-demand effectively. The prior estimation of workloads helps in addressing the scaling issues arises due to dynamic nature of the resource demands. In this paper, we evaluate six different forecasting approaches over real world workload data traces of web and cloud servers. The entire analysis is carried out three times as three different functions are used to measure the deviation in forecasts. The three forecast error measures are root mean squared error, mean absolute error, and mean absolute scaled error. We also carried out a statistical evaluation using Friedman test and Finner post-hoc analysis. The study concludes that the auto ARIMA process outperforms other models and achieves the best rank in the statistical analysis.

Keywords Workload prediction · Resource demand · Time series · Cloud computing · Statistical analysis

1 Introduction

The cloud architectures provide a way to share resources in the form of virtual machines (VMs) among multiple users independent to their geographical locations. Every individual is moving towards cloud technology and getting benefited in a number of ways such as on demand services, elasticity, flexibility, availability and many more. Elasticity is one of the key properties of cloud which provides the flexibility of stretching and compressing virtual hardware resources at any point of time in the lifetime of hosted applications. However, the cloud resource management is complex and challenging task [1, 2] that has raised the possibility of dynamic resource scaling. The resource scaling helps in upholding the quality of services (QoS) promised to its subscribers and can be achieved by two different methods. First, a reactive resource scaling mechanism can be adapted where resources get scaled

✉ Jitendra Kumar
jitendra@nitt.edu

[1] Department of Computer Applications, National Institute of Technology Tiruchirappalli, Tiruchirappalli, India

[2] Department of Computer Applications, National Institute of Technology Kurukshetra, Kurukshetra, India

after arrival of demands. In reactive scaling, users have to wait for certain amount of time to get their requested resources allocated due to the fact that VM initialization is not instantaneous. Unlike reactive methods, the proactive scaling approaches minimize the waiting time by allocating the requested resources in no time. These approaches utilize the estimated workload information for upcoming time instances to scale the resources in advance before actual demand arrives. The forecasts used in resource provisioning have been classified into application and host load predictions [3]. The application predictions are further categorized into three levels i.e. performance, SLA parameters, and workload. The data sets considered in the presented work fall into workload category. The process of workload forecasting involves the use of historical information to extract the patterns of applications running on the server and time series forecasting methods have been extensively explored in workload prediction applications. Some of the key contributions in the domain are listed below.

The web server workload forecast is carried out using an auto regression approach [4]. The model provides the long term and short term estimations and considers the correlation and non stationarity of the workload data. The auto regressive integrated moving average (ARIMA) and its variants have been extensively used to develop forecasting methodologies. For instance, Roy et al. [5] highlighted the challenges in auto scaling of cloud resources and developed a model-predictive estimation approach for resource auto scaling. It also shows that the distributed resources can be allocated in a way that achieves the satisfactory QoS and low operational cost. Similarly, Tran et al. [6] proposed a predictive model for server workload forecasting using seasonal variant of ARIMA process. The approach is one of the integral parts in the project EnergeTic-FUI, France. The accuracy of the model is validated with a wide set of experiments on benchmark datasets. The impact of workload estimations has been analyzed by Calheiros et al. in [7]. The authors modeled the ARIMA process to estimate the workload information of cloud servers. They also studied the aspects of predictive resource management and analyzed the accuracy of predictive model in terms of resource utilization and QoS. It has been claimed that the predictive model receives an accuracy upto 91%. Similarly, a bandwidth allocation scheme using predictive analytic was developed for cloud architectures enabled with software defined networks [8]. The approach collects the data from virtual machines executing on cloud servers and analyzes the collected data to estimate the bandwidth utilization of each VM for upcoming time instance. Then the predicted information is incorporated to generate the bandwidth allocation pattern and allocates the bandwidth in advance to improve the network performance in terms of bandwidth utilization and capacity. Khan et al. advocated the use of multiple time series and grouping the similar applications to improve the forecast accuracy [9]. The approach examines the correlation among VM workload to extract the repeatable patterns of workloads. It also uses a clustering approach to group the applications exhibiting the similar behavior. Then it introduces a predictive method to estimate the variation in workload patterns. The long term QoS requirements has been estimated using multivariate time series while short term advertisements are predicted using univariate time series [10]. Intra correlation of QoS attributes is incorporated to improve the accuracy of predictions and inter correlation helps in obtaining the best possible service composition through time series group similarity.

The time series forecasting methods have also been used in combination with machine learning approaches. For instance, Messias et al. [11] advised to learn the suitable time series forecasting model using a genetic algorithm based learning mechanism as it is challenging task to identify which model suits to workload data under consideration. It becomes more complex if enough historical data is not available. Liu et al. classified the workloads into a number of classes using 0-1 integer programming and each class is assigned with a prediction approach to forecast the workload on the servers [12]. Cetinski and Juric proposed a hybrid approach to forecast the workload on cloud data centers by combining the statistical methods with learning approaches [13]. Rahman et al. proposed to monitor the QoS received by the user to keep an eye on the QoS being provided and received [14]. Khan et al. explored the auto scaling techniques adopted by renowned cloud service providers including Amazon, Google, and Microsoft [15]. The study observed the features and entities of auto scaling approaches. The presented approach allows proactive analysis of workload patterns and estimates the responsiveness of auto scaling operations. Baldán et al. used ARIMA and ES in combination with learning approaches to forecast the workload instances of cloud data centers [16]. The model successfully reduced the under and over provisioning cost up to 30%. Kim et al. carried out an experimental study using a number of different models [17]. The experiments were carried out on an application under four realistic workload patterns, two billing patterns, and three types of predictive scaling. The study revealed that none of the approach is universally best for each workload. Kumar and Mazumdar compared the performance of the different variants of ARMA class such as ARIMA, SARIMA (Seasonal ARIMA), ARFIMA (Fractionally Integrated ARIMA) with SSA (Singular Spectrum Analysis) on CPU, Memory and bandwidth data traces obtained from Wikimedia grid [18]. The performance of ARIMA was found best on network data trace while SSA outperformed other models over CPU and Memory data traces. In general, a number of approaches have been explored to improve the estimation accuracy but every approach lags in producing 100% accurate results [19].

The aim of this article is to assess the performance of the time series forecasting models in workload estimation of different type of workloads from cloud environment. The study comprises six different time series models and their forecast ability is evaluated over five real world data traces. A comprehensive statistical evaluation is also carried out using Friedman test with Finner post-hoc analysis that ranks each model based its forecast accuracy. In addition, the forecast accuracy of each model is compared and statistically evaluated against each other. Further, this paper is organized as follows: Section 2 discusses the workload prediction and its role in cloud data center management. It also lists out the performance evaluation metrics used to evaluate the accuracy of forecasting approaches in this study. The time series prediction models are discussed in Sect. 3. The results of experimental study are presented and discussed in Sect. 5 followed by an analytical study of the experimental findings in Sect. 6. Finally, the paper is wrapped up with conclusive remarks in Sect. 7.

2 Preliminaries

2.1 Workload Prediction

The forecasting plays a vital role in the growth of an organization by estimating the future trends in advance that helps in developing an effective strategic planning [20–25]. Cloud service providers can also use predictive analytics to estimate the future demands for improving the quality of services (QoS) and quality of experience (QoE). A general architecture of cloud system with forecasting module is depicted in Fig. 1 where a device called *resource manager* takes inputs from $\mathcal{F}$ and *current state monitor* to manage the cloud resources effectively. The accuracy of $\mathcal{F}$ is one of the important aspects in improving the

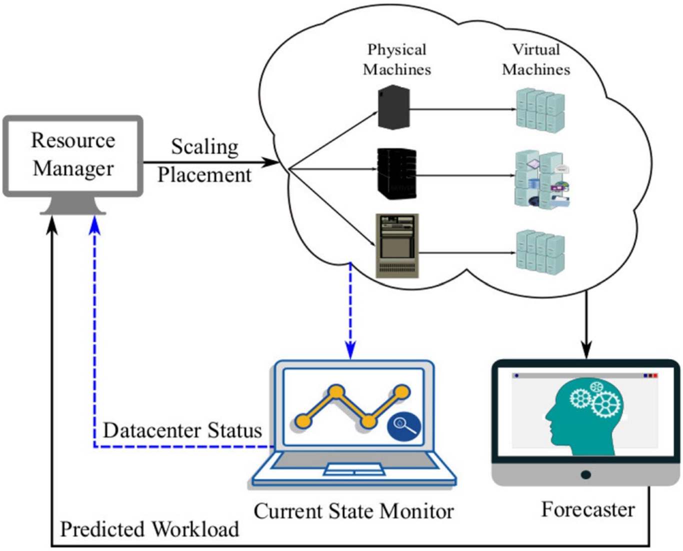


Fig. 1 A view of cloud system with workload forecaster

cloud resource management operations which include scaling, virtual machine placement, minimizing the active physical machines, and others.

Let Y is a set of workload values over time t, where y_t denotes the workload at time t. Also, consider that f is a function of y that analyzes the previous workload instances to estimate the outcome of future events (Eq. (1)). The forecasts are compared against the actual values and forecast error (e_{t+1}) is computed.

$$\hat{y}_{t+1} = f\left(y_t, y_{t-1}, \dots, y_1\right) \tag{1}$$

2.2 Performance Measure Indicators

The accuracy analysis using the individual forecast errors is a challenging task. Therefore, a metric that gives a single number to measure the forecast accuracy of any prediction model is required. A wide range of metrics have been proposed in the literature and each of them carry their distinct merits and demerits, for instance, RMSE is highly sensitive towards the outliers. Thus, we have adopted three evaluation metrics i.e. Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and Mean Absolute Scaled Error (MASE) to measure the accuracy of models.

2.2.1 Root Mean Squared Error (RMSE)

It is one of the widely used metrics and it penalizes a forecast models for large errors [26]. The model is considered to be more accurate if its score is closer to 0. The mathematical representation of the metric is mentioned in (2) where n is the number of data points in workload trace.

$$RMSE(y, \hat{y}) = \sqrt{\frac{1}{n} \sum_{t=1}^{n} (y_t - \hat{y}_t)^2} \tag{2}$$

2.2.2 Mean Absolute Error (MAE)

In root mean squared error the square of higher error values will get more weight which may influence the accuracy of forecaster. While MAE gives equal weight to each error component and measures the accuracy of forecasting model by computing the mean of absolute differences between actual and predicted workloads as shown in (3). The MAE is better indicator of average error then RMSE [27, 28]. It produces a non negative number to evaluate the forecast accuracy and if it is close to 0, forecasts are very much similar to actual values.

$$MAE(y, \hat{y}) = \frac{1}{n} \sum_{t=1}^{n} |y_t - \hat{y}_t| \tag{3}$$

2.2.3 Mean Absolute Scaled Error (MASE)

The metric was proposed by Rob J. Hyndman and Anne B. Koehler to use as a substitute of percentage error metrics [29]. Unlike other accuracy metrics, it produces a good accuracy estimate of a forecasting model when comparing across a number of series of different scale. It scales the forecast errors based on the training mean absolute error from a naïve forecast method and can be computed using (4), where m denotes the seasonal term. We computed MASE using *accuracy* function provided in R and used to select the best model over a number of traces.

$$MASE(y, \hat{y}) = \frac{1}{n} \sum_{t=1}^{n} \left(\frac{|y_t - \hat{y}_t|}{\frac{1}{n-m} \sum_{t=m+1}^{n} |y_t - y_{t-1}|} \right) \tag{4}$$

3 Time Series Forecasting Models

The analysis of time series data began a long back ago in 1927 [30]. The common aim of time series analysis is the forecasting of the trends by extracting meaningful statistics and other characteristics of the data. A number of time series forecasting methods have been derived and this study focuses on six different models.

3.1 Autoregressive (AR)

An autoregressive (AR) states that the future outcome of a variable is a function of its previous values and a white noise term. Since, the variable y is regressed on past occurrences of itself, the method is named as *autoregressive*. Let us consider the past values of a workload variable y at equally spaced time interval $t, t-1, \dots, 1$ are $y_t, y_{t-1}, \dots, y_1$ then an AR process of order p can be expressed as given in Eq. (5) [31].

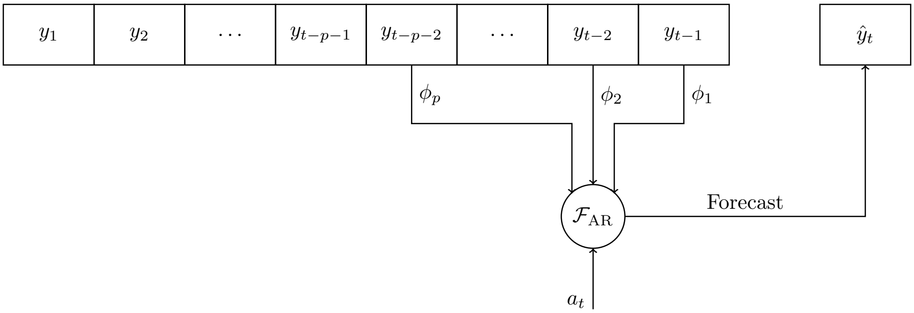


Fig. 2 AR process of order p

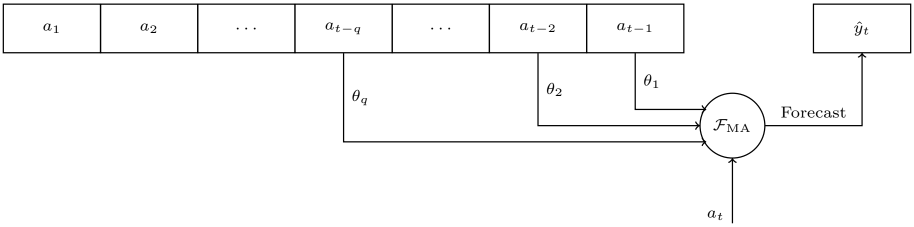


Fig. 3 MA of order q

$$\hat{y}_t = \phi_1 \times y_{t-1} + \phi_2 \times y_{t-2} + \ldots + \phi_p \times y_{t-p} + a_t \tag{5}$$

Figure 2 depicts an AR process of order p, where y_t and $\hat{y}_t$ are the actual and predicted workloads respectively at time instance t. ϕ_i $(i = 1, 2, \ldots, p)$ are model parameters that are restricted to lie between -1 and +1 while a_t is random noise term.

3.2 Moving Average (MA)

Unlike AR process that uses the previous actual values to forecast the upcoming instances, Moving Average (MA) process considers that an observable time series (y_t) can be modeled and generated from a series of independent white noise (a_t), if successive values of y_t highly depends on previous values [32]. These white noise terms are randomly generated using a fixed distribution, and usually having mean zero. A moving average method of order q can be represented as mentioned in Eq. (6), where θ_j $(j = \{1, 2, \ldots, q\})$ are the model parameters (weights) and required neither to be total unity nor to be positive [32]. Figure 3 graphically illustrates the MA process of order q, where $\mathcal{F}_{MA}$ is a function that models random white noise terms to anticipate future values.

$$\hat{y}_t = a_t - \theta_1 \times a_{t-1} - \theta_2 \times a_{t-2} - \ldots - \theta_q \times a_{t-q} \tag{6}$$

3.3 Autoregressive Moving Average (ARMA)

The Autoregressive Moving Average (ARMA) process combines both autoregressive and moving average terms in the model which is sometimes favorable to achieve a greater fit of the actual data. This method parsimoniously describes a time series using two different polynomials, one for auto regression and other one for moving average model. Commonly this method is preferred for modeling of a noisy time series data. Fig. 4 shows the working of an ARMA process with p autoregressive terms and q moving average terms which can be represented using expression (7).

$$\hat{y}_t = \phi_1 \times y_{t-1} + \phi_2 \times y_{t-2} + \cdots + \phi_p \times y_{t-p} + a_t - \theta_1 \times a_{t-1} - \theta_2 \times a_{t-2} - \cdots - \theta_q \times a_{t-q} \tag{7}$$

3.4 Autoregressive Integrated Moving Average (ARIMA)

The Autoregressive Integrated Moving Average (ARIMA) model is more general form of ARMA model. It has three different components namely autoregressive (AR), Integration (I), and Moving Average (MA). It is preferably applied over a non-stationary time series data which can be transformed into a stationary data by applying integration or differencing a number of times. A general ARIMA model of order (p, d, q) is represented as ARIMA(p, d, q) and shown in Fig. 5. Where p, d and q are the number of autoregressive terms, number of differences required, and number of moving average terms in the prediction equation. The differencing operation of order d is denoted using δ^d and $\tilde{y}_t$ represents the value of differenced time series at time t. The simplest form of ARIMA(1,1,1) can be written as mentioned in Eq. (8), where B represents backward shift $(B \times y_t = y_{t-1})$ operator [31].

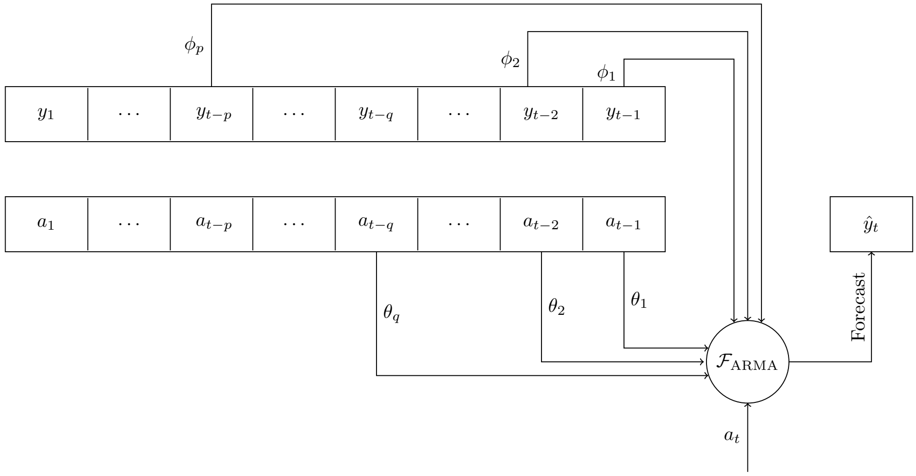


Fig. 4 ARMA of order p and q

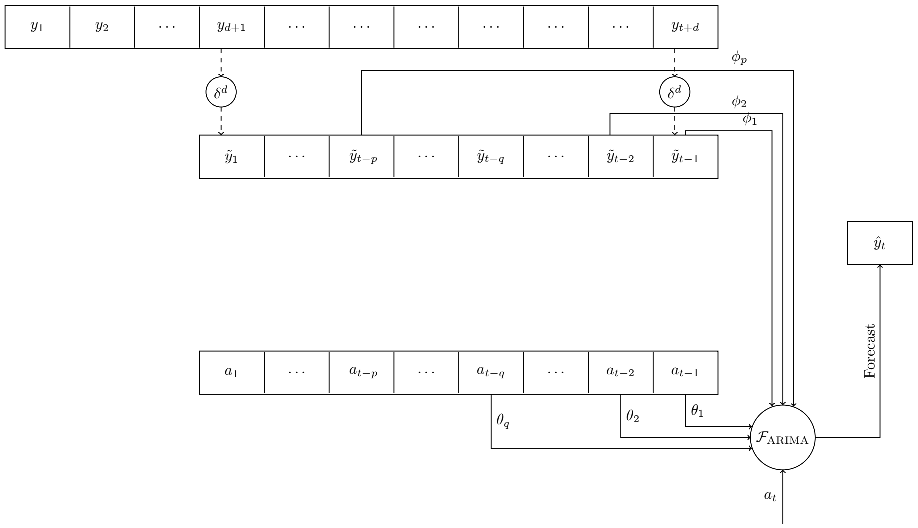


Fig. 5 ARIMA of order p, d, and q

$$\underbrace{(1-\theta_1 B)}_{\text{AR(1)}}\ \underbrace{(1-B)y_t}_{\text{First Difference}} = \underbrace{(1-\theta_1 B)a_t}_{\text{MA(1)}} \tag{8}$$

In addition, the auto ARIMA (referred as M_5) function provided in the `forecast` package of R [33] is also considered to analyze the effect of parameter optimization. It analyzes the data series and finds the optimal values for regression, differencing, and moving average terms. Based on these terms, the order of ARIMA model is identified which is later applied to forecast the trends on the data series.

3.5 Exponential Smoothing (ES)

Exponential smoothing (ES) advocates that the most recent forecasts highly affect the next forecast. The weights of previous predictions reduces exponentially as the data becomes older. The general model for exponential smoothing is given in Eq. (9), where α is the smoothing constant or model's parameter. The visual representation of the model is shown in Fig. 6. The key benefit of the model is that it does not need to store high amount of data to make forecasts.

$$\hat{y}_t = \alpha y_{t-1} + (1-\alpha)\hat{y}_{t-1} \tag{9}$$

3.6 An Illustration

Let Y is a set of randomly sampled workload instances that represents the memory requested by an application. The simulation is carried out using `R` programming language and workload is generated using `sample(min:max, n, replace =`

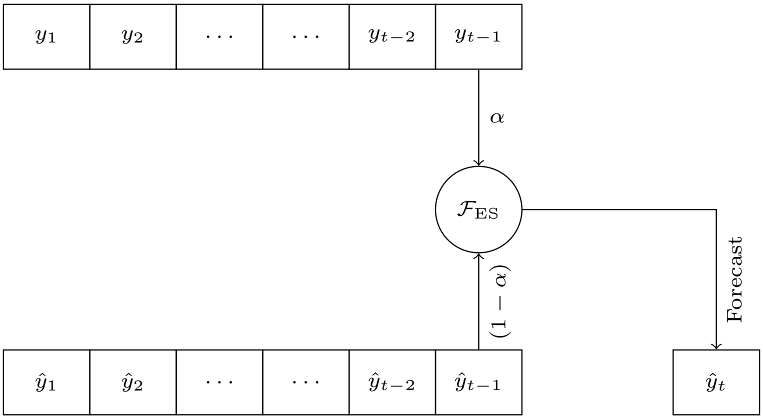


Fig. 6 Exponential smoothing

`FALSE)`. Since, the requested memory is sampled at uniform interval, it is converted into a time series object. Here, *min* and *max* defines the minimum and maximum values in the workload trace respectively. We generated Y of length $n = 40$ with replacement option being *FALSE* that means no value is repeated in generation of workload data. An random forecasting model (M_0 i.e. $\hat{y}_t = y_{t-1} + rand$) is applied over Y to estimate the future values listed as $\hat{Y}^{M_0}$, where *rand* is a random number between 0 and 1. The forecast error ($e_t = y_t - \hat{y}_t$) is measured and listed out as e^{M_0}. The random forecasting model (M_0) estimates the workload with 35.64, 29.42 and 0.98 forecast accuracy as per the RMSE, MAE and MASE metrics that represent a large amount of error in estimations.

$$\begin{aligned}Y = \{&25.00, 21.00, 75.00, 13.00, 44.00, 83.00, 99.00, 30.00, 54.00, 41.00, 90.00, 96.00, 43.00, 46.00,\\ &22.00, 57.00, 28.00, 56.00, 64.00, 61.00, 16.00, 70.00, 32.00, 18.00, 78.00, 69.00, 66.00,\\ &94.00, 12.00, 42.00, 49.00, 87.00, 100.00, 67.00, 37.00, 8.00, 14.00, 35.00, 50.00, 7.00\}\end{aligned}$$

$$\begin{aligned}\hat{Y}^{M_0} = \{&NA, 25.12, 21.90, 75.74, 13.50, 44.84, 83.17, 99.65, 30.09, 54.09, 41.52, 90.04, 96.89,\\ &43.19, 46.21, 22.64, 57.13, 28.08, 56.28, 64.83, 61.81, 16.74, 70.91, 32.62, 18.17, 78.64,\\ &69.21, 66.90, 94.26, 12.91, 42.25, 49.80, 87.79, 100.73, 67.64, 37.25, 8.58, 14.03, 35.19, 50.35\}\end{aligned}$$

$$\begin{aligned}e^{M_0} = \{&NA, -4.12, 53.10, -62.74, 30.50, 38.16, 15.83, -69.65, 23.91, -13.09, 48.48, 5.96,\\ &-53.89, 2.81, -24.21, 34.36, -29.13, 27.92, 7.72, -3.83, -45.81, 53.26, -38.91, -14.62,\\ &59.83, -9.64, -3.21, 27.10, -82.26, 29.09, 6.75, 37.20, 12.21, -33.73, -30.64, -29.25, 5.42, 20.97,\\ &14.81, -43.35\}\end{aligned}$$

Similarly, we evaluate each time series forecasting model on random workload data. The estimations obtained from an autoregressive process (M_1) are listed as $\hat{Y}^{M_1}$ and e^{M_1} represents the forecast errors. The forecasting model randomly selects the values of two parameters $\phi_1 = -0.0093$ and $c = 49.365$. The model generates the 27.81, 23.78, and 0.78 for RMSE, MAE and MASE correspondingly that shows the better forecasts than M_0.

$$\hat{Y}^{M_1} = \{NA, 49.17, 48.67, 49.24, 48.96, 48.59, 48.44, 49.09, 48.86, 48.98, 48.53, 48.47, 48.97, 48.94, 49.16, 48.83, 49.10, 48.84, 48.77, 48.80, 49.22, 48.71, 49.07, 49.20, 48.64, 48.72, 48.75, 48.49, 49.25, 48.97, 48.91, 48.56, 48.44, 48.74, 49.02,$$

$$e^{M_1} = \{NA, -28.17, 26.33, -36.24, -4.96, 34.41, 50.56, -19.09, 5.14, -7.98, 41.47, 47.53, -5.97, -2.94, -27.16, 8.17, -21.10, 7.16, 15.23, 12.20, -33.22, 21.29, -17.07, -31.20, 29.36, 20.28, 17.25, 45.51, -37.25, -6.97, 0.09, 38.44, 51.57, 18.26, -12.02, -41.29, -35.23, -14.04, 1.10, -42.30\}$$

Similarly on applying the first order MA (M_2) with randomly chosen $\theta_1 = -0.126$ and $c = 49.365$ over Y, it forecasts with 28.13, 23.40, and 0.78 forecast errors obtained using RMSE, MAE, and MASE respectively that shows the similar accuracy as of M_1.

$$\hat{Y}^{M_2} = \{NA, 49.37, 52.94, 46.59, 53.60, 50.57, 45.28, 42.60, 50.95, 48.98, 50.37, 44.37, 42.86, 49.35, 49.79, 52.87, 48.84, 51.99, 48.86, 47.46, 47.66, 53.35, 47.27, 51.29, 53.56, 46.29, 46.50, 46.91, 43.43, 53.33, 50.79, 49.59, 44.65, 42.39, 46.26, 50.53, 54.72, 54.50, 51.82, 49.59\}$$

$$e^{M_2} = \{0.00, -28.37, 22.06, -33.59, -9.60, 32.43, 53.72, -12.60, 3.05, -7.98, 39.63, 51.63, 0.14, -3.35, -27.79, 4.13, -20.84, 4.01, 15.14, 13.54, -31.66, 16.65, -15.27, -33.29, 24.44, 22.71, 19.50, 47.09, -31.43, -11.33, -1.79, 37.41, 55.35, 24.61, -9.26, -42.53, -40.72, -19.50, -1.82, -42.59\}$$

After applying an ARMA process (M_3) on Y with p and q both terms being 1 where ϕ_1, θ_1 and c are same as selected in M_1 and M_2, the forecasts have RMSE, MAE, and MASE equal to 27.34, 23.17, and 0.77 respectively. The forecasts of M_3 are bit more accurate than M_1 and M_2.

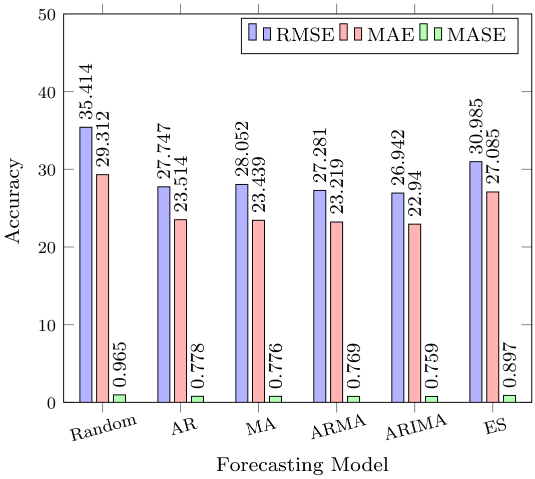


Fig. 7 Forecast accuracy on random data

Table 1 List of data traces and corresponding acronyms

Type	Dataset	Workload characterstic	Acronym
Web Server Logs	NASA Server	Number of HTTP Requests	D_1
	Calgary Server	Number of HTTP Requests	D_2
	Saskatchewan Server	Number of HTTP Requests	D_3
Google Cluster Resource Traces	CPU trace	Amount of CPU Cores Requested	D_4
	Memory trace	Amount of RAM (MB) requested	D_5

$$\begin{aligned}\hat{Y}^{M_3} = \{&NA, 49.13, 45.62, 52.36, 44.28, 48.92, 52.88, 54.25, 46.03, 49.86, 47.86, 53.83,\\ &53.78, 47.60, 48.73, 45.79, 50.24, 46.30, 50.06, 50.52, 50.11, 44.91, 51.87, 46.56,\\ &45.59, 52.72, 50.77, 50.67, 53.95, 43.96, 48.72, 48.94, 53.35, 54.31, 50.34, 47.34,\\ &44.33, 45.41, 47.72, 49.18\}\\ e^{M_3} = \{&0.00, -28.13, 29.38, -39.36, -0.28, 34.08, 46.12, -24.25, 7.97, -8.86, 42.14,\\ &42.17, -10.78, -1.60, -26.73, 11.21, -22.24, 9.70, 13.94, 10.48, -34.11, 25.09,\\ &-19.87, -28.56, 32.41, 16.28, 15.23, 43.33, -41.95, -1.96, 0.28, 38.06, 46.65, 12.69,\\ &-13.34, -39.34, -30.33, -10.41, 2.28, -42.18\}\end{aligned}$$

We applied ARIMA(1,1,1) (M_4) also over Y which improves the accuracy more by producing forecasts with 26.78, 22.35, and 0.76 values of RMSE, MAE, and MASE respectively.

$$\begin{aligned}\hat{Y}^{M_4} = \{&NA, NA, 51.20, 53.66, 46.62, 50.55, 54.89, 56.57, 48.46, 51.64, 49.95, 55.76, 56.18,\\ &50.00, 50.63, 47.78, 52.00, 48.41, 51.86, 52.62, 52.25, 47.02, 53.56, 48.80, 47.41,\\ &54.46, 53.08, 52.82, 56.09, 46.37, 50.34, 50.93, 55.36, 56.67, 52.77, 49.46, 46.21,\\ &47.05, 49.45, 51.09\}\end{aligned}$$

$$\begin{aligned}e^{M_4} = \{&NA, 0.00, 23.80, -40.66, -2.62, 32.45, 44.11, -26.57, 5.54, -10.64, 40.05, 40.24,\\ &-13.18, -4.00, -28.63, 9.22, -24.00, 7.59, 12.14, 8.38, -36.25, 22.98, -21.56,\\ &-30.80, 30.59, 14.54, 12.92, 41.18, -44.09, -4.37, -1.34, 36.07, 44.64, 10.33,\\ &-15.77, -41.46, -32.21, -12.05, 0.55, -44.09\}\end{aligned}$$

We evaluated an exponential smoothing model (M_6) over Y with randomly selected $\alpha = 0.5$ which forecasts the workload trace with RMSE, MAE, and MASE equal to 31.12, 27.44, and 0.90 values. The forecasts of M_6 are better than M_0 only. However, it may produce more accurate forecasts with better values of α.

$$\begin{aligned}\hat{Y}^{M_6} = \{&NA, 49.98, 35.49, 55.24, 34.12, 39.06, 61.03, 80.02, 55.01, 54.50, 47.75, 68.88,\\ &82.44, 62.72, 54.36, 38.18, 47.59, 37.79, 46.90, 55.45, 58.22, 37.11, 53.56,\\ &42.78, 30.39, 54.19, 61.60, 63.80, 78.90, 45.45, 43.72, 46.36, 66.68, 83.34, 75.17,\\ &56.09, 32.04, 23.02, 29.01, 39.51\}\end{aligned}$$

$$\begin{aligned}e^{M_6} = \{&NA, -28.98, 39.51, -42.24, 9.88, 43.94, 37.97, -50.02, -1.01, -13.50, 42.25,\\ &27.12, -39.44, -16.72, -32.36, 18.82, -19.59, 18.21, 17.10, 5.55, -42.22, 32.89,\\ &-21.56, -24.78, 47.61, 14.81, 4.40, 30.20, -66.90, -3.45, 5.28, 40.64, 33.32, -16.34,\\ &-38.17, -48.09, -18.04, 11.98, 20.99, -32.51\}\end{aligned}$$

Table 2 Performance of AR(1)

PWS	Error metric	D_1	D_2	D_3	D_4	D_5
5	RMSE	5.10E+01	1.60E+01	2.76E+01	5.87E+02	4.38E+02
	MAE	3.72E+01	1.13E+01	1.97E+01	2.60E+02	2.12E+02
	MASE	1.01E+00	7.74E-01	8.39E-01	1.02E+00	1.02E+00
10	RMSE	8.56E+01	3.21E+01	6.86E+01	1.25E+03	8.47E+02
	MAE	6.23E+01	1.95E+01	4.28E+01	5.23E+02	4.06E+02
	MASE	1.02E+00	7.93E-01	1.02E+00	1.07E+00	1.03E+00
20	RMSE	1.50E+02	9.22E+01	1.51E+02	2.82E+03	1.84E+03
	MAE	1.10E+02	8.30E+01	1.25E+02	1.20E+03	8.67E+02
	MASE	1.02E+00	1.04E+00	1.18E+00	1.15E+00	1.05E+00
30	RMSE	2.26E+02	4.93E+01	6.33E+01	3.98E+03	2.59E+03
	MAE	1.60E+02	3.64E+01	3.89E+01	1.85E+03	1.29E+03
	MASE	1.02E+00	8.78E-01	9.77E-01	1.14E+00	1.02E+00
60	RMSE	1.65E+02	7.43E+01	8.69E+01	7.10E+03	4.77E+03
	MAE	1.21E+02	5.98E+01	5.92E+01	3.90E+03	2.73E+03
	MASE	1.17E+00	8.76E-01	9.73E-01	1.06E+00	9.80E-01

Table 3 Performance of MA(1)

PWS	Error metric	D_1	D_2	D_3	D_4	D_5
5	RMSE	7.86E+01	1.62E+01	2.82E+01	7.11E+02	5.53E+02
	MAE	6.17E+01	1.14E+01	2.04E+01	3.80E+02	3.08E+02
	MASE	1.68E+00	7.81E-01	8.68E-01	1.50E+00	1.49E+00
10	RMSE	1.48E+02	3.22E+01	6.99E+01	1.37E+03	9.91E+02
	MAE	1.17E+02	1.96E+01	4.48E+01	7.02E+02	5.62E+02
	MASE	1.92E+00	7.93E-01	1.06E+00	1.44E+00	1.43E+00
20	RMSE	2.79E+02	9.26E+01	1.54E+02	2.97E+03	2.13E+03
	MAE	2.23E+02	8.43E+01	1.32E+02	1.47E+03	1.16E+03
	MASE	2.06E+00	1.05E+00	1.25E+00	1.40E+00	1.40E+00
30	RMSE	3.95E+02	5.08E+01	7.11E+01	4.16E+03	2.92E+03
	MAE	3.18E+02	3.80E+01	4.89E+01	2.19E+03	1.64E+03
	MASE	2.04E+00	9.16E-01	1.23E+00	1.35E+00	1.30E+00
60	RMSE	1.75E+02	7.58E+01	1.11E+02	7.47E+03	5.18E+03
	MAE	1.41E+02	6.11E+01	8.18E+01	4.40E+03	3.25E+03
	MASE	1.36E+00	8.96E-01	1.34E+00	1.20E+00	1.17E+00

Figure 7 shows the forecast accuracy of each model on random data. It can be seen that the ARIMA process generates better forecasts according to all three evaluation metrics i.e. RMSE, MAE and MASE. We conducted a comprehensive experimental study on various data sets to analyze the behavior of each model.

Table 4 Performance of ARMA(1,1)

PWS	Error metric	D_1	D_2	D_3	D_4	D_5
5	RMSE	4.56E+01	1.56E+01	2.52E+01	5.87E+02	4.29E+02
	MAE	3.23E+01	1.10E+01	1.79E+01	2.60E+02	2.02E+02
	MASE	8.79E-01	7.54E-01	7.63E-01	1.02E+00	9.73E-01
10	RMSE	7.95E+01	3.19E+01	6.40E+01	1.25E+03	8.42E+02
	MAE	5.66E+01	1.95E+01	3.89E+01	5.20E+02	3.96E+02
	MASE	9.25E-01	7.90E-01	9.25E-01	1.06E+00	1.01E+00
20	RMSE	1.47E+02	9.16E+01	1.44E+02	2.81E+03	1.81E+03
	MAE	1.07E+02	8.20E+01	1.13E+02	1.17E+03	8.62E+02
	MASE	9.92E-01	1.02E+00	1.07E+00	1.12E+00	1.04E+00
30	RMSE	2.25E+02	4.87E+01	5.79E+01	3.95E+03	2.59E+03
	MAE	1.59E+02	3.60E+01	3.57E+01	1.81E+03	1.29E+03
	MASE	1.02E+00	8.69E-01	8.96E-01	1.12E+00	1.02E+00
60	RMSE	1.62E+02	7.42E+01	8.43E+01	7.03E+03	4.74E+03
	MAE	1.19E+02	5.98E+01	5.86E+01	3.88E+03	2.75E+03
	MASE	1.15E+00	8.77E-01	9.63E-01	1.06E+00	9.87E-01

4 Simulation Environment and Datasets

The experiments are carried out on a machine equipped with Intel® Core™ I7-7700 processor of 3.60GHz clock speed. The machine consists 8 GB of main memory. The models are implemented using R. This analytical study is performed over five different benchmark data sets. The data sets are NASA server HTTP Trace, Calgary HTTP Trace, Saskatchewan server HTTP Trace, [34] and Google Cluster Trace [35]. We selected CPU and memory

Table 5 Performance of ARIMA(1,1,1)

PWS	Error metric	D_1	D_2	D_3	D_4	D_5
5	RMSE	4.54E+01	1.58E+01	2.52E+01	5.84E+02	4.30E+02
	MAE	3.17E+01	1.12E+01	1.80E+01	2.55E+02	2.02E+02
	MASE	8.64E-01	7.69E-01	7.68E-01	1.00E+00	9.77E-01
10	RMSE	7.97E+01	3.20E+01	6.43E+01	1.24E+03	8.42E+02
	MAE	5.56E+01	1.97E+01	3.84E+01	5.07E+02	3.99E+02
	MASE	9.08E-01	7.98E-01	9.12E-01	1.04E+00	1.01E+00
20	RMSE	1.48E+02	9.18E+01	1.47E+02	2.79E+03	1.83E+03
	MAE	1.05E+02	8.15E+01	1.07E+02	1.15E+03	8.63E+02
	MASE	9.73E-01	1.02E+00	1.02E+00	1.10E+00	1.04E+00
30	RMSE	2.28E+02	4.87E+01	5.86E+01	3.94E+03	2.58E+03
	MAE	1.55E+02	3.60E+01	3.60E+01	1.78E+03	1.30E+03
	MASE	9.92E-01	8.69E-01	9.05E-01	1.10E+00	1.02E+00
60	RMSE	1.65E+02	7.35E+01	8.63E+01	7.06E+03	4.79E+03
	MAE	1.19E+02	5.92E+01	5.95E+01	3.78E+03	2.76E+03
	MASE	1.15E+00	8.67E-01	9.79E-01	1.03E+00	9.89E-01

Table 6 Performance of Auto Arima

PWS	Error metric	D_1	D_2	D_3	D_4	D_5
5	RMSE	4.53E+01	1.58E+01	2.52E+01	5.82E+02	4.25E+02
	MAE	3.17E+01	1.12E+01	1.80E+01	2.56E+02	2.02E+02
	MASE	8.63E-01	7.69E-01	7.68E-01	1.01E+00	9.75E-01
10	RMSE	7.97E+01	3.19E+01	6.43E+01	1.23E+03	8.20E+02
	MAE	5.56E+01	1.96E+01	3.84E+01	5.04E+02	3.92E+02
	MASE	9.08E-01	7.95E-01	9.12E-01	1.03E+00	9.96E-01
20	RMSE	1.48E+02	9.14E+01	1.47E+02	2.79E+03	1.79E+03
	MAE	1.05E+02	8.09E+01	1.07E+02	1.15E+03	8.61E+02
	MASE	9.72E-01	1.01E+00	1.02E+00	1.10E+00	1.04E+00
30	RMSE	2.25E+02	4.83E+01	5.86E+01	3.94E+03	2.56E+03
	MAE	1.54E+02	3.58E+01	3.60E+01	1.78E+03	1.30E+03
	MASE	9.89E-01	8.63E-01	9.05E-01	1.10E+00	1.02E+00
60	RMSE	1.70E+02	7.26E+01	8.63E+01	7.03E+03	4.71E+03
	MAE	1.11E+02	5.83E+01	5.95E+01	3.82E+03	2.71E+03
	MASE	1.08E+00	8.55E-01	9.79E-01	1.04E+00	9.73E-01

resource requests data traces from Google cluster traces. In this paper, we refer these traces as D_1, D_2, D_3, D_4, and D_5 respectively as listed out in Table 1.

NASA server traces are two HTTP request traces consisting of two months requests to the NASA Kennedy Space Center WWW server in Florida. In our experiments, we have considered only one month request data. The data logs are stored in ASCII file format distributed in five different fields. Each record contains *host*, *timestamp*, *request*, *HTTP reply code*, and *bytes*. Here, *host* indicates the hostname or an IP address of the requesting

Table 7 Performance of exponential smoothing

PWS	Error metric	D_1	D_2	D_3	D_4	D_5
5	RMSE	4.57E+01	1.53E+01	2.51E+01	6.24E+02	4.39E+02
	MAE	3.19E+01	1.10E+01	1.85E+01	2.41E+02	1.97E+02
	MASE	2.55E+00	9.72E-01	9.72E-01	1.41E+00	1.45E+00
10	RMSE	7.99E+01	3.25E+01	4.75E+01	1.32E+03	8.74E+02
	MAE	5.57E+01	1.94E+01	2.33E+01	4.78E+02	3.84E+02
	MASE	3.00E+00	7.89E-01	1.47E+00	1.77E+00	1.69E+00
20	RMSE	1.48E+02	1.97E+00	2.76E+00	3.06E+03	1.94E+03
	MAE	1.05E+02	3.07E+03	5.37E+03	1.03E+03	8.23E+02
	MASE	3.26E+00	9.08E-01	8.00E-01	1.09E+01	9.68E+00
30	RMSE	2.28E+02	7.44E-01	1.19E+00	4.46E+03	2.78E+03
	MAE	1.55E+02	2.31E+03	2.60E+03	1.62E+03	1.27E+03
	MASE	3.33E+00	1.12E+00	3.46E+00	1.48E+00	5.03E+00
60	RMSE	8.08E+00	7.71E+01	3.38E-01	8.12E+03	5.18E+03
	MAE	1.14E+04	6.16E+01	1.33E+04	3.66E+03	2.78E+03
	MASE	2.22E+00	1.04E+00	4.19E+00	1.02E+00	1.15E+00

machine, *timestamp* denotes the time of request in the format “DAY MON DD HH:MM:SS YYYY” with timezone − 0400, *request* is mentioned in quotes, next field indicates the reply code, and last field shows the bytes sent in reply of the request. Similarly, Calgary and Saskatchewan traces contain HTTP requests worth of one year and seven month’s HTTP requests to the University of Calgary’s department of computer science WWW server and University of Saskatchewan’s WWW server respectively. The both servers are respectively located at Calgary, Alberta, Canada and Saskatoon, Saskatchewan, Canada. The other dataset i.e. Google cluster trace was released by Google in 2011 that incorporates the data of 29 days collected from organization’s cluster cell. The data contains 10, 388 machines, over 0.67 million jobs, 20 million tasks. Here, a job may consist multiple tasks and each task may further have multiple processes and arranged in a set of six different tables.

5 Forecast Evaluation

We carried out a number of experiments with different prediction window size (PWS) to study the performance behavior of time series forecasting models. The prediction window size defines the forecast time interval e.g. if the length of the window is 10 minutes, the model predicts upcoming workload on the server after each 10 min. The experiments are performed with prediction window of 5, 10, 20, 30, and 60 min duration. The 60% of the data is used to estimate the model parameters and the remaining data is used to test the forecast accuracy of estimated model.

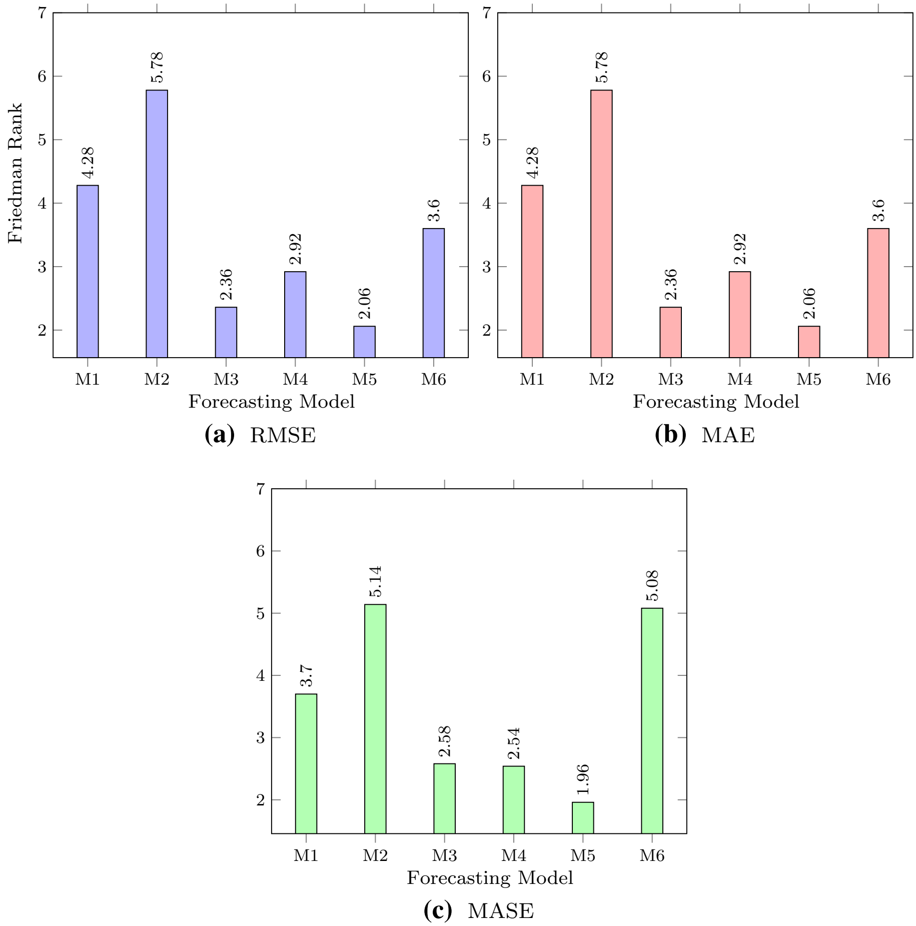
The forecast error in the experiments carried out with first order autoregressive model are reported in Table 2. The model produces least root mean squared error over Calgary traces (D_2) in all experiments with different lengths of prediction window. In the case of mean absolute error, the least error was measured again on D_2 for 5, 10, 20 and 40 minutes of prediction window. The Saskatchewan trace (D_3) receives the least mean absolute error when the length of window is 60 minutes. The mean absolute scaled error measured the minimum forecast error in the predictions of D_2 for the cases when prediction window is of length 5, 10, 30, and 60 minutes. The NASA trace (D_1) achieves the better forecast accuracy than other traces for 20 minutes of PWS. It is also observed that the first order autoregressive model performs better on web server workloads rather than cloud server workloads. In cloud server workloads, the memory trace i.e. D_5 attains better results. Based on MASE results, it can be generalized that AR(1) performs better on Calgary trace.

The similar results were observed in the experiments performed with first order moving average forecasting model. The measured error in forecasts of MA process are tabulated in Table 3 which shows that the model attained better accuracy over Calgary trace (D_2) for all values of prediction window. The forecasts of web server workloads were found more accurate than cloud server workloads. On comparing the performance of the model over different cloud workloads, the memory trace (D_5) achieved better forecasts than D_4 i.e. CPU demand trace.

We performed the same set of experiments with ARMA(1,1) and the results are listed in Table 4. The achieved forecast errors indicate that the model produces lower root mean squared error for D_2 forecasts in all instances of prediction window. The Saskatchewan trace received lower mean absolute error for prediction lookup window of 30 and 60 min. For other values of PWS D_2 got better forecasts according to mean absolute error. D_2 also witnesses better results in terms of mean absolute scaled error except one instance i.e.

Table 8 Friedman test ($\alpha = 0.05$)

Accuracy metric	F-Statistics	p-value	H_0 Result
RMSE	28.67	0.0	$H_0.R$
MAE	19.46	0.0	$H_0.R$
MASE	27.63	0.0	$H_0.R$



(a) RMSE

(b) MAE

(c) MASE

Fig. 8 Friedman test ranks

when prediction lookup window is 20 min. The experimental results depict similar trend in the accuracy of web and cloud server workloads as shown in previous two models.

We performed experiments with ARIMA(1,1,1) and auto ARIMA also. The forecast errors measured in forecasts of both models are shown in Tables 5 and 6 respectively. The results indicate that both models produced forecasts with similar accuracy. In both cases, the root mean squared error and mean absolute error of Calgary trace forecasts were found better than other traces. According to mean absolute scaled error metric, the Saskatchewan

Table 9 Post hoc analysis using Finner test (Controle Method=M_5)

Comparison	RMSE			MAE			MASE		
	Statistic	p-value	H_0^p Result	Statistic	p-value	H_0^p Result	Statistic	p-value	H_0^p Result
M_1 vs M_2	2.83	0.00858	$H_0^p.R$	2.61	0.01942	$H_0^p.R$	2.72	0.01216	$H_0^p.R$
M_1 vs M_3	3.63	0.00071	$H_0^p.R$	2.31	0.03497	$H_0^p.R$	2.12	0.04647	$H_0^p.R$
M_1 vs M_4	2.57	0.01688	$H_0^p.R$	2.80	0.01285	$H_0^p.R$	2.19	0.04225	$H_0^p.R$
M_1 vs M_5	4.20	0.00010	$H_0^p.R$	3.86	0.00035	$H_0^p.R$	3.29	0.00216	$H_0^p.R$
M_1 vs M_6	1.29	0.24195	$H_0^p.A$	2.49	0.02352	$H_0^p.R$	2.61	0.01513	$H_0^p.R$
M_2 vs M_3	6.46	0.00000	$H_0^p.R$	4.91	0.00000	$H_0^p.R$	4.84	0.00000	$H_0^p.R$
M_2 vs M_4	5.40	0.00000	$H_0^p.R$	5.40	0.00000	$H_0^p.R$	4.91	0.00000	$H_0^p.R$
M_2 vs M_5	7.03	0.00000	$H_0^p.R$	6.46	0.00000	$H_0^p.R$	6.01	0.00000	$H_0^p.R$
M_2 vs M_6	4.12	0.00011	$H_0^p.R$	5.10	0.00000	$H_0^p.R$	0.11	0.92397	$H_0^p.A$
M_3 vs M_4	1.06	0.30707	$H_0^p.A$	0.49	0.67571	$H_0^p.A$	0.08	0.93974	$H_0^p.A$
M_3 vs M_5	0.57	0.57075	$H_0^p.A$	1.55	0.17621	$H_0^p.A$	1.17	0.29194	$H_0^p.A$
M_3 vs M_6	2.34	0.02853	$H_0^p.R$	0.19	0.85011	$H_0^p.A$	4.72	0.00001	$H_0^p.R$
M_4 vs M_5	1.63	0.13922	$H_0^p.A$	1.06	0.34817	$H_0^p.A$	1.10	0.30784	$H_0^p.A$
M_4 vs M_6	1.29	0.24195	$H_0^p.A$	0.30	0.78555	$H_0^p.A$	4.80	0.00000	$H_0^p.R$
M_5 vs M_6	2.91	0.00772	$H_0^p.R$	1.36	0.22898	$H_0^p.A$	5.90	0.00000	$H_0^p.R$

trace and NASA trace forecasts were better when the prediction window was 5 and 20 min duration.

The last model we experimented with is exponential smoothing and the forecasting errors are listed in the Table 7. Unlike the previous models, it produces the mix results. For instance, according to root mean square metric the D_2 forecasts are less erroneous for prediction window of length 5, 10, 20 and 30 minutes. For the case of 60 minutes, Saskatchewan trace receives better forecast. And MAE indicates that D_2 forecasts are better when prediction window duration is 5, 10 and 60 min and D_1 forecasts are better in other cases. On comparing the MASE, the model performs better on D_2 in case of 10 and 30 min prediction window. In case of 20 and 60 minutes interval, the model performs better on Saskatchewan and CPU trace respectively. While it performs equally better on D_2 and D_3 for 5 minutes prediction window. Again the model performs better on web server workloads in most of the cases and it gives mixed performance over cloud server traces.

6 Statistical Analysis

This section analyzes the statistics of forecast accuracy observed by different prediction models that are reported in previous section. The experimental findings report the forecast accuracy on a particular data trace only and it becomes difficult to rank the models. The statistical analysis helps in finding out whether the difference in results are significant or not. We have used the Friedman test [36] due the fact that it is considered to be most powerful test when five or more models' performance is compared [37]. The Friedman test considers a null hypothesis (H_0) that assumes the accuracy of each model is same in nature and no significant difference in the results is present. The alternate hypothesis (H_1)

advocates that the experimental findings of at least one model are significantly different than others. Provided k models' performance is evaluated on d different datasets, a rank ($\mathcal{R}_{\xi_{ij}}$) for an experimental finding i.e. forecast accuracy of j^{th} forecasting model (M_j) on i^{th} dataset (D_i) is computed for all datasets. These ranks are summed up to compute the rank of a forecasting model ($\mathcal{R}_{M_j}$) as shown in Eq. (10). The test further computes the Friedman statistic commonly referred as F-Statistic as given in Eq. (11), where Q can be obtained from Eq. (12). The F-Statistic is tested against the F-quantiles for a given α with degree of freedom, $f_1 = k - 1$ and $f_2 = (d-1)(k-1)$, where α is the significance level being considered. The study includes six models ($k = 6$) and five data traces ($d = 5$). The statistical results are reported with $\alpha = 0.05$ as standard.

$$\mathcal{R}_{M_j} = \sum_{i=1}^{d} \mathcal{R}_{\xi_{ij}}; j \in \{1, 2, \dots, k\} \tag{10}$$

$$F - Statistic = \frac{(d-1)Q}{d(k-1) - Q} \tag{11}$$

$$Q = \frac{12}{dk(k+1)} \sum_{j=1}^{k} \left(\mathcal{R}_{M_j} - \frac{d(k+1)}{2} \right)^2 \tag{12}$$

We applied the statistical test using the STAC web platform [38]. The Friedman test observed the 28.67 and 0.0 values of F-Statistic and p-value respectively that resulted in the rejection of null hypothesis H_0 for RMSE based forecast accuracy as it finds the significant difference in the predictions of different models. Similarly, it also rejects the H_0 for MAE and MASE based forecast accuracy evaluation. The F-Statistics and p-values for both findings are given in Table 8, where $H_0.R$ denotes the rejection of H_0. The Friedman ranks obtained by each model are shown in Fig. 8, where Figs. 8a,b and c depict the ranks given to the forecasting models for RMSE, MAE and MASE based accuracy evaluation respectively.

The Friedman test reports the presence of any significant difference in the results and provides a rank to each model based on the experimental results. However, post-hoc analysis methods can be used to explore the statistical properties of experimental findings further. We have used Finner post-hoc analysis [39] that performs multiple comparisons. The test considers one of the methods under evaluation as a control method. It works around a null hypothesis (H_0^p) where p denotes the null hypothesis for post-hoc analysis. The H_o^p assumes that the mean of the control method's result is equal with each other group member under test. The pairwise comparison results obtained from Finner post-hoc method are listed out in Table 9, where $H_0^p.R$ and $H_0^p.A$ represent the rejection and acceptance of H_0^p. It can be observed that the analysis rejects the H_0^p in most of the comparisons. Based on these statistical findings, it can be observed that M_3, M_4 and M_5 produced the similar forecast accuracy and presence of significant difference is not noticed. The rankings of these models also exhibit the similar behavior as no major difference in the rankings is observed. However, M_5 received the best ranking among these models.

7 Conclusions

Forecasting have been always beneficial to a wide range of applications in their business decision making. This work studies the performance behavior of time series forecasting schemes for server workload estimation in cloud environment. The study analyses the performance of six different forecasting models on three web server workload traces and two cloud server workload traces. We have analyzed the experimental findings of each model using statistical tests including Friedman test and Finner post-hoc analysis. The study also finds that the three models M_3, M_4 and M_5 produced more or less similar forecast accuracy. But the performance of M_5 is observed to be best among the considered models.

Acknowledgements The authors would like to thank Ministry of Electronics and Information Technology (MeitY), Govt. of India for financial support to carry out this work.

References

1. Sebrechts, M., Van Seghbroeck, G., Wauters, T., Volckaert, B., & De Turck, F. (2018). Orchestrator conversation: Distributed management of cloud applications. *International Journal of Network Management*, *28*(6), e2036.
2. Singh, A. K., & Kumar, J. (2019). Secure and energy aware load balancing framework for cloud data centre networks. *Electronics Letters*, *55*(9), 540–541.
3. Amiri, M., & Mohammad-Khanli, L. (2017). Survey on prediction models of applications for resources provisioning in cloud. *Journal of Network and Computer Applications*, *82*(Supplement C), 93–113.
4. Li, T.-H. (2005). A hierarchical framework for modeling and forecasting web server workload. *Journal of the American Statistical Association*, *100*(471), 748–63.
5. Roy, N., Dubey, A., & Gokhale, A. (2011). Efficient autoscaling in the cloud using predictive models for workload forecasting. In 2*011 IEEE 4th International Conference on Cloud Computing*, pp. 500–507.
6. Tran, V. G., Debusschere, V., & Bacha, S. (2012). Hourly server workload forecasting up to 168 hours ahead using seasonal arima model. *IEEE International Conference on Industrial Technology*, *2012*, 1127–1131.
7. Calheiros, R. N., Masoumi, E., Ranjan, R., & Buyya, R. (2015). Workload prediction using arima model and its impact on cloud applications' qos. *IEEE Transactions on Cloud Computing*, *3*(4), 449–458.
8. Cao, J., Zhu, X., Dong, F., Liu, B., Ma, Z., & Min, H. (2016). Time series based bandwidth allocation strategy in cloud datacenter. *International Conference on Advanced Cloud and Big Data (CBD)*, *2016*, 228–233.
9. Khan, A., Yan, X., Tao, S., & Anerousis, N. (2012). Workload characterization and prediction in the cloud: A multiple time series approach. *IEEE Network Operations and Management Symposium*, *2012*, 1287–1294.
10. Ye, Z., Mistry, S., Bouguettaya, A., & Dong, H. (2016). Long-term qos-aware cloud service composition using multivariate time series analysis. *IEEE Transactions on Services Computing*, *9*(3), 382–393.
11. Messias, V. R., Estrella, J. C., Ehlers, R., Santana, M. J., Santana, R. C., & Reiff-Marganiec, S. (2016). Combining time series prediction models using genetic algorithm to autoscaling web applications hosted in the cloud infrastructure. *Neural Computing and Applications*, *27*(8), 2383–2406.
12. Liu, C., Liu, C., Shang, Y., Chen, S., Cheng, B., & Chen, J. (2017). An adaptive prediction approach based on workload pattern discrimination in the cloud. *Journal of Network and Computer Applications*, *80*(C), 35–44.
13. Cetinski, K., & Juric, M. B. (2015). Ame-wpc: Advanced model for efficient workload prediction in the cloud. *Journal of Network and Computer Applications*, *55*, 191–201.
14. Rahman, Z.U., Hussain, O.K., & Hussain, F.K. (2014). Time series qos forecasting for management of cloud services. In *2014 Ninth International Conference on Broadband and Wireless Computing, Communication and Applications*, pp. 183–190.

15. Khan, M.N.A.H., Liu, Y., Alipour, H., & Singh, S. (2015). Modeling the autoscaling operations in cloud with time series data. In *Proceedings of the 2015 IEEE 34th Symposium on Reliable Distributed Systems Workshop (SRDSW'15)*, IEEE Computer Society, Washington, DC, USA, pp. 7–12.
16. Baldan, F. J., Ramirez-Gallego, S., Bergmeir, C., Herrera, F., & Benitez, J. M. (2016). A forecasting methodology for workload forecasting in cloud systems. *IEEE Transactions on Cloud Computing*, *6*(4), 929–941.
17. Kim, I.K., Wang, W., Qi, Y., & Humphrey, M. (2016). Empirical evaluation of workload forecasting techniques for predictive cloud resource scaling. In *2016 IEEE 9th International Conference on Cloud Computing (CLOUD)*, (pp. 1–10).
18. Kumar, A. S., & Mazumdar, S. (2016). Forecasting hpc workload using arma models and ssa. *International Conference on Information Technology (ICIT)*, *2016*, 294–297.
19. Kumar, J., & Singh, A. K. (2018). Workload prediction in cloud using artificial neural network and adaptive differential evolution. *Future Generation Computer Systems*, *81*, 41–52.
20. Bibal Benifa, J. V., & Dharma, D. (2018). Has: Hybrid auto-scaler for resource scaling in cloud environment. *Journal of Parallel and Distributed Computing*, *120*, 1–15.
21. Kaur, G., Bala, A., & Chana, I. (2019). An intelligent regressive ensemble approach for predicting resource usage in cloud computing. *Journal of Parallel and Distributed Computing*, *123*, 1–12.
22. Mohiuddin, I., & Almogren, A. (2019). Workload aware vm consolidation method in edge/cloud computing for IoT applications. *Journal of Parallel and Distributed Computing*, *123*, 204–214.
23. Moghaddam, S. K., Buyya, R., & Ramamohanarao, K. (2019). Acas: An anomaly-based cause aware auto-scaling framework for clouds. *Journal of Parallel and Distributed Computing*, *126*, 107–120.
24. Kumar, J., Goomer, R., & Singh, A. K. (2018). Long short term memory recurrent neural network (LSTM-RNN) based workload forecasting model for cloud datacenters. *Procedia Computer Science*, *125*, 676–682.
25. Kumar, J., & Singh, A. K. (2016). Dynamic resource scaling in cloud using neural network and black hole algorithm. *Fifth International Conference on Eco-friendly Computing and Communication Systems (ICECCS)*, *2016*, 63–67.
26. Pontius, R. G., Thontteh, O., & Chen, H. (2008). Components of information for multiple resolution comparison between maps that share a real variable. *Environmental and Ecological Statistics*, *15*(2), 111–142.
27. Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate Research*, *30*, 79–82.
28. Willmott, C. J., & Matsuura, K. (2006). On the use of dimensioned measures of error to evaluate the performance of spatial interpolators. *International Journal of Geographical Information Science*, *20*(1), 89–102.
29. Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, *22*(4), 679–688.
30. Tsay, R. S. (2000). Time series and forecasting: Brief history and future research. *Journal of the American Statistical Association*, *95*(450), 638–643.
31. Makridakis, S., Wheelwright, S. C., & Hyndman, R. J. (1998). *Forecasting: Methods and applications* (3rd ed.). New York: Wiley.
32. Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (2008). *Time series analysis*. New York: Wiley.
33. Hyndman, R. J., & Khandakar, Y. (2008). Automatic time series forecasting: the forecast package for R. *Journal of Statistical Software*, *26*(3), 1–22.
34. Traces available in the internet traffic archive, ftp://ita.ee.lbl.gov/html/contrib/. Accessed on 1 May 2019.
35. Reiss, C., Wilkes, J., & Hellerstein, J. L. (2011). Google cluster-usage traces: Format + schema. Google Inc., White Paper, 1–14.
36. Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, *32*(200), 675–701.
37. Conover, W. J., & Conover, W. J. (1980). *Practical nonparametric statistics*. New York: Wiley.
38. Rodríguez-Fdez, I., Canosa, A., Mucientes, M., & Bugarín, A. (2015). STAC: a web platform for the comparison of algorithms using statistical tests. In *Proceedings of the 2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*.
39. Finner, H. (1993). On a monotonicity problem in step-down multiple test procedures. *Journal of the American Statistical Association*, *88*(423), 920–923.

Jitendra Kumar is working as Assistant Professor in the Department of Computer Applications, National Institute of Technology Tiruchirappalli, Tamilnadu, India. He earned his PhD from National Institute of Technology Kurukshetra, India (an institution of national importance). He received M. Sc and M. Tech both in Computer Science from Dayalbagh Educational Institute, Agra, India in 2011 and 2013 respectively. He is recipient of Directors medal for securing the first rank in the university examination of MSc at Dayalbagh Educational Institute, Agra, India. He has experience of more than 2 years in academics and research. He has published papers in several journals and conferences of high repute. His current research interests include cloud computing, machine learning, big data, parallel processing.

Ashutosh Kumar Singh is working as a Professor and Head in the Department of Computer Applications, National Institute of Technology Kurukshetra, India. He has more than 18 years research and teaching experience in various Universities of the India, UK, and Malaysia. He received his Ph.D in Electronics Engineering from Indian Institute of Technology, BHU, India and Post Doc from Department of Computer Science, University of Bristol, UK. He is also Charted Engineer from UK. His research area includes Verification, Synthesis, Design and Testing of Digital Circuits, Data Science, Cloud Computing, Machine Learning, Security, Big Data. He has published more than 160 research papers in different journals, conferences and news magazines. He is the co-author of six books which includes 'Web Spam Detection Application using Neural Network', 'Digital Systems Fundamentals' and 'Computer System Organization & Architecture'. He has worked as an Editorial Board Member of International Journal of Networks and Mobile Technologies, International journal of Digital Content Technology and its Applications. Also he has shared his experience as a Guest Editor for Pertanika Journal of Science and Technology. He is involved in reviewing process of different journals and conferences such as; IEEE transaction of computer, IET, IEEE conference on ITC, ADCOM etc.