Computing (2023) 105:353–374
https://doi.org/10.1007/s00607-022-01129-7

REGULAR PAPER

Time series-based workload prediction using the statistical hybrid model for the cloud environment

K. Lalitha Devi[1] · **S. Valli**[1]

Received: 11 July 2020 / Accepted: 20 October 2022 / Published online: 9 November 2022

Abstract

Resource management is addressed using infrastructure as a service. On demand, the resource management module effectively manages available resources. Resource management in cloud resource provisioning is aided by the prediction of central processing unit (CPU) and memory utilization. Using a hybrid ARIMA–ANN model, this study forecasts future CPU and memory utilization. The range of values discovered is utilized to make predictions, which is useful for resource management. In the cloud traces, the ARIMA model detects linear components in the CPU and memory utilization patterns. For recognizing and magnifying nonlinear components in the traces, the artificial neural network (ANN) leverages the residuals derived from the ARIMA model. The resource utilization patterns are predicted using a combination of linear and nonlinear components. From the predicted and previous history values, the Savitzky–Golay filter finds a range of forecast values. Point value forecasting may not be the best method for predicting multi-step resource utilization in a cloud setting. The forecasting error can be decreased by introducing a range of values, and we employ as reported by Engelbrecht HA and van Greunen M (in: Network and Service Management (CNSM), 2015 11th International Conference, 2015) OER (over estimation rate) and UER (under estimation rate) to cope with the error produced by over or under estimation of CPU and memory utilization. The prediction accuracy is tested using statistical-based analysis using Google's 29-day trail and BitBrain (BB).

Keywords Auto-regression integrated moving average (ARIMA) · Artificial neural network (ANN) · Savitzky–Golay filter · Time series forecasting · CPU · Memory usage · Cloud computing

Mathematics Subject Classification 68

✉ K. Lalitha Devi
lalithavelu10@gmail.com

[1] Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, India

1 Introduction

Cloud computing has risen to prominence as a new form of distributed computing. It delivers physical resources such as processing, storage, and bandwidth capability as an on-demand service for customers, with scalability and reliability to meet the quality of service (QoS) limitations stipulated in the service level agreement, using the virtualization technique (SLA). The infrastructure in terms of CPU capacity, memory, and I/O is required for any application to be deployed in the cloud. The IAAS supplier provides this as a service. Workload prediction in cloud computing forecasts the physical machine's future workload based on previous workload traces. Accurate workload prediction methods would aid the IAAS provider in effectively managing cloud data centre resources through efficient capacity planning. To maintain SLAs, most service providers assign more resources than demand; this over-provisioning of resources reduces physical resource consumption.

The available cloud workload prediction algorithms [1] are based on point value prediction, which is appropriate for predicting short-term CPU load. Although point value prediction works well in a centralized computer system, it fails to anticipate workload in a wide distributed context. So, in a large distributed environment, a prediction model is designed to provide a range of interval values for both short-term (one step) and long-term (multistep ahead) CPU and memory utilization of the future workload. Furthermore, existing grid or cloud workload forecasting methodologies are solely dependent on single-model forecasting. This model is a statistics-based hybrid model for estimating an application's CPU and memory utilization in the cloud.

One or more tasks make up a job. A separate physical machine is assigned to each task. When submitting a job to a provider, cloud customers must specify their resource requirements in terms of CPU, memory, and I/O in order to finish the work. The person who submits the job may be unaware of the resources needed to complete the task. As a result, forecasting the data center's future CPU and memory utilization is difficult. CPU and memory utilization have an impact on work and cloud performance, and it is the most important criteria for a cloud service provider. The workload pattern submitted to the cloud could be constant, trending, seasonal, bursty, or erratic. All of these things must be modeled in the prediction model. As a result, a hybrid ARIMA–ANN model is built to forecast the cloud data center's future CPU and memory utilization. The jobs make considerable use of CPU and memory resources. So, using statistical approaches, the Google traces version 2 [2] task utilization table is evaluated to predict CPU and RAM usage. A model is a mathematical description of a process that generates a time series in time series forecasting. Forecasting utilizes historical data as input and finds a model that fits the data, with forecasts as the model's output.

This hybrid ARIMA model forecasts the linear components, and the ANN forecasts the nonlinear components both from the original data and the residuals obtained from the ARIMA model. The error value may emphasize the nonlinear components in the original data. The outputs from both the models are considered the final predicted CPU and memory usage load values. These ARIMA–ANN forecasted CPU and memory load values are combined with the past history of the CPU and memory load values to create a new time series. This new time series is passed through the Savitzky–Golay filter to produce a range of values for future CPU and memory usage prediction. The

future CPU and memory usage of the cloud workload is predicted to improve the CPU and memory utilization to control cloud QoS accuracy.

In the next section, the related work is reviewed. The basic concepts of time series models, linear ARIMA model and the nonlinear ANN model are presented in Sect. 3. The developed hybrid model is introduced in Sect. 4. Experimental evaluation is discussed in Sects. 5, and 6 contains the concluding remarks.

2 Highlights of the proposed work

1. Propose a hybrid ARIMA and ANN model for resource prediction in cloud datacenter.
2. Main contribution in the proposed model is implementation of combined linear and nonlinear prediction method for the cloud environment.
3. The proposed model used for prediction of load in cloud datacenter.
4. Estimation and comparison of the proposed hybrid model with different standard technique.
5. Extensive experimental evaluation using publicly available google cluster trace and Bitbrain data sets for different datacenters in cloud environment.

3 Related work

In recent years, cloud service providers address the dual problem of providing QoS while competing for available resources. Statistical methods address this problem. Accurate forecasting is needed in sales forecasting, marketing research, financial forecasting, and so on. Time series forecasting is very popular in wind speed prediction [3, 4], electric load prediction [5, 6], and crime forecasting [7]. De Gooijer and Hyndman [8] have presented a review on 25 years of time series forecasting. They have discussed the effectiveness of ARIMA and the ANN models.

Many of the research groups tried to predict the CPU usage of the cloud. The cloud workload varies with time and is correlated over time spans. Thus, the CPU load can be predicted from the history of CPU usage data. Wu et al. [9] developed an auto regression(AR)–based adaptive hybrid prediction scheme (AH Model) for predicting the n-step ahead CPU load in the computational grid and they proposed an adaptive confidence window approach using the Savitzky–Golay filter to achieve good performance in the grid. They integrated mean and historical interval value adaptive parameters for predicting the future CPU load. Mean Square Error (MSE) analysis controls the interval of history used in prediction.

The CPU and memory usage of the cloud workload fluctuates over time; it may have both linear and non-linear components. Hybrid models are better than the individual models for forecasting. Many research groups developed several hybrid ARIMA–ANN models by introducing modifications in the Zhang's model [10]. The resultant models were used in stock market prediction, wind speed prediction and electric load prediction. Different mixtures of hybrid techniques have been proposed to overcome the

difficulties of the single models. Combining models in forecasting exploits the unique strength of each model to find the uncertainties and the nonlinear patterns in the data. Hybrid models are homogeneous as in the case of differently configured ANNs or heterogeneous as in the case of combining both the linear and the nonlinear forecasting models.

Zhang proposed a hybrid ARIMA–ANN model [10] for time series forecasting. This model assumes that the time series data may have both linear and nonlinear components. First, the ARIMA model is fit to predict the linear components and the residuals are given as input to the ANN to predict the nonlinear patterns. Both the outputs are combined to obtain a final forecast. The effectiveness of the model is demonstrated by evaluating three well-known data sets–the Wolf's Sunspot data, the Canadian lynx data, and the British pound/US dollar exchange rate. This model performed better than the individual ARIMA and ANN models in one-step ahead and multi-step ahead prediction.

The authors proposed another hybrid ARIMA–ANN [11] model for time series forecasting. This model also assumes that the time series data are a combination of linear and nonlinear components. In this hybrid model, the ARIMA model is first fit to the time series data to forecast the linear data point. Then the original data, the value forecasted, and the residuals from the ARIMA model are given as inputs to the ANN model. The output from the ANN is the final forecasted value. The ARIMA model magnifies the linear patterns of the input and is taken as the input for the ANN. Babu CN et al. [12] proposed the moving average(MA) filter–based hybrid ARIMA–ANN model for forecasting the time series data. This hybrid method decomposes the given data into linear and nonlinear components. The ARIMA model is fitted directly to the linear data and the ANN model to the nonlinear data. The combined output is the final forecasted value. Kang S et al. [13] proposed a bandwidth prediction scheme for the variable-bit-rate (VBR) video traffic with the regular group of pictures (GOP) pattern. Kalman filter was deployed over the GOP pattern and ARIMA was used for accurate prediction. Liu H et al. [3] constructed a combination of the ARIMA–ANN and ARIMA–Kalman hybrid models for predicting the wind speed and they compared the performance of both the models.

AA EI Desouky and MM EIKateb [6] used five different neural network configurations depending on the architecture and the number of inputs to the ANN with an adaptive learning algorithm to predict the monthly electric load for Jeddah city in Saudi Arabia. The ANN and the hybrid ARIMA–ANN achieved error reduction in forecasting. Valensula et al. [14], automated the approach to obtain the structure of the ARIMA model using a fuzzy logic– based expert system and genetic algorithm (GA) optimizes the weights associated with each fuzzy rule of the expert system. Shukur OB et al. [4] used the hybrid Kalman Filter (KF) –ANN model based on ARIMA and improved the accuracy of the wind speed forecasting.

Tran et al. [15] designed an online time series framework to model and predict the various temporal input/output (I/O) patterns; they used the ARIMA time series model to predict the temporal patterns of I/O requests. ARIMA predictions were used to vary the number of pre-fetched blocks adapting to fluctuating demands. Combining the temporal ARIMA predictions with spatial Markov predictions enabled an experimental parallel file system to achieve performance relative to the standard Linux file systems.

Yan et al. [16] used the hybrid ARIMA–ANN models for forecasting the resource consumption of an Internet Information Server (IIS), a web server that is subjected to software ageing problems. They used the built-in windows counter to obtain the IIS parameters. They focused on available memory and.NET CLR (common language run time) memory in all the heaps.

In the IAAS cloud environment, some of the authors have used the ARIMA model [17] to predict the future application workload behavior to calculate the required VM configuration in the data center. Hu et al. [18] developed a multi-step ahead CPU load prediction method based on the support vector regression and Kalman smoothing technique. Jiang et al. [19] addressed VM types and request timestamps. Two-level ensemble algorithm was used in predicting the multi-type VM demands based on the history. They introduced the cloud prediction cost measure to evaluate the quality of the prediction result. Caron et al. [20] used the Knuth-Morris-Pratt(KMP) string matching algorithm to predict the CPU usage of the cloud client by finding similar usage patterns in the previous usage traces.

Mao and Humphrey [21] first estimated the number of VMs needed for each VM type based on the workload. Subsequently, they determined whether two or more existing VMs can be consolidated. Earliest Deadline First (EDF) algorithm has been used for scheduling the tasks on each VM. Many of the research groups tried to predict the CPU usage of the cloud workload. The cloud workload varies with time and is correlated over time spans. Thus, the CPU load can be predicted from the history of the CPU usage data. Predicting the irregular CPU and memory usage pattern of the cloud workload optimizes the resource allocation and its utilization in the dynamic cloud environment. Khasei et al. [11, 22] used the predicted present value, the past errors and the past actual data values as inputs to the ANN. The ANN output is taken as the final predicted value for one step ahead prediction. Babu et al. [12] analyzed and proved that Khasei model is not suitable for the multi-step ahead prediction. They have suggested an idea to use the past predicted value instead of the past actual value. Based on the suggestion, the past predicted value is used for prediction in our developed work.

Buyuksahin and Ertekin [23] present a new ARIMA- ANN hybrid method that uses the strategies for decomposing the original data and for combining linear and nonlinear models throughout. The hybridization process is a key factor in the forecasting performance of the methods. Abdulhakim et al. [24] 2019 used a hybrid ARIMA-ANN model for Short-term forecasting of water levels in the Changhua River that is important for flood prevention strategies. Hryhorkiv et al. [25] proposed an advanced hybrid forecasting model based on the combination of ARIMA and ANN. ARIMA's statistical properties are used to obtain S&P 500 stock index forecasts. Toga et al. [26] applied time-series-based models such as ARIMA and ANN to analyse the effect of Covid-19 prevalence in Turkey.

The authors [27] utilised the hybrid model to use all the potential of the ARIMA, ANN, and ETS models in the time series forecasting of COVID-19 trends. Utilizing the most recent daily data for COVID-19 globally for the upcoming days, the forecasting performance of various models was evaluated based on the residuals. To address a single model's shortcoming, the authors [28] combine linear and non-linear models such the auto-regressive integrated moving average (ARIMA) and artificial neural network (ANN). The ARIMA-ANN model is referred to in the combined approach.

We use data on Pakistan's gold prices from 1/7/2003 to 1/6/2021 for empirical study. The objective [29] is to develop a comprehensive hybrid model that integrates ARIMA, MLP, ARIMA-MLP, and MLP-ARIMA to analyse all conceivable combinations of linear and nonlinear patterns. To find mixed linear/nonlinear patterns in time series, two models were developed: ARIMA-MLP and MLP-ARIMA. The ideal hybrid model must be able to handle the following four types of patterns: (1) pure linear, (2) pure nonlinear, (3) linear-nonlinear, and (4) nonlinear-linear. Matoussi and Hamrouni [30] proposed a new approach to predict the number of requests arriving at a SaaS service in order to prepare the virtualized resources necessary to respond to user requests. The prediction will be established based on the temporal locality principle and the dynamic assignment of weights to different data points in recent history. Yadav and Yadav [31], presents a predictive analysis of time series forecasting using deep learning method (LSTM) to predict the future load over servers. Based on that analysis it allocates the number of containers for running an application to manage the fluctuating workload. Then this model performs the hybrid vertical and horizontal elasticity. Al-Sayed [32], proposed workload sequence prediction is treated as a translation problem. Attention Seq2Seq-based technique is proposed for predicting cloud resources' workloads. Chen et al. [33] proposed a resource usage prediction method – RPTCN based on a deep learning method - temporal convolutional networks (TCNs) in cloud systems. In order to explore the relationship between the usage of different resources in the temporal dimension, the authors used correlation analysis to screen performance indicators as multidimensional feature input for prediction. The prediction model presented in this study [34] predicts the real resource used for a variety of time intervals, including daily, hourly, and minute usage. The SARIMA model predicts the future resource with accuracy.

In the literature, there are studies that show the success of linear and nonlinear methods over each other, [5, 17] states that statistical and linear models give better results than ANNs. On the other hand, [35] indicates that ANN performs better than linear models when data has high inter-correlations among two or more independent variables. Many researchers in time series forecasting show that hybrid models improve forecasting performances [3, 4, 14, 22]. By taking the advantage of each individual method in a combined model, the error risk of using an inappropriate method is reduced and more accurate results are obtained.

4 Time series forecasting models

Time series is a sequential set of data points observed over successive time periods [36] for a set of data y(t), where t represents the time interval and it varies from 0, 1, 2, and so on. It is an indication of how often the observations are recorded. A time series containing records of a single variable is univariate and if the time series contains records of more than one variable then it is multivariate. Time series can be continuous or discrete. When the time series is continuous, the observations are recorded at every instance of time, and observations are recorded at equally spaced time intervals for discrete time series. The time series have four components. They are trendy, cyclic, seasonal, and irregular and can be removed from the observed data. Considering the

effects of these four components, the multiplicative and the additive models are used for determining the time series data.

$$\text{Multiplicative Model } Y(t) = T(t) * S(t) * C(t) * I(t) \tag{1}$$

$$\text{Additive Model } Y(t) = T(t) + S(t) + C(t) + I(t) \tag{2}$$

Y(t) in Eqs. (1) and (2) is the observation, T(t) is the trend component, S(t) is the seasonal component, C(t) is the cyclic component, and I(t) is the irregular variations at time t. In the multiplicative model, all the four components can affect one another, whereas in the additive model, they are independent of each other.

4.1 The ARIMA model

In the past, ARIMA model was used for forecasting the time series [10]. In an ARIMA (p, d, q) model, the future value of a variable is assumed to be a linear function of the several past observations and random errors. The time series generated takes the form as in Eq. (3).

$$\begin{aligned} y_t = &\theta_0 + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} \ldots + \varphi_p y_{t-p} \\ &+ \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} \ldots + \theta_q \varepsilon_{t-q} \end{aligned} \tag{3}$$

y_t and ε_t in Eq. (3) are the actual value and the random error at time period t respectively, $\varphi_i (i = 1, 2 \ldots . p)$ and $\theta_i (i = 1, 2 \ldots . q)$ are the model parameters. In the auto regression AR(p) model, the current value of the time series is expressed as a linear aggregation of the p previous values and the error term and the moving average MA(q) model is expressed as the current value of time series as an error at time t and q previous error terms.

4.2 The ANN model

The ANN structure is very similar to the neuron structure of the human brain. ANNs approximate the various nonlinearities in the data [11]. To model the time series data, the relationship between the output (y_t) and the inputs (y_{t-1}, $y_{t-2} \ldots y_{t-p}$) is expressed using Eq. (4).

$$y_t = W_0 + \sum_{j=1}^{q} w_j . g\left(w_{0j} + \sum_{i=1}^{p} w_{ij} \cdot y_{t-i} \right) + \varepsilon_t \tag{4}$$

w_{ij} (i = 0,1,2,…,p,j = 1,2,…,q) and w_j(j = 1,2,..,q) in Eq. (4) are the model parameters and called the connection weights; p is the number of input nodes and q is the number of hidden nodes; g denotes the transfer function of the hidden layer. The logistic

function is the hidden layer transfer function given by Eq. (5).

$$g(u) = \frac{1}{1 + \exp(-u)} \tag{5}$$

This logistic function is used for modeling time data because they are nonlinear and continuously differentiable which is a desirable property for network learning. Hence, the ANN model performs a nonlinear functional mapping between the past observations and t

by Eq. (6); f in Eq. (6) is a non-linear function, ε_t is the error term and w is a weight vector of all the parameters.

$$y_t = f\left(y_{t-1}, y_{t-2}, \ldots, y_{t-p}, w\right) + \varepsilon_t \tag{6}$$

The ANN model is equivalent to a nonlinear autoregressive model. This models' coefficient is obtained by giving the observed data sequence as input to the ANN, and the ANN is trained with this sample sequence. After training, the performance is tested and validated.

5 The system architecture

The hybrid ARIMA and ANN model's system architecture for predicting the range of confidence values for CPU and memory utilization is depicted in Fig. 1. The analysis relies on the Google cluster data collection [2]. The null values are removed during data preprocessing, and the data is turned into a time series as a result. The time series is then sent into the ARIMA model, which predicts CPU and memory utilization for the next several requests based on the data. To determine whether the original CPU and memory consumption time series is stationary, ARIMA employs the Dickey Fuller test. The nonlinear components that the ARIMA model cannot model are referred to as

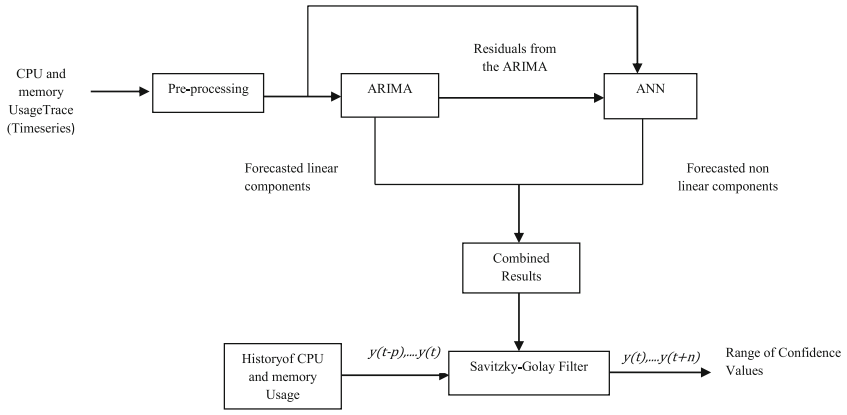

Fig. 1 The hybrid ARIMA–ANN model

residues. The residuals are extracted and combined with the original data sequence as input to the ANN model. It generates a collection of CPU and memory use predictions. To get the final prediction values, the predicted values from both models are added together. The new time series is then created by combining these values with the previous history. Recent past data, in particular, may provide more useful information than data from the distant past [37]. To eliminate any inaccuracies, this new sequence of data points is fed into the Savitzky–Golay filter [9]. The range of confidence values for the n-step ahead CPU and memory use estimate is derived using the smoothed data.

5.1 The hybrid ARIMA–ANN model

The hybrid model integrates ARIMA and ANN within an estimated range of confidence values to yield accurate results in prediction. It forecasts the workload n-step ahead from the current time. The time series gen as input to the hybrid model in Eq. (7) is a function of linear and nonlinear components.

$$y_t = f(L_t, N_t) \tag{7}$$

L_t in Eq. (7) represents the linear components and N_t denotes the nonlinear components. These two components are estimated from the data. ARIMA model finds the linear component. The residuals from the ARIMA model are the nonlinear components of the original data.

$$L_{t(ARIMA)} = \widehat{L_t} + e_t \tag{8}$$

$\widehat{L_t}$ in Eq. (8) is the forecast of CPU and memory load value at time t and e_t is the residual at time t from the linear model [10]. Residuals may still contain the linear correlation structure. Residual analysis may not find the nonlinear correlation structure and no standard statistical technique is available to find the nonlinear correlation structure in the data [11]. ARIMA may not be adequate to handle the nonlinear components. The ANN in Fig. 1 performs the nonlinear modeling [11]. Multilayer perceptron is used in modeling the nonlinear components given by Eqs. (9)–(11). Let f^1, f^2, f denote the nonlinear functions obtained from the neural network which represents the nonlinear relationship existing among the residuals of linear modeling and the original data:

$$N_t^1 = f^1(e_{t-1}, \ldots, e_{t-n}) \tag{9}$$

$$N_t^2 = f^2(z_{t-1}, \ldots, z_{t-m}) \tag{10}$$

$$\hat{N}_{t(ANN)} = f\left(N_t^1, N_t^2\right) \tag{11}$$

n and m specify the order of the model. The combined forecast $\widehat{y_t}$ is given by Eq. (12).

$$\hat{y}_t = \widehat{L_t} + \widehat{f\left(N_t^1, N_t^2\right)} = L_t + f\left(e_{t-1}, \ldots e_{t-N_1}, z_{t-1}, \ldots z_{t-M_1}\right) \tag{12}$$

n1 and m1 in Eq. (12) are determined using the neural network; e_t that is, $\{e_i(i = t-1, \ldots .t - N_1)\}$ the residual component and $\{z_j(j = t-1, \ldots .t - m1)\}$ the portion of original data in Eq. (12) denoted as N_t^1 and N_t^2 are used in modeling the nonlinear component. If the data contains only pure linear structure, e_t may be missing in the input to the ANN. Function f is a nonlinear function determined by ANN and it is interpreted as the estimated value of N_t. So the combined forecast is given by Eq. (13)

$$\hat{y}_t = \left(\hat{L}_t, \hat{N}_{t(ANN)}\right) \tag{13}$$

$\widehat{L}$ and $\hat{N}$ in Eq. (13) is the output of the ARIMA model and ANN model respectively.

1 Algorithm for prediction using the hybrid ARIMA– ANN model

Input: Workload traces time span m

Prediction look ahead span n

Output: The range of values for n-step look ahead prediction.

1. Transform the input into time series

2. **for** i =1 to m

Find $\hat{L}_t$ using Equation (8)

end for.

3. **for** i=1 to m

Find $\widehat{N}_{t(ANN)}$ using Eq. (11)

end for.

4. **Find** $\hat{y}_t = (\hat{L}_t, \widehat{N}_{t\,(ANN)})$ using Eq. (12)

5. **for** i=1 to m

$A(i) = y_{(t-p),}y_{(t-p+1),} \ldots \ldots \ldots y_{(t)}\ ,\hat{Y}_{(t+1),}\hat{Y}_{(t+2),} \ldots \ldots \ldots \hat{Y}_{(n)}$

Compute $y_l(t+n) = A_{ls}(n)$

$y_u(t+n) = A_{us}(n)$

end for.

Although Zhang's hybrid ARIMA–ANN model takes advantage of ARIMA and the ANN model's unique strengths in determining linear and nonlinear patterns, there are no assumptions in modeling the linear and nonlinear patterns of data. As a result, the ANN is trained using the nonlinear components, which are the ARIMA model's error values. This ANN model captures the nonlinear components as well as the linear components ignored by ARIMA. As a result, this constructed model better reflects the linear and nonlinear correlation structures in the data than separate models, and the

integration of the confidence range values for prediction improves model accuracy. Algorithm 1 describes a hybrid model for predicting CPU and memory utilization n steps ahead.

6 Results and analysis

We have conducted our analysis and experiments on workload traces for a compute cluster of Google trace and BitBrain. This trace contains the task demand and usage for CPU, memory and disk. The usage of each type of resource is reported at every five minutes' interval. We mainly focused on CPU and memory, and our approach can be extended to consider resources such as disk space.

6.1 Experimental setup

Google released the trace version 2 on November 2011, which represents 29 days of the computing cell information on single cluster. Google provides schema to describe the data set [2]. The data set contains different tables [38] as in Table 1.

The data set contains workload traces and its resource demands and actual resource usage records for a job and task over a time period of 29 days. The workload traces of Google compute cluster consist of all the different task types [38, 39]. This Google trace provides information of each task's demand and usage of CPU, memory, and I/O disk time. Every record in the dataset has a timestamp attribute which is in microseconds and is recorded as a 64bit integer. Every job and the machine are represented by a unique 64bit identifier.

A time series involving CPU and memory usage data points is constructed from the Google cloud trace in this work. Sample time series data are shown in Fig. 2. Time series data points are divided into the training period and the testing period. The training data points fit the models and the testing data points validate the models.

For ARIMA model forecasting, a suitable model order was found using the R software package. It predicts the future linear model. MATLAB's *ntstool* is used in modeling the nonlinear time series data. The ANN and the hybrid model are implemented using MATLAB; a total of 100 data points are used in n-step prediction. The

Table 1 Database schema

S.no	Table name
1	Machine event
2	Machine attribute
3	Job event
4	Task event
5	Task constraints
6	Task usage

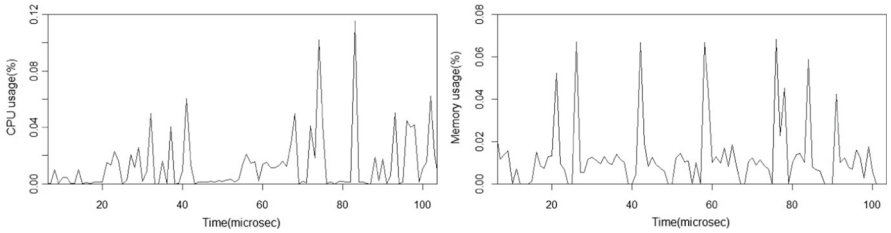

Fig. 2 Sample Google cloud trace data

forecast horizon is 10. The multi-step ahead prediction is performed using the different look ahead spans.

In order to forecast the usage of CPU and memory, we split the data set into training data set and testing data set. Training data are transformed to a time series process. The values of p, d and q of the ARIMA models are obtained using the R software package [40]. The inputs to the ANN are CPU and memory usage time series and their respective error values [35]. The number of input nodes and hidden nodes correspond to the lagged observations used in finding the underlying pattern in a time series. The optimum ANN network configuration is implemented using the Levenberg–Marquardt algorithm. Different network configurations are evaluated. The network configuration with best forecasting accuracy is selected as the final model to fit the data. At run time, the model is constantly updated. Whenever new requests arrive, their CPU and memory usage values are incorporated into the time series and the fitting process is repeated, which may lead to changes in the network configuration of the ANN and the values of p, d and q of the ARIMA model. The predicted CPU and memory usage data points along with the past history data points are used in creating the time series. The Savitzky–Golay filter smoothens this series to find a range of predicted workload. The frame size f is set as 51 and the polynomial degree d is taken as 4 [9].

6.2 Evaluation metrics

The prediction capability of the developed hybrid model is compared with the AR (30) [41] using Google cloud trace version 2 data set. The RMSE value measures the difference between the values predicted by the model and the actual values observed. RMSE is defined as the square root of the mean square error. In point value prediction, the error is calculated as the difference between the actual workload and the predicted point value. For window based prediction [9], Y(t) is the true load value at time t, and error is zero if it is inside the confidence window. Otherwise, the error is calculated using Eq. (14).

$$Error(t) = \begin{cases} 0 \; if \; Y_l(t) \leq Y(t) \leq Y_u(t) \\ \min\{|Y(t) - Y_l(t)|, |Y(t) - Y_u(t)|\} \end{cases} \tag{14}$$

Equations (15) and (16) are used to calculate the MSE (Mean Square Error) and RMSE (Relative Mean Square Error). RMSE, according to Engelbrecht et al. [41], solely indicates the error distance between the forecasted and true values. As a result, they developed ES, a new metric for determining if the error distance is overestimated by 10% more than the genuine value or underestimated by 10% less. Equation (17) denotes the overestimation rate, which is given by Eq. (18), and UER denotes the underestimation rate, which is given by Eq. (19).

$$MSE = \frac{1}{t+1} \sum_{1}^{t} [Error(i)]^2 \tag{15}$$

$$RMSE = \sqrt{MSE} \tag{16}$$

$$ES = \frac{1}{2}(OER) + \frac{1}{2}(UER) \tag{17}$$

$$OER = \frac{Number \; of \; over \; predicted \; values}{Total \; forecasted \; values} \tag{18}$$

$$UER = \frac{Number \; of \; under \; predicted \; values}{Total \; forecasted \; values} \tag{19}$$

To measure the accuracy of the work, metrics such as Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) is also 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 y_i^p and the actual value y_i^a and is calculated using Eq. (20). MAPE is used in forecasting accuracy of a prediction model and is calculated using Eq. (21). Lesser MAPE value indicates better prediction accuracy in terms of percentage. In Eq. (21) y_i^p is the predicted value, y_i^a is the actual value and N is the number of the predicted values in the dataset.

$$MAE = \frac{1}{N} \sum_{i=1}^{N} \left(y_i^p - y_i^a \right) \tag{20}$$

$$MAPE = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{y_i^p - y_i^a}{y_i^a} \right) * 100 \tag{21}$$

RMSE tells the how far the distance between actual and predicted observations of CPU and memory usage. It suggests the user that whether we can go with the particular prediction model for forecasting the given attribute values. MAPE measures the prediction accuracy forecasting model. So that MAPE metrics are used to find the forecasting accuracy of the hybrid ARIMA-ANN model.

6.3 Results and discussion

The use of the CPU and RAM is examined. When the data is highly fluctuating, AR and ARIMA may be unable to model the nonlinear components. The subsequent error will have a significant impact. As a result, AR and ARIMA may not be appropriate for cloud multi-step ahead load forecasting. Nonlinear patterns are better predicted by ANN. Prediction, however, necessitates the use of training data. The performance of ANN may be harmed as a result of this constraint. As a result, the proposed hybrid model combines the best features of these two models to better capture both linear and nonlinear components in the data than the other models.

6.3.1 Forecasting the CPU usage

Figure 3a and b depicts the actual observations as well as the expected CPU utilisation values for different input data points. The best-fitting ARIMA model for the sample CPU consumption is for the linear data (5, 0, 0). Five input nodes, five hidden nodes, and one output neuron make up the fitted ANN for nonlinear data. N is how it is written (5-5-1). The CPU use series' future behaviour is predicted by the trained hybrid model.

The RMSE of CPU consumption for various look-ahead spans is shown in Fig. 4a. When compared to the hybrid model, ARIMA produces a higher RMSE value. For the second look ahead span, the ANN model generates the best RMSE values, implying that ANN has strong predicting accuracy for short horizons. The performance of ANN is not particularly outstanding when the forecast look-ahead span is increased to three, four, or five. Figure 4b shows the MAPE values of the CPU usage.

6.3.2 Forecasting the memory usage

Figure 5a and b depicts the actual observations as well as the projected memory utilisation numbers for different data points. Memory consumption forecasting uses the same hybrid model configurations as are used to predict CPU utilization. The new approach is compared to previous research based on ARIMA and ANN, as well as linear and nonlinear regression methods. The CPU and memory use patterns are used to evaluate the models. The estimation score (ES), the root mean square error (RMSE), and the proper estimation rate are all compared. The accurate value is one that is anticipated to be within a percent of the true value. The RMSE of the suggested hybrid model is lower than those of ARIMA and ANN.

The RMSE of memory utilization is shown in Fig. 6a. When compared to the other two models, ARIMA has the lowest RMSE value during the third look-ahead span, indicating that the testing data is linear, and it has higher RMSE values for the other look-ahead spans. For the duration of second look-ahead span, the ANN and ARIMA models have the best RMSE values. The performance of ANN degrades as the look-ahead span is expanded to three, four, and five. As a result, individual models may not detect all of the data's patterns. As a result, integrating the two models may be the

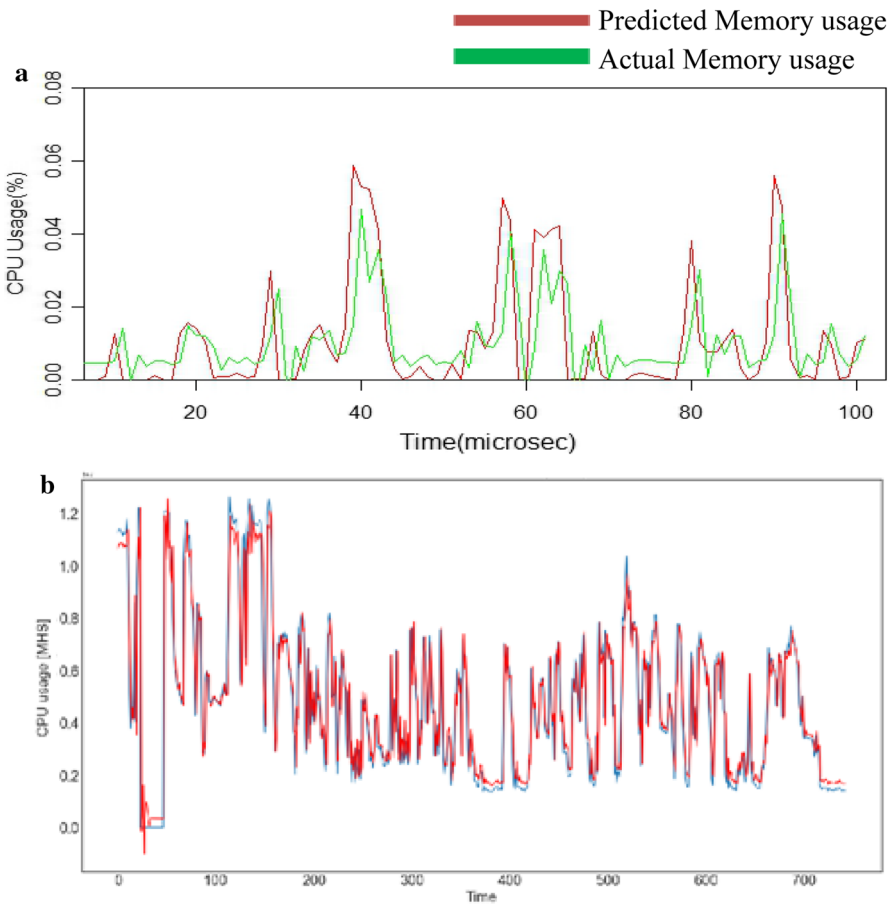

Fig. 3 **a**, **b** The hybrid model prediction of CPU usage

most effective strategy to address this performance degradation. Figure 6a show that the created hybrid model outperforms the ARIMA and ANN models in predicting CPU and memory utilization over various look-ahead spans. In terms of CPU and memory consumption forecasting, the hybrid model outperforms the other two models in both the short- and long-term look-ahead periods.5.6 Relationship between RMSE and the Look-Ahead Span. Figure 6b shows the MAPE values of memory usage data.

Iteratively running one step ahead prediction yields a multi-step ahead prediction. The assumption that the projected data are the real data for predicting the next step is based on time series data related to n-step ahead prediction. This assumption, however, will not always be true because of the accumulation of prediction errors. As a result, for improvements over longer time horizons, we used the range of expected values. The projected RMSE and predicted steps are shown in Fig. 7. It means RMSE is proportional to prediction steps in a linear way.

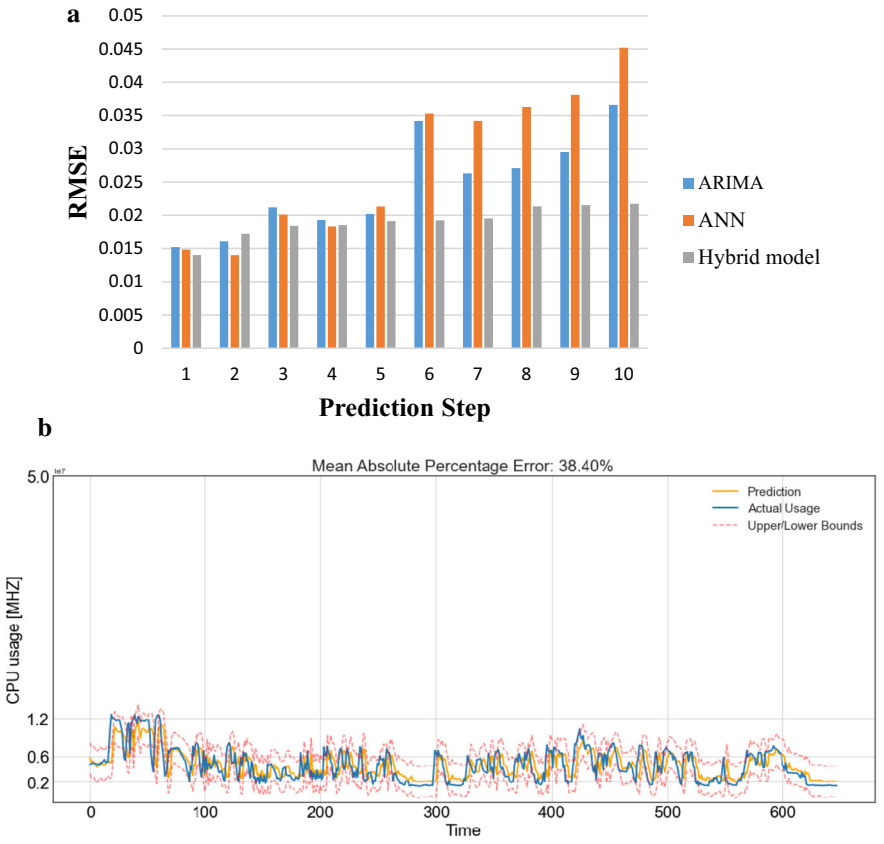

Fig. 4 **a** RMSE of CPU usage. **b** MAPE of CPU usage

6.4 Comparison with the other models

Figures 8 and 9 show how the hybrid ARIMA–ANN model compares to the other models in terms of ES and correct prediction values. The generated model's performance gains are compared to those of other models. The linear ARIMA model and nonlinear ANN model combination found different patterns in the CPU and memory utilization time series data in our designed hybrid model. In addition, when the prediction horizon is extended, the Savitzky–Golay filter filters the errors in the time series data to determine the range of forecasting values.

In Support vector regression model, SVR the training phase is the costliest part, and lots of research are going on to develop better way to do it. Selection of kernel is important because if we choose a kernel like the Gaussian, which starts giving zero as the distance between the parameter grows then as we start to move away from the training data, the machine will start returning the mean value of the training set. The selection of the kernel determines the estimator's asymptotic performance. However,

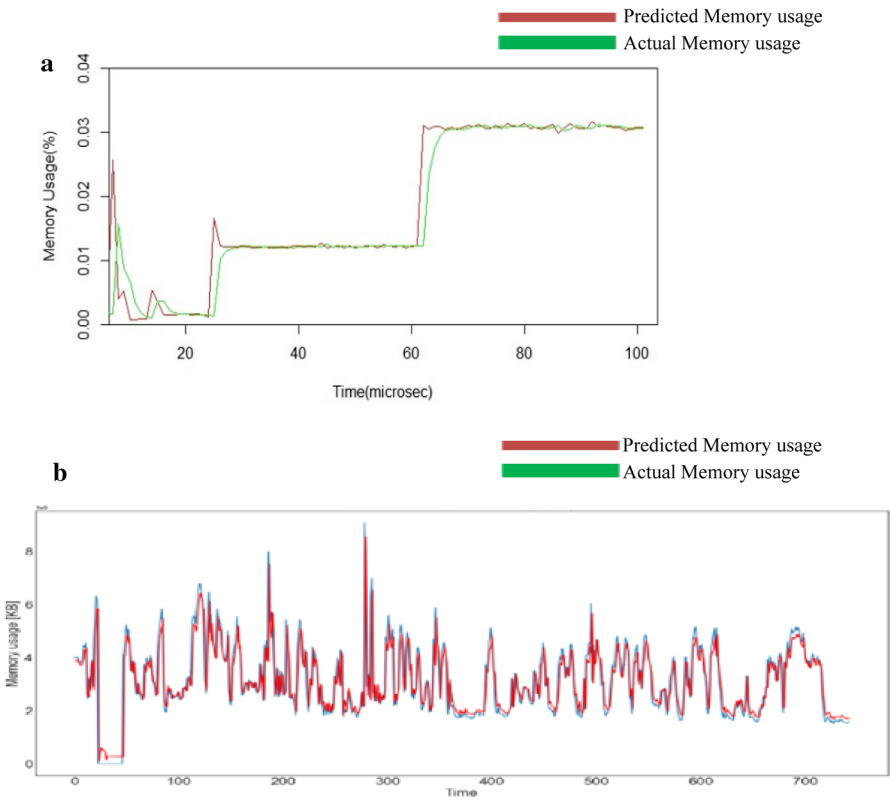

Fig. 5 **a**, **b** The hybrid model prediction of memory usage

in the hybrid prediction model, the prediction modeling is simply fitted and accurate prediction modeling is obtained.

The Bitbrains fast storage trace [42] contains data from 1250 virtual machines (VMs) connected to fast Storage Area Network devices. The dataset is available for 30 days, with a 5 min sampling rate and one file per virtual machine. The provisioned CPU capacity, CPU utilization, provisioned memory capacity, actual memory usage, network I/O throughput, disc I/O throughput, and the number of cores provisioned are all contained in each file.

The proposed method was tested using multi-attribute resource workload data obtained from the BB trace. MSE, MAE, and MAPE metrics are used to assess the accuracy of the prediction. The accuracy of the SVRT and LRT algorithms on the test workload data is shown in Figs. 10 and 11. In both one-step and four-steps forward forecasts, the hybrid ARIMA-ANN model has a lower error ratio than the SVRT and LRT methods, as shown in Figs. 10 and 11.

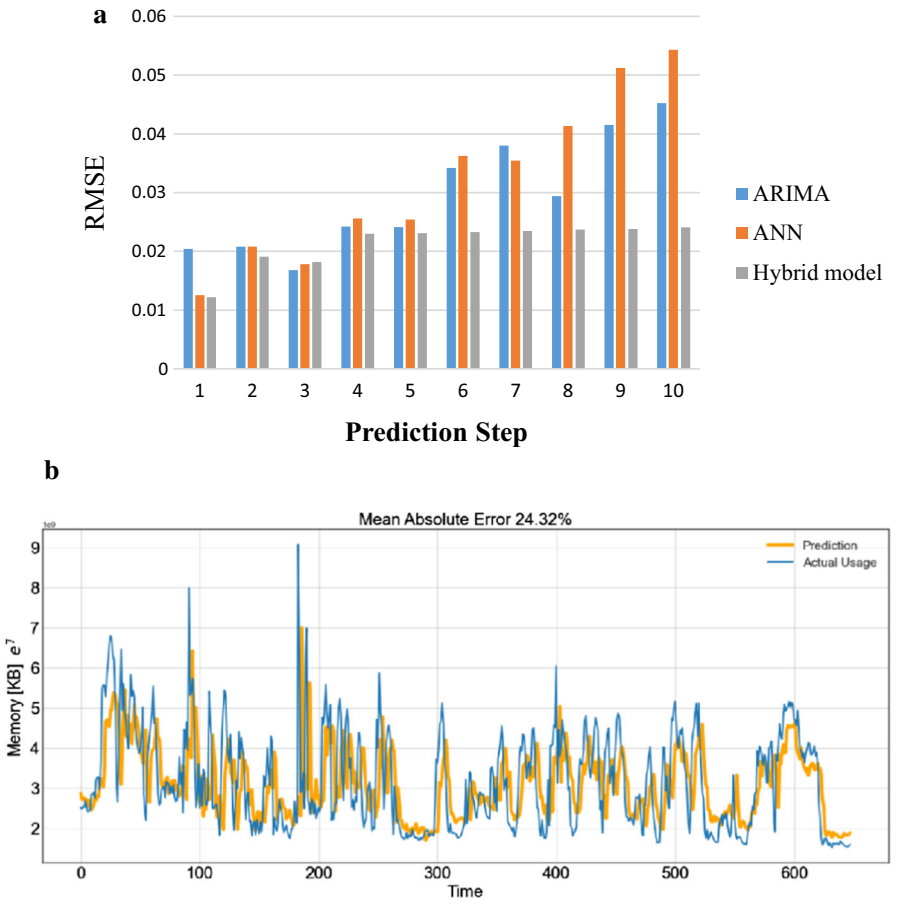

Fig. 6 **a** RMSE of memory usage. **b** MAPE of memory usage

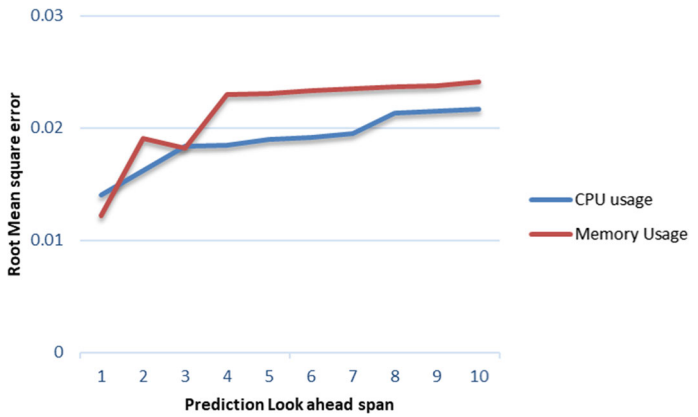

Fig. 7 Relationship between RMSE and predicted steps

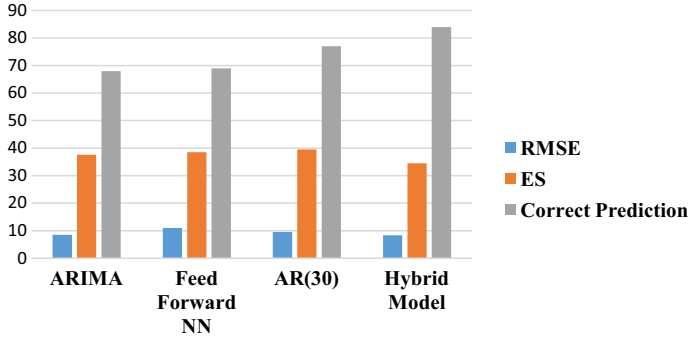

Fig. 8 Comparison of CPU usage

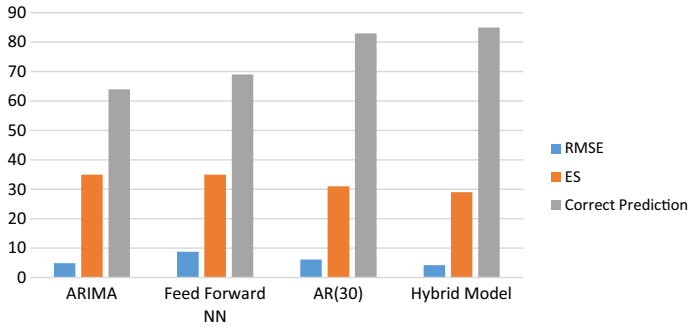

Fig. 9 Comparison of memory usage

7 Conclusion

Accurate forecasting of future workload is necessary to achieve improved QoS and increase resource usage in the cloud environment. Based on the hybrid ARIMA–ANN technique, this research develops a multi-step forward CPU and memory load prediction model. This concept has been found to work well in a dynamic cloud computing environment through testing. It combines ARIMA and ANN technologies with the Savitzky–Golay filter. The model's forecasting accuracy is verified using publicly available Google trace data and BitBrain data. The hybrid ARIMA–ANN exhibits better RMSE, ES, and correct prediction values when compared to AR (30), Feed Forward NN, MSE, MAE, MAPE, and conventional ARIMA. The service provider will utilize the prediction results to forecast demand and make the necessary preparations. If the burden completely shifts to a different pattern, dynamic cloud workloads may have substantial forecast errors. In that circumstance, a new predictive model must be built in order to achieve high forecast accuracy. To adapt to such situations, a dynamic workload must be capable of recognizing previously unnoticed new workload patterns. It can also adaptively retrain its model to deal with such drastic pattern changes. Cloud providers can use this predictive analysis to avoid different types of losses such

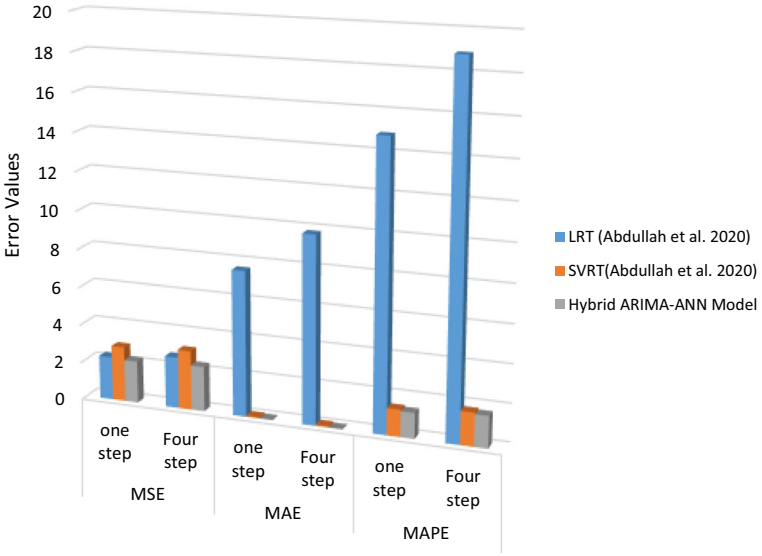

Fig. 10 Comparison of CPU utilization – BB trace

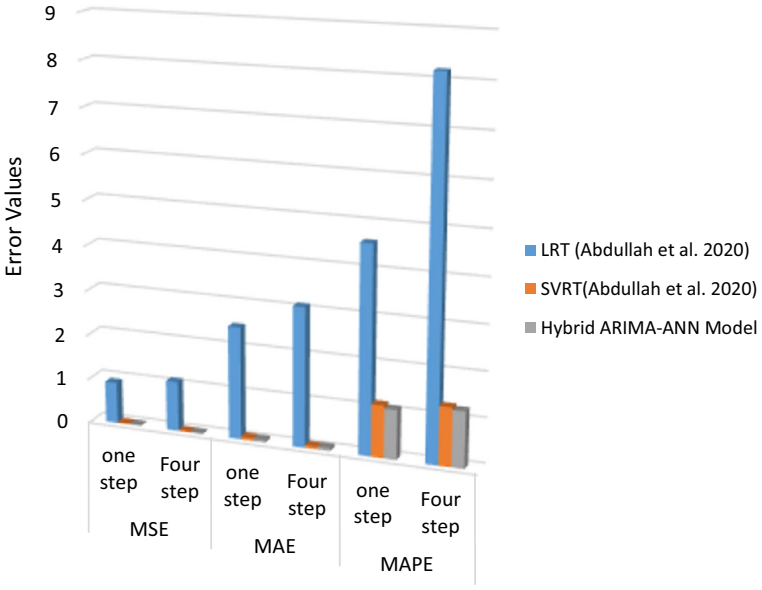

Fig. 11 Comparison of memory utilization – BB trace

as services unavailability, maximum energy consumption and customer's loss and it prevents excessive or insufficient provisioning of cloud resources. In future dynamizing the size of the sliding window associated to the recent history to be analyzed and dynamic assignment of weights to different data points in recent history to improve the prediction accuracy [43, 44].

References

1. Shyam GK, Manvi SS (2016) Virtual resource prediction in cloud environment: a Bayesian approach. J Netw Comput Appl 65:144–154
2. https://console.cloud.google.com/storage/browser/clusterdata-2011-2.
3. Liu H, Tian HQ, Li YF (2012) Comparison of two new ARIMA-ANN and ARIMA-Kalman hybrid methods for wind speed prediction. Appl Energy 98:415–424
4. Shukur OB, Lee MH (2015) Daily wind speed forecasting through hybrid KF-ANN model based on ARIMA. Renew Energy 76:637–647
5. Contreras J, Espinola R, Nogales FJ, Conejo AJ (2003) ARIMA models to predict next- day electricity prices. IEEE Trans Power Syst 18(3):1014–1020
6. El Desouky AA, Elkateb MM (2000) Hybrid adaptive techniques for electric-load forecast using ANN and ARIMA. In: IEE Proceedings-Generation, Transmission and Distribution 147(4): 213–217
7. Noor NMM, Retnowardhani A, Abd ML, Saman MYM (2013) Crime Forecasting using ARIMA Model and Fuzzy Alpha-cut. J Appl Sci 13(1):167–172
8. Gooijer De, Jan G, Rob JH (2006) 25 years of time series forecasting. Int J Forecast 22(3):443–473
9. Wu Y, Hwang K, Yuan Y, Zheng W (2010) Adaptive workload prediction of grid performance in confidence windows. IEEE Trans Parallel Distrib Syst 21(7):925–938
10. Zhang GP (2003) Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing 50:159–175
11. Mehdi K, Bijari M (2011) A novel hybridization of artificial neural networks and ARIMA models for time series forecasting. Appl Soft Comput 11(2):2664–2675
12. Babu CN, Reddy BE (2014) A moving-average filter based hybrid ARIMA–ANN model for forecasting time series data. Appl Soft Comput 23:27–38
13. Kang S, Lee S, Won Y, Seong B (2010) On-line prediction of nonstationary variable-bit-rate video traffic. IEEE Trans Signal Process 58(3):1219–1237
14. Valenzuela O, Rojas I, Rojas F, Pomares H, Herrera LJ, Guillén A, Marquez L, Pasadas M (2008) Hybridization of intelligent techniques and ARIMA models for time series prediction. Fuzzy Sets Syst 159(7):821–845
15. Tran N, Reed DA (2004) Automatic ARIMA time series modeling for adaptive I/O prefetching. IEEE Trans Parallel Distrib Syst 15(4):362–377
16. Yan Y, Guo P, Liu L (2014) A novel hybridization of artificial neural networks and ARIMA models for forecasting resource consumption in an IIS web server. In: Software Reliability Engineering Workshops pp 437–442
17. Calheiros RN, Masoumi E, Ranjan R, Buyya R (2015) Workload prediction using ARIMA model and its impact on cloud applications' QoS. IEEE Trans Cloud Comput 3(4):449–458
18. Hu R, Jiang J, Liu G, Wang L (2013) CPU load prediction using support vector regression and Kalman smoother for cloud. In: IEEE 33rd International Conference on Distributed Computing Systems Workshops, pp 88–92
19. Jiang Y, Perng CS, Li T, Chang R (2011) Asap: a self-adaptive prediction system for instant cloud resource demand provisioning. In: IEEE 11th International Conference on Data Mining, pp 1104–1109
20. Caron E, Desprez F, Muresan A (2010) Forecasting for grid and cloud computing on-demand resources based on pattern matching. In: 2010 IEEE Second International Conference on CloudCom, pp. 456–463
21. Mao M, Humphrey M (2011) Auto-scaling to minimize cost and meet application deadlines in cloud workflows. In: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, pp 1–12
22. Khashei M, Bijari M (2010) An artificial neural network (p, d, q) model for timeseries forecasting. Expert Syst Appl 37(1):479–489

23. Buyuksahin UC, Ertekin S (2019) Improving forecasting accuracy of time series data using a new ARIMA-ANN hybrid method and empirical mode decomposition. Neurocomputing 361:151–163
24. Abdulhakim F, Jun F (2019) Prediction of flow flooding in Changhua river based on time series models. In: IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference, 1:1663-1669
25. Hryhorkiv V, Buiak L, Verstiak A, Hryhorkiv, M, Verstiak O, Tokarieva K (2020) Forecasting financial time series using combined ARIMA-ANN algorithm. In: IEEE 10th International Conference on Advanced Computer Information Technologies, pp. 455–458
26. Toga G, Atalay B, Toksari MD (2021) COVID-19 prevalence forecasting using autoregressive integrated moving average (ARIMA) and artificial neural networks (ANN): case of Turkey. J Infect Public Health
27. Safi SK, Sanusi OI (2021) A hybrid of artificial neural network, exponential smoothing, and ARIMA models for COVID-19 time series forecasting. Model Assist Stat Appl 16(1):25–35
28. Khan F, Urooj A, Muhammadullah S (2021) An ARIMA-ANN hybrid model for monthly gold price forecasting: empirical evidence from Pakistan. Pakistan Econ Rev 4(1):pp 61–75
29. Hajirahimi Z, Khashei M (2022) A novel parallel hybrid model based on series hybrid models of ARIMA and ANN models. Neural Processing Letters, Springer, pp 1–19
30. Matoussi W, Hamrouni T (2022) A new temporal locality-based workload prediction approach for SaaS services in a cloud environment. J King Saud Univ Comput Inf Sci 34(7):3973–3987
31. Yadav MP, Yadav DK (2021) Workload prediction for cloud resource provisioning using time series data. Soft computing for problem solving. Springer, Singapore, pp 447–459
32. Al-Sayed MM (2022) Workload time series cumulative prediction mechanism for cloud resources using neural machine translation technique. J Grid Comput 20(2):1–29
33. Chen W, Lu C, Ye K, Wang Y, Xu CZ (2021) RPTCN: Resource Prediction for High-dynamic Workloads in Clouds based on Deep Learning. In: IEEE International Conference on Cluster Computing, pp 59–69
34. Anupama KC, Shivakumar BR, Nagaraja R (2021) Resource utilization prediction in cloud computing using hybrid model. Int J Adv Comput Sci Appl 12:4
35. Zhang G, Patuwo BE, Hu MY (1998) Forecasting with artificial neural networks– the state of the art. Int J Forecast 14(1):35–62
36. Adhikari R, Agrawal RK (2013) An introductory study on time series modeling and forecasting. *arXiv preprint* arXiv:1302.6613.
37. Hu R, Jiang J, Liu G, Wang L (2014) Efficient resources provisioning based on load forecasting in cloud. Sci World J
38. Rasheduzzaman M, Islam MA, Islam T, Hossain T, Rahman RM (2014) Task shape classification and workload characterization of Google cluster trace. In: Advance Computing Conference (IACC), pp 893–898
39. Moreno IS, Garraghan P, Townend P, Xu J (2014) Analysis, modeling and simulation of workload patterns in a large-scale utility cloud. IEEE Trans Cloud Comput 2(2):208–221
40. http://robjhyndman.com/hyndsight/forecast4
41. Engelbrecht HA, van Greunen M (2015) Forecasting methods for cloud hosted resources, a comparison. In: Network and Service Management (CNSM), 11th International Conference on, pp 29–35
42. http://gwa.ewi.tudelft.nl/datasets/gwa-t-12-bitbrains.
43. Abdullah L, Li H, Al-Jamali S, Al-Badwi A, Ruan C (2020) Predicting multi-attribute host resource utilization using support vector regression technique. IEEE Access 8:66048–66067
44. Alrweili H, Fawzy H (2022) Forecasting crude oil prices using an ARIMA-ANN hybrid model. J Stat Appl Probab 11(3):845–855. https://doi.org/10.18576/jsap/110308