Analysing Cloud QoS Prediction Approaches and Its Control Parameters: Considering Overall Accuracy and Freshness of a Dataset

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 31, 2019, accepted June 14, 2019, date of publication June 19, 2019, date of current version July 9, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2923706

Analysing Cloud QoS Prediction Approaches and Its Control Parameters: Considering Overall Accuracy and Freshness of a Dataset

WALAYAT HUSSAIN[1] AND OSAMA SOHAIB[1,2]
[1]Faculty of Engineering and IT, University of Technology Sydney, Sydney, NSW 2007, Australia
[2]School of Information, Systems and Modelling, University of Technology Sydney, Sydney, NSW 2007, Australia

Corresponding author: Osama Sohaib (osama.sohaib@uts.edu.au)

This work was supported by the School of Information, Systems and Modelling, FEIT–UTS.

ABSTRACT service level agreement (SLA) management is one of the key issues in cloud computing. The primary goal of a service provider is to minimize the risk of service violations, as these results in penalties in terms of both money and a decrease in trustworthiness. To avoid SLA violations, the service provider needs to predict the likelihood of violation for each SLO and its measurable characteristics (QoS parameters) and take immediate action to avoid violations occurring. There are several approaches discussed in the literature to predict service violation; however, none of these explores how a change in control parameters and the freshness of data impact prediction accuracy and result in the effective management of an SLA of the cloud service provider. The contribution of this paper is two-fold. First, we analyzed the accuracy of six widely used prediction algorithms—simple exponential smoothing, simple moving average, weighted moving average, Holt–Winter double exponential smoothing, extrapolation, and the autoregressive integrated moving average—by varying their individual control parameters. Each of the approaches is compared to 10 different datasets at different time intervals between 5 min and 4 weeks. Second, we analyzed the prediction accuracy of the simple exponential smoothing method by considering the freshness of a data; i.e., how the accuracy varies in the initial time period of prediction compared to later ones. To achieve this, we divided the cloud QoS dataset into sets of input values that range from 100 to 500 intervals in sets of 1–100, 1–200, 1–300, 1–400, and 1–500. From the analysis, we observed that different prediction methods behave differently based on the control parameter and the nature of the dataset. The analysis helps service providers choose a suitable prediction method with optimal control parameters so that they can obtain accurate prediction results to manage SLA intelligently and avoid violation penalties.

INDEX TERMS Cloud computing, QoS prediction, SLA violation, prediction accuracy, data acquisition, protocols, mining.

I. INTRODUCTION

Cloud computing is increasingly recognized and popular among business communities due to its elastic architecture and economical, easily accessible, scalable and flexible nature. In a press release from April 2019, Gartner [1] claimed that the cloud market would grow exponentially by 2022 and predicted that worldwide cloud market would increase by 17.5% from $182.4 billion in 2018 to $214.3 billion by the end of 2019. Among different cloud service markets, the leading service market is for Infrastructure as a Service (IaaS). IaaS is predicted to grow its revenue by 27.5% to $38.9 billion in 2019, an increase of 8.4% from 2018. When businesses and consumers use cloud services, they benefit by increasing their capacity in several ways. Cloud computing provides the architecture through which a consumer can store, retrieve and process data as well as execute their application anytime and anywhere, regardless of the physical location of the servers. A cloud can be considered as a huge pool of virtualized resources, such as a platform, infrastructure and services that can be easily accessed and used by consumers under the agreed standards of service delivery. Often called service level objectives or SLOs, these are the performance metrics of service level agreements (SLAs).

An SLA is an important legal contract between the cloud service provider and consumer that outlines the obligations, commitment and penalties of each party. For example, in

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaofan He.

their SLAs, Amazon Elastic Compute Cloud (EC2) and Amazon Elastic Block Store (EBS) commit to providing an uptime of at least 99.99% for their services in a monthly billing cycle [2] and are liable for a 10% service credit if the monthly uptime of EC2 is less than 99.9% but equal to or greater than 99.0%. EC2's SLA also contains information about their service commitment and procedure for compensating the consumer if the commitment is not fulfilled. IBM provides SLAs for high availability and non-high availability zones. According to the IBM cloud service description published in April 2019 [3], for all cloud services except IaaS, the service provider is liable for 10% of service credit if the monthly uptime is less than 99.95% for services in both high availability and non-high availability zones and 25% of service credits for services if it is less than 99.90% for high availability zone and less than 99.0% for non-high availability zones. However, there are exclusions of no credit for failure to meet the SLA due to causes such as technology, design, unsupported system, hardware, facility and client system administration. Microsoft Azure offers different SLAs for its services. It commits to a monthly uptime of at least 99.99% for Azure Active Directory [4] both for basic and premium services and if the uptime percentage is less than 99.9% then the provider is liable for a service credit of 25%. The service credit will increase to 100% if the uptime percentage drops to less than 95%.

An SLA is comprised of one or many performance metrics called service level objectives (SLOs), which are further composed of one or many low-level resource metrics or quality of service (QoS) parameters. To avoid a service violation and penalties, the service provider needs to predict QoS parameters beforehand and in the case of discrepancies take immediate action to avoid a violation. Quality of service (QoS) is the measurable characteristics on which the overall performance of the cloud service depends. It is one of the key factors used to measure the SLOs in an SLA. A critical parameter combines with other QoS parameters to form a performance metric. Therefore, effective prediction methods are needed by which the service provider can predict instances of deviations in the QoS that will be delivered and take appropriate steps so that quality promised in the SLAs is maintained. There are many techniques that are used to predict QoS parameters for future intervals; however, each technique behaves differently depending on the choice of the prediction method, the data patterns for input, the prediction method parameters and considering the distant or near past data for the prediction. The choice of a prediction algorithm with appropriate control parameters for each method plays an important role in service providers managing SLAs and avoiding SLA violations. Not meeting the agreed-upon QoS parameters results in violation penalties and damage to a provider's reputation. A provider can reduce their risk of service violation by following more formal quantitative prediction methods [5] and by selecting an optimal parameter for the related prediction method. Therefore, in order to manage the risk of SLA violation and to avoid violation penalties, it is vital that the service provider determine the appropriate QoS prediction method based on its prediction accuracy at different time intervals.

In the authors' previous work [6], [7], we analysed the accuracy of the time series and machine learning prediction approaches and ranked them according to their prediction accuracy. The contribution of this paper is two-fold. First, we demonstrate that how prediction accuracy changes by varying the value of the individually control parameters of the related algorithm; for example, how the simple exponential smoothing prediction method behaves while the value of a smoothing parameter or control parameter is changing, i.e. the value of α from 1 to 9. Choosing an optimal parameter for the associated prediction method assists in minimizing the mean square error (MSE), taking the cut-off frequency and the computational limitation of the transfer function, among other advantages [8], that resulted in minimized errors and maximized prediction accuracy. Secondly, we analysed the prediction accuracy by considering the freshness factor of data. By freshness, we mean how a prediction algorithm performs in an earlier time period as compared to the following ones. Usually, prediction accuracy changes by considering data from previous intervals to predict the future interval [7]. For this study, six of the most commonly used prediction methods are considered: simple exponential smoothing, simple moving average, weighted moving average, Holt-Winter double exponential smoothing, extrapolation and the autoregressive integrated moving average (ARIMA). Prediction accuracy is determined by changing each control parameter on a real cloud dataset from Amazon EC2 IaaS cloud services. Three QoS parameters are considered: central processing unit (CPU), memory and input-output (I/O). The QoS values of these parameters are predicted and then the predicted values are compared with the actually observed ones. The evaluation benchmark for comparison is mean square error, root means square error (RMSE) and mean absolute deviation (MAD). To consider various possible patterns in the input QoS data and determine their effect on the output values, each dataset is divided into 10-time intervals, starting from 5 minutes to 4 weeks, which provides datasets which contain different patterns.

A. THE GAPS IN THE LITERATURE

Firstly, it has been observed that while the existing literature evaluates various prediction approaches including machine learning, stochastic and time series prediction, none of them discusses how the different control parameters of each approach impact the prediction accuracy of cloud QoS data. Secondly, none of the existing studies explores how the freshness of data impacts on prediction accuracy. Finally, none of the previous research examines how the prediction algorithms and each individual control parameters behave on cloud QoS data with different data patterns such as horizontal, cyclic and sessional at different time intervals (from 5 min to 4 weeks) and look at the prediction algorithms from a cloud SLA management perspective. This paper addresses all three gaps.

B. CONTRIBUTIONS OF THE PAPER

This paper aids in the understanding of the existing time series prediction algorithms by analysing how different prediction algorithms behave when the values of different control parameters are varied at different data patterns. The second contribution of this work is to analyse prediction accuracy by considering the freshness of data that means that how the prediction algorithm responds by considering data from earlier interval to predict for future intervals. This paper analyses the prediction approaches for 10 different time intervals between 5 minutes and 4 weeks. Consumers usually request for a new virtual machine about 12 – 15 minutes before they need [9] and it takes about 5 to 10 minutes to set up a new virtual machine. Therefore, we choose a minimum of 5 minutes and increased the time intervals up to 4 weeks to analyse how each prediction approach behaves with various control parameters.

C. SIGNIFICANCE OF THE PAPER

This study is significant for the following reasons. Firstly, by knowing the prediction algorithm with optimal control parameters to detect possible service violations before the actual violation occurs, the cloud service provider would be able to optimally manage their SLA to avoid violations. Secondly, the paper assists the cloud provider in choosing the optimum prediction method for different data patterns at varying time intervals. Thirdly, the cloud provider can improve its reputation in the market by achieving high consumer satisfaction, eventually converting potential consumers to regular consumers. Finally, the discussed approaches assist interacting parties in proactively managing SLA, not only on single services but in managing combined services such as in a Cloud of Things (CoT) environment. In CoT, the required services are combined from different services from different regions. Therefore QoS parameters such as availability and response time need to be accurately predicted to assist in service formation and protective management.

The structure of the paper is organized as follows. Section 2 discusses and critically analyses related studies from the literature in the area of this work. Section 3 describes the adopted approaches and the benchmark used to measure the prediction accuracy. Section 4 presents the overall prediction accuracy of each approach with varying control parameters. Section 5 presents the prediction accuracy by considering the freshness of data. Section 6 presents the findings and discussion and, finally, Section 7 concludes the paper.

II. RELATED STUDIES

Researchers have used many techniques for QoS prediction in recent times. The current approaches used for QoS prediction for cloud services will be reviewed in this section. The QoS prediction approaches are an effective way to predict the near-future values of cloud services [10]. Predictions are based on an analysis of previous QoS data. Typical QoS prediction techniques that have been proposed are neural network and artificial intelligence [11], collaborative filtering technology [12], case-based reasoning [13], Bayesian networks [14] and combinational prediction techniques [15].

Kumar *et al.* [16] propose an artificial neural network model using past QoS performance parameter data to predict missing QoS parameters. The performance was analysed using a comparison of three training algorithms: Levenberg-Marquardt (LM), Bayesian regularization (BR) and the scaled conjugate gradient (SCG). The results show that the BR algorithm is more precise in predicting the QoS parameters in cloud computing environments; however, there is a need to develop models with varying neural network values and different neural network architectures for more accurate prediction results. To resolve the issue of overload information, the QoS approaches for service recommendations have been incorporated into cloud service marketplaces [17]–[19]. To predict QoS and ranking of cloud services, the authors [20] applied the Spearman coefficient on QoS similarity computing in the typical collaborative filtering (CF) model. However, all of the approaches discussed above [20], [21] fail to consider the fact that QoS values are not constant and are dependent on the time factor, instead of focusing only on QoS information for the service recommendation.

Furthermore, the QoS values prediction problem is closely related to matrix factorization methods [22]. The matrix and the collaborative filtering sparse problems impact the prediction accuracy and overall recommendation quality of QoS values [23]. For these reasons, trust-aware collaborative filtering methods such as [24], [25] have gained attention in recent times. Liu *et al.* [23] proposed a novel clustering-based and trust-aware method for personalized and reliable QoS values prediction. Moreover, Wu *et al.* [26] proposed a context-aware prediction model that provides a more effective approach for the QoS prediction in the case of sparse data. Ma and Shan [22] proposed a general collaborative filtering (GFC) method based on a neural network to model the user-service interactions. The QoS values from 339 users on 5,825 web services were evaluated and the results showed better prediction accuracy than existing collaborative filtering methods. However, the trials only focused on response time QoS prediction.

Zhang *et al.* [27] suggested a time-aware personalized QoS prediction framework (WSPred) to predict unknown QoS values. With the help of a tensor factorization model, a time-sensitive QoS prediction method was developed. This method is based on a 3D matrix involving the dimensions of user, service and time. To study the relationship between the trio (user, service and invocation time), Zhang *et al.* [28] widened the work in [27] by forming a non-negative tensor factorization model. These time-sensitive, CF-based QoS prediction methods [27], [28] neglect the fact that predictable QoS values of target user at particular time period will be influenced both by the QoS values of previous time intervals and by other similar users' QoS values. Also, Lo *et al.* [29] suggested a framework of extended matrix factorization (EMF) along with the relational regularization. For the same purpose, some researchers explore incorporating EMF by

adding information such as geographical location, time and reputation. The survey of QoS prediction is reported in [30]–[32]. The performance prediction model is considered in [30]. To assess and predict the performance of servers employed in cloud infrastructure, the authors utilize the Markovian arrival process (MAP) and a MAP/MAP/1 queuing model. The QoS requirements are met by resolving the problem of QoS optimization at runtime in [33]. To fulfil the particular QoS requirements of service-oriented systems and to identify a runtime variation methodology, a linear programming optimization problem is implemented in [34]. To develop QoS adaptive service-based systems for meeting the QoS attributes defined earlier, a multi-objective optimization problem is suggested in [33]. Gallotti *et al.* [35] proposed QoS prediction based on the model checking solution to assist in QoS prediction at the earliest possible time.

Wu *et al.* [36] put forward a learning neighbourhood-based prediction method. In this approach, the previous profile record is critical for the prediction of a service violation. The service brought forward by Romano *et al.* [37] was QoS monitoring as a service (QoS-MONaaS). It consists of four elements, which are highly functional. Since QoS-MONaaS elements operate in an inconsistent cloud environment, they are capable of managing functions as per time use [38]. ur Rehman *et al.* [39] proposed the service management model. This framework allows the end user of a service to not only analyse the efficiency of services with the help of predictable QoS results but also helps them to decide whether to continue or discontinue the use. Chaudhuri *et al.* [40] used earlier service records to predict the QoS parameters. A flexible method of computation has been used for the confirmation of this approach on the public dataset. A method called local neighbourhood matrix factorization (LoNMF) was proposed by Lo *et al.* [41] for forecasting QoS parameters. The integration of the matrix factorization method with the network and service neighbourhood information by Qi *et al.* [42] made possible the prediction of personalized QoS parameters. Zheng *et al.* [43] brought forward a prediction method by merging item-based and user-based collaborative filtering methods. Sun *et al.* [44] utilized the memory-based collaborative filtering method and QoS web services' characteristics for the similarity measurement. Shao *et al.* [45] employed the method of collaborative filtering for similarity mining based on earlier performance. The evaluation of time series is the procedure used to measure the parameters at a particular time, such as hourly, daily, weekly, monthly or any regular time interval. The data obtained from the evaluation of time series not only reveals the data patterns in a time series but also proposes a proper method for predicting future data and provides information about the system's previous behaviour [46]. Because each of the predictable revealed patterns in time series data exhibits particular characteristics, it helps in selecting an optimal prediction method [47]. Additionally, current methods fail to predict the variation of dynamic web service QoS parameters. Moreover, the average QoS parameters are described by the historical data, however QoS parameters fluctuate based on different locations and networks. Therefore, to predict dynamic web services, Song *et al.* [48] suggest a new technique for personalized QoS parameters. However, the authors used a small dataset, which may limit the development of QoS value prediction.

The main document to look at in order to examine the commitment of services' source and the end-user is the service level agreement (SLA). Hussain *et al.* [49] provide a comparative analysis of the SLA violation prediction model depending on the profile record. Kumar *et al.* [1] developed a model to predict 15 QoS parameters of web services based on 37 source code metrics. The performance of the matrices was measured with six different sets as input and assessed using extreme learning machines (ELM) with various kernel functions. The results show that the performance of the predictive model differs with the different sets of feature selection technique, software metrics and the kernel functions. In another study, Hussain *et al.* [6] compared the time series with machine learning-based prediction approaches. The authors provided a comprehensive evaluation of existing SLA management approaches. The above-discussed studies give predictable QoS values; however, these approaches do not fully take into account the impact and significance of QoS attributes and resources of the core cloud architecture.

Although the approaches discussed in this section assist different stakeholders in a cloud environment to predict QoS parameters and help them in making the decision to mitigate it, to the best of our knowledge the approaches are lacking in the following areas:

- None of the approaches discusses QoS prediction with varying data patterns at different time intervals;
- None of the prediction approaches discusses how prediction algorithms behave by changing each individual control parameter of the related prediction algorithm;
- None of the approaches discusses how prediction accuracy is impacted by considering data from different time intervals;
- None of the approaches demonstrates how different data patterns impact output results; and
- Very few of the approaches discuss predictions from the perspective of cloud small scale service providers for SLA management while using a real cloud dataset.

III. PREDICTION APPROACHES AND ACCURACY BENHMARK

There are several types of prediction algorithms available in the literature for time series predictive modelling with varying degree of prediction accuracy. For this study, we have selected six commonly used prediction methods – simple exponential smoothing, simple moving average, weighted moving average, Holt-Winter double exponential smoothing (HWDES), extrapolation and the autoregressive integrated moving average (ARIMA). The reason for choosing those methods because these methods have been used widely in time series dataset [7], [50]–[52], and give optimum

prediction results. The authors [8], [53]–[56] used simple exponential smoothing, simple moving average, weighted moving average method to get best prediction results, and the authors [57]–[59] used ARIMA and HWDES as the prediction method to get an ideal result. A brief explanation of these prediction approaches and their related methods is provided below.

A. SIMPLE EXPONENTIAL SMOOTHING METHOD

Exponential smoothing is an optimal forecasting approach for state-space models [60]–[64]. Exponential smoothing was proposed by Brown [65] for smoothing and predicting time series data. The basic purpose of this method is to smooth random variations in time series data and give optimal results for short-range forecasting [66].

The method predicts the forthcoming data by taking the weighted average of all previous data where its weights decrease on an exponential basis over time. The smoothing function starts from the second observation and needs an initial value that most of the time is chosen as the first value of the series $F_{t-1} = y_t$.

These weights are determined by a smoothing constant, as presented in Equation 1:

$$\hat{K}_{a+1} = \propto \boldsymbol{K_a} + (\mathbf{1} - \propto)\hat{K}_{\boldsymbol{a}} \tag{1}$$

where $\hat{K}_{a+1}$ is the forecasted value at interval a+1, $\hat{K}_a$ is the forecasted value at time interval a, K_a is the actual value at interval 'a' and $'\propto'$ is a value -smoothing constant that ranges between 0 and 1 i.e. $0 < \propto < 1$.

B. SIMPLE MOVING AVERAGE METHOD

This prediction method considers the data from the earlier time intervals, averages them and then uses the result to predict the upcoming time interval [67]. The working of a simple moving average is presented in Equation 2.

$$\hat{S}_{t+1} = \frac{\sum_{i=t-j+1}^{t} S_t}{j} \tag{2}$$

where $\hat{S}_{t+1}$ is the predicted result for future interval $t + 1$ and j is the total time intervals. Each of the j previous values has a weight of *1/j*. When the size of the previous record j becomes larger, each individual value of the recent past is assigned a lesser weight in order to have a smooth series forecast graph. The first period in S_{t-j+1} is one stage old. The second period is two stages old and so on till j term. The phrase *term moving* is used because each time a new value replaces the previous value in the equation, a new average is calculated.

The average for each period changes or is moved based on the new data. The problem with this method is that it always lags behind the actual data. To use the moving average, we need to select the number of time series j. The observation at j depends on the relevance of the number of previous values. For a small number of previous values, a small value of j is considered and for a large number of previous values, a larger value of j is considered. Therefore, for a smaller number of datasets, j will track shifts more quickly in a dataset and a larger value of j gives the optimal result for smoothing random fluctuation.

C. WEIGHTED MOVING AVERAGE METHOD

The prediction method gives a higher weight to the nearest past data rather than the older data to calculate the average [68]. In this method, a set of weighting factors are selected such as $w_1, w_2, w_3 \ldots, w_k$, with the sum of all these weights being equal to 1, as presented in Equation 3.

$$\sum_{i=1}^{k} w_i = 1 \tag{3}$$

The weights are used to determine the smoothed statistics value s_t, as presented in Equation 4.

$$s_t = \sum_{i=1}^{N} w_i a_{t+1-i} = w_1 a_t + w_2 a_{t-1} + \ldots + w_N a_{t-N+1} \tag{4}$$

where a is a raw time series and w is a weighting factor.

Many technical analysts believe that assigning a greater weight to the recent past data rather than older data produces good prediction results. When using this method, the system reacts quickly when it detects any change. Equation 5 presents the weighted moving average:

$$\boldsymbol{F_t} = \frac{\sum_{i=1}^{N} \boldsymbol{w_i} * \boldsymbol{A_t}}{\sum_{i=1}^{N} \boldsymbol{wi}} \tag{5}$$

where w is the weighting factor, A is the actual data, F is the average data and N is the total time period.

D. HOLT-WINTER DOUBLE EXPONENTIAL SMOOTHING METHOD

This prediction method deals with data that have a trend and seasonality. Seasonal data are time-series data that repeat after every N time interval. The Holt-Winter method [69] comprises a prediction equation and a smoothing equation for level, seasonality and trend. There are two methods in the Holt-Winter model that vary from each other based on seasonal components. These methods are the multiplicative seasonal component and the additive seasonal component [70]. The multiplicative seasonal component is used when seasonal data changes proportionally with the time-series data or when there is a multiplicative change in seasonality, as presented in Equation 6.

$$y_t = (p_1 + p_2 t) * SF_t + e_t \tag{6}$$

where p_1 is the permanent factor, p_2 is the linear trend factor, SF_t is a seasonal factor and e_t is the error factor. The additive seasonal component is used when there is a constant seasonal change in the data, irrespective of the overall level of time-series data, as presented in Equation 7.

$$y_t = p_1 + p_2 t + SF_t + e_t \tag{7}$$

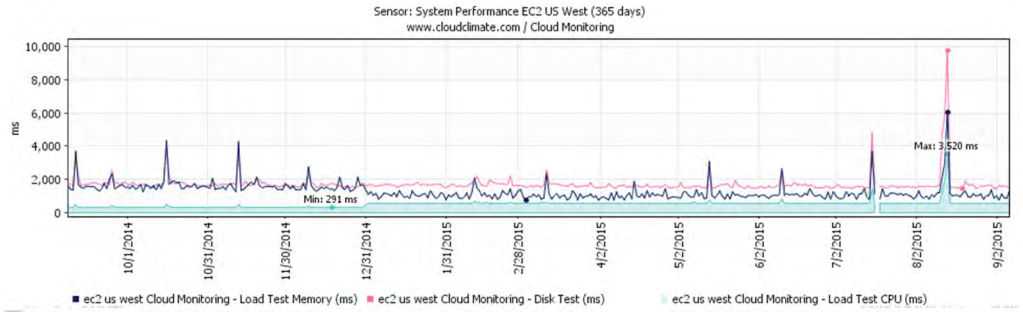


FIGURE 1. QoS parameters of EC2 US West.

E. AUTOREGRESSIVE INTEGRATED MOVING AVERAGE METHOD (ARIMA)

The method was formulated by mathematical statisticians George and Gwilym in the 1970s [71] to use with business and economic data. This is one of the most efficient of the autoregressive moving average (ARMA) methods that include the seasonality component [72].

A non-seasonal ARIMA model is represented by ARIMA (p, d, q) such that p, q and d are positive integers and p represents autoregressive (AR), d represents the level of differencing and q represents moving average (MA).

The ARIMA method is presented in Equation 8.

$$\begin{aligned}\check{y}_t &= a + \psi_1 y_{t-1} + \psi_2 y_{t-2} + \psi_3 y_{t-3} + \ldots + \psi_n y_{t-n} \\ &\quad + b + \hbar_1 e_{t-1} + \hbar_2 e_{t-2} + \hbar_3 e_{t-3} + \ldots + \hbar_n e_{t-n} + e_t \end{aligned} \tag{8}$$

where $\check{y}_t$ is the predicted value. The sequence of the AR model, the number of differencing and the sequence of the MA model is presented as ARIMA (p, d, q). Therefore, ARIMA (1, 1, 2) is presented as AR $= 1$, MA $= 2$ and the difference of 1.

F. EXTRAPOLATION METHOD

The prediction method predicts forthcoming data based on previously available data and considers all data including data beyond the range of known data points. This method produces better results for the short range than the long range because irrelevant previous data make the long-range results noisy and insignificant.

The method is reliable, inexpensive, quick and effortlessly automated; however, the process of extrapolation can only be applied to historical data. Short-range data, in which the values that have been collected are less than a year old, are adjusted seasonally adjusted by a seasonal adjustment to reduce error in prediction [73]. Some of the common methods of extrapolation are linear extrapolation, polynomial extrapolation and conic extrapolation. In the linear extrapolation method, a tangent line is drawn which extends outside the limit of a series, as presented in Equation 9.

$$b\,(\tilde{a}) = b_1 + \frac{\tilde{a} - a_1}{a_2 - a_1} * (b_2 - b_1) \tag{9}$$

where (a_1, b_1) and (a_2, b_2) are the end point of a series, $b\,(\tilde{a})$ is the predicted value at point $\tilde{a}$. The polynomial extrapolation determines the function value at some point $\tilde{a}$ on the x-axis, which is in the range of dataset n value. The conic extrapolation selects five nearest points around the known data which is performed by using a template known as ''conic section template'' [74].

G. ACCURACY BENCHMARK FOR MEASURING PREDICTION ACCURACY

We analysed the prediction accuracy of each method using mean square error (MSE), root mean square error (RMSE) and mean absolute deviation (MAD). MSE is the average of the squared predicted errors as presented in Equation 10.

$$MSE = \frac{\sum_{t=b+1}^{a} e_t}{a - b} \tag{10}$$

MSE gives a quadratic loss function as it squares and averages the different errors. MSE is therefore advantageous at the point when we would be concerned about huge errors whose negative results are proportionately greater than the equivalent smaller ones [75]. RMSE is calculated by taking the square root of the MSE as presented in Equation 11.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{\sum_{t=b+1}^{a} e_t}{a - b}} \tag{11}$$

MAD is also referred to as mean absolute error (MAE), which is the mean absolute value of the forecast error. MAD does not consider positive or negative forecast errors, as presented in Equation 12.

$$MAD = \frac{\sum_{t=v+1}^{x} |e_t|}{x - v} \tag{12}$$

Prediction accuracy depends on forecast error, which is the degree of alteration between two values – predicted and observed. If Z_t and $\check{Z}_t$ represent the observed and predicted values respectively at time interval t, then prediction error e_t is calculated using Equation 13.

$$e_{\mathrm{t}} = Z_{\mathrm{t}} - \check{Z}_{\mathrm{t}} \tag{13}$$

A positive error indicates that the forecast method has underestimated the actual observation, and a negative error

TABLE 1. Data patterns for all datasets.

QoS attribute	5 mins	10 mins	20 mins	1 hr	4 hrs	12 hrs	1 day	1 week	2 weeks	4 weeks
CPU	HOZ	HOZ	HOZ	HOZ	HOZ	HOZ	RND	CYC	SNL	SNL
Memory	CYC	CYC	CYC	CYC	CYC	RND	RND	TRD	SNL	SNL
I/O	RND	RND	RND	RND	HOZ	HOZ	HOZ	RND	SNL	SNL

TABLE 2. Prediction using the SES method for CPU.

Time interval	Method 1: Simple Exponential Smoothing-CPU									
5 minutes	α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
	MSE	5,814.78	6,127.30	6,477.62	6,871.47	7,317.54	7,827.20	8,415.24	9,101.31	9,912.08
	RMSE	76.25	78.28	80.48	82.89	85.54	88.47	91.73	95.4	99.56
	MAD	9.92	9.93	9.98	10.05	10.13	10.2	10.25	10.3	10.35
10 minutes	α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
	MSE	43.57	43.67	44.8	46.53	48.78	51.55	54.87	58.84	63.57
	RMSE	6.6	6.61	6.69	6.82	6.98	7.18	7.41	7.67	7.97
	MAD	4.59	4.57	4.57	4.58	4.62	4.67	4.75	4.82	4.9
20 minute	α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
	MSE	27.99	27.15	27.34	27.98	28.94	30.22	31.82	33.8	36.23
	RMSE	5.29	5.21	5.23	5.29	5.38	5.5	5.64	5.81	6.02
	MAD	3.81	3.72	3.7	3.69	3.71	3.77	3.83	3.91	4
1 hour	α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
	MSE	19.81	16.19	15.43	15.39	15.68	16.22	16.97	17.95	19.19
	RMSE	4.45	4.02	3.93	3.92	3.96	4.03	4.12	4.24	4.38
	MAD	3.22	2.87	2.81	2.81	2.85	2.9	2.97	3.04	3.15
4 hours	α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
	MSE	420.83	413.31	424.71	445.18	472.04	504.88	544.17	590.98	646.94
	RMSE	20.51	20.33	20.61	21.1	21.73	22.47	23.33	24.31	25.44
	MAD	8.1	6.76	6.11	5.8	5.67	5.58	5.53	5.5	5.5
12 hours	α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
	MSE	238.79	216.12	206.02	201.68	201.89	206.24	214.49	226.63	242.95
	RMSE	15.45	14.7	14.35	14.2	14.21	14.36	14.65	15.05	15.59
	MAD	9.96	8.69	7.87	7.28	6.9	6.59	6.38	6.21	6.21
1 day	α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
	MSE	90,26.24	94,89.07	101,39.25	108,80.19	116,93.45	125,74.30	135,25.61	145,51.94	156,63.40
	RMSE	30.43	30.04	31.42	32.85	34.95	35.61	36.77	38.47	39.77
	MAD	15.65	17.75	18.69	19.51	20.24	21.61	22.76	24.72	25.79
1 week	α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
	MSE	12,924.12	7,157.09	4,981.36	3,898.21	3,263.83	2,856.92	2,581.83	2,390.80	2,258.07
	RMSE	113.68	84.6	70.58	62.44	57.13	53.45	50.81	48.9	47.52
	MAD	80.72	54.21	41.05	33.92	29.77	26.9	24.82	23.39	22.42
2 weeks	α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
	MSE	11,404.40	6,619.06	4,561.19	3,494.42	2,851.81	2,425.29	2,125.19	1,907.26	1,747.38
	RMSE	106.79	81.36	67.54	59.11	53.4	49.25	46.1	43.67	41.8
	MAD	80.53	50.26	36.35	28.72	24.57	21.9	20.12	18.77	17.86
4 weeks	α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
	MSE	16,357.11	11,822.96	8,864.75	7,004.97	5,809.26	5,005.89	4,443.18	4,036.18	3,736.51
	RMSE	127.89	108.73	94.15	83.7	76.22	70.75	66.66	63.53	61.13
	MAD	109.07	81.4	62.5	49.87	41.57	35.57	31.29	28.07	25.84

indicates that the forecast method has overestimated the actual observation for time period t.

IV. ANALYSING OVERALL ACCURACY of PREDICTION ALGORITHMS with VARYING CONTROL PARAMETERS

To analyse the prediction accuracy of the previously mentioned approaches, we use a dataset from Amazon EC2 US West for a period of three years starting from 28-03-2013 to 28-03- 2016. The data were collected from CloudClimate [76] using the PRTG monitoring service [77].

The QoS parameters considered for this study are CPU, memory and I/O. Figure 1 represents a part of the PRTG network dataset from 01-01-2014 to 09-02-2015 for the CPU, memory and I/O.

A. MEASUREMENT INTERVAL OF QoS PARAMETERS AND DETERMINING THE PATTERNS IN THEM

We divided the measurement intervals of a dataset into 10 subsets i.e. 5 minutes, 10 minutes, 20 minutes, 1 hour, 4 hours, 12 hours, 1 day (24 hours), 1 week, 2 weeks and

TABLE 3. Prediction using the SMA method for CPU.

Time interval	Method 2: Simple moving average method - CPU											
5 minutes	K	2	193	384	575	766	957	1148	1339	1530	1721	1912
	MSE	3669.63	5486.13	5511.18	5514.3	5516.05	5519.17	5521.13	5521.29	5522.52	5522.39	5522.5
	RMSE	60.58	74.07	74.24	74.26	74.27	74.29	74.3	74.31	74.31	74.31	74.31
	MAD	6.89	10.46	10.89	10.8	10.78	10.79	10.78	10.76	10.75	10.74	10.74
10 minutes	K	2	98	194	290	386	482	578	674	770	866	
	MSE	24.57	49.25	58.59	66.89	73.59	78.85	82	83.95	85.45	86.52	
	RMSE	4.96	7.02	7.65	8.18	8.58	8.88	9.06	9.16	9.24	9.3	
	MAD	3.22	5.13	5.9	6.37	6.67	6.86	6.98	7.04	7.09	7.13	
20 minutes	k	2	50	98	146	194	242	290	338	386	434	
	MSE	14.41	32.44	41.59	49.72	56.46	61.7	64.8	66.76	68.25	69.28	
	RMSE	3.8	5.7	6.45	7.05	7.51	7.86	8.05	8.17	8.26	8.32	
	MAD	2.59	4.15	4.86	5.32	5.61	5.81	5.93	6	6.05	6.08	
1 hour	k	2	18	34	50	66	82	98	114	130	146	
	MSE	7.65	19.17	28.3	36.1	42.79	47.78	50.63	52.63	54.06	55.09	
	RMSE	2.77	4.38	5.32	6.01	6.54	6.91	7.12	7.25	7.35	7.42	
	MAD	2.01	3.04	3.78	4.23	4.53	4.73	4.82	4.89	4.94	4.98	
4 hours	k	2	20	38	56	74	92	110	128	146	164	182
	MSE	231.35	409.87	439.66	443.32	460.68	473.8	487.32	488.28	492.66	494.3	493.68
	RMSE	15.21	20.25	20.97	21.06	21.46	21.77	22.08	22.1	22.2	22.23	22.22
	MAD	3.84	8.59	9.62	10.41	11.21	12.05	12.95	13.12	13.32	13.39	13.34
12 hours	k	2	8	14	20	26	32	38	44	50	56	
	MSE	93.42	193.83	207.02	224.08	234.69	247.27	256.59	259.14	263.17	263.29	
	RMSE	9.67	13.92	14.39	14.97	15.32	15.72	16.02	16.1	16.22	16.23	
	MAD	4.33	8.12	8.97	10.04	10.8	11.67	12.44	12.55	12.73	12.76	
1 day	k	2	5	8	11	14	17	20	23	26	29	
	MSE	63670.95	66687.11	73782.67	69522.18	70676.63	69877.87	70270.89	70131.15	70191.73	70173.93	
	RMSE	252.33	258.24	271.63	263.67	265.85	264.34	265.09	264.82	264.94	264.9	
	MAD	152.96	155.5	156.29	145.37	147.34	147.11	150.36	148.85	149.32	148.99	
1 week	k	2	4	6	8	10	12	14	16	18	20	22
	MSE	1359.15	2989.67	4749.1	6522.72	8352.68	10058.03	11736.31	13191.2	14320.2	15130.1	15643.45
	RMSE	36.87	54.68	68.91	80.76	91.39	100.29	108.33	114.85	119.67	123	125.07
	MAD	17.71	27.04	38.25	47.96	58.3	67.08	75.85	82.29	86.87	90	91.96
2 weeks	k	2	12	22	32	42	52	62	72	82	92	102
	MSE	1120.64	9354.04	15450.27	17891.26	17376.5	14254.08	13482.06	13978.8	14527.24	14810.67	14411.74
	RMSE	33.48	96.72	124.3	133.76	131.82	119.39	116.11	118.23	120.53	121.7	120.05
	MAD	13.81	61.54	95.35	111.81	115.29	105.62	101.71	103.06	105.22	107	106.04
4 weeks	k	2	7	12	17	22	27	32	37	42	47	52
	MSE	2367.97	10155.94	15324.11	17370	15914.55	13084.02	12931.61	13501.46	14055.19	14161.09	13799.12
	RMSE	48.66	100.78	123.79	131.8	126.15	114.39	113.72	116.2	118.55	119	117.47
	MAD	22.35	68.63	97.97	111.98	109.86	100.06	98.84	100.60	102.95	104.08	103.10

4 weeks. The minimum dataset is of 5 minutes and we chose it because it takes approximately 5 to 10 minutes for a service request to result in the requested resources [9]. Therefore, a time interval of 5 minutes is the minimum possible time for the provider to take appropriate mitigating action when it detects that a violation is likely to occur. Each of the time intervals has different data patterns which are presented in Table 1. We observed five different data patterns in a dataset: trend (TRD), horizontal (HOZ), random (RND), sessional (SNL) and cyclic (CYC).

1) QoS PREDICTION ACCURACY USING THE SIMPLE EXPONENTIAL SMOOTHING (SES) METHOD

In this subsection, we analyse the prediction precision of the SES method to predict the QoS parameters. The existing literature advocates that different observations of α be used for prediction to represent the sensitivity of a forecast.

Chopra and Meindle [78] suggest that a value of $\alpha = 0.2$ is the optimal parameter value that generates an accurate result in the SES method. Schroeder *et al.* [79] recommend that when the value of α is set between $\alpha = 0.1$ and $\alpha = 0.3$, it generates an optimal result in SES. Heizer *et al.* [80] propose that when the value of α is set between $\alpha = 0.05$ to 0.5, the SES produces an optimal prediction result.

To observe the effect of α on the prediction accuracy of a cloud dataset, we analyse the prediction result with all nine possible values for the variable. We start with the value of 0.1 and increase it to 0.9.

TABLE 4. Prediction results using the WMA method for CPU, memory and I/O.

Time interval	k	α	CPU Test			Memory Test			I/O test		
			MSE	RMSE	MAD	MSE	RMSE	MAD	MSE	RMSE	MAD
5 minutes	2	0.5	6311.04	79.44	8.92	12774721.5	3574.17	543.73	36220.42	190.32	61.91
5 minutes	2	1.5	2312.15	48.08	5.44	4684977.79	2164.48	325.53	12773.97	113.02	37.06
5 minutes	2	2	1566.36	39.58	4.43	3174781.89	1781.79	264.51	8553.48	92.49	30.2
5 minutes	2	5	352.28	18.77	2.01	714546.05	845.31	119.68	515.02	22.69	7.18
5 minutes	2	10	98.23	9.91	1.03	199295.26	446.42	61.53	1869.15	43.23	13.85
5 minutes	900	0.5	5578.8	74.69	13.07	14973518.27	3869.56	671.04	47574.63	218.12	93.25
5 minutes	900	1.5	2919.58	54.03	4.71	3147991.85	1774.26	221.91	11588.11	107.65	24.68
5 minutes	900	2	1820.03	42.66	3.58	1952605.32	1397.36	163.71	7062.94	84.04	18.81
5 minutes	900	5	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
5 minutes	900	10	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
5 minutes	1910	0.5	5599.86	74.83	15.33	12039412.91	3469.79	430.82	49081.23	221.54	91.73
5 minutes	1910	1.2	0.0014	0.0369	0.0033	0.36	0.6	0.04	30	5.48	0.18
5 minutes	1910	2	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
5 minutes	1910	5	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
5 minutes	1910	10	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
10 minutes	2	0.5	30.78	5.55	3.34	5329348.86	2308.54	440.42	15036.31	122.62	45.26
10 minutes	2	1.5	11.08	3.33	2.01	1918565.59	1385.12	264.25	5413.07	73.57	27.16
10 minutes	2	2	7.7	2.77	1.67	1332337.21	1154.27	220.21	3759.08	61.31	22.63
10 minutes	2	5	1.92	1.39	0.84	333084.3	577.13	110.1	939.77	30.66	11.32
10 minutes	2	10	0.57	0.76	0.46	99099.46	314.8	60.06	279.6	16.72	6.17
10 minutes	480	0.5	127.86	11.31	8.62	8261169.04	2874.22	653.31	33271.3	182.4	91.06
10 minutes	480	1.5	8.76	2.96	1.35	1344374.7	1159.47	203.59	7003.64	83.69	22.57
10 minutes	480	2	5.37	2.32	1.03	813530.58	901.96	147.66	4270.24	65.35	17.19
10 minutes	480	5	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
10 minutes	480	10	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
10 minutes	950	0.5	150.65	12.27	9.49	6923553.4	2631.26	438.42	34638.38	186.11	87.98
10 minutes	950	1.2	0.00283	0.05	0.00705	12.19	3.49	0.64	42.66	6.53	0.38
10 minutes	950	2	0.00092	0.03	0.00257	0.19	0.44	0.04	14.5	3.81	0.16
10 minutes	950	5	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
10 minutes	950	10	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
20 minutes	2	0.5	158.37	12.58	9.79	4272631.52	2067.03	619.39	13664.65	116.9	52.31
20 minutes	2	1.5	15.3193	3.91	2.5364	1559267.79	1248.71	373.3	5017.94	70.84	32
20 minutes	2	2	10.3023	3.21	2.0687	1055095.41	1027.18	303.51	3401.49	58.32	26.18
20 minutes	2	5	2.2758	1.51	0.94321	236629.83	486.45	136.83	766.07	27.68	12.09
20 minutes	2	10	0.62993	0.79	0.48701	65904.27	256.72	70.39	213.71	14.62	6.3
20 minutes	240	0.5	114.6	10.71	8.29	4831640.19	2198.1	650.13	23292.68	152.62	84.85
20 minutes	240	1.5	14.19	3.767	2.156	1032723.49	1016.23	243.05	4532.16	67.32	21.63
20 minutes	240	2	8.609	2.934	1.626	616540.78	785.2	175.41	2767.77	52.61	15.94
20 minutes	240	5	1.67	1.29	0.67	116333.39	341.08	67.95	544.37	23.33	6.62
20 minutes	240	10	0.45	0.67	0.34	31290.04	176.89	33.99	147.94	12.16	3.38
20 minutes	480	0.5	127.86	11.31	8.62	4372559.71	2091.07	457.93	26662.04	163.29	85.45
20 minutes	480	1.2	12.92	3.59	1.69	114.26	10.69	480	35.03	5.92	0.52
20 minutes	480	2	5.37	2.32	1.03	0.09	0.29	0.02	12.73	3.57	0.18
20 minutes	480	5	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

The prediction accuracy for each case is observed by ascertaining MAD, MSE and RMSE. Due to space limitations, we present only the prediction results for CPU in Table 2; however, a comparative analysis of the other two QoS parameters, memory and I/O, is presented in Table 8.

2) QoS PREDICTION ACCURACY USING THE SIMPLE MOVING AVERAGE (SMA) METHOD

This subsection determines the prediction accuracy of the SMA method to forecast the QoS factors as previously mentioned. Subject to the size of a dataset, we test the results with different numerical values of k where k is the number of observations. Due to space limitation, we cannot present prediction accuracy for each observation, therefore, to analyse prediction accuracy we start with two entries and then divide 1912 entries of the CPU dataset into ten equal intervals that begin with 193 and end with the last entry 1912. The values for MAD, MSE and RMSE for each time interval with a variable value of k are shown in Table 3. Due to space limitations, we only present the prediction results for CPU. However, a comparative analysis of the other two QoS parameters, memory and I/O, is presented in Table 8.

3) QoS PREDICTION ACCURACY USING THE WEIGHTED MOVING AVERAGE (WMA) METHOD

In this subsection, we determine the prediction accuracy of the WMA method to forecast the QoS parameters. To analyse the impact of the number of observations and an increasing factor, the system takes two inputs from a user: the number

TABLE 5. Prediction results using the Exp method for CPU, memory and I/O.

Time interval	Prediction benchmark	CPU	Memory	I/O
5 minutes	MSE	32,468.230	70,198,694.170	4,199,716.751
5 minutes	RMSE	180.190	8,378.466	2,049.320
5 minutes	MAD	20.210	1,500.633	1,994.841
10 minutes	MSE	201.130	40,582,824.490	413,970.568
10 minutes	RMSE	14.180	6,370.465	2,032.725
10 minutes	MAD	9.180	1,568.985	1,994.105
20 minutes	MSE	111.680	25,193,532.050	4,078,063.168
20 minutes	RMSE	10.570	5,019.316	2,019.422
20 minutes	MAD	7.100	1,643.599	1,988.803
1 hour	MSE	58.920	12,115,868.430	4,006,667.120
1 hour	RMSE	7.680	3,480.785	2,001.666
1 hour	MAD	5.740	17,250.578	1,973.186
4 hours	MSE	2,174.310	35,566,415.510	5,113,263.770
4 hours	RMSE	46.630	1,888.495	2,261.253
4 hours	MAD	10.550	1,370.484	2,148.640
12 hours	MSE	767.320	1,419,041.564	4,766,165.430
12 hours	RMSE	27.700	1,191.235	2,183.154
12 hours	MAD	11.260	938.564	2,108.726
1 day	MSE	546.467	1,366,942.300	4,669,856.767
1 day	RMSE	23.377	1,169.163	2,160.985
1 day	MAD	11.267	954.833	2,041.900
1 week	MSE	3,646.626	4,174,309.000	6,297,784.430
1 week	RMSE	60.387	2,043.110	2,509.539
1 week	MAD	36.330	1,777.600	2,377.028
2 weeks	MSE	2,298.490	4,705,600.319	6,806,946.164
2 weeks	RMSE	47.940	2,169.239	2,609.013
2 weeks	MAD	27.450	2,012.755	2,561.520
4 weeks	MSE	5,166.067	4,651,048.301	6,695,974.526
4 weeks	RMSE	71.875	2,156.628	2,587.658
4 weeks	MAD	37.168	1,985.936	2,519.932

of observations k that is used to consider average and weight factor α.

A weight factor is used to assign the highest weight to recent past data and a lower weight to distant past data and the sum of all the weight factors is equal to one. To analyse the impact of different values of k with respect to the weight factor α, we take 15 entries for each time interval. Subject to the size of a dataset, the value of k (number of observations) and a weighted factor are selected. We select three values of k for each time interval in such a way that the first value of k takes the first two observations, the second value of k takes the mid-value of the dataset and the third value of k takes the last value of the dataset. For each value of k, we evaluate it with five values of weighted factors – 0.5, 1.5, 2, 5 and 10. The values for MAD, MSE and RMSE for CPU, memory and I/O are presented in Table 4. Due to space limitations, we only present the prediction results for three-time intervals: 5, 10 and 20 minutes. However, a comparative analysis of other time intervals is presented in Table 9.

As discussed earlier, due to the larger input for k and the large value of the increasing factor, a smaller weight factor is generated which is smaller than the smallest non-zero floating point value. Therefore, the system did not produce a value, as indicated by N/A in Table 4. From Table 4, we observe that with a higher number of observations and a higher value of alpha, we obtain better prediction accuracy at every time interval.

4) QoS PREDICTION ACCURACY USING THE EXTRAPOLATION (Exp) METHOD

In this subsection, we determine the prediction accuracy of the extrapolation method to forecast the QoS parameters. Table 5 presents the prediction results for CPU, I/O and memory using the extrapolation method.

5) QoS PREDICTION ACCURACY USING THE HOLT-WINTER DOUBLE EXPONENTIAL SMOOTHING (HWDES) METHOD

This subsection determines the prediction precision of the Holt-Winter double exponential smoothing method to forecast the QoS parameters. To analyse the impact of smoothing factor α and trend smoothing factor β on prediction accuracy, we take all possible values of α ($0 < \alpha < 1$) and β ($0 < \beta < 1$) for each time interval and take the value of MAD, RMSE and MSE. Each time interval has 81 entries with all possible values of α and β.

The values for MAD, MSE and RMSE for CPU, memory and I/O are shown in Table 6 and Figure 2a, 2b and 2c. Due to space limitations, we present the prediction results for 5 minutes only; however, a comparative analysis of the other time intervals is presented in Table 9.

TABLE 6. Prediction results using the HWDES method for CPU, memory and I/O.

Time interval			CPU Test			Memory Test			I/O test		
	α	β	MSE	RMSE	MAD	MSE	RMSE	MAD	MSE	RMSE	MAD
5 minutes	0.1	0.1	4961.90	70.44	10.69	10615404.51	3258.13	843.83	30281.04	174.01	67.72
5 minutes	0.1	0.2	5220.43	72.25	11.69	11250389.13	3354.16	962.53	32021.44	178.95	73.59
5 minutes	0.1	0.3	5479.12	74.02	12.69	11880022.66	3446.74	1071.25	33862.01	184.02	78.87
5 minutes	0.1	0.4	5716.66	75.61	13.42	12499339.54	3535.44	1165.74	35507.56	188.43	83.46
5 minutes	0.1	0.5	5994.37	77.42	14.18	13052750.52	3612.86	1242.77	37078.64	192.56	86.97
5 minutes	0.1	0.6	6261.28	79.13	15.00	13553233.32	3681.47	1309.06	38981.22	197.44	90.78
5 minutes	0.1	0.7	6486.90	80.54	15.54	14029038.58	3745.54	1361.65	41103.93	202.74	94.48
5 minutes	0.1	0.8	6732.35	82.05	15.98	14483454.90	3805.71	1405.82	43066.75	207.53	98.02
5 minutes	0.1	0.9	7028.84	83.84	16.58	14925318.27	3863.33	1445.24	44841.47	211.76	100.73
5 minutes	0.2	0.1	4144.52	64.38	8.89	8635225.48	2938.58	634.36	24383.07	156.15	57.23
5 minutes	0.2	0.2	4369.92	66.11	9.59	9113842.81	3018.91	704.64	25881.56	160.88	61.22
5 minutes	0.2	0.3	4597.91	67.81	10.20	9563411.35	3092.48	762.21	27404.29	165.54	64.54
5 minutes	0.2	0.4	4828.85	69.49	10.74	9991768.94	3160.98	805.46	28961.49	170.18	67.65
5 minutes	0.2	0.5	5064.51	71.17	11.27	10411555.88	3226.69	841.02	30516.93	174.69	70.48
5 minutes	0.2	0.6	5303.60	72.83	11.77	10833018.77	3291.36	871.43	32074.55	179.09	73.15
5 minutes	0.2	0.7	5546.96	74.48	12.21	11268079.26	3356.80	897.90	33644.01	183.42	75.68
5 minutes	0.2	0.8	5793.25	76.11	12.69	11725198.45	3424.21	923.05	35237.34	187.72	78.16
5 minutes	0.2	0.9	6040.87	77.72	13.10	12204024.42	3493.43	948.39	36854.62	191.98	80.70
5 minutes	0.3	0.1	3365.64	58.01	7.65	6926714.88	2631.87	517.53	19383.87	139.23	49.38
5 minutes	0.3	0.2	3560.80	59.67	8.14	7316507.49	2704.90	562.66	20633.33	143.64	52.33
5 minutes	0.3	0.3	3759.95	61.32	8.59	7700735.05	2775.02	597.93	21901.74	147.99	54.93
5 minutes	0.3	0.4	3963.27	62.95	9.02	8090482.93	2844.38	627.24	23182.25	152.26	57.32
5 minutes	0.3	0.5	4170.45	64.58	9.43	8494179.60	2914.48	650.37	24473.45	156.44	59.72
5 minutes	0.3	0.6	4381.07	66.19	9.84	8915800.45	2985.93	676.03	25772.87	160.54	61.94
5 minutes	0.3	0.7	4594.88	67.79	10.18	9355651.45	3058.70	701.32	27074.23	164.54	63.98
5 minutes	0.3	0.8	4811.88	69.37	10.53	9812718.71	3132.53	727.11	28378.02	168.46	66.13
5 minutes	0.3	0.9	5032.23	70.94	10.93	10285898.03	3207.16	757.42	29696.26	172.33	68.26
5 minutes	0.4	0.1	2632.63	51.31	6.51	5385960.81	2320.77	428.38	14897.49	122.06	42.28
5 minutes	0.4	0.2	2795.68	52.87	6.89	5711497.15	2389.87	459.17	15899.28	126.09	44.62
5 minutes	0.4	0.3	2963.06	54.43	7.26	6043885.09	2458.43	482.24	16917.93	130.07	46.88
5 minutes	0.4	0.4	3134.72	55.99	7.60	6389101.40	2527.67	504.80	17951.62	133.98	48.89
5 minutes	0.4	0.5	3310.57	57.54	7.92	6749574.59	2597.99	528.37	19000.50	137.84	50.65
5 minutes	0.4	0.6	3490.68	59.08	8.25	7125368.12	2669.34	552.15	20067.15	141.66	52.56
5 minutes	0.4	0.7	3675.29	60.62	8.51	7515710.91	2741.48	573.53	21157.50	145.46	54.34
5 minutes	0.4	0.8	3864.73	62.17	8.76	7919674.52	2814.19	595.70	22277.85	149.26	56.00
5 minutes	0.4	0.9	4059.38	63.71	9.07	8336162.79	2887.24	621.52	23431.63	153.07	57.90
5 minutes	0.5	0.1	1954.88	44.21	5.42	3986180.97	1996.54	348.35	10885.32	104.33	35.53
5 minutes	0.5	0.2	2084.66	45.66	5.73	4247288.65	2060.90	371.64	11652.85	107.95	37.48
5 minutes	0.5	0.3	2218.89	47.11	6.01	4518322.86	2125.63	389.86	12438.68	111.53	39.10
5 minutes	0.5	0.4	2357.68	48.56	6.28	4801641.13	2191.26	408.78	13243.67	115.08	40.65
5 minutes	0.5	0.5	2501.23	50.01	6.51	5097583.05	2257.78	425.81	14070.24	118.62	42.27
5 minutes	0.5	0.6	2649.85	51.48	6.78	5405578.59	2324.99	446.98	14921.53	122.15	43.90
5 minutes	0.5	0.7	2803.89	52.95	7.03	5724753.47	2392.65	465.73	15799.81	125.70	45.49
5 minutes	0.5	0.8	2963.74	54.44	7.26	6054108.98	2460.51	481.15	16705.49	129.25	46.99
5 minutes	0.5	0.9	3129.76	55.94	7.48	6392652.67	2528.37	496.41	17637.34	132.81	48.43
5 minutes	0.6	0.1	1344.56	36.67	4.34	2735647.04	1653.98	274.41	7372.56	85.86	28.76
5 minutes	0.6	0.2	1440.81	37.96	4.59	2929879.29	1711.69	292.42	7921.81	89.00	30.14
5 minutes	0.6	0.3	1541.33	39.26	4.80	3133437.51	1770.15	306.00	8489.45	92.14	31.47
5 minutes	0.6	0.4	1646.38	40.58	5.02	3347209.34	1829.54	321.09	9077.08	95.27	32.90
5 minutes	0.6	0.5	1756.30	41.91	5.22	3571202.41	1889.76	334.85	9686.64	98.42	34.23
5 minutes	0.6	0.6	1871.47	43.26	5.39	3805175.28	1950.69	346.80	10319.77	101.59	35.49
5 minutes	0.6	0.7	1992.30	44.64	5.58	4048958.98	2012.20	358.58	10977.44	104.77	36.70
5 minutes	0.6	0.8	2119.24	46.04	5.82	4302652.02	2074.28	372.39	11660.38	107.98	37.99
5 minutes	0.6	0.9	2252.74	47.46	6.06	4566780.04	2137.00	387.11	12369.78	111.22	39.36
5 minutes	0.7	0.1	817.59	28.59	3.27	1660905.23	1288.76	203.73	4416.40	66.46	21.87
5 minutes	0.7	0.2	881.15	29.68	3.44	1789161.59	1337.60	216.69	4767.09	69.04	22.89
5 minutes	0.7	0.3	948.34	30.80	3.61	1924904.14	1387.41	227.46	5133.60	71.65	24.00
5 minutes	0.7	0.4	1019.46	31.93	3.76	2068683.66	1438.29	235.67	5517.47	74.28	25.04
5 minutes	0.7	0.5	1094.89	33.09	3.92	2220825.28	1490.24	246.07	5920.35	76.94	26.07
5 minutes	0.7	0.6	1175.06	34.28	4.11	2381756.44	1543.29	256.73	6343.86	79.65	27.14
5 minutes	0.7	0.7	1260.43	35.50	4.29	2552179.72	1597.55	266.51	6789.83	82.40	28.21
5 minutes	0.7	0.8	1351.53	36.76	4.45	2733166.34	1653.23	275.57	7260.53	85.21	29.22
5 minutes	0.7	0.9	1448.98	38.07	4.61	2926192.89	1710.61	283.85	7758.96	88.08	30.21

TABLE 6. *(Continued.)* **Prediction results using the HWDES method for CPU, memory and I/O.**

Time interval			CPU Test			Memory Test			I/O test		
	α	β	MSE	RMSE	MAD	MSE	RMSE	MAD	MSE	RMSE	MAD
5 minutes	0.8	0.1	395.54	19.89	2.19	802757.60	895.97	135.31	2105.40	45.88	14.84
5 minutes	0.8	0.2	429.22	20.72	2.30	870703.66	933.12	143.63	2285.46	47.81	15.54
5 minutes	0.8	0.3	465.34	21.57	2.42	943573.07	971.38	150.34	2476.26	49.76	16.30
5 minutes	0.8	0.4	504.18	22.45	2.55	1021863.12	1010.87	157.24	2679.07	51.76	17.05
5 minutes	0.8	0.5	546.07	23.37	2.67	1106113.90	1051.72	163.86	2895.33	53.81	17.84
5 minutes	0.8	0.6	591.38	24.32	2.79	1197039.87	1094.09	169.55	3126.73	55.92	18.58
5 minutes	0.8	0.7	640.57	25.31	2.89	1295592.39	1138.24	174.26	3375.28	58.10	19.29
5 minutes	0.8	0.8	694.17	26.35	2.99	1402984.78	1184.48	178.36	3643.48	60.36	20.00
5 minutes	0.8	0.9	752.80	27.44	3.11	1520698.14	1233.17	186.34	3934.28	62.72	20.82
5 minutes	0.9	0.1	108.53	10.42	1.10	220176.23	469.23	67.65	569.37	23.86	7.58
5 minutes	0.9	0.2	118.75	10.90	1.17	240824.47	490.74	71.91	622.47	24.95	7.95
5 minutes	0.9	0.3	129.92	11.40	1.24	263362.19	513.19	75.32	679.74	26.07	8.36
5 minutes	0.9	0.4	142.15	11.92	1.30	288058.38	536.71	78.45	741.78	27.24	8.78
5 minutes	0.9	0.5	155.63	12.48	1.36	315241.29	561.46	81.75	809.36	28.45	9.22
5 minutes	0.9	0.6	170.55	13.06	1.45	345331.69	587.65	86.05	883.38	29.72	9.69
5 minutes	0.9	0.7	187.16	13.68	1.54	378863.18	615.52	90.63	964.99	31.06	10.20
5 minutes	0.9	0.8	205.77	14.34	1.63	416501.07	645.37	95.79	1055.60	32.49	10.74
5 minutes	0.9	0.9	226.77	15.06	1.73	459069.49	677.55	101.65	1156.99	34.01	11.32

From the prediction results, we observe that each time interval where the value of α is 0.9 and the value of β is 0.1 gives an optimal prediction result, because the value of MSE, RMSE and MAD values for CPU, memory and I/O are lowest among all other values. Therefore, for optimal value in each time interval, each value of α and β should be 0.9 and 0.1 respectively.

6) QoS PREDICTION ACCURACY USING THE ARIMA METHOD

This subsection determines the prediction accuracy of the ARIMA method to forecast the QoS parameters. The system takes as inputs the order of ARIMA (p), the degree of differencing (d) and the order of MA (q). Due to space limitations, we consider only eight combinations of p, d and q, these being (0,0,0), (0,0,1), (0,1,0), (0,1,1), (1,0,0), (1,0,1), (1,1,0) and (1,1,1) for only two time intervals, 5 and 10 minutes.

However, a comparative analysis of the other time intervals is presented in Table 9. The values for MAD, MSE and RMSE for different arrangements of p, d and q are presented in Table 7.

V. ANALYSING THE SES ALGORITHM BASED ON THE FRESHNESS OF DATA

The second part of this research analyses the prediction accuracy of the SES by considering the freshness of data. Freshness represents the accuracy of results in the initial time periods of prediction as compared to later ones. As mentioned in the literature [7], [81]–[83], the prediction accuracy varies with training dataset. The accuracy of the prediction approaches decreases with an increase in time [7], so the freshness criterion aims to determine which parameter and variable in the prediction approach gives the most accurate results for the initial time slots.

In this section, we present our observations on the best input parameters to use for QoS prediction using the SES method. To achieve this, we divide the dataset into sets of input values that range from 100 to 500 in sets of 1–100, 1–200, 1–300, 1–400 and 1–500. For each input value, we first determine the error value at time slot t1 to t10 and then plot the change in the error value over the time slots as a percentage of deviation with respect to the error value observed at time slot t1. The plot of the error value over the nine time slots (future intervals), taking the inputs of 1–100, 1–200, 1–300, 1–400 and 1–500 for different values of alpha, is shown in Figures 3 to 10.

Figures 9 and 10 show the averaged error over the predicted time slots for datasets 1–400 and 1–500 respectively.

VI. DISCUSSION

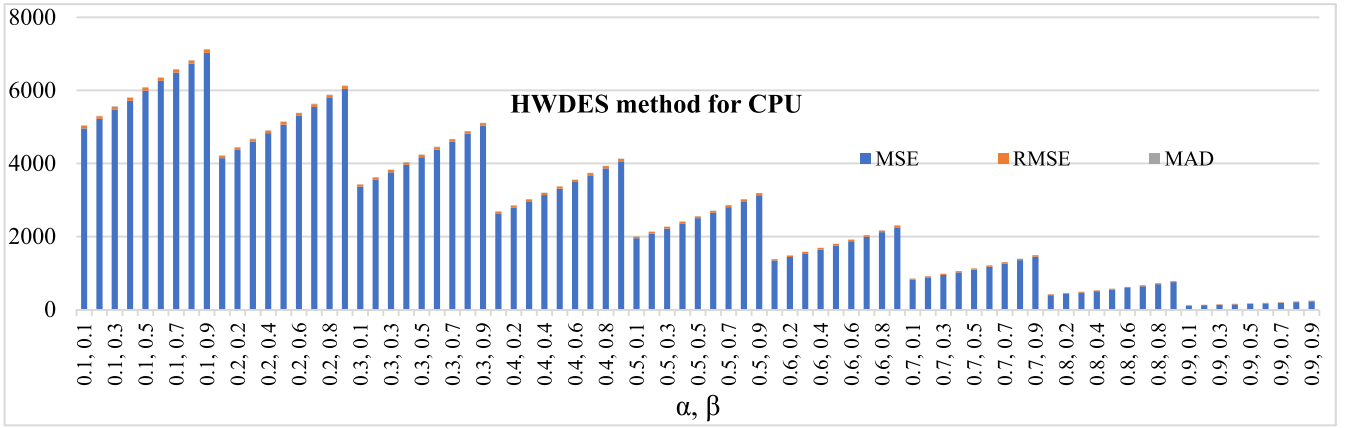
This section presents the results of the above-mentioned prediction algorithms based on two criteria: overall accuracy and freshness of prediction result. The overall accuracy analyses the prediction accuracy of each method with its optimal control parameters determined from earlier experiments. The second area of discussion is the freshness of data and how it impacts prediction accuracy.

A. OVERALL ACCURACY

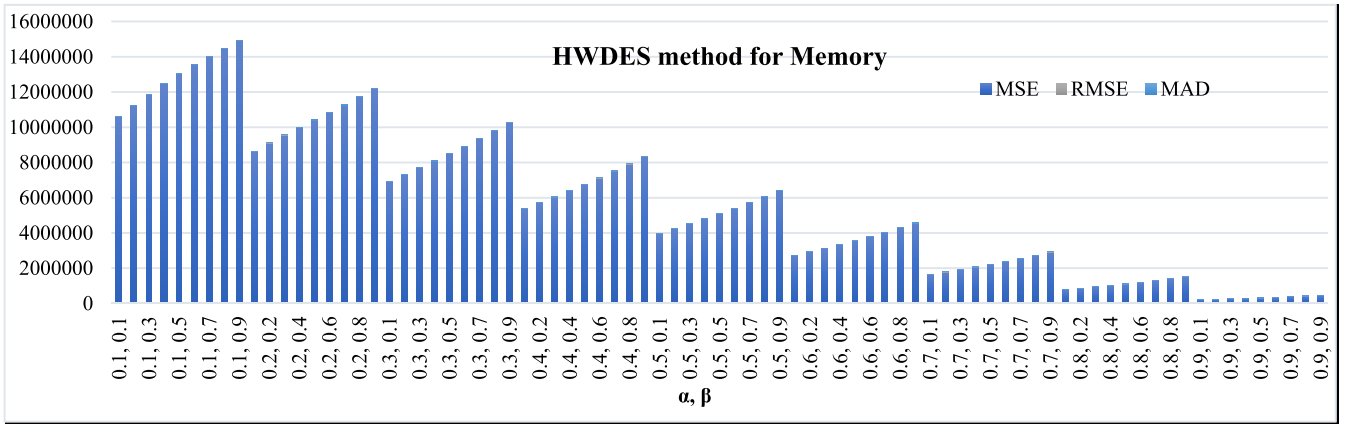
We evaluate and compare the overall accuracy of all discussed approaches with their optimal control parameters on 10×3 different datasets considering three QoS parameters: CPU, memory and I/O. We present following terms for each prediction methods. MT- 1 as SES, MT-2 as SMA, MT-3 as WMA, MT-4 as Exp, MT-5 as HWDES and MT-6 as ARIMA. The comparative analysis is presented in Table 8.

From the table 8 we observe the following findings:

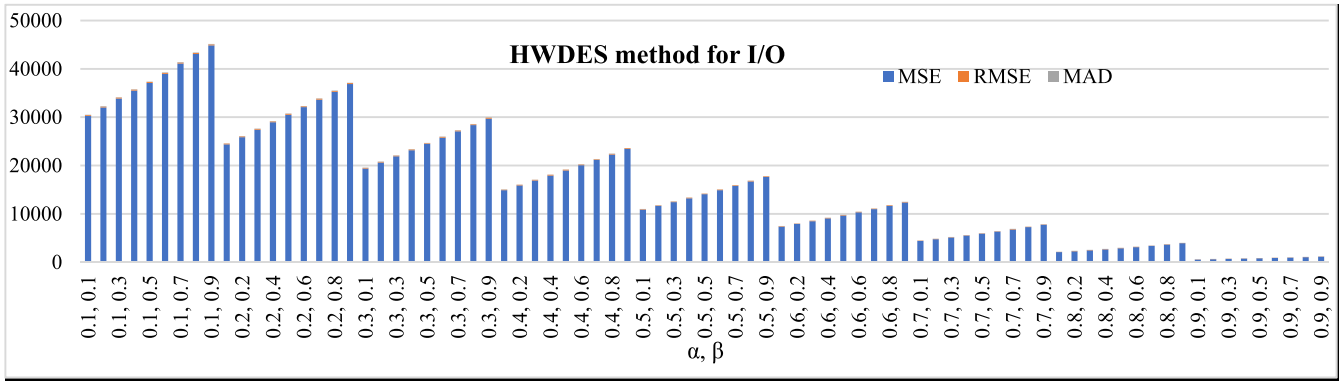
The SES algorithm generates good results for a dataset that does not have any patterns. Furthermore, we see that when a dataset has a seasonality and trend pattern then the accuracy decreases, as can be seen for CPU data for weeks 1, 2 and 4 where their dataset follows a seasonal pattern.



(a)



(b)



(c)

FIGURE 2. (a) HWDES method for CPU. (b) HWDES method for memory. (c) HWDES method for I/O.

We observe that the smoothing factor α impacts prediction accuracy. The sensitivity of prediction accuracy is directly proportional to the value of α. When $\alpha = 1$, it is only lacking one behind the naive forecast, which abruptly changes with a sudden change in the dataset.

Depending on the nature of the dataset, different values of α give optimal results. To analyse the impact of α, we evaluated the nine possible combinations of α ranges from 0.1 to 0.9 for all 10 types of datasets. From the prediction results, we observe that there is no specific value of α which produces an optimal result and for each dataset with its own pattern, different values of α generate optimal results.

a) The SMA generates good results for a dataset that has random variations and the prediction accuracy highly depends on the size of its control parameter k, which is the number of records for the calculating mean. To analyse the impact of parameter k on prediction accuracy, we divide the dataset into 10 subsets and ponder the

TABLE 7. Prediction results using the ARIMA method for CPU, memory and I/O.

Time interval	p	d	q	CPU Test			Memory Test			I/O test		
				MSE	RMSE	MAD	MSE	RMSE	MAD	MSE	RMSE	MAD
5 minutes	0	1	0	0.0100	0.0900	0.0400	62.2100	7.8900	0.8100	0.2300	0.4800	0.1600
5 minutes	0	1	1	5370.0400	73.2800	9.7400	10333890.4900	3214.6400	906.4700	25862.7900	160.8200	59.9200
5 minutes	1	1	1	5572.3800	74.6500	9.4400	9245027.4600	3040.5600	861.6800	19541.2700	139.7900	55.2100
5 minutes	1	0	0	5613.8600	74.9300	9.6200	8985012.1100	2997.5000	671.9700	18292.7300	135.2500	67.7500
5 minutes	0	0	0	5522.2416	74.3118	10.7035	11847222.0683	3441.9794	763.8845	42122.7690	205.2383	99.4288
5 minutes	0	0	1	5607.2026	74.8813	9.8501	9106816.9165	3017.7503	687.9435	22783.2652	150.9413	77.9168
5 minutes	1	1	0	5158.1580	71.8203	5.8434	6638977.5550	2576.6213	319.6355	12295.6266	110.8856	31.9978
5 minutes	1	0	1	5914.8569	76.9081	9.8149	9084603.1499	3014.0675	651.9676	23932.7113	154.7020	57.0296
10 minutes	0	1	0	0.0000	0.0700	0.0300	142.4100	11.9300	1.5900	0.4600	0.6700	0.2600
10 minutes	0	1	1	29.5700	5.4400	3.6700	5864081.5000	2421.5900	908.2400	13333.9700	115.4700	52.1700
10 minutes	1	1	1	22.8900	4.7800	3.2700	5088754.5900	2255.8300	853.1000	10728.2300	103.5800	47.8000
10 minutes	1	0	0	37.4100	6.1200	4.7600	5217467.7600	2284.1800	675.4800	9802.5900	99.0100	57.2700
10 minutes	0	0	0	86.7450	9.3137	7.1046	6717479.6611	2591.8101	763.6912	28367.4402	168.4264	96.5679
10 minutes	0	0	1	52.7754	7.2647	5.6114	5052548.2002	2247.7874	672.6463	14515.7596	120.4814	71.7941
10 minutes	1	1	0	14.3889	3.7933	2.2676	3381301.7161	1838.8316	327.1896	6601.6521	81.2506	27.2436
10 minutes	1	0	1	28.0249	5.2939	3.8809	5145496.2656	2268.3686	659.7672	12604.3650	112.2692	49.9596

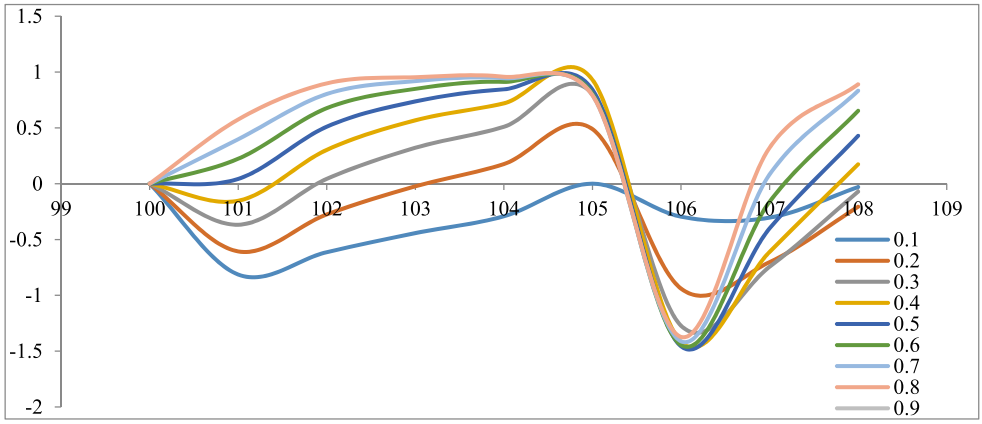


FIGURE 3. Prediction error for predicting nine future intervals (100–108) by taking CPU data with values of 1–100.

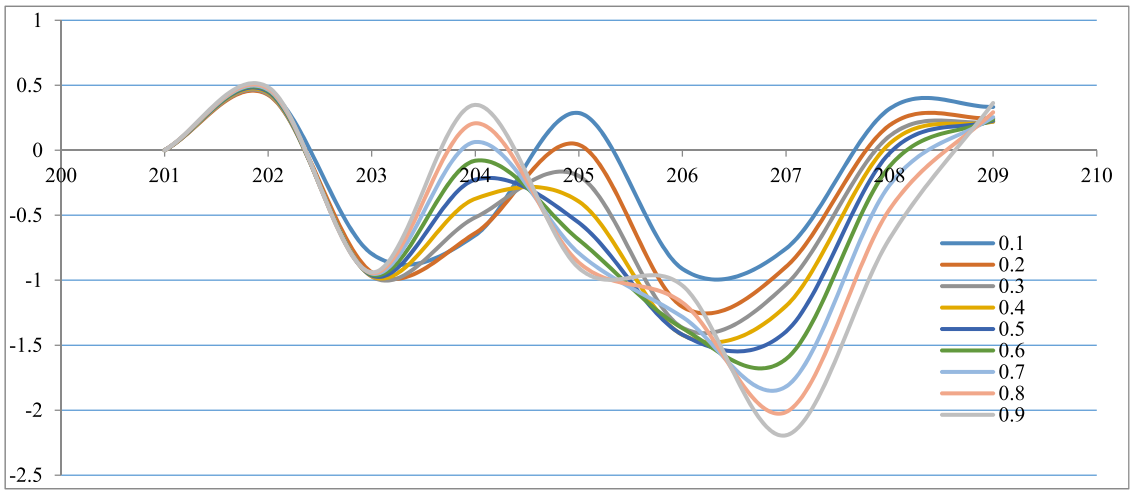


FIGURE 4. Prediction error for predicting nine future intervals (201–209) by taking CPU data with values 1–200.

different value of k subject to the number of records in a dataset. For each time interval, the analysis starts by two record - $k = 2$ and increases at different intervals until the end of a dataset. From the obtained results, we observe that the prediction accuracy is inversely proportional to the size of k because smaller time

TABLE 8. Comparative analysis of the six prediction methods with their optimal control parameters.

		CPU test				Memory test				I/O test			
Time interval	Method	MSE	RMSE	MAD	at	MSE	RMSE	MAD	at	MSE	RMSE	MAD	at
5 minutes	MT-1	5814.78	76.25	9.92	α=0.1	12237648.4	3498.24	694.3	α=0.1	35020	187.14	65.78	α=0.1
	MT-2	3669.63	60.58	6.89	k=2/1919	7432322.47	2726.23	414.9	k=2/1919	20609.1	143.56	47.05	k=2/1919
	MT-3	0.0014	0.04	0.0033	k=1910/1919,α=1.2	0.36	0.6	0.04	k=1910/1919,α=1.2	30	5.48	0.18	k=1910/1919,α=1.2
	MT-4	32468.23	180.19	20.21		65964876.61	8121.88	1186		160781	400.98	132.71	
	MT-5	108.53	10.42	1.1	α=0.9,β=0.1	220176.23	469.23	67.65	α=0.9,β=0.1	569.37	23.86	7.58	α=0.9,β=0.1
	MT-6	0.01	0.09	0.04	p=0,d=1,q=0	62.21	7.89	0.81	p=0,d=1,q=0	0.23	0.48	0.16	p=0,d=1,q=0
10 minutes	MT-1	43.57	6.6	4.57	α=0.2	7008225.09	2647.31	652.5	α=0.6	21917.4	148.05	62	α=0.3
	MT-2	24.57	4.96	3.22	k=2/960	4156421.14	2038.73	447.6	k=2/960	12400.3	111.36	44.03	k=2/960
	MT-3	0.0009	0.0303	0.0026	k=950/960,α=2	0.19	0.44	0.04	k=950/960,α=2	14.5	3.81	0.16	k=950/960,α=2
	MT-4	201.13	14.18	9.18		35255761.54	5937.66	1242		96362.2	310.42	122.31	
	MT-5	0.7	0.83	0.51	α=0.9,β=0.1	119974.63	346.37	71.79	α=0.9,β=0.1	341.76	18.49	6.97	α=0.9,β=0.1
	MT-6	0.0042	0.07	0.03	p=0,d=1,q=0	142.41	11.93	1.59	p=0,d=1,q=0	0.46	0.67	0.26	p=0,d=1,q=0
20 minutes	MT-1	27.15	5.21	3.69	α=0.4	4395116.12	2096.45	707	α=0.1	15148.1	123.08	59.5	α=0.5
	MT-2	14.41	3.8	2.59	k=2/481	2478822.13	1574.43	474.4	k=2/481	7956.57	89.2	40.4	k=2/481
	MT-3	0.45	0.67	0.34	k=240/481,α=10	0.09	0.29	0.02	k=240/481,α=10	12.73	3.57	0.18	k=240/481,α=10
	MT-4	111.68	10.57	7.1		21669336.8	4655.03	1366		70749.7	265.99	119.27	
	MT-5	0.4	0.63	0.42	α=0.9,β=0.1	72892.21	269.99	77.45	α=0.9,β=0.1	235.86	15.36	6.66	α=0.9,β=0.1
	MT-6	0.01	0.07	0.04	p=0,d=1,q=0	294.36	17.16	3.12	p=0,d=1,q=0	1.23	1.11	0.47	p=0,d=1,q=0
1 hour	MT-1	15.39	3.92	2.81	α=0.4	1952440.66	1397.3	763.7	α=0.1	8274.6	90.96	57.91	α=0.7
	MT-2	7.65	2.77	2.01	k=2/161	1268127.97	1126.11	533.7	k=2/161	4087.48	63.93	40.53	k=2/161
	MT-3	0.004	0.0632	0.0081	k=155/161,α=10	497.57	22.31	2.64	k=155/161,α=10	2.05	1.43	0.23	k=155/161,α=10
	MT-4	58.92	7.68	5.74		10551904.78	3248.37	1533		24605.7	156.86	95.45	
	MT-5	0.21	0.46	0.33	α=0.9,β=0.1	36283.73	190.48	85.07	α=0.9,β=0.1	99.15	9.96	6.08	α=0.9,β=0.1
	MT-6	0.01	0.1	0.07	p=0,d=1,q=0	588.84	24.27	6.99	p=0,d=1,q=0	4.62	2.15	1.13	p=0,d=1,q=0
4 hours	MT-1	413.31	20.33	5.5	α=0.2	466636.37	683.11	481.5	α=0.1	78195.8	279.64	99.57	α=0.5
	MT-2	231.35	15.21	3.84	k=2/186	333686.64	577.66	418.2	k=2/186	45471.9	213.24	70.33	k=2/186
	MT-3	0.0087	0.0934	0.0114	k=180/181,α=10	252.55	15.89	2.21	k=180/181,α=10	0.81	0.9	0.15	k=180/181,α=10
	MT-4	2174.31	46.63	10.55		2894397.14	1701.29	1263		408213	638.92	191.24	
	MT-5	7.07	2.66	0.58	α=0.9	9743.62	98.71	68.08	α=0.9,β	1355.6	36.82	10.86	α=0.9,

TABLE 8. *(Continued.)* **Comparative analysis of the six prediction methods with their optimal control parameters.**

		CPU test				Memory test				I/O test			
Time interval	**Method**	**MSE**	**RMSE**	**MAD**	**at**	**MSE**	**RMSE**	**MAD**	**at**	**MSE**	**RMSE**	**MAD**	**at**
					,β=0.1				=0.1	8			β=0.1
	MT-6	0.2	0.45	0.2	p=0,d=1,q=0	138.59	11.77	5.14	p=0,d=1,q=0	16.99	4.12	1.4	p=0,d=1,q=0
12 hours	MT-1	201.68	14.2	6.21	α=0.4	126211.77	355.26	239.25	α=0.1	30755.31	175.37	94.4	α=0.3
	MT-2	93.42	9.67	4.33	k=2/62	60694.92	246.36	184.68	k=2/62	13498.48	116.18	63.74	k=2/62
	MT-3	0.14	0.37	0.07	k=60/62 , α=10	79.68	8.93	1.99	k=60/62 , α=10	0.73	0.85	0.14	k=60/62 , α=10
	MT-4	767.32	27.7	11.26		794249.5	891.21	648.85		139100.21	372.96	168.63	
	MT-5	2.65	1.63	0.68	α=0.9 , β=0.1	2302.08	47.98	34.98	α=0.9, β=0.1	436.95	20.9	10.13	α=0.9, β=0.1
	MT-6	0.36	0.6	0.29	p=0, d=1, q=0	273.75	16.55	10.22	p=0, d=1, q=0	46.28	6.8	2.75	p=0, d=1, q=0
1 day	MT-1	90260.24	300.43	159.65	α=0.1	36542.23	191.16	89.91	α=0.1	20.36	4.51	0.9	α=0.9
	MT-2	63670.95	252.33	145.37	k=2/31	21788.12	147.61	76.81		7.28	2.7	0.67	
	MT-3	12.36	3.52	0.85	k=30/31, α=10	21.62	4.65	1.09	k=30/31 , α=10	1.69E-28	1.30E-14	1.19E-14	k=30/31, α=10
	MT-4	456537.19	675.68	448.61		170563.84	412.99	203.19		20.16	4.49	0.81	
	MT-5	1718.08	41.45	27.15	α=0.9 , β=0.1	602.19	24.54	12.04	α=0.9, β=0.1	0.21	0.46	0.16	α=0.9, β=0.1
	MT-6	348.94	18.68	7.09	p=0, d=1, q=0	135.58	11.64	3.76	p=0, d=1, q=0	0.81	0.9	0.7	p=0, d=1, q=0
1 week	MT-1	2258.07	47.52	22.42	α=0.9	46576.34	215.82	162.57	α=0.5	18485.61	135.96	97.02	α=0.1
	MT-2	1359.15	36.87	17.71	k=2/26	21834.27	147.76	105.73	k=2/26	10647.83	103.19	76.22	k=4/26
	MT-3	0	0.02	0.01	k=26/26, α=10	0.1	0.32	0.07	k=26/26 , α=10	0.56	0.75	0.21	k=26/26, α=10
	MT-4	18698.78	136.74	76.65		184055.43	429.02	334.44		229356.71	478.91	277.84	
	MT-5	92.95	9.64	4.71	α=0.9 , β=0.1	687.54	26.22	18.71	α=0.9, β=0.1	1052.61	32.44	17.37	α=0.9, β=0.1
	MT-6	69.39	8.33	5.87	p=0, d=1, q=0	170.71	13.07	9.26	p=0, d=1, q=0	7.93	2.82	1.84	p=0, d=1, q=0
2 weeks	MT-1	1747.38	41.8	17.86	α=0.9	20191.53	142.1	110.22	α=0.6	7890.94	88.83	69.76	α=0.4
	MT-2	1120.64	33.48	13.81	k=2/105	8325.92	91.25	68.08	k=2/105	2867.03	53.54	42.06	k=2/105
	MT-3	0.01	0.11	0.02	k=104/105 , α=10	2.71	1.65	0.21	k=104/105, α=10	0.04	0.2	0.03	k=104/105, α=10
	MT-4	2298.49	47.94	27.45		68895.15	262.48	213.17		43696.96	209.04	155.42	
	MT-5	17.88	4.23	2.12	α=0.9 , β=0.1	232.72	15.26	11.83	α=0.9, β=0.1	119.22	10.92	8.37	α=0.9, β=0.1
	MT-6	34.89	5.91	4.07	p=0, d=1, q=0	161.28	12.7	8.49	p=0, d=1, q=0	19.03	4.36	2.75	p=0, d=1, q=0
4 weeks	MT-1	3736.51	61.13	25.84	α=0.9	20753.18	144.06	97.09	α=0.9	3900.34	62.45	55.88	α=0.9
	MT-2	2367.97	48.66	22.35	k=2/53	12359.18	111.17	76.19	k=2/53	2312.63	48.09	42.5	k=2/53
	MT-3	0.01	0.09	0.02	k=52/53,	0.27	0.52	0.1	k=52/53 , α=10	2.11	1.45	0.25	k=52/53,

TABLE 8. *(Continued.)* **Comparative analysis of the six prediction methods with their optimal control parameters.**

		CPU test				Memory test				I/O test			
Time interval	Method	MSE	RMSE	MAD	at	MSE	RMSE	MAD	at	MSE	RMSE	MAD	at
					α=10								α=10
	MT-4	3945.95	62.82	33.12		51464.27	226.86	177.88		26003.22	161.26	114.22	
	MT-5	29.53	5.43	2.68	α=0.9, β=0.1	229.07	15.13	11.18	α=0.9, β=0.1	83.1	9.12	7.2	α=0.9, β=0.1
	MT-6	51.16	7.15	4.86	p=0, d=1, q=0	205.64	14.34	9.98	p=0, d=1, q=0	15.78	3.97	2.93	p=0, d=1, q=0
5 minutes	MT-1	5814.78	76.25	9.92	α=0.1	12237648.4	3498.24	694.3	α=0.1	35020	187.14	65.78	α=0.1
	MT-2	3669.63	60.58	6.89	k=2/1919	7432322.47	2726.23	414.9	k=2/1919	20609.1	143.56	47.05	k=2/1919
	MT-3	0.0014	0.04	0.0033	k=1910/1919,α=1.2	0.36	0.6	0.04	k=1910/1919,α=1.2	30	5.48	0.18	k=1910/1919,α=1.2
	MT-4	32468.23	180.19	20.21		65964876.61	8121.88	1186		160781	400.98	132.71	
	MT-5	108.53	10.42	1.1	α=0.9,β=0.1	220176.23	469.23	67.65	α=0.9,β=0.1	569.37	23.86	7.58	α=0.9, β=0.1
	MT-6	0.01	0.09	0.04	p=0,d=1,q=0	62.21	7.89	0.81	p=0,d=1,q=0	0.23	0.48	0.16	p=0,d=1,q=0

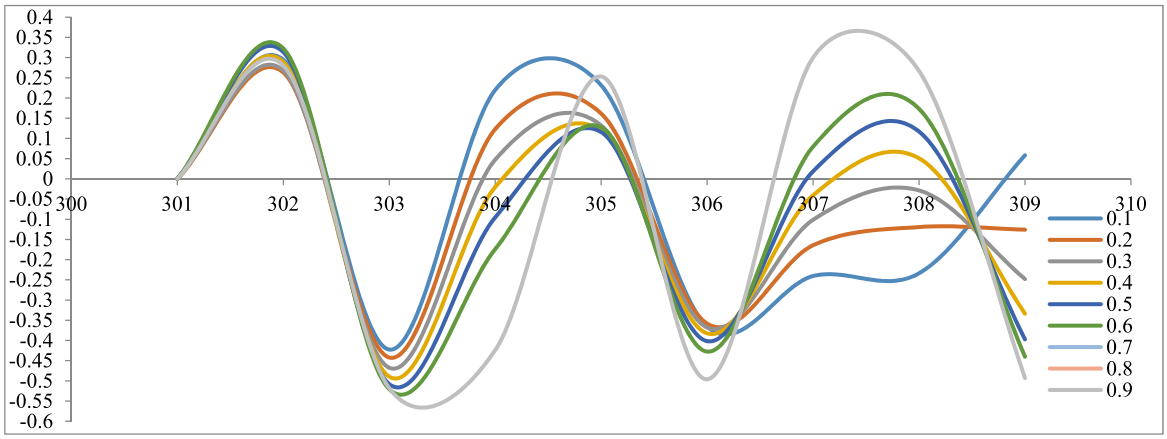


FIGURE 5. Prediction error for predicting nine future intervals (301–309) by taking CPU data with values 1–300.

intervals are more sensitive to prediction accuracy and it alters abruptly compared to longer time intervals when k generates more smooth data. Therefore, the minimum value for k, which is 2, is the most optimal parameter in the SMA algorithm.

b) The prediction accuracy of WMA is analysed by varying two parameters – the number records k and the increasing factor α, which is the difference in weight between recent past and distant past. To achieve this, we take three values of k – initial k = 2/3, mid k = N (total number of records)/2 and final k = N − 1/N − 2 – and a random value of α– 0.5, 1.2, 1.5, 2, 5 and 10. When the value of $\alpha = 0.5$, then it means that the weight of the recent past record is 0.5 times greater than the distant past record, and when the sum of all weights is equal to 1, it means that by increasing the value of α it gives higher weight to the most recent data. From the above result, we observe that the prediction accuracy is directly proportional to the value of k and α, which means that a large dataset and higher weights to the most recent record generate more accurate results. The N/A in a table indicates that the weight factor is smaller than the smallest non-zero floating-point value in MATLAB and it does not generate any output.

c) The extrapolation algorithm generates accurate results on different data patterns. From the above results, we see that a dataset with time intervals of 1 hour,

TABLE 9. Accuracy ranking of prediction algorithms at different time intervals.

Dataset	CPU		Memory		I/O	
	Accuracy rank	Method	Accuracy rank	Method	Accuracy rank	Method
5 minutes	1	MT-3	1	MT-3	1	MT-6
	2	MT-6	2	MT-6	2	MT-3
	3	MT-5	3	MT-5	3	MT-5
	4	MT-2	4	MT-2	4	MT-2
	5	MT-1	5	MT-1	5	MT-1
	6	MT-4	6	MT-4	6	MT-4
10 minutes	1	MT-3	1	MT-3	1	MT-6
	2	MT-6	2	MT-6	2	MT-3
	3	MT-5	3	MT-5	3	MT-5
	4	MT-2	4	MT-2	4	MT-2
	5	MT-1	5	MT-1	5	MT-1
	6	MT-4	6	MT-4	6	MT-4
20 minutes	1	MT-6	1	MT-3	1	MT-6
	2	MT-5	2	MT-6	2	MT-3
	3	MT-3	3	MT-5	3	MT-5
	4	MT-2	4	MT-2	4	MT-2
	5	MT-1	5	MT-1	5	MT-1
	6	MT-4	6	MT-4	6	MT-4
1 hour	1	MT-3	1	MT-3	1	MT-3
	2	MT-6	2	MT-6	2	MT-6
	3	MT-5	3	MT-5	3	MT-5
	4	SMA	4	MT-2	4	MT-2
	5	MT-2	5	MT-1	5	MT-1
	6	MT-4	6	MT-4	6	MT-4
4 hours	1	MT-3	1	MT-6	1	MT-3
	2	MT-6	2	MT-3	2	MT-6
	3	MT-5	3	MT-5	3	MT-5
	4	MT-2	4	MT-2	4	MT-2
	5	MT-1	5	MT-1	5	MT-1
	6	MT-4	6	MT-4	6	MT-4
12 hours	1	MT-3	1	MT-3	1	MT-3
	2	MT-6	2	MT-6	2	MT-6
	3	MT-5	3	MT-5	3	MT-5
	4	MT-2	4	MT-2	4	MT-2
	5	MT-1	5	MT-1	5	MT-1
	6	MT-4	6	MT-4	6	MT-4
1 day	1	MT-3	1	MT-3	1	MT-3
	2	MT-6	2	MT-6	2	MT-5
	3	MT-5	3	MT-5	3	MT-6
	4	MT-2	4	MT-2	4	MT-2
	5	MT-1	5	MT-1	5	MT-1
	6	MT-4	6	MT-4	6	MT-4
1 week	1	MT-3	1	MT-3	1	MT-3
	2	MT-6	2	MT-6	2	MT-6
	3	MT-5	3	MT-5	3	MT-5
	4	MT-2	4	MT-2	4	MT-2
	5	MT-1	5	MT-1	5	MT-1
	6	MT-4	6	MT-4	6	MT-4
2 weeks	1	MT-3	1	MT-3	1	MT-3
	2	MT-5	2	MT-6	2	MT-6
	3	MT-6	3	MT-5	3	MT-5
	4	MT-2	4	MT-2	4	MT-2
	5	MT-1	5	MT-1	5	MT-1
	6	MT-4	6	MT-4	6	MT-4
4 weeks	1	MT-3	1	MT-3	1	MT-3
	2	MT-5	2	MT-6	2	MT-6
	3	MT-6	3	MT-5	3	MT-5
	4	MT-2	4	MT-2	4	MT-2
	5	MT-1	5	MT-1	5	MT-1
	6	MT-4	6	MT-4	6	MT-4

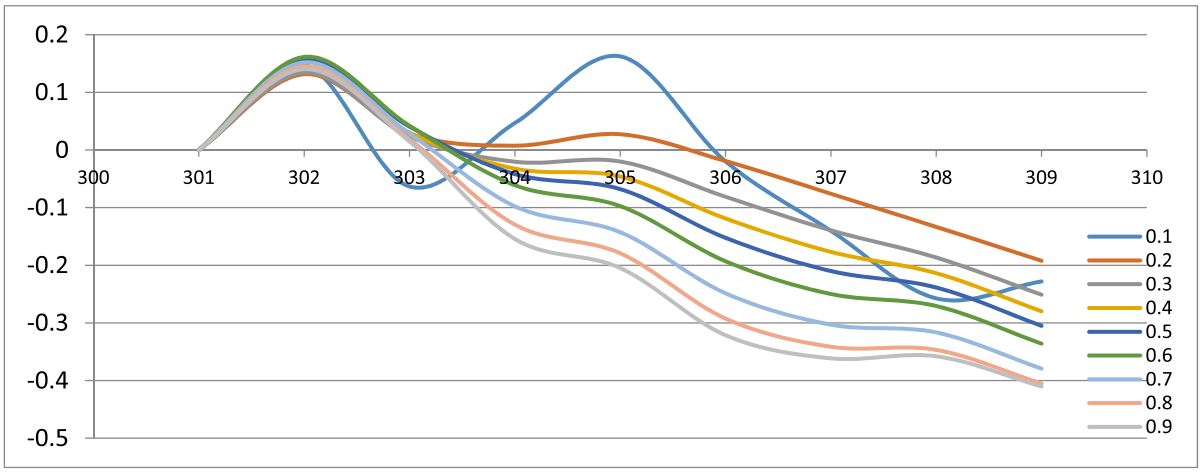


FIGURE 6. Averaged error over the predicted time slots for dataset 1–300.

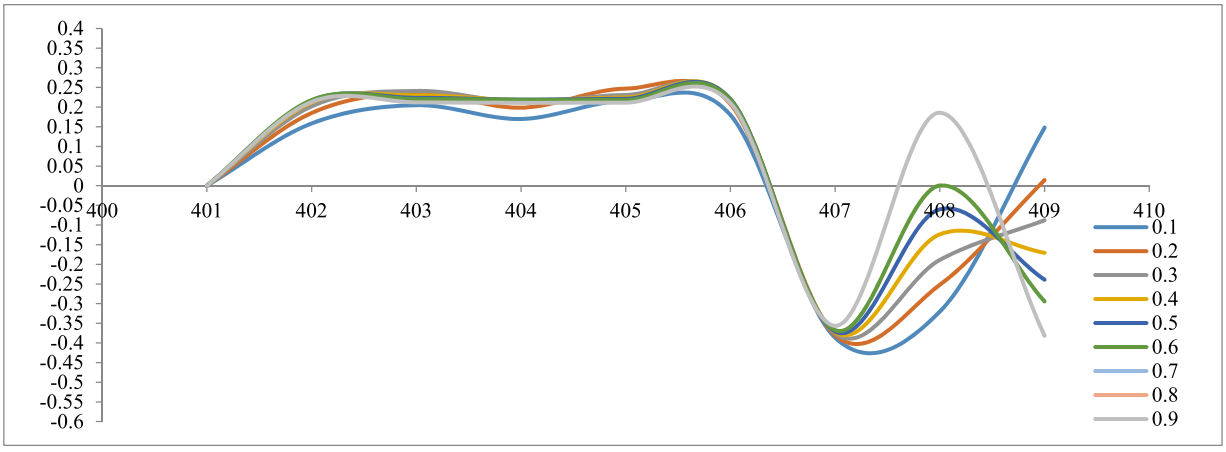


FIGURE 7. Prediction error for predicting nine future intervals (401–409) by taking CPU data with values 1–400.

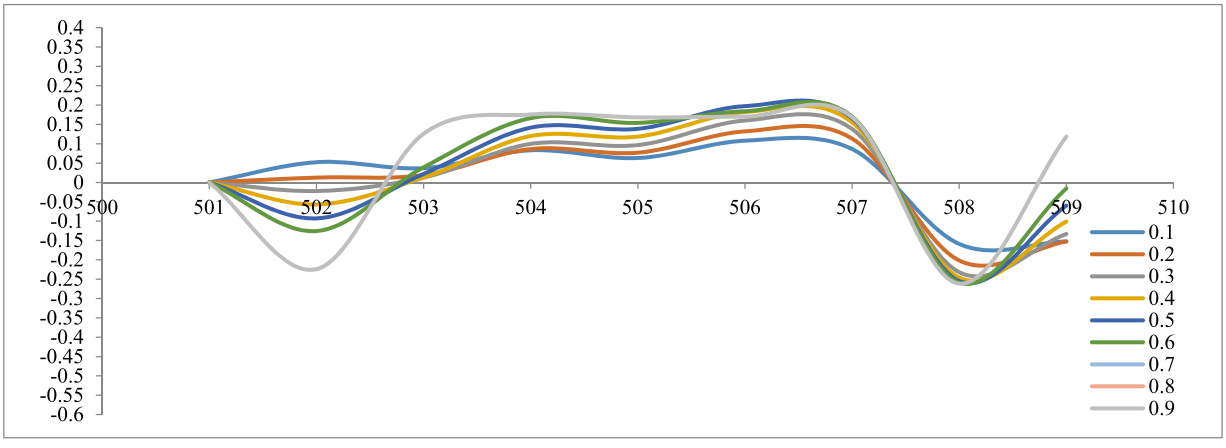


FIGURE 8. Prediction error for predicting nine future intervals (501–509) by taking CPU data with values 1–500 values.

4 weeks and 1 day for CPU, memory and I/ O respectively generates the most accurate results.

d) The prediction accuracy of the Holt-Winter double exponential smoothing algorithm is analysed by varying two parameters: α and β. To achieve this, we analyse these parameters on $9\times9 = 81$ different cases by varying the values of α and β with 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 and 0.9. Table 8 shows the

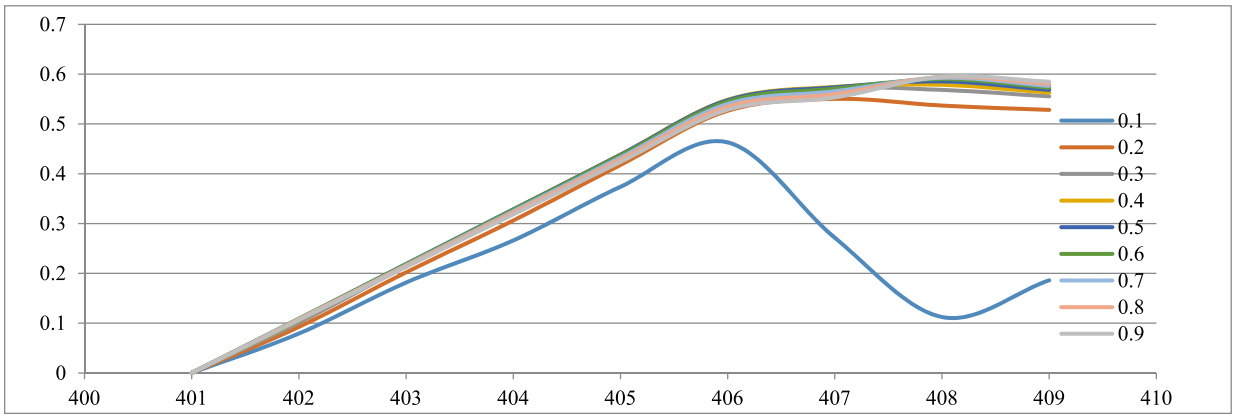


FIGURE 9. Averaged error over the predicted time slot for dataset 1–400.

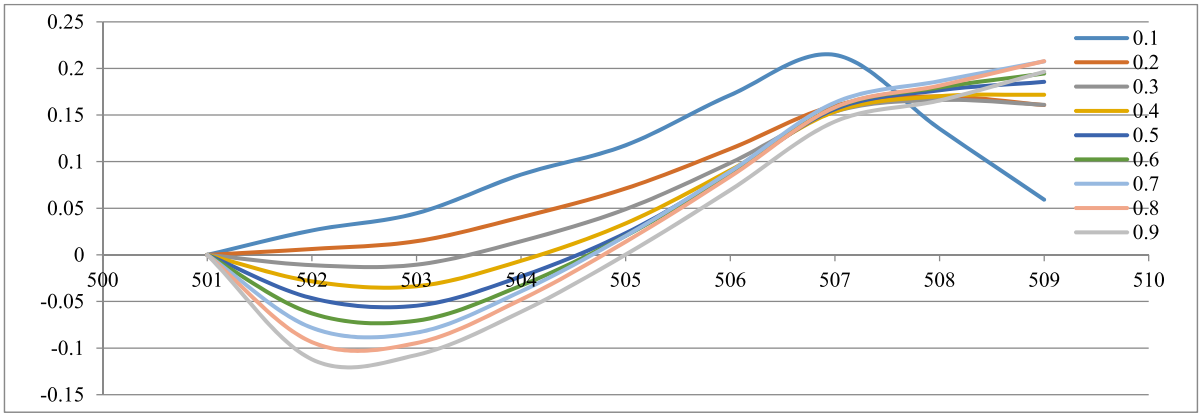


FIGURE 10. Averaged error over the predicted time slot for dataset 1–500.

most accurate results of the value of $\alpha = 0.9$ and $\beta = 0.1$.

e) The last algorithm for our study is ARIMA and for this method we consider three control parameters – the order of ARIMA (p), the degree of differencing (d) and the order of MA (q) – and consider eight sets of these parameters (p, d, q) with the values (0,0,0), (0,0,1),(0,1,0), (0,1,1), (1,0,0), (1,0,1), (1,1,0) and (1,1,1). From the above analysis, we observe that in all cases by setting the value of p = 0, d = 1 and q = 0, i.e. (0,1,0), gives the most accurate result.

f) Using optimal control parameter for each prediction methods we compare the accuracy of six methods at different time intervals and ranked the approaches in ascending order as presented in Table 9. We observed that the Weighted Moving Average method (MT-3) has the best prediction result in 25 out of 30 cases, followed by ARIMA that get best result in 5 out of 30 cases. From the above analysis we determined that WMA and ARIMA methods are the two most accurate prediction methods at different time intervals as presented in Table 10. The first three accuracy ranking are presented in Table 11.

TABLE 10. Most accurate prediction result at 30 different datasets.

Method	Frequency
WMA (MT-3)	25
ARIMA (MT-6)	5

TABLE 11. Accuracy ranking of prediction approaches.

Accuracy Rank	Methods
1	WMA
2	ARIMA
3	HWDES, SMA, SES, EXP

B. THE FRESHNESS OF DATA

When we analyse the SES algorithm by considering the freshness factor, we find out the following observations:

a) From Figure 3, we see that when the deviation in error in time slot t2 exceeds 0.5%, then the value of α which gives the highest positive deviation (that is, shows an improvement in the prediction results) from the error observed in the first time slot results in having the best sustained prediction in future time slots from time slot t1 (100 in the figure).

b) This observation holds true when we consider datasets 1–200 and predict the future QoS values using the SES algorithm, as shown in Figure 4. From this figure, we note that even though the alpha value of 0.9 leads to the highest possible positive deviation, exceeding 0.5%, it goes into the negative region between time slots t2 and t3 but it is the first value to come back in the positive region. Thus, when the freshness of the prediction results is more important, then using the value of alpha which gives the largest deviation, exceeding more than 0.5% in time slot t2, ensures the most optimal prediction results.

c) Figure 5 shows the prediction results over the dataset with inputs 1–300. Unlike the previous two cases, it can be noted that the deviation of change in the error in time slot t2 is less than 0.5, so the observation made earlier does not hold in this case.

d) But, we note that the alpha value that results in the lowest deviation in time slot t2 gives a prediction result that stays in the positive range over the predicted time slots, as shown in Figure 6. The curve for alpha value 0.2 shows the predicted error being in the positive range for the longest period of time over which the prediction is done.

e) From the analysis, we observe that for the input data of 1–400 and 1–500, the prediction results follow the same common pattern. When the input data is non-cyclic and the deviation is not more than 0.5%. The alpha value, which gives the highest positive change and has a sustained increase in the predicted accuracy as the time slots increase, as shown in Figure 7 and Figure 8 which illustrates the deviation of error in time slot t2 with respect to time slot t1 for datasets 1–400 and 1–500 respectively.

VII. CONCLUSION AND FUTURE WORK

An SLA is a key document between a consumer and the service provider that outlines service objectives, business terms, service relations, obligations and the possible actions to be taken in the case of SLA violations. An SLA violation causes penalties in terms of money, service credit or loss of reputation. To avoid SLA violations, the service provider should have an optimal SLA management framework that intelligently predicts discrepancies in SLO and QoS parameters and, in the case of violation detection, alerts the service provider to take appropriate action before the actual violation occurs.

In conclusion, we analysed the prediction accuracy of six widely used prediction methods based on overall accuracy and the freshness of data. We examined the control parameters of each approach and examined how a variation in control parameters impacts the prediction accuracy. Our analysis allows the cloud provider to identify any discrepancies in SLOs and manage the SLA optimally.

In future work, we will evaluate our approach in a Cloud of Thing environment where a requested service is composed of a variety of services from different regions and it's very important for a service provider to choose optimal prediction method that generates accurate future QoS parameters to manage services ideally. We will analyse that using discussed prediction methods, how the cloud provider can better manage its resources in cloud-of-thing environment.

REFERENCES

[1] *Gartner Forecasts Worldwide Public Cloud Revenue to Grow 17.5 Percent in 2019*, Gartner, Stratford, CT, USA, 2019.

[2] (2019). *Amazon Compute Service Level Agreement 2019*. Accessed: Apr. 22, 2019. [Online]. Available: https://aws.amazon.com/compute/sla/

[3] *IBM Cloud Service Description*, IBM, Armonk, NY, USA, 2019, p. 6.

[4] M. Azure. (2019). *SLA Summary for Azure Services*. Accessed: Apr. 22, 2019. [Online]. Available: https://azure.microsoft.com/en-au/support/legal/sla/summary/

[5] D. Waters, *Supply Chain Risk Management: Vulnerability and Resilience in Logistics*. London, U.K.: Kogan Page, 2011.

[6] W. Hussain, F. K. Hussain, M. Saberi, O. K. Hussain, and E. Chang, "Comparing time series with machine learning-based prediction approaches for violation management in cloud SLAs," *Future Gener. Comput. Syst.*, vol. 89, pp. 464–477, Dec. 2018.

[7] W. Hussain, F. Hussain, and O. Hussain, "QoS prediction methods to avoid SLA violation in post-interaction time phase," in *Proc. IEEE 11th Conf. Ind. Electron. Appl. (ICIEA)*, Hefei, China, Jun. 2016, pp. 32–37.

[8] D. Chaudhuri, M. Mukherjee, M. H. Khondekar, and K. Ghosh, "Simple exponential smoothing and its control parameter: A reassessment," in *Recent Trends in Signal and Image Processing* (Advances in Intelligent Systems and Computing), vol. 922, S. Bhattacharyya, S. Pal, I. Pan, and A. Das, Eds. Singapore: Springer, 2019, pp. 63–77.

[9] S. Islam, J. Keung, K. Lee, and A. Liu, "Empirical prediction models for adaptive resource provisioning in the cloud," *Future Generat. Comput. Syst.*, vol. 28, no. 1, pp. 155–162, 2012.

[10] S. Li, J. Wen, F. Luo, and G. Ranzi, "Time-aware QoS prediction for cloud service recommendation based on matrix factorization," *IEEE Access*, vol. 6, pp. 77716–77724, 2018.

[11] A. Bestavros and O. Krieger, "Toward an open cloud marketplace: Vision and first steps," *IEEE Internet Comput.*, vol. 18, no. 1, pp. 72–77, Jan. 2014.

[12] C. Krintz, "The AppScale cloud platform: Enabling portable, scalable Web application deployment," *IEEE Internet Comput.*, vol. 17, no. 2, pp. 72–75, Mar. 2013.

[13] Z. Ye, S. Mistry, A. Bouguettaya, and H. Dong, "Long-term QoS-aware cloud service composition using multivariate time series analysis," *IEEE Trans. Services Comput.*, vol. 9, no. 3, pp. 382–393, May/Jun. 2016.

[14] C. Lee, C. Wang, E. Kim, and S. Helal, "Blueprint flow: A declarative service composition framework for cloud applications," *IEEE Access*, vol. 5, pp. 17634–17643, 2017.

[15] I. A. Ridhawi, Y. Kotb, and Y. A. Ridhawi, "Workflow-net based service composition using mobile edge nodes," *IEEE Access*, vol. 5, pp. 23719–23735, 2017.

[16] S. Kumar, M. K. Pandey, A. Nath, and K. Subbiah, "Performance analysis of ensemble supervised machine learning algorithms for missing value imputation," in *Proc. 2nd Int. Conf. Comput. Intell. Netw. (CINE)*, Jan. 2016, pp. 160–165.

[17] J. Xu, Z. Zheng, Z. Fan, and W. Liu, "Online personalized QoS prediction approach for cloud services," in *Proc. 4th Int. Conf. Cloud Comput. Intell. Syst. (CCIS)*, Aug. 2016, pp. 32–37.

[18] Y. Xu, J. Yin, W. Lo, and Z. Wu, "Personalized location-aware QoS prediction for Web services using probabilistic matrix factorization," in *Web Information Systems Engineering—WISE*. Berlin, Germany: Springer, 2013.

[19] X. Kong, F. Xia, J. Wang, A. Rahim, and S. K. Das, "Time-location-relationship combined service recommendation based on taxi trajectory data," *IEEE Trans. Ind. Informat.*, vol. 13, no. 3, pp. 1202–1212, Jun. 2017.

[20] X. Zheng, L. Da Xu, and S. Chai, "Ranking-based cloud service recommendation," in *Proc. IEEE Int. Conf. Edge Comput. (EDGE)*, Jun. 2017, pp. 136–141.

[21] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "QoS-aware Web service recommendation by collaborative filtering," *IEEE Trans. Services Comput.*, vol. 4, no. 2, pp. 140–152, Apr./Jun. 2011.

[22] W. Ma, R. Shan, and M. Qi, "General collaborative filtering for Web service QoS prediction," *Math. Problems Eng.*, vol. 2018, Dec. 2018, Art. no. 5787406.

[23] J. Liu and Y. Chen, "A personalized clustering-based and reliable trust-aware QoS prediction approach for cloud service recommendation in cloud manufacturing," *Knowl.-Based Syst.*, vol. 174, pp. 43–56, Jun. 2019.

[24] G. Guo, J. Zhang, and D. Thalmann, "Merging trust in collaborative filtering to alleviate data sparsity and cold start," *Knowl.-Based Syst.*, vol. 57, pp. 57–68, Feb. 2014.

[25] C. Park, D. Kim, J. Oh, and H. Yu, "Improving top-K recommendation with truster and trustee relationship in user trust network," *Inf. Sci.*, vol. 374, pp. 100–114, Dec. 2016.

[26] H. Wu, K. Yue, B. Li, B. Zhang, and C.-H. Hsu, "Collaborative QoS prediction with context-sensitive matrix factorization," *Future Gener. Comput. Syst.*, vol. 82, pp. 669–678, May 2018.

[27] Y. Zhang, Z. Zheng, and M. R. Lyu, "WSPred: A time-aware personalized QoS prediction framework for Web services," in *Proc. IEEE 22nd Int. Symp. Softw. Rel. Eng. (ISSRE)*, Nov./Dec. 2011, pp. 210–219.

[28] W. Zhang, "Temporal QoS-aware Web service recommendation via non-negative tensor factorization," in *Proc. 23rd Int. Conf. World Wide Web*, Seoul, South Korea, 2014, pp. 585–596.

[29] W. Lo, J. Yin, S. Deng, Y. Li, and Z. Wu, "An extended matrix factorization approach for QoS prediction in service selection," in *Proc. IEEE 9th Int. Conf. Services Comput.*, Jun. 2012, pp. 162–169.

[30] S. Pacheco-Sanchez, G. Casale, B. Scotney, S. McClean, G. Parr, and S. Dawson, "Markovian workload characterization for QoS prediction in the cloud," in *Proc. IEEE 4th Int. Conf. Cloud Comput.*, Jul. 2011, pp. 147–154.

[31] P. Leitner, A. Michlmayr, F. Rosenberg, and S. Dustdar, "Monitoring, prediction and prevention of SLA violations in composite services," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jul. 2010, pp. 369–376.

[32] P. Leitner, B. Wetzstein, F. Rosenberg, A. Michlmayr, S. Dustdar, and F. Leymann, "Runtime prediction of service level agreement violations for composite services," in *Proc. Service-Oriented Comput., ICSOC/ServiceWave Workshops*. Springer, 2010, pp. 176–186.

[33] S. S. Yau, N. Ye, H. Sarjoughian, and D. Huang, "Developing service-based software systems with QoS monitoring and adaptation," in *Proc. IEEE 12th Int. Workshop Future Trends Distrib. Comput. Syst.*, Oct. 2008, pp. 74–80.

[34] V. Cardellini, E. Casalicchio, V. Grassi, F. L. Presti, and R. Mirandola, "Qos-driven runtime adaptation of service oriented architectures," in *Proc. 7th Joint Meeting Eur. Softw. Eng. Conf. ACM Sigsoft Symp. Found. Softw. Eng.*, Amsterdam, The Netherlands, 2009, pp. 131–140.

[35] S. Gallotti, C. Ghezzi, R. Mirandola, and G. Tamburrelli, "Quality prediction of service compositions through probabilistic model checking," in *Quality of Software Architectures. Models and Architectures*. Berlin, Germany: Springer, 2008.

[36] H. Wu, J. He, B. Li, and Y. Pei, "Personalized QoS prediction of cloud services via learning neighborhood-based model," 2015, *arXiv:1508.04537*. [Online]. Available: https://arxiv.org/abs/1508.04537

[37] L. Romano, D. De Mari, Z. Jerzak, and C. Fetzer, "A novel approach to QoS monitoring in the cloud," in *Proc. 1st Int. Conf. Data Compress., Commun. Process. (CCP)*, Jun. 2011, pp. 45–51.

[38] G. Cicotti, L. Coppolino, S. D'Antonio, and L. Romano, "How to monitor QoS in cloud infrastructures: The QoSMONaaS approach," *Int. J. Comput. Sci. Eng.*, vol. 11, no. 1, pp. 29–45, 2015.

[39] Z. U. Rehman, O. K. Hussain, F. K. Hussain, E. Chang, and T. Dillon, "User-side QoS forecasting and management of cloud services," *World Wide Web*, vol. 18, no. 6, pp. 1677–1716, 2015.

[40] A. Chaudhuri, S. Maity, and S. K. Ghosh, "QoS prediction for network data traffic using hierarchical modified regularized least squares rough support vector regression," in *Proc. 30th Annu. ACM Symp. Appl. Comput.*, 2015, pp. 659–661.

[41] W. Lo, J. Yin, Y. Li, and Z. Wu, "Efficient Web service QoS prediction using local neighborhood matrix factorization," *Eng. Appl. Artif. Intell.*, vol. 38, pp. 14–23, Feb. 2015.

[42] K. Qi, H. Hu, W. Song, J. Ge, and J. Lü , "Personalized QoS prediction via matrix factorization integrated with neighborhood information," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, Jun./Jul. 2015, pp. 186–193.

[43] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "WSRec: A collaborative filtering based Web service recommender system," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jul. 2009, pp. 437–444.

[44] H. Sun, Z. Zheng, J. Chen, and M. R. Lyu, "Personalized Web service recommendation via normal recovery collaborative filtering," *IEEE Trans. Services Comput.*, vol. 6, no. 4, pp. 573–579, Oct./Dec. 2013.

[45] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, "Personalized QoS prediction for Web services via collaborative filtering," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jul. 2007, pp. 439–446.

[46] S. Bisgaard and M. Kulahci, "Quality quandaries*: Time series model selection and parsimony," *Qual. Eng.*, vol. 21, no. 3, pp. 341–353, 2009.

[47] N. R. Herbst, N. Huber, S. Kounev, and E. Amrehn, "Self-adaptive workload classification and forecasting for proactive resource provisioning," *Concurrency, Comput., Pract. Exper.*, vol. 26, no. 12, pp. 2053–2078, 2014.

[48] Y. Song, L. Hu, and M. Yu, "A novel QoS-aware prediction approach for dynamic Web services," *PLoS ONE*, vol. 13, no. 8, 2018, Art. no. e0202669.

[49] W. Hussain, F. K. Hussain, and O. K. Hussain, "Comparative analysis of consumer profile-based methods to predict SLA violation," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Aug. 2015, pp. 1–8.

[50] O. K. Hussain, Z. Rahman, F. K. Hussain, J. Singh, N. K. Janjua, and E. Chang, "A user-based early warning service management framework in cloud computing," *Comput. J.*, vol. 58, no. 3, pp. 472–496, Mar. 2014.

[51] A. Amin, A. Colman, and L. Grunske, "An approach to forecasting QoS attributes of Web services based on ARIMA and GARCH models," in *Proc. IEEE 19th Int. Conf. Web Services (ICWS)*, Jun. 2012, pp. 74–81.

[52] X. Ren, R. Lin, and H. Zou, "A dynamic load balancing strategy for cloud computing platform based on exponential smoothing forecast," in *Proc. IEEE Int. Conf. Cloud Comput. Intell. Syst. (CCIS)*, Sep. 2011, pp. 220–224.

[53] S. Chowhan, S. Shirwaikar, and A. Kumar, "Predictive modeling of service level agreement parameters for cloud services," *Int. J. Next-Gener. Comput.*, vol. 7, no. 2, pp. 115–129, 2016.

[54] K. P. Tran, P. Castagliola, G. Celano, and M. B. C. Khoo, "Monitoring compositional data using multivariate exponentially weighted moving average scheme," *Qual. Rel. Eng. Int.*, vol. 34, no. 3, pp. 391–402, 2018.

[55] H. D. Nguyen, K. P. Tran, and C. Heuchenne, "Monitoring the ratio of two normal variables using variable sampling interval exponentially weighted moving average control charts," *Qual. Rel. Eng. Int.*, vol. 35, no. 1, pp. 439–460, 2019.

[56] G. Sbrana and A. Silvestrini, "Random switching exponential smoothing: A new estimation approach," *Int. J. Prod. Econ.*, vol. 211, pp. 211–220, May 2019.

[57] S. Fatima, S. S. Ali, S. S. Zia, E. Hussain, T. R. Fraz, and M. S. Khan, "Forecasting carbon dioxide emission of Asian countries using ARIMA and simple exponential smoothing models," *Int. J. Econ. Environ. Geol.*, vol. 10, no. 1, pp. 64–69, 2019.

[58] S. Ding, Y. Li, D. Wu, Y. Zhang, and S. Yang, "Time-aware cloud service recommendation using similarity-enhanced collaborative filtering and ARIMA model," *Decis. Support Syst.*, vol. 107, pp. 103–115, Mar. 2018.

[59] Y. Wang, C. Wang, C. Shi, and B. Xiao, "Short-term cloud coverage prediction using the ARIMA time series model," *Remote Sens. Lett.*, vol. 9, no. 3, pp. 274–283, 2018.

[60] J. K. Ord, A. B. Koehler, and R. D. Snyder, "Estimation and prediction for a class of dynamic nonlinear statistical models," *J. Amer. Stat. Assoc.*, vol. 92, no. 440, pp. 1621–1629, 1997.

[61] R. J. Hyndman, A. B. Koehler, R. D. Snyder, and S. Grose, "A state space framework for automatic forecasting using exponential smoothing methods," *Int. J. Forecasting*, vol. 18, no. 3, pp. 439–454, Jul./Sep. 2000.

[62] R. J. Hyndman and A. V. Kostenko, "Minimum sample size requirements for seasonal forecasting models," *Foresight*, vol. 6, pp. 12–15, Jun. 2007.

[63] E. S. Gardner, Jr., "Exponential smoothing: The state of the art," *J. Forecasting*, vol. 4, no. 1, pp. 1–28, 1985.

[64] R. J. Hyndman and Y. Khandakar, "Automatic time series for forecasting: The forecast package for R," Dept. Econ. Bus. Statist., Monash Univ., Melbourne, VIC, USA, 2007.

[65] R. G. Brown, *Statistical Forecasting for Inventory Control*. New York, NY, USA: McGraw-Hill, 1959.

[66] J. D. Camm, J. J. Cochran, M. J. Fry, J. W. Ohlmann, and D. R. Anderson, *Essentials of Business Analytics*, 1st ed. Boston, MA, USA: Cengage, 2014, p. 696.

[67] C. A. Ellis and S. A. Parbery, "Is smarter better? A comparison of adaptive, and simple moving average trading strategies," *Res. Int. Bus. Finance*, vol. 19, no. 3, pp. 399–411, 2005.

[68] J. M. Lucas and M. S. Saccucci, "Exponentially weighted moving average control schemes: Properties and enhancements," *Technometrics*, vol. 32, no. 1, pp. 1–12, 1990.
[69] P. R. Winters, "Forecasting sales by exponentially weighted moving averages," *Manage. Sci.*, vol. 6, no. 3, pp. 324–342, 1960.
[70] P. S. Kalekar, "Time series forecasting using holt-winters exponential smoothing," *Kanwal Rekhi School Inf. Technol.*, vol. 4329008, pp. 1–13, Dec. 2004.
[71] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, vol. 734. Hoboken, NJ, USA: Wiley. 2011
[72] E. R. Ziegel, "Forecasting and time series: An applied approach," *Technometrics*, vol. 36, no. 4, p. 434, 1994.
[73] J. S. Armstrong, "Extrapolation for time-series and cross-sectional data," in *Principles of Forecasting* (International Series in Operations Research & Management Science), vol. 30, J. S. Armstrong, Ed. Boston, MA, USA: Springer, 2001, pp. 217–243.
[74] D. E. Myers, "Spatial interpolation: An overview," *Geoderma*, vol. 62, no. 1, pp. 17–28, 1994.
[75] S. Makridakis, "Accuracy measures: Theoretical and practical concerns," *Int. J. Forecasting*, vol. 9, no. 4, pp. 527–529, 1993.
[76] CloudClimate. *Watching the Cloud*. Accessed: Mar. 3, 2019. [Online]. Available: http://www.cloudclimate.com
[77] P. N. Monitor. *PRTG Network Monitor*. Accessed: Mar. 4, 2019. [Online]. Available: https://prtg.paessler.com
[78] S. Chopra and P. Meindl, "Supply chain management. Strategy, planning & operation," in *Das Summa Summarum des Management*, C. Boersch and R. Elschen, Eds. Springer, 2007, pp. 265–275.
[79] R. G. Schroeder, M. J. Rungtusanatham, and S. M. Goldstein, *Operations Management in the Supply Chain: Decisions and Cases*, 6th ed. New York, NY, USA: McGraw-Hill, 2013.
[80] J. H. Heizer, B. Render, and H. J. Weiss, *Operations Management*, vol. 8. Upper Saddle River, NJ, USA: Prentice-Hall, 2004.
[81] D. R. B. Stockwell and A. T. Peterson, "Effects of sample size on accuracy of species distribution models," *Ecolog. Model.*, vol. 148, no. 1, pp. 1–13, 2002.
[82] J. Cho, K. Lee, E. Shin, G. Choy, and S. Do, "How much data is needed to train a medical image deep learning system to achieve necessary high accuracy?" 2015, *arXiv:1511.06348*. [Online]. Available: https://arxiv.org/abs/1511.06348
[83] M. Johnson and D. Q. Nguyen. *How Much Data is Enough? Predicting How Accuracy Varies With Training Data Size*. Accessed: May 5, 2019. [Online]. Available: http://web.science.mq.edu.au/~mjohnson/papers/Johnson17Power-talk.pdf

WALAYAT HUSSAIN received the Ph.D. degree from the University of Technology Sydney. He was a Lecturer and an Assistant Professor with BUITEMS for many years. He is currently a Lecturer with the Faculty of Engineering and IT, University of Technology Sydney, Australia. He published in various top-ranked reputable journals and conferences such as the *Computer Journal*, *Information Systems*, IEEE Access, *Future Generation Computer Systems*, *Computers & Industrial Engineering*, *Mobile Networks and Applications*, the *Journal of Ambient Intelligence and Humanized Computing*, FUZZ-IEEE, and ICONIP. His research interests include business intelligence, cloud computing, and usability engineering by focusing on providing an informed decision to different stakeholders. He was a recipient of three international and one national research awards and recognitions till date from his research. He was also a recipient of 2016 FEIT HDR Publication Award by the University of Technology Sydney.

OSAMA SOHAIB received the Ph.D. degree in information systems from the University of Technology Sydney (UTS), in 2015, where he is currently a Lecturer with the School of Information, Systems and Modelling, Faculty of Engineering and Information Technology. His work has published in various reputable journals, such as *Computers & Industrial Engineering*, IEEE Access, *Mobile Networks and Applications*, the *International Journal of Disaster Risk Reduction*, the *Journal of Ambient Intelligence and Humanized Computing*, the *Journal of Global Information Management, and Sustainability*. His research interests include decision-making, e-services, HCI, and survey methods.