Comparing time series with machine learning-based prediction approaches for violation management in cloud SLAs

Hussain, Walayat, Hussain, Farookh Khadeer, Saberi, Morteza, Hussain, Omar Khadeer and Chang, Elizabeth

Hussain, W., Hussain, F. K., Saberi, M., Hussain, O. K. and Chang, E. (2018). Comparing time series with machine learning-based prediction approaches for violation management in cloud SLAs. *Future Generation Computer Systems, 89*, pp. 464-477. https://doi.org/10.1016/j.future.2018.06.041

Comparing time series with machine learning-based prediction approaches for violation management in Cloud SLAs

WALAYAT HUSSAIN, School of Systems, Management and Leadership, Faculty of Engineering and Information Technology, University of Technology Sydney, NSW, Australia
FAROOKH KHADEER HUSSAIN, Centre for Artificial Intelligence, School of Software, Faculty of Engineering and Information Technology, University of Technology Sydney, NSW, Australia
MORTEZA SABERI, School of Business, University of New South Wales, Canberra, Australia
OMAR KHADEER HUSSAIN, School of Business, University of New South Wales, Canberra, Australia
ELIZABETH CHANG, School of Business, University of New South Wales, Canberra, Australia

Corresponding Authors:

Walayat Hussain: WALAYAT.HUSSAIN@UTS.EDU.AU
Farookh Khadeer Hussain: FAROOKH.HUSSAIN@UTS.EDU.AU

Abstract:

In cloud computing, service level agreements (SLAs) are legal agreements between a service provider and consumer that contain a list of obligations and commitments which need to be satisfied by both parties during the transaction. From a service provider's perspective, a violation of such a commitment leads to penalties in terms of money and reputation and thus has to be effectively managed. In the literature, this problem has been studied under the domain of cloud service management. One aspect required to manage cloud services after the formation of SLAs is to predict the future Quality of Service (QoS) of cloud parameters to ascertain if they lead to violations. Various approaches in the literature perform this task using different prediction approaches however none of them study the accuracy of each. However, it is important to do this as the results of each prediction approach vary according to the pattern of the input data and selecting an incorrect choice of a prediction algorithm could lead to service violation and penalties. In this paper, we test and report the accuracy of time series and machine learning-based prediction approaches. In each category, we test many different techniques and rank them according to their order of accuracy in predicting future QoS. Our analysis helps the cloud service provider to choose an appropriate prediction approach (whether time series or machine learning based) and further to utilize the best method depending on input data patterns to obtain an accurate prediction result and better manage their SLAs to avoid violation penalties.

Keywords:

Cloud computing, SLA monitoring, QoS prediction methods, Machine learning prediction algorithms, Time series prediction approaches, SLA management, prediction accuracy, cloud service provider

Comparing time series with machine learning-based prediction approaches for violation management in Cloud SLAs

WALAYAT HUSSAIN, School of Systems, Management and Leadership, Faculty of Engineering and Information Technology, University of Technology Sydney, NSW, Australia
FAROOKH KHADEER HUSSAIN, Centre for Artificial Intelligence, School of Software, Faculty of Engineering and Information Technology, University of Technology Sydney, NSW, Australia
MORTEZA SABERI, School of Business, University of New South Wales, Canberra, Australia
OMAR KHADEER HUSSAIN, School of Business, University of New South Wales, Canberra, Australia
ELIZABETH CHANG, School of Business, University of New South Wales, Canberra, Australia

Abstract: In cloud computing, service level agreements (SLAs) are legal agreements between a service provider and consumer that contain a list of obligations and commitments which need to be satisfied by both parties during the transaction. From a service provider's perspective, a violation of such a commitment leads to penalties in terms of money and reputation and thus has to be effectively managed. In the literature, this problem has been studied under the domain of cloud service management. One aspect required to manage cloud services after the formation of SLAs is to predict the future Quality of Service (QoS) of cloud parameters to ascertain if they lead to violations. Various approaches in the literature perform this task using different prediction approaches however none of them study the accuracy of each. However, it is important to do this as the results of each prediction approach vary according to the pattern of the input data and selecting an incorrect choice of a prediction algorithm could lead to service violation and penalties. In this paper, we test and report the accuracy of time series and machine learning-based prediction approaches. In each category, we test many different techniques and rank them according to their order of accuracy in predicting future QoS. Our analysis helps the cloud service provider to choose an appropriate prediction approach (whether time series or machine learning based) and further to utilize the best method depending on input data patterns to obtain an accurate prediction result and better manage their SLAs to avoid violation penalties.

Key Words:
Cloud computing, SLA monitoring, QoS prediction methods, Machine learning prediction algorithms, Time series prediction approaches, SLA management, prediction accuracy, cloud service provider.

1. INTRODUCTION

Cloud computing is being adapted by a growing number of individuals and enterprises due to its wide range of services, including the elastic scaling of resources, automatic service deployment, and virtualized resources with its benefits of being economical and easily manageable in nature. Due to these features, cloud computing has become the first choice for many small to large organizations [1]. Gartner Research states that cloud computing is a rapidly emerging technology on which organizations spent an estimated $677 billion from 2013 to 2016 [2]. According to a survey conducted by an IT decision maker for large companies, more than half of the respondents (68%) expected that 50% of their I.T. resources would be migrated to a cloud platform [3].

An SLA is a legal contract which includes service obligations, deliverability, service objectives and service violation penalties [4-6]. An SLA is not only used to measure the performance of the provider, it also helps to resolve disputes regarding consumer duties [7]. An SLA comprises one or more objectives, called service level objectives (SLO), which comprise one or many low-level metrics [8]. Due to the elastic nature of the cloud, it is very important that a small- to medium-sized cloud service provider allocates its 'marginal resources' wisely and forms intelligent viable SLAs [9]. When a provider and a consumer agree on all the SLOs and form an SLA, it is very important that the service provider continually monitors the Quality of Service (QoS) parameters being delivered against the QoS parameters defined in the SLA to ascertain instances of SLA violation in order to take immediate remedial action to lower its impact. This research comes under the area of cloud service management. As shown in Figure 1, there are two phases to manage cloud services – the pre-interaction time phase and the post-interaction time phase [10]. The pre-interaction time phase is the time duration before finalizing an SLA with a provider (from time t-m till t) as presented in Figure 1. In this time period, a consumer requests a service and assesses the capability of the provider to commit to it. In the case of small- to medium-sized cloud service providers, this time period is used to decide with which user to form an SLA along with the level of commitment to each SLO. The post-interaction time phase starts when a consumer and a provider have finalized and

formed their SLA (from time t till t+n). When multiple SLAs have been formed with multiple users, the cloud provider needs a system that detects possible violations to the SLA and alerts the service provider so that action can be taken to prevent it [11]. To determine SLA violations, the process of predicting future QoS is important because depending on the prediction results, the service provider will be able to manage its resources and avoid violations. There are many prediction algorithms that undertake agreed QoS parameters and predict future intervals, however, the results of each prediction approach vary, depending on the data pattern of the input data. The wrong choice of a prediction algorithm could lead to service violation and violation penalties. Hence, it is very important to determine the accuracy of different prediction approaches, depending on the input QoS patterns and the pattern of input data.

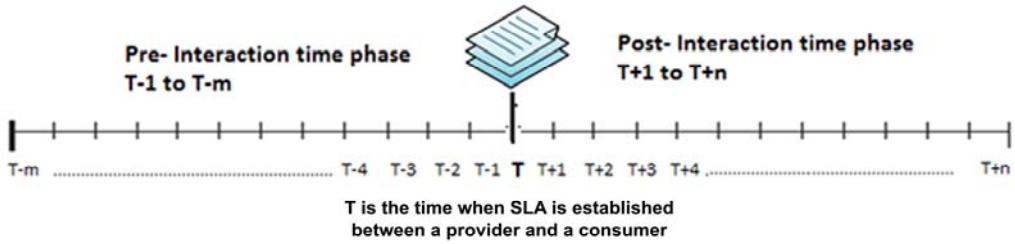

Figure 1: The two-time phases of cloud service management

To address this issue, in this paper, we investigate the accuracy of the prediction methods in two different categories, namely time series (TS) and machine learning (ML) -based methods. Our objective is to test which of these give the most accurate results depending on the patterns in the inputs. In each category, we test different techniques to determine which ones rank higher than the others. The significance of this work is that it assists the cloud provider, particularly small- and medium-sized cloud providers, to optimally manage their SLAs and risk of SLA violation to avoid violation penalties. There are many approaches in the literature that utilize TS- and ML-based prediction approaches for managing cloud services but to the best of our knowledge, there is no existing work that discusses the accuracy of each category of approaches. Furthermore, there is also no other work which performs such computations on real cloud QoS data. However, such analysis is important as it guides the service provider to manage its resources wisely and avoid SLA violations. The methods that we consider in each category are shown in Figure 2. We determine and test the prediction accuracy of these techniques in each category on a real-cloud dataset from Amazon EC2 IaaS cloud services. We consider three QoS parameters, namely CPU, memory and I/O to learn about the cloud services and then predict their QoS values before comparing them with the actual observed values. The choice of only these three cloud QoS parameters is due to their availability from a real cloud provider and their common use in forming SLAs. To consider various possible patterns in the input data and determine their effect on the output values, we divide the dataset into 9 time intervals, starting from 5 minutes to 4 weeks.

This paper is arranged as follows: Section 2 defines the data patterns in the time series data and the related work from the literature that emphasizes the importance of QoS prediction in cloud service management. Section 3 describes the adopted approach in comparing the performance of the TS and ML-based approaches. Section 4 presents the obtained prediction results on a real-world cloud QoS dataset by both types of approaches and Section 5 presents a comparative evaluation of the different techniques used in them. Section 6 presents the need for the data analyst to know beforehand which prediction method gives the best result in predicting QoS based on the specifics of the past input series. Section 7 concludes the paper with future research directions.

2. RELATED STUDIES

This section defines the data patterns in the time series data and the related work from the literature that emphasizes the importance of QoS prediction in cloud service management.

2.1 Different possible patterns in time series data

Time series evaluation is the process of measuring variables at a set period of time, which could be hourly, daily, weekly, monthly or some other regular time interval. Time series data provides information on the previous behavior of a system or the presence of data patterns in a time series and suggests an appropriate method for future data prediction [12, 13]. Patterns in time series data help the selection of an optimal prediction method, because each pattern has definite characteristics that can be predicted using a certain prediction method [14]. Several common types of data patterns in time series data are shown in Figure 3 and are described as follows.

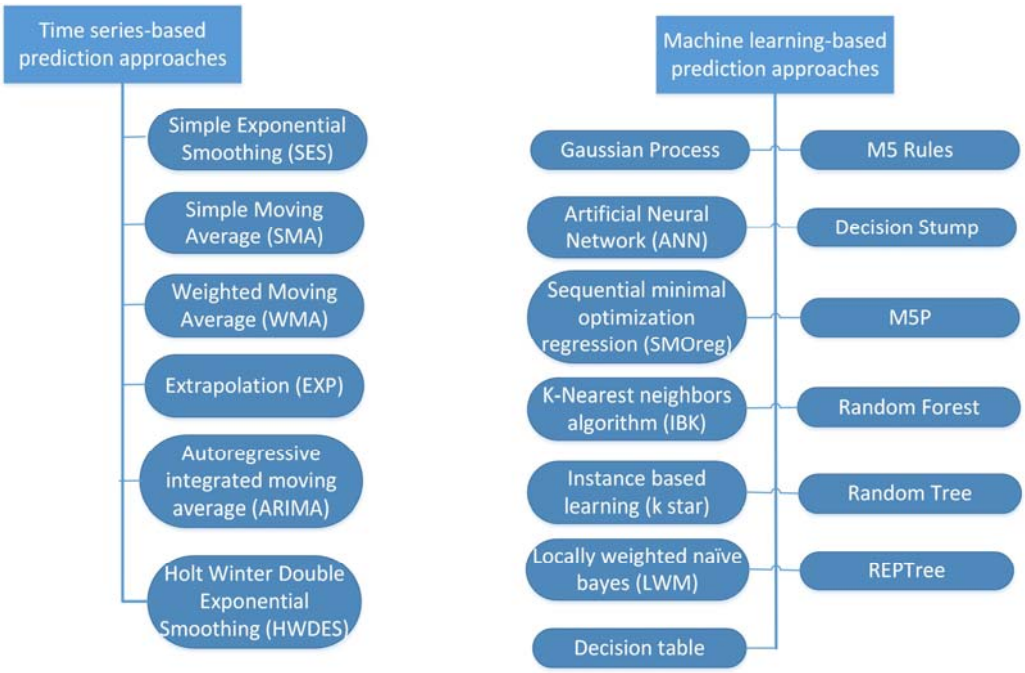

Figure 2: Different prediction techniques considered under time series and machine learning prediction approaches

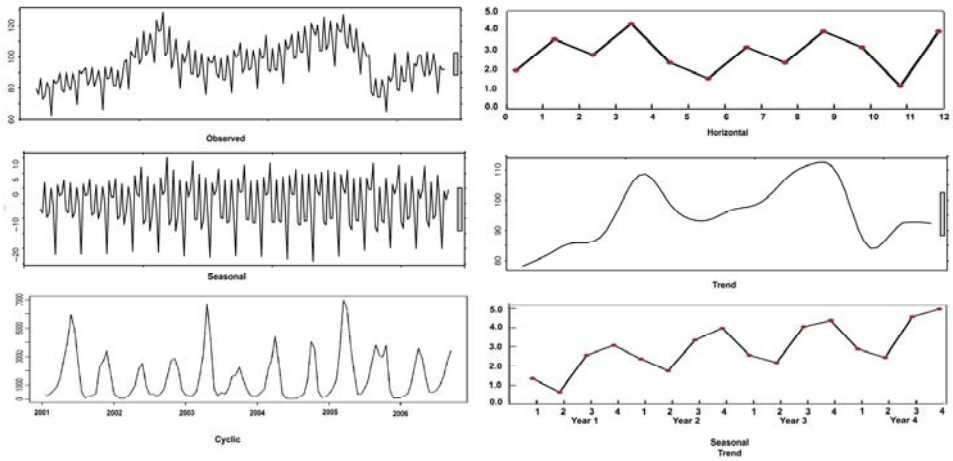

Figure 3: Different types of patterns in a time series

a) Horizontal pattern: This occurs when data change arbitrarily over a certain period of time with a constant mean. These data follow a stationary time series in which the statistical properties of data are independent of the series [15].

b) Trend pattern: When data show a shift towards higher or lower values over longer periods of time, this is called a trend pattern. Trends in data are due to an increase or decrease in values due to certain factors, such as population increases or decreases or a variation in consumer preferences [16].
c) Seasonal pattern: This occurs when data contain certain patterns that repeat themselves after a consecutive period of time due to seasonal variations.
d) Trend and seasonal pattern: This occurs when data have characteristics of both trend and seasonality, for example, some data have an increasing or decreasing trend in certain seasons [15, 16].
e) Cyclic pattern: This occurs when data have a repeated pattern above and below the trend-line but not for a fixed period with no seasonality [16].

2.2 QoS Prediction in Cloud Service Management

In this section, we review some of the related literature that discusses SLA management and QoS monitoring in cloud computing. A QoS prediction method was proposed by Cicotti et al. [17] that links a monitoring approach with the probabilistic model-checking method. A system parametric model, which includes the SLA, reference system and big data analytics, and is based on real-time monitoring was proposed that leads to a model-checking platform and as an end product, gives the probability of a service violation occurring. The effectiveness of the approach is validated using a probabilistic checking model, PRISM, and a Smart Grid case study. Wu et al. [18] proposed a learning neighborhood-based prediction method to predict personalized QoS parameters. In their proposed model, machine-learning methods are used to build a neighborhood-based approach which gives optimal results compared to other collaborative filtering methods, such as a memory-based method. The previous profile history plays an important role in predicting service violation. Chaudhuri et al. [19] proposed HMRLSRSVR and considered the previous service history to predict QoS parameters. A soft computing approach is used to validate their approach on a public dataset. In our earlier work, we proposed a profile-based SLA violation prediction model [20]. By considering the profiles of consumers, we determine how many resources should be offered to consumers depending on their previous SLA adherence history and the provider-side's available resources [21]. From the experiment results, we observed that by considering the consumer's profile along with their nearest neighbors' profiles, we obtain optimal prediction results. The authors of [22] proposed a cross-layer multi-cloud application monitoring and benchmarking approach as a service for efficient QOS monitoring and benchmarking. The proposed framework monitors the QoS of application components that may be deployed across multiple cloud platforms (Amazon EC2 and Microsoft Azure). The work in [23] proposed thirteen different kinds of correlated ranking algorithms which improves the accuracy of QoS ranking prediction. These QoS ranking predictions examine the order of services being considered by a particular user. Lo et al. [24] proposed a local neighborhood matrix factorization (LoNMF) method to predict QoS parameters. The authors used a two-level selection process to select the nearest neighbors. After the selection of the nearest neighbors, the system combines domain knowledge and an extensive matrix-factorization method for customized QoS prediction. Qi et al. [25] integrated information from the network and from neighbors with the matrix-factorization approach to forecast customized QoS parameters. A prediction method that combines both collaborative methods is proposed by Zheng et al. [26]. The authors observed that by combining both filtering approaches, they obtained optimal results for QoS prediction and have better service selection compared to using an individual approach. Sun et al. [27] applied the features of web services' QoS to measure similarity using the memory-based collaborative filtering method. Shao et al. [28] used the collaborative filtering method for similarity mining, depending on previous behavior. Their proposed approach finds the similarity among users from QoS data and based on this similarity, predicts future QoS values. QoS monitoring as a service (QoS-MONaaS) was proposed by Romano et al. [29]. The model provides a continuous QoS monitoring facility for all consumers. It is implemented on top of the SRT-15 platform. It performs three operations, the service provider operation, the service consumer operation and the QoS-MONaaS operation. The QoS-MONaaS comprises four components which are capable of operating

in a heterogeneous cloud environment and achieves high performance by managing tasks according to their time consumption [30]. A credible aware QoS prediction method was proposed by Wu et al. [31] in which they used a two-phased k-means clustering method which groups the QoS data together and uses this as input to determine an unreliable consumer index and then the unreliable consumer index is used to identify unreliable consumers and find their nearest neighbors, using the two phased k-means clustering method. Verba et al. [32] proposed a messaging-based modular gateway platform that allows devices on multiple platforms to connect to the same device via a messaging service which lessens the intricacy of any application. The platform allows a wide range of systems to deploy applications on devices without worrying about the platform. The proposed framework provides a generic platform for IoT devices that encapsulates a wide range of containers and drivers from more than one user and allows applications to communicate directly with the cloud provider, clients and peer applications without worrying about the underlying protocol. The QoS value is predicted from the values of closely related users. Related web services which have the same physical properties are clustered at one place [33]. The degree of closeness between consumers is calculated depending on the formed clusters and then the collaborative filtering method is utilized to predict the QoS value.

With reference to QoS prediction in cloud SLA management from the SME perspective, we compare the existing approaches (shown in Table 1) based on their QoS prediction performance for service providers to accurately predict QoS parameters. For each approach, we determine the prediction category used and if any, the analysis is done by the authors before using this prediction category in their approach. From the analysis in Table 1, while it can be noted that QoS prediction is being undertaken, there is no work that compares the performance of these approaches thereby showing to the SMEs which one to use to best avoid SLA violations. In the next section, we discuss our adopted approach to apply the thirteen prediction methods on a real cloud dataset and test their prediction accuracy.

3. APPROACH FOR TESTING THE ACCURACY OF TIME SERIES AND MACHINE LEARNING BASED QoS PREDICTION APPROACHES

In this section, we present our adopted approach for testing the accuracy of six time-series and thirteen machine learning-based prediction approaches. Section 3.1 focusses on the time series-based approaches and section 3.2 focusses on the machine learning-based approaches.

3.1 Approach for testing the accuracy of time series-based prediction approaches

Our approach for testing the accuracy of the six time series-based prediction approaches consists of four steps (TS-1 to TS-4) as follows. Some of the steps, such as data collection, may have a similar broad objective with the machine learning-based prediction approaches but their methods of implementation differ. The steps are explained in detail below.

Step TS-1: Data Collection

In this step, time series data is collected from reliable sources and is divided into two parts, namely input data and testing data. Input data constitutes 80% of the collected data and is used by the prediction algorithms to predict the future QoS of each input. The predicted values are then tested against the testing data to determine the accuracy of the predicted results.

Step TS-2: Input selection

In this step, the inputs required from the input data to predict the future QoS are selected. This step needs to be applied to each of the time series prediction methods under consideration as their required inputs vary. For example, simple moving average, one of the most basic prediction methods, considers data from previous N time intervals, averages them and then uses the result to

Table 1. Critical evaluation of existing SLA management approaches

Source	Are future QoS values predicted?	Prediction category used (ML / TS / Other)?	Analysis done to determine the best category of approach?	Determining optimal parameters for prediction algorithm	QoS prediction with varying data patterns at different time intervals
Cicotti et al. [17]	✓	TS	×	×	×
Wu et al. [18]	✓	Other	✓	×	×
Chaudhuri et al. [19]	✓	ML	✓	×	×
Alhamazani et al. [22]	×	None	×	×	×
Jayapriya et al. [23]	✓	ML, Other	✓	×	×
Lo et al. [24]	✓	ML	✓	×	×
Qi et al. [25]	✓	Other	×	×	×
Zheng et al. [26]	✓	Other	×	×	×
Shao et al. [28]	✓	Other	×	×	×
Romano et al. [29]	✓	Other	×	×	×
Cicotti et al. [30]	✓	TS	×	×	×
Wu et al. [31]	✓	ML	✓	×	×
Chen et al. [33]	✓	Other	×	×	×
Zheng et al. [34]	✓	Other	×	×	×
Ding et al. [35]	✓	Other	×	×	✓
Zhang et al. [36]	✓	ML	×	×	✓
Zheng et al. [37]	✓	ML	×	×	×
Tang et al. [38]	✓	ML	×	×	✓
Zhang et al. [39]	✓	Other	×	×	✓
Lo et al. [40]	✓	ML	×	×	✓

predict the future time interval. On the other hand, the extrapolation method unlike interpolation which considers previous data between two known data points, considers data beyond the range of known data points. Hence, data needs to be selected appropriately according to the specifics of the time series method considered to obtain an accurate prediction result.

Step TS-3: Implementing prediction methods

In this step, the six different time series prediction approaches considered in this work are applied on the input data to predict the future QoS. MATLAB is used to apply the different algorithms in this step.

Step TS-4: **Comparing methods and result analysis**

In this final step, root mean square error (RMSE) is used to examine the accuracy of the prediction methods on the inputs dataset in determining the future QoS.

3.2 Approach for testing the accuracy of machine learning-based prediction approaches

Our approach for testing the accuracy of thirteen machine learning-based prediction approaches consists of five steps (ML-1 to ML-5). As mentioned in Section 3.1, some of the steps such as data collection may have a similar broad objective as the time series-based prediction approaches but the way they are implemented differs. Hence, these are explained in detail according to how they are applied in the following.

Step ML-1: Data Collection

In this step, the collected data is divided into two, namely, training and testing data. The training data is used to train the prediction methods while the testing data allows us to apply the learned methods, test the accuracy of each and find the most accurate one. We use the cross-validation method to assess the accuracy of the prediction methods. The advantage of using cross-validation is that there is a good variety from the inputs when the testing is done which addresses the sensitivity of training and test partitioning. Cross-validation derives a better model in measuring the methods' prediction accuracy.

Step ML-2: Stationary-based transformation

Stationary time series data have uniform statistical properties over time. This data does not have any trends or seasonality over the time series. Stationary data have irregular cyclic behavior throughout the time series and there is a constant variation in fluctuation over the time period. In many cases, the process is unable to make accurate predictions when the data doesn't satisfy stationarity conditions. So, to test the accuracy of the prediction methods under optimal conditions, using two subsequent transformations, we make the collected data stationary by using equation 1. This subsequent transformation is referred to as difflog transformation.

$$Y_t = log(Z_t) - log(Z_{t-1}) \qquad (1)$$

Step ML-3: Input selection

There is no specific method in most of the mentioned prediction methods to determine the models' inputs. However, the right selection of inputs is important to have an accurate prediction output. To achieve this, we leverage the power of traditional time series modeling for the input selection by using the autocorrelation function (ACF). It should be noted that selecting the model inputs is not an easy task as the feasible solution space is quite large. If we only consider 12 lags as input candidates and assume two inputs, the size of a feasible solution space is 66[1]. Taking the specifics of the dataset considered in this work, 792 models[2] need to be constructed and tested.

Step ML-4: Implementing prediction methods

By completing the first three steps, the different prediction methods can be learned and applied on the testing dataset to measure their accuracy. WEKA is used to perform the tasks in this step.

Step ML-5: **Comparing methods and result analysis**

In this final step, root mean square error (RMSE) is used to determine the accuracy of the prediction methods on datasets of different types in determining the future QoS.

4. ANALYSIS OF DIFFERENT TIME SERIES AND MACHINE LEARNING PREDICTION METHODS ON QoS DATA

In this section, we apply the aforementioned steps for QoS prediction and accuracy determination by time series and machine learning approaches on a real dataset from Amazon EC2 IaaS cloud services – EC2 US West. The dataset covers a period of three years starting from 28 March 2013 to 28 March 2016 and was collected from CloudClimate [41] using the PRTG monitoring service [42]. The QoS parameters considered for this study are CPU, memory and I/O performance. Figure 4 shows the QoS parameters for the part of the whole time period i.e. from 01/01/2014 to 09/02/2015 in a graphical format as represented by the PRTG network monitor. To analyze the deviation between actual and predicted observations and to measure prediction accuracy at different time intervals, we divide the measurement interval of a dataset into nine subsets ranging from 5 minutes, 10 minutes, 20 minutes, 1 hour, 4 hours, 12 hours, 1 day (24 hours), 1 week, and 4 weeks to see the patterns and the accuracy of the prediction approaches. A minimum time interval

[1] $\binom{12}{2}$

[2] 12*66

of 5 minutes is chosen because, as mentioned in the literature, it takes anytime between 5-15 minutes for the cloud service provider to set up a virtual machine [43]. Therefore, a time interval of 5 minutes is considered the required time for the provider to take appropriate mitigating action when it detects that a violation is likely to occur. Table 2 shows the data patterns observed for three QoS parameters – I/O, CPU and memory in all nine datasets. In Table 2, HZ represents a horizontal pattern, CY represents a cyclic pattern, SS refers to a seasonal pattern, TD refers to a trend and RD shows a random pattern. Section 4.1 discusses the results of the prediction approaches based on the time series method, and Section 4.2 discusses the results from machine-learning based prediction approaches.

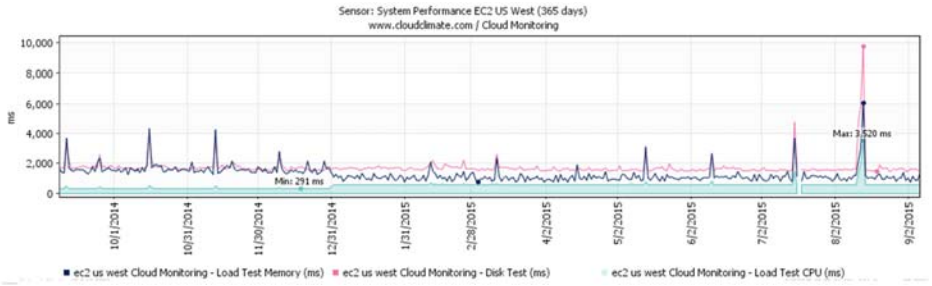

Figure 4: QoS parameters of EC2 US West [42]

Table 2. Data patterns observed in the nine datasets

Parameter	5 mins	10 mins	20 mins	1 hr	4 hrs	12 hrs	1 day	1 week	4 weeks
CPU	HZ	HZ	HZ	HZ	HZ	HZ	RD	CY	SS
Memory	CY	CY	CY	CY	CY	RD	RD	TD	SS
I/O	RD	RD	RD	RD	HZ	HZ	HZ	RD	SS

4.1 Application of time series-based approaches for predicting the future QoS – Applying, comparing the results and ranking them according to their accuracy

Table 3-5 represents the accuracy of the six time series-based prediction methods in predicting each of the three QoS in each dataset. The simple exponential smoothing (SES) method is suitable for a dataset that does not follow any pattern. When there is a trend and seasonality in a dataset, its prediction accuracy decreases as observed for the CPU dataset in the time periods of 1 week and 4 weeks, both of which follow a seasonal pattern and therefore their RMSE is relatively larger than the other datasets. The value of smoothing factor α impacts significantly on prediction accuracy. When the value of α is 0, it is insensitive and when the value of α increases, it become more sensitive. The simple moving average (SMA) method is suitable for data that have random variations. The prediction accuracy depends on the size of *k* (number of entries for mean), therefore we analyzed the prediction accuracy with a different number of *k*. We consider 10 to 11 different values of *k* to analyze the prediction precision, subject to the size of the dataset. The datasets are divided into 10 sub-sections that start from two entries i.e. k=2 and then after some time intervals till the last value in a dataset. For each dataset (time intervals) when the size of *k* increases, the prediction accuracy decreases because when a shorter time span is used to find the average, it is more sensitive, and when there is any change in a dataset, it changes immediately. The longer the time span, the less sensitive it is, therefore a larger *k* produces smoother data. Therefore, for each dataset, the smallest *k*, which we set as 2, gives the most optimal result. The weighted moving average (WMA) method was used to analyze the prediction accuracy for 10 time intervals with two variable parameters, the number of observation *k* and increasing factor α. Depending on the size of the dataset, we selected three values of *k*, the starting point being k=2 or k=3, the mid-value of the dataset k=m/2 (m is size of the dataset) and the end point k=m-1 or k=m-2. To analyze the impact of the weight factor, we randomly set the value of α as 0.5, 1.2, 1.5, 2, 5 and 10. The value of α in our experiment is the difference in the

weight between recent past data and distant past data. This means that when the value of α is 0.5, the weight of every recent past data is 0.5 times greater than the distant past. When α is 10, this means that the weight of every recent past data is 10 times greater than the distant past and the sum of all the weights is equal to 1. Therefore, the value of α=10 gives a higher weight to the most recent data than the value of α=0.5. Therefore, from the prediction results, we observed that when the value of *k* and α increases, we obtain better result because using larger data and assigning a higher weight to recent past data generates an optimal prediction result. We observed that in some entries, the weight factor is smaller than the smallest nonzero floating-point value in MATLAB, therefore it does not generate any prediction results. The extrapolation (EXP) method, is used to compare the results of the 9 datasets, the results showing that a time period of 1 hour for CPU, a time period of 4 weeks for memory and a time period of 1 day for I/O give most optimal prediction results. The Holt-Winters double exponential smoothing (HWDES) method was used to analyze the prediction accuracy in 10 time intervals with two variable parameters α and β. In each dataset, we analyzed 9x9=81 cases for a set of α and β with values of α as = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 and values of β as = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9. In each dataset, we observed that with values of α=0.9 and β=0.1, we obtain the optimal prediction result. The ARIMA method has a wide range of parameters which affects prediction. We considered eight sets of three parameters (p,d,q) which are = (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). In each dataset, we observed that the set of p,d,q = (0,1,0) gives the most optimal prediction result.

Based on the optimal prediction result for each prediction method with the optimal parameters, we select and compare all six prediction methods to analyze their prediction accuracy. Table 6 presents a comparative analysis of the six prediction methods for the 9 time intervals. Table 6 shows that the WMA and the ARIMA method give the most optimal prediction results at different time intervals. The Holt-Winters double exponential smoothing method generates the second best optimal prediction results, followed by the SMA method, then the SES method and the Exp method giving the least optimal prediction results.

Table 3. Accuracy of the time series-based prediction methods in the form of RMSE in forecasting CPU QoS in each time period

Methods	5 mins	10 mins	20 mins	1 hr	4 hrs	12 hrs	1 day	1 week	4 weeks
SES	76.25	6.6	5.21	3.92	20.33	14.2	300.43	22.42	25.84
SMA	60.58	4.96	3.8	2.77	15.21	9.67	252.33	17.71	22.35
WMA	0.04	0.0303	0.67	0.0632	0.0934	0.37	3.52	0.01	0.02
EXP	180.19	14.18	10.57	7.68	46.63	27.7	675.68	76.65	33.12
HWDES	10.42	0.83	0.63	0.46	2.66	1.63	41.45	4.71	2.68
ARIMA	0.09	0.07	0.07	0.1	0.45	0.6	18.68	5.87	4.86

Table 4. Accuracy of the time series-based prediction methods in the form of RMSE in forecasting memory QoS in each time period

Methods	5 mins	10 mins	20 mins	1 hr	4 hrs	12 hrs	1 day	1 week	4 weeks
SES	3498.24	2647.31	2096.45	1397.3	683.11	355.26	191.16	215.82	144.06
SMA	2726.23	2038.73	1574.43	1126.11	577.66	246.36	147.61	147.76	111.17
WMA	0.6	0.44	0.29	22.31	15.89	8.93	4.65	0.32	0.52
EXP	8121.88	5937.66	4655.03	3248.37	1701.29	891.21	412.99	429.02	226.86
HWDES	469.23	346.37	269.99	190.48	98.71	47.98	24.54	26.22	15.13
ARIMA	7.89	11.93	17.16	24.27	11.77	16.55	11.64	13.07	14.34

Table 5. Accuracy of the time series-based prediction methods in the form of RMSE in forecasting I/O performance QoS in each time period

Methods	5 mins	10 mins	20 mins	1 hr	4 hrs	12 hrs	1 day	1 week	4 weeks
SES	187.14	148.05	123.08	90.96	279.64	175.37	4.51	135.96	62.45
SMA	143.56	111.36	89.2	63.93	213.24	116.18	2.7	103.19	48.09
WMA	5.48	3.81	3.57	1.43	0.9	0.85	1.30E-14	0.75	1.45
EXP	400.98	310.42	265.99	156.86	638.92	372.96	4.49	478.91	161.26
HWDES	23.86	18.49	15.36	9.96	36.82	20.9	0.46	32.44	9.12
ARIMA	0.48	0.67	1.11	2.15	4.12	6.8	0.9	2.82	3.97

Table 6. Time series-based prediction algorithm at different time intervals arranged in ascending order

Time interval	CPU		Memory		I/O Performance	
	Prediction accuracy order	Prediction method	Prediction accuracy order	Prediction method	Prediction accuracy order	Prediction method
5 minutes	1	WMA	1	WMA	1	ARIMA
	2	ARIMA	2	ARIMA	2	WMA
	3	HWDES	3	HWDES	3	HWDES
	4	SMA	4	SMA	4	SMA
	5	SES	5	SES	5	SES
	6	EXP	6	EXP	6	EXP
10 minutes	1	WMA	1	WMA	1	ARIMA
	2	ARIMA	2	ARIMA	2	WMA
	3	HWDES	3	HWDES	3	HWDES
	4	SMA	4	SMA	4	SMA
	5	SES	5	SES	5	SES
	6	EXP	6	EXP	6	EXP
20 minutes	1	ARIMA	1	WMA	1	ARIMA
	2	HWDES	2	ARIMA	2	WMA
	3	WMA	3	HWDES	3	HWDES
	4	SMA	4	SMA	4	SMA
	5	SES	5	SES	5	SES
	6	EXP	6	EXP	6	EXP
1 hour	1	WMA	1	WMA	1	WMA
	2	ARIMA	2	ARIMA	2	ARIMA
	3	HWDES	3	HWDES	3	HWDES
	4	SMA	4	SMA	4	SMA
	5	SES	5	SES	5	SES
	6	EXP	6	EXP	6	EXP
4 hours	1	WMA	1	ARIMA	1	WMA
	2	ARIMA	2	WMA	2	ARIMA
	3	HWDES	3	HWDES	3	HWDES
	4	SMA	4	SMA	4	SMA
	5	SES	5	SES	5	SES
	6	EXP	6	EXP	6	EXP
12 hours	1	WMA	1	WMA	1	WMA
	2	ARIMA	2	ARIMA	2	ARIMA
	3	HWDES	3	HWDES	3	HWDES
	4	SMA	4	SMA	4	SMA
	5	SES	5	SES	5	SES
	6	EXP	6	EXP	6	EXP
1 day	1	WMA	1	WMA	1	WMA
	2	ARIMA	2	ARIMA	2	HWDES
	3	HWDES	3	HWDES	3	ARIMA
	4	SMA	4	SMA	4	SMA
	5	SES	5	SES	5	SES
	6	EXP	6	EXP	6	EXP
1 week	1	WMA	1	WMA	1	WMA
	2	ARIMA	2	ARIMA	2	ARIMA
	3	HWDES	3	HWDES	3	HWDES
	4	SMA	4	SMA	4	SMA
	5	SES	5	SES	5	SES
	6	EXP	6	EXP	6	EXP
4 weeks	1	WMA	1	WMA	1	WMA
	2	HWDES	2	ARIMA	2	ARIMA
	3	ARIMA	3	HWDES	3	HWDES
	4	SMA	4	SMA	4	SMA
	5	SES	5	SES	5	SES
	6	EXP	6	EXP	6	EXP

4.2 Application of machine learning-based approaches for predicting the future QoS

4.2.1 Step ML-2: Stationary-based transformation

Figure 4 shows the values for the input parameter, *CPU,* for a week before and after applying difflog transformation to make it stationary. As indicated by the blue line, CPU difflog is much closer to the stationary process by maintaining a constant average over the time. It should be noted that we normalize CPU to put both lines in the same figure. However, this transformation does not change the properties of the curve.

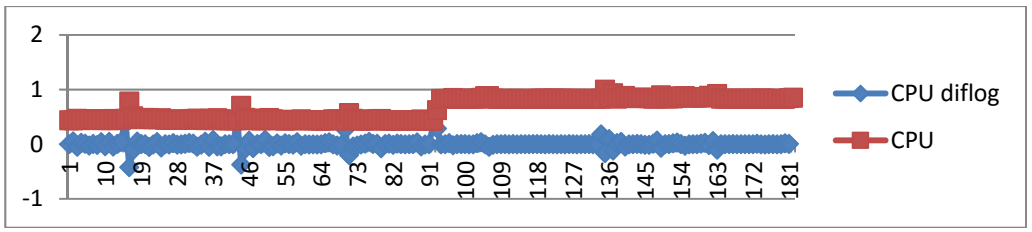

Figure 5. Comparison of QoS of input parameter CPU over a time period of a week before and after making it stationary

Figure 5 shows the transformation process on the performance of input parameter I/O on an interval of 12 hours. Similar to what was observed in Figure 4, Figure 5 shows that difflog makes the average I/O performance much more stable but it is not able to make the variance stable.

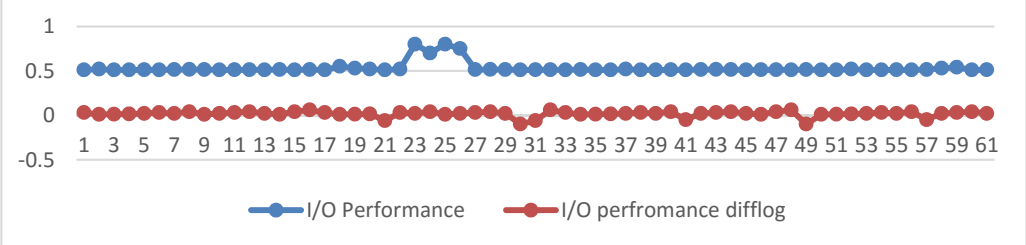

Figure 6. Comparison of QoS of the performance of input parameter I/O over a time period of 12 hours before and after making it stationary

4.2.2 Step ML-3: Input selection

To select the most appropriate combination of inputs to predict its future QoS, we run ACF over the input's dataset to find the model inputs. Figure 6 shows the results of the ACF on input parameter CPU over an interval of a day.

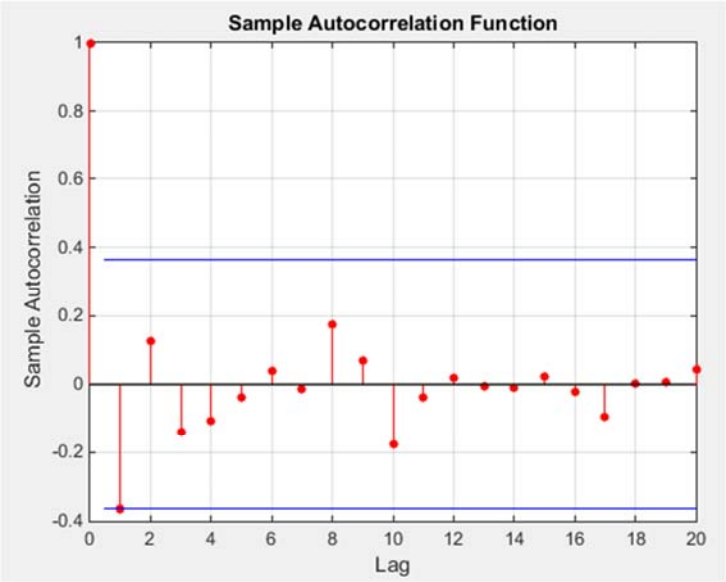

Figure 7. ACF Plot

This figure shows the first lag has a higher correlation with the current time which means the inputs for this should be of the following model:

$$Y_t = \alpha Y_{t-1} + \beta$$

Table 7 shows the combination of the series of inputs for the nine datasets used to predict the future QoS values.

Table 7. Inputs for various time intervals

Parameter	5 mins	10 mins	20 mins	1 hr	4 hrs	12 hrs	1 day	1 week	4 weeks
CPU	Y_{t-1}	Y_{t-1}	Y_{t-1}	Y_{t-1}	Y_{t-1}	Y_{t-1}	Y_{t-1}	Y_{t-1}, Y_{t-2}	Y_{t-1}
Memory	Y_{t-1}	Y_{t-1}, Y_{t-2}	Y_{t-1}	Y_{t-1}	Y_{t-1}, Y_{t-2}, Y_{t-1}, Y_{t-3}	Y_{t-1}, Y_{t-2}, Y_{t-1}, Y_{t-3}	Y_{t-1}	Y_{t-1}, Y_{t-2}	Y_{t-1}
I/O	Y_{t-1}, Y_{t-2}	Y_{t-1}, Y_{t-2}	Y_{t-1}	Y_{t-1}, Y_{t-2}	Y_{t-1}	Y_{t-1}, Y_{t-2}, Y_{t-1}, Y_{t-3}	Y_{t-1}	Y_{t-1}, Y_{t-2}	Y_{t-1}, Y_{t-2}

4.2.3 Steps ML-4 and ML-5: Implementing the prediction methods, comparing the results and ranking them according to accuracy

Thirteen prediction methods were applied to predict the three QoS parameters, namely, CPU, memory and I/O performance over a nine time periods. The accuracy of the prediction methods in the form of RMSE in estimating CPU, memory and I/O performance are reported in Tables 8, 9 and 10, respectively. The prediction accuracy of each method for each time interval is arranged in ascending order, as shown in Table 11.

Table 8. Accuracy of the machine learning-based prediction methods in the form of RMSE in forecasting CPU QoS in each time period

Methods	5 mins	10 mins	20 mins	1 hr	4 hrs	12 hrs	1 day	1 week	4 weeks
Gaussian process	0.0954	0.0142	0.0108	0.0079	0.0396	0.0269	0.0246	0.0889	0.0855
ANN	0.0815	0.0127	0.0101	0.0079	0.0401	0.0295	0.0249	0.0893	0.0752
SMOreg	0.0857	0.0128	0.0099	0.0072	0.0385	0.0256	0.024	0.0869	0.0816

IBK	0.1069	0.0123	0.0099	0.0079	0.0481	0.0349	0.0286	0.1011	0.048
Kstar	0.0916	0.013	0.01	0.0076	0.0397	0.0277	0.0245	0.0926	0.0786
LWL	0.0865	0.0132	0.0101	0.008	0.0397	0.0278	0.0252	0.763	0.0716
Decision table	0.0854	0.0123	0.01	0.0076	0.0394	0.0281	0.0263	0.0809	0.0717
M5rules	0.0807	0.0122	0.0096	0.0073	0.0402	0.0259	0.0332	0.0913	0.0701
Decision Stump	0.0881	0.014	0.0103	0.0082	0.0394	0.0281	0.0228	0.0761	0.0737
M5P	0.0797	0.0122	0.0095	0.0073	0.04	0.0254	0.0323	0.091	0.0703
Random Forest	0.0879	0.0121	0.0096	0.0074	0.0429	0.031	0.0253	0.0878	0.0538
Random Tree	0.1011	0.0123	0.0099	0.0079	0.0481	0.0349	0.0286	0.0886	0.048
REPtree	0.0844	0.0123	0.0096	0.0079	0.0395	0.0262	0.0245	0.0874	0.0715

Table 9. Accuracy of the machine learning-based prediction methods in the form of RMSE in forecasting memory in each time period

Methods	**5 mins**	**10 mins**	**20 mins**	**1 hr**	**4 hrs**	**12 hrs**	**1 day**	**1 week**	**4 weeks**
Gaussian process	0.0103	0.580	0.691	0.839	0.727	0.344	0.315	0.272	0.580
ANN	0.011	0.543	0.628	0.721	0.516	0.310	0.360	0.270	0.250
SMOreg	0.012	0.559	0.671	0.832	0.521	0.311	0.318	0.233	0.257
IBK	0.010	0.557	0.666	0.823	0.565	0.440	0.361	0.306	0.076
Kstar	0.010	0.559	0.640	0.711	0.433	0.345	0.332	0.262	0.206
LWL	0.011	0.523	0.606	0.685	0.548	0.390	0.346	0.254	0.226
Decision table	0.010	0.524	0.589	0.695	0.496	0.360	0.313	0.244	0.208
M5rules	0.013	0.519	0.588	0.675	0.466	0.324	0.333	0.230	0.206
Decision Stump	0.010	0.522	0.615	0.687	0.585	0.380	0.331	0.247	0.231
M5P	0.013	0.504	0.589	0.671	0.468	0.321	0.333	0.230	0.201
Random Forest	0.0101	0.526	0.655	0.754	0.471	0.348	0.329	0.249	0.096
Random Tree	0.0104	0.555	0.666	0.823	0.591	0.419	0.361	0.313	0.066
REPtree	0.009	0.521	0.586	0.709	0.501	0.352	0.313	0.257	0.183

Table 10. Accuracy of the machine learning-based prediction methods in the form of RMSE in forecasting I/O performance in each time period

Methods	**5 mins**	**10 mins**	**20 mins**	**1 hr**	**4 hrs**	**12 hrs**	**1 day**	**1 week**	**4 weeks**
Gaussian process	0.019	0.087	0.081	0.829	0.136	0.106	0.127	0.142	0.133
ANN	0.034	0.074	0.077	0.711	0.124	0.119	0.140	0.137	0.123
SMOreg	0.016	0.078	0.072	0.829	0.127	0.112	0.136	0.131	0.122
IBK	0.024	0.106	0.088	0.822	0.114	0.113	0.186	0.171	0.044
Kstar	0.019	0.083	0.073	0.732	0.136	0.117	0.135	0.143	0.106
LWL	0.020	0.078	0.069	0.684	0.123	0.121	0.120	0.124	0.111
Decision table	0.023	0.075	0.070	0.712	0.117	0.155	0.115	0.119	0.105
M5rules	0.017	0.072	0.063	0.711	0.105	0.108	0.136	0.118	0.111
Decision Stump	0.020	0.078	0.069	0.687	0.104	0.106	0.114	0.126	0.111
M5P	0.017	0.072	0.063	0.702	0.105	0.108	0.136	0.119	0.105
Random Forest	0.019	0.790	0.076	0.731	0.128	0.109	0.155	0.137	0.053
Random Tree	0.019	0.109	0.088	0.804	0.148	0.117	0.186	0.151	0.042
REPtree	0.017	0.078	0.068	0.719	0.137	0.106	0.127	0.122	0.103

Table 11. Machine learning-based prediction algorithm at different time intervals arranged in ascending order

Time interval	CPU		Memory		I/O performance	
	Prediction accuracy order	Prediction method	Prediction accuracy order	Prediction method	Prediction accuracy order	Prediction method
5 minutes	1	M5P	1	Decision table	1	SMOreg
	2	M5rules	2	Decision Stump	2	REPtree
	3	ANN	3	Kstar	3	M5rules
	4	REPtree	4	Random forest	4	M5P
	5	Decision table	5	Gaussian process	5	Random forest
	6	SMOreg	6	IBK	6	Gaussian process
	7	LWL	7	Random tree	7	Random tree
	8	Random forest	8	LWL	8	Kstar
	9	Decision Stump	9	ANN	9	LWL
	10	Kstar	10	SMOreg	10	Decision Stump
	11	Gaussian process	11	M5P	11	Decision table
	12	Random tree	12	M5rules	12	IBK
	13	IBK	13	REPtree	13	ANN
10 minutes	1	Random forest	1	M5P	1	M5P

Time interval	CPU		Memory		I/O performance	
	Prediction accuracy order	Prediction method	Prediction accuracy order	Prediction method	Prediction accuracy order	Prediction method
	2	M5rules	2	M5rules	2	M5rules
	3	M5P	3	REPtree	3	ANN
	4	IBK	4	Decision Stump	4	Decision table
	5	Decision table	5	LWL	5	LWL
	6	Random tree	6	Decision table	6	REPtree
	7	REPtree	7	Random forest	7	SMOreg
	8	ANN	8	ANN	8	Decision Stump
	9	SMOreg	9	Random tree	9	Kstar
	10	Kstar	10	IBK	10	Gaussian process
	11	LWL	11	Kstar	11	IBK
	12	Decision Stump	12	SMOreg	12	Random tree
	13	Gaussian process	13	Gaussian process	13	Random forest
1 hour	1	SMOreg	1	M5P	1	LWL
	2	M5rules	2	M5rules	2	Decision Stump
	3	M5P	3	LWL	3	M5P
	4	Random forest	4	Decision Stump	4	M5rules
	5	Kstar	5	Decision table	5	ANN
	6	Decision table	6	REPtree	6	Decision table
	7	Gaussian process	7	Kstar	7	REPtree
	8	ANN	8	ANN	8	Random forest
	9	IBK	9	Random forest	9	Kstar
	10	Random tree	10	IBK	10	Random tree
	11	REPtree	11	Random tree	11	IBK
	12	LWL	12	SMOreg	12	SMOreg
	13	Decision Stump	13	Gaussian process	13	Gaussian process
4 hours	1	SMOreg	1	Kstar	1	Decision Stump
	2	Decision table	2	M5rules	2	M5P
	3	Decision Stump	3	M5P	3	M5rules
	4	REPtree	4	Random forest	4	IBK
	5	Gaussian process	5	Decision table	5	Decision table
	6	Kstar	6	REPtree	6	LWL
	7	LWL	7	ANN	7	ANN
	8	M5P	8	SMOreg	8	SMOreg
	9	ANN	9	LWL	9	Random forest
	10	M5rules	10	IBK	10	Gaussian process
	11	Random forest	11	Decision Stump	11	Kstar
	12	IBK	12	Random tree	12	REPtree
	13	Random tree	13	Gaussian process	13	Random tree
12 hours	1	M5P	1	ANN	1	Gaussian process
	2	SMOreg	2	SMOreg	2	Decision Stump
	3	M5rules	3	M5P	3	REPtree
	4	REPtree	4	M5rules	4	M5rules
	5	Gaussian process	5	Gaussian process	5	M5P
	6	Kstar	6	Kstar	6	Random forest
	7	LWL	7	Random forest	7	SMOreg
	8	Decision table	8	REPtree	8	IBK
	9	Decision Stump	9	Decision table	9	Random tree
	10	ANN	10	Decision Stump	10	Kstar
	11	Random forest	11	LWL	11	ANN
	12	IBK	12	Random tree	12	LWL
	13	Random tree	13	IBK	13	Decision table
1 day	1	Decision Stump	1	Decision table	1	Decision Stump
	2	SMOreg	2	REPtree	2	Decision table
	3	Kstar	3	Gaussian process	3	LWL
	4	REPtree	4	SMOreg	4	Gaussian process
	5	Gaussian process	5	Random forest	5	REPtree
	6	ANN	6	Decision Stump	6	Kstar
	7	LWL	7	Kstar	7	SMOreg
	8	Random forest	8	M5rules	8	M5rules
	9	Decision table	9	M5P	9	M5P
	10	IBK	10	LWL	10	ANN
	11	Random tree	11	ANN	11	Random forest
	12	M5P	12	Random tree	12	IBK

Time interval	CPU		Memory		I/O performance	
	Prediction accuracy order	Prediction method	Prediction accuracy order	Prediction method	Prediction accuracy order	Prediction method
	13	M5rules	13	IBK	13	Random tree
1 week	1	Decision Stump	1	M5rules	1	M5rules
	2	Decision table	2	M5P	2	Decision table
	3	SMOreg	3	SMOreg	3	M5P
	4	REPtree	4	Decision table	4	REPtree
	5	Random forest	5	Decision Stump	5	LWL
	6	Random tree	6	Random forest	6	Decision Stump
	7	Gaussian process	7	LWL	7	SMOreg
	8	ANN	8	REPtree	8	Random forest
	9	M5P	9	Kstar	9	ANN
	10	M5rules	10	ANN	10	Gaussian process
	11	Kstar	11	Gaussian process	11	Kstar
	12	IBK	12	IBK	12	Random tree
	13	LWL	13	Random tree	13	IBK
4 weeks	1	IBK	1	Randomtree	1	Randomtree
	2	Random tree	2	IBK	2	IBK
	3	Random forest	3	Randomforest	3	Randomforest
	4	M5rules	4	REPtree	4	REPtree
	5	M5P	5	M5P	5	M5P
	6	REPtree	6	M5rules	6	Decision table
	7	LWL	7	Kstar	7	Kstar
	8	Decision table	8	Decision table	8	LWL
	9	Decision Stump	9	LWL	9	Decision Stump
	10	ANN	10	Decision Stump	10	M5rules
	11	Kstar	11	ANN	11	SMOreg
	12	SMOreg	12	SMOreg	12	ANN
	13	Gaussian process	13	Gaussian process	13	Gaussian process

5. COMPARING THE RESULTS FROM THE DIFFERENT PREDICTION METHODS AND RESULTS ANALYSIS

In this section, we compare the performance of time series-based and machine learning-based approaches in terms of their prediction accuracy on the testing data. Table 12 shows the most accurate prediction results in terms of RMSE for each prediction approach in each of the nine time intervals for the three considered QoS inputs. For each time interval, it can be seen that ML-based approaches outperform the TS-based approach. There are a handful of cases in which the TS-based approach outperforms the ML-based ones, however when the ML result for that time period is considered with the average of the TS-based approach for that SLO, the ML-result is much more accurate.

Table 12. Comparative analysis of the RMSE for each time period by time series and machine learning-based prediction methods

Time interval	Minimum RMSE for CPU prediction by ML-based prediction approach	Minimum RMSE for CPU prediction by TS-based prediction approach	Best prediction result given by ML or TS-based approach?	Minimum RMSE for Memory prediction by ML-based prediction approach	Minimum RMSE for Memory prediction by TS-based prediction approach	Best prediction result given by ML or TS-based approach?	Minimum RMSE for I/O performance prediction by ML-based prediction approach	Minimum RMSE for I/O performance prediction by TS-based prediction approach	Best prediction result given by ML or TS-based approach?
5 mins	0.0797	0.04	TS-based	0.0096	0.6	ML-based	0.0156	0.48	ML-based
10 mins	0.0121	0.0303	ML-based	0.5036	0.44	TS-based	0.0716	0.67	ML-based
20 mins	0.0095	0.07	ML-based	0.5857	0.29	TS-based	0.0627	1.11	ML-based
1 hr	0.0072	0.07	ML-based	0.6709	22.31	ML-based	0.6843	1.43	ML-based
4 hrs	0.0385	0.0934	ML-based	0.4333	11.77	ML-based	0.1037	0.9	ML-based
12 hrs	0.0254	0.37	ML-based	0.3103	8.93	ML-based	0.1056	0.85	ML-based

1 day	0.0228	3.52	ML-based	0.3133	4.65	ML-based	0.114	1.30E-14	TS- based
1 week	0.0761	0.01	ML-based	0.2299	0.32	ML-based	0.1184	0.75	ML-based
4 weeks	0.048	0.02	ML-based	0.0661	0.52	ML-based	0.0417	1.45	ML-based
Average	**0.0355**	**0.4693**		**0.3470**	**5.53**		**0.1464**	**0.84**	

Tables 13(a) and (b) show the best prediction method for each input parameter for each time period in the ML-based and TS-based categories. It can be seen from the results that some methods are repeated more than others and we show their frequency in Table 14. From the ML-based results, we note that M5P outperformed the other 12 methods in 7 of the 27 cases. The second most accurate method is Decision Stump which had the best performance four times. In the TS-based techniques, the WMA approach has the best performance in 22 of the 27 cases. Table 15 shows all the prediction approaches ranked in order of accuracy for all time periods. As IBK was not able to outperform the other methods in any ML-based cases, it is ranked in the last position. Six methods are ranked equally in fifth place. In TS-based cases, WMA is ranked the highest, followed by ARIMA and the other four approaches are equally ranked in third position.

Table 13(a). Most accurate ML-based prediction method for each QoS parameter in each time interval

Time interval	CPU	Memory	I/O performance
5 mins	M5P	REP tree	SMO reg
10 mins	Random forest	M5P	M5P
20 mins	M5P	REP tree	M5rules
1 hr	SMO reg	M5P	LWL
4 hrs	SMO reg	Kstar	Decision Stump
12 hrs	M5P	ANN	Gaussian process
1 day	Decision Stump	Decision table	Decision Stump
1 week	Decision Stump	M5P	M5rules
4 weeks	Random tree	Random tree	Random tree

Table 13(b). Most accurate TS-based prediction method for each QoS parameter in each time interval

Time interval	CPU	Memory	I/O performance
5 mins	WMA	WMA	ARIMA
10 mins	WMA	WMA	ARIMA
20 mins	ARIMA	WMA	ARIMA
1 hr	WMA	WMA	WMA
4 hrs	WMA	ARIMA	WMA
12 hrs	WMA	WMA	WMA
1 day	WMA	WMA	WMA
1 week	WMA	WMA	WMA
4 weeks	WMA	WMA	WMA

Table 14(a). Frequency of 13 most accurate ML-based prediction methods

Prediction Methods	Frequency
M5P	7
Decision Stump	4
SMOreg	3
Random Tree	3
REP tree	2
M5rules	2
Gaussian process	1
ANN	1
Kstar	1
LWL	1
Decision table	1
Random Forest	1
IBK	0

Table 14(b). The two most accurate TS-based prediction methods in each time period

Prediction Methods	Frequency

WMA	22
ARIMA	5

Table 15(a). 13 ML-based prediction algorithms for each input for each time period ranked in order of accuracy

Rank	1	2	3	4	5	6
Prediction method	M5P	Decision Stump	SMOreg, Random Tree	REPtree, M5rules	Gaussian process, ANN, Kstar, LWL, Decision Table, Random forest	IBK

Table 15(b). 6 TS-based prediction algorithms for each input for each time period ranked in order of accuracy

Rank	1	2	3
Prediction method	WMA	ARIMA	HWDES, SMA, SES, EXP

6. APPLICATIONS OF THE ACCURACY OF PREDICTION RESULTS

The accurate prediction of a service's quality values is not only important in the domain of cloud computing but also in many other areas. In the following, we discuss other areas in which QoS prediction is important and the analysis in this paper can be utilized and applied for accurate service management of service.

- Cloud of Things (CoT) environment.

In the recent past, cloud computing in combination with the Internet of Things has given rise to a new and dynamic area, namely the Cloud of Things (CoT) for service delivery [44, 45]. Despite the various benefits such a paradigm provides, it also brings with it challenges that need to be managed under a dynamic environment to achieve the service aims. This dynamism in QoS is not only observed during the formation of the SLAs but also at run-time. Hence, frequent changes in QoS which are both expected according to a pattern or dependent on other external conditions need to be captured and managed to avoid service violations. To manage the QoS according to a pattern, QoS prediction is one of the critical tasks. Furthermore, in the CoT, as different services from different regions are amalgamated to achieve the required service, the predicted QoS should not only be for individual services but also for the combined ones. But before this can take place, QoS attributes such as response time and service availability need to be predicted over a period both before and after service formation to proactively manage the risk of service violations. To achieve this, service providers need to choose an appropriate prediction approach which, according to the past characteristics of the input's QoS, gives the most accurate future QoS values.

- Proactive healthcare management

With the increase in the population and the strain it places on the health care system, the focus these days is on transforming from a reactive sick care to a proactive health care system [46]. In a proactive model, the objective is to identify various factors such as at-risk individuals based on their current health record data, predict the onset of diseases and predict the risks of individuals being exposed to certain chronic conditions. To achieve these goals, predictive and descriptive types of data analytics have been utilized to predict and categorize patients in these risk profiles [47]. Having such insights is also critical for better government planning and management so that resources can be allocated appropriately. To achieve this, the recent focus on healthcare has shifted towards predictive analytics. The objective is to use statistical methods to predict outcomes for specific patients in certain conditions [48]. The objective is not to replace the main role of the physician but to provide him with superior tools and methods that will help them to better and more proactively manage a patient's health. To assist these goals, the predicted results need to be accurate hence, using the correct algorithm is key.

- Stock market prediction

Stock markets are volatile and investors need appropriate sophisticated prediction techniques that will pre-determine how the markets will behave. Such techniques are also beneficial to the regulators in helping them to take corrective measures. To achieve this, using prediction methods that can

capture the existing patterns and trends in the previous patterns and use these to forecast future trends is critical. Existing methods have utilized time series methods [49] whereas other methods have utilized ANN [50]. However, a comparison of the methods to determine the accuracy of each is missing. The presented analyses in this paper assists in addressing this gap and thereby helping in the accurate prediction of stock market patterns and trends.

7. CONCLUSION

QoS prediction is an essential element in the SLA management framework to predict service violation and avoid violation penalties. It is crucial for a service provider to understand the likely behavior of a service consumer and they must know when to take appropriate remedial action once it detects a possible service violation. Different prediction methods produce outputs with different levels of accuracy, depending on the nature of the dataset. In the existing literature, there is no work which compares the different approaches for QoS prediction to enable the service provider to form a viable SLA. To address this gap, this paper compared the prediction results from time series and machine learning-based prediction approaches and evaluated them on three QoS parameters using 9 time series datasets from a real cloud provider. From the comparative analyses, we observed that ML-based approaches outperform the TS-based approaches in accurately predicting the future QoS. Of the ML-based approaches, M5P and the Decision Stump method give the most optimal results at different time intervals and can help the service provider to avoid service violation and violation penalties. In the future, we will evaluate these optimal prediction methods on the developed approaches for the management of cloud services SLAs in the post-interaction phase and determine if it leads to a change in recommending an appropriate action to the service provider for SLA violation management.

ACKNOWLEDGEMENTS

The first author acknowledges that this work was started when he was at the School of Software, University of Technology Sydney for which they provided proofreading assistance.

8. REFERENCES

[1] M. Cunha, N. Mendonça, and A. Sampaio, "Cloud Crawler: a declarative performance evaluation environment for infrastructure□as□a□service clouds," *Concurrency and Computation: Practice and Experience,* vol. 29, no. 1, 2017.

[2] I. Gartner, "Forecast: Public Cloud Services, Worldwide, 2013-2019," ed: Gartner 2016.

[3] B. Narasimhan and R. Nichols, "State of cloud applications and platforms: The cloud adopters' view," *Computer,* no. 3, pp. 24-28, 2011.

[4] D. Greenwood, G. Vitaglione, L. Keller, and M. Calisti, "Service level agreement management with adaptive coordination," in *International conference on Networking and Services*, Silicon Valley, CA, USA, 2006, pp. 45-45: IEEE.

[5] B. R. Kandukuri, V. R. Paturi, and A. Rakshit, "Cloud security issues," in *IEEE International Conference on Services Computing*, Bangalore, India, 2009, pp. 517-520: IEEE.

[6] W. Hussain, F. K. Hussain, and O. K. Hussain, "Maintaining Trust in Cloud Computing through SLA Monitoring," in *Neural Information Processing*, 2014, pp. 690-697: Springer.

[7] J. Yan, R. Kowalczyk, J. Lin, M. B. Chhetri, S. K. Goh, and J. Zhang, "Autonomous service level agreement negotiation for service composition provision," *Future Generation Computer Systems,* vol. 23, no. 6, pp. 748-759, 2007.

[8] V. C. Emeakaroha, I. Brandic, M. Maurer, and S. Dustdar, "Low level metrics to high level SLAs-LoM2HiS framework: Bridging the gap between monitored metrics and SLA parameters in cloud environments," in *International Conference on High Performance Computing and Simulation (HPCS)*, Caen, France, 2010, pp. 48-54: IEEE.

[9] W. Hussain, F. K. Hussain, and O. K. Hussain, "Profile-based viable Service Level Agreement (SLA) Violation Prediction Model in the Cloud," in *INTERNATIONAL CONFERENCE ON P2P, PARALLEL, GRID, CLOUD AND INTERNET COMPUTING*, KRAKOW, POLAND, 2015: Conference Publishing Service.

[10] O. K. Hussain, T. Dillon, F. K. Hussain, and E. Chang, *Risk assessment and management in the networked economy*. Springer, 2012.

[11] G. Katsaros, G. Kousiouris, S. V. Gogouvitis, D. Kyriazis, A. Menychtas, and T. Varvarigou, "A Self-adaptive hierarchical monitoring mechanism for Clouds," *Journal of Systems and Software,* vol. 85, no. 5, pp. 1029-1041, 2012.

[12] S. Bisgaard and M. Kulahci, "Quality Quandaries*: Time series model selection and parsimony," *Quality Engineering,* vol. 21, no. 3, pp. 341-353, 2009.

[13] a. J. C. Jeffrey D. Camm, Michael J. Fry , effrey W. Ohlmann, David R. Anderson *Essentials of Business Analytics*. Cengage Learning, 2015, p. 701.

[14] N. R. Herbst, N. Huber, S. Kounev, and E. Amrehn, "Self□adaptive workload classification and forecasting for proactive resource provisioning," *Concurrency and computation: practice and experience,* vol. 26, no. 12, pp. 2053-2078, 2014.

[15] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM computing surveys (CSUR),* vol. 31, no. 3, pp. 264-323, 1999.

[16] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2014.

[17] G. Cicotti, L. Coppolino, S. D'Antonio, and L. Romano, "Runtime Model Checking for SLA Compliance Monitoring and QoS Prediction," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA),* vol. 6, no. 2, pp. 4-20, 2015.

[18] H. Wu, J. He, B. Li, and Y. Pei, "Personalized QoS Prediction of Cloud Services via Learning Neighborhood-based Model," in *11 International Conference on Collaborative Computing: Networking, Applications and Worksharing*, Wuhan, 2015, pp. 106-117.

[19] 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 *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, Salamanca, Spain, 2015, pp. 659-661: ACM.

[20] W. Hussain, F. K. Hussain, O. Hussain, and E. Chang, "Profile-based viable Service Level Agreement (SLA) Violation Prediction Model in the Cloud," presented at the 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), Krakow, Poland, NOVEMBER 4-6, 2015.

[21] W. Hussain, F. K. Hussain, and O. K. Hussain, "Comparative Analysis of Consumer Profile-based Methods to Predict SLA Violation," in *IEEE International Conference on Fuzzy Systems*, Istanbul, Turkey, 2015: IEEE.

[22] K. Alhamazani *et al.*, "Cross-layer multi-cloud real-time application QoS monitoring and benchmarking as-a-service framework," *IEEE Transactions on Cloud Computing,* 2015.

[23] K. Jayapriya, N. A. B. Mary, and R. Rajesh, "Cloud Service Recommendation Based on a Correlated QoS Ranking Prediction," *Journal of Network and Systems Management,* vol. 24, no. 4, pp. 916-943, 2016.

[24] W. Lo, J. Yin, Y. Li, and Z. Wu, "Efficient web service QoS prediction using local neighborhood matrix factorization," *Engineering Applications of Artificial Intelligence,* vol. 38, pp. 14-23, 2015.

[25] K. Qi, H. Hu, W. Song, J. Ge, and J. Lu, "Personalized QoS Prediction via Matrix Factorization Integrated with Neighborhood Information," in *IEEE International Conference on Services Computing (SCC)*, New York, NY, USA, 2015, pp. 186-193: IEEE.

[26] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "Wsrec: A collaborative filtering based web service recommender system," in *IEEE International Conference on Web Services*, Los Angeles, CA, USA, 2009, pp. 437-444: IEEE.

[27] H. Sun, Z. Zheng, J. Chen, and M. R. Lyu, "Personalized web service recommendation via normal recovery collaborative filtering," *IEEE Transactions on Services Computing,* vol. 6, no. 4, pp. 573-579, 2013.

[28] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, "Personalized qos prediction forweb services via collaborative filtering," in *IEEE International Conference on Web Services*, Salt Lake City, UT, USA, 2007, pp. 439-446: IEEE.

[29] L. Romano, D. De Mari, Z. Jerzak, and C. Fetzer, "A novel approach to QoS monitoring in the cloud," in *First International Conference on Data Compression, Communications and Processing (CCP)*, Palinuro, Italy, 2011, pp. 45-51: IEEE.

[30] G. Cicotti, L. Coppolino, S. D'Antonio, and L. Romano, "How to monitor QoS in cloud infrastructures: the QoSMONaaS approach," *International Journal of Computational Science and Engineering,* vol. 11, no. 1, pp. 29-45, 2015.

[31] C. Wu, W. Qiu, Z. Zheng, X. Wang, and X. Yang, "QoS Prediction of Web Services Based on Two-Phase K-Means Clustering," in *2015 IEEE International Conference on Web Services (ICWS)*, New York, NY, USA, 2015, pp. 161-168: IEEE.

[32] N. Verba, K.-M. Chao, A. James, D. Goldsmith, X. Fei, and S.-D. Stan, "Platform as a service gateway for the Fog of Things," *Advanced Engineering Informatics,* vol. 33, pp. 243-257, 2017.

[33] F. Chen, S. Yuan, and B. Mu, "User-QoS-Based Web Service Clustering for QoS Prediction," in *IEEE International Conference on Web Services (ICWS),* , New York, NY, USA, 2015, pp. 583-590: IEEE.

[34] Z. Zheng, X. Wu, Y. Zhang, M. R. Lyu, and J. Wang, "QoS ranking prediction for cloud services," *IEEE transactions on parallel and distributed systems,* vol. 24, no. 6, pp. 1213-1222, 2013.

[35] S. Ding, S. Yang, Y. Zhang, C. Liang, and C. Xia, "Combining QoS prediction and customer satisfaction estimation to solve cloud service trustworthiness evaluation problems," *Knowledge-Based Systems,* vol. 56, pp. 216-225, 2014.

[36] Y. Zhang, Z. Zheng, and M. R. Lyu, "Exploring latent features for memory-based QoS prediction in cloud computing," in *2011 30th IEEE Symposium on Reliable Distributed Systems (SRDS)*, 2011, pp. 1-10: IEEE.

[37] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "Collaborative web service qos prediction via neighborhood integrated matrix factorization," *IEEE Transactions on Services Computing,* vol. 6, no. 3, pp. 289-299, 2013.

[38] M. Tang, T. Zhang, J. Liu, and J. Chen, "Cloud service QoS prediction via exploiting collaborative filtering and location□based data smoothing," *Concurrency and Computation: Practice and Experience,* vol. 27, no. 18, pp. 5826-5839, 2015.

[39] Y. Zhang, Z. Zheng, and M. R. Lyu, "WSPred: A time-aware personalized QoS prediction framework for Web services," in *Software Reliability Engineering (ISSRE), 2011 IEEE 22nd International Symposium on*, 2011, pp. 210-219: IEEE.

[40] W. Lo, J. Yin, S. Deng, Y. Li, and Z. Wu, "Collaborative Web Service QoS Prediction with Location-Based Regularization," in *2012 IEEE 19th International Conference on Web Services*, 2012, pp. 464-471.

[41] CloudClimate. *Watching the Cloud.* Available: http://www.cloudclimate.com

[42] P. N. Monitor. Available: https://prtg.paessler.com

[43] S. Islam, J. Keung, K. Lee, and A. Liu, "Empirical prediction models for adaptive resource provisioning in the cloud," *Future Generation Computer Systems,* vol. 28, no. 1, pp. 155-162, 2012.

[44] M. Aazam, I. Khan, A. A. Alsaffar, and E. N. Huh, "Cloud of Things: Integrating Internet of Things and cloud computing and the issues involved," in *Proceedings of 2014 11th International Bhurban Conference on Applied Sciences & Technology (IBCAST) Islamabad, Pakistan, 14th - 18th January, 2014*, 2014, pp. 414-419.

[45] A. Botta, W. de Donato, V. Persico, and A. Pescapé, "Integration of Cloud computing and Internet of Things: A survey," *Future Generation Computer Systems,* vol. 56, pp. 684-700, 2016.

[46] F. R. Vogenberg, "Predictive and Prognostic Models: Implications for Healthcare Decision-Making in a Modern Recession," *American Health & Drug Benefits,* vol. 2, no. 6, pp. 218-222, Sep-Oct 2009.

[47] Z. Hu *et al.*, "Online Prediction of Health Care Utilization in the Next Six Months Based on Electronic Health Record Information: A Cohort and Validation Study," *Journal of Medical Internet Research,* vol. 17, no. 9, p. e219, 2015.

[48] https://www.elsevier.com/connect/seven-ways-predictive-analytics-can-improve-healthcare.

[49] D. Banerjee, "Forecasting of Indian stock market using time-series ARIMA model," in *2nd International Conference on Business and Information Management (ICBIM)*, Durgapur, India, 2014, pp. 131-135.

[50] L. Wang and Q. Wang, "Stock Market Prediction Using Artificial Neural Networks Based on HLP," in *Third International Conference on Intelligent Human-Machine Systems and Cybernetics*, Zhejiang, China, 2011, vol. 1, pp. 116-119.

Walayat Hussain has completed his PhD in 2016 from the University of Technology Sydney, Sydney, Australia. He has the MS (Computer Science), Postgraduate Diploma (Computer Science) and BS (Software Engineering) degrees. He is currently working as a Lecturer at the School of Systems, Management and Leadership, UTS, Australia and IIBIT- Federation University, Australia. Prior to joining UTS he worked as an Assistant Professor at the BUITEMS, Pakistan and a Lecturer at the University of Balochistan for seven years. Walayat's research area are distributed computing, cloud computing, fog computing, decision support system, big data, usability engineering and SLA management.

.