ISSN 2007-9737

An LSTM Based Time Series Forecasting Framework for Web Services Recommendation

Vijendra Pratap Singh, Manish Kumar Pandey, Pangambam Sendash Singh, Subbiah Karthikeyan

Banaras Hindu University, Institute of Science, Department of Computer Science, India

{vijendrap.singh4, pangambams.singh4, karthik}@bhu.ac.in, pandey.manish@live.com

Abstract. The convergence of Social Mobility Analytics and Cloud (SMAC) technologies gives rise to an unforeseen aggrandization of the web services on the internet. The resilience and payment-based approach of the cloud makes it an obvious choice for the deployment of web services-based applications. Out of available web services, to gratify the similar functionalities, the choice of the web service based on the personalized quality of service (QoS) parameters plays an important role in determining the selection of the web service. The role of time is rarely being discussed in deciding the QoS of web services. The delivery of QoS is not made as declared due to the non-functional performance of web services correlated behavior with the invocation time. This happens because service status usually changes over time. Hence, the design of the time aware web service recommendation system based on the personalized QoS parameters is very crucial and becomes a challenging research issue. In this study, LSTM based deep learning models were used for the prediction of these time aware QoS parameters and the results are compared with the previous approaches. The experimental results show that the LSTM based Time Series Forecasting Framework is performing better. The RMSE, MAE, and MAPE are used as an evaluation metric and their value for the prediction of Response time (RT) is found to be 0.030269, 0.02382 and 0.59773 respectively with adaptive moment estimation as the training option and is found to be 0.66988, 0.66465 and 27.9934 respectively with root mean square propagation as the training option. The RMSE, MAE, and MAPE value for the prediction of throughput (TP) is found to be 0.77787, 0.4792 and 1.61 respectively with adaptive moment estimation as the training option and is found to be 0.2.7087, 1.4076and 7.1559 respectively with root mean square propagation as the training option respectively. Thus, the experimental results show that the LSTM model of Time Series Forecasting for Web Services Recommendation Framework is performing better as compared to previous methods.

Keywords. Time-Aware web services recommendation, QoS-Prediction, LSTM, SMAC, cloud services.

1 Introduction

The convergence of SMAC technologies begins a new era known as the third computing paradigm [28, 31]. This era resulted in humongous growth of multifarious web services on the internet. These web services are invoked in a variety of applications ranging from social networking sites, mobility-based applications to a sensor-based application such as the Internet of Things, Internet of Vehicles [30], Healthcare Streams [29], etc.

The advent of cloud computing services leads to the further exponential growth of web services that are deployed on cloud platforms. In [27], authors emphasize the role of Information Security Management System (ISMS) standards in handling the security challenges faced by Cloud Service Providers (CSP) during cloud engineering. Web services evaluation in real time on a large-scale is a tedious task. From the user's point of view, Evaluation of Web services with user-dependent QoS properties are important to find out the optimal choice among them.

The performance of web services in a cloud computing setting has significantly affected by the Quality of Service which in turn depends on

various quality factors. These quality factors are categorized into two types of viz. Functional and non-functional performance parameters [24]. The non-functional performance parameter is further subdivided into two groups viz. user-independent and user-dependent. The user-independent parameters, like price and popularity, are not significant in making the recommendation of the web services for users. On the other hand, the user-dependent parameters such as response time, failure probability and throughput play a very crucial role in deciding the choice of the web services which can be a recommendation to the users [2, 8, 46, 51, 52]. Recommendation of appropriate web services is a multi-criteria decision process which makes it a very challenging research problem.

There exist various types of QoS-motivated research-based approaches which are listed in the Related Work section. One of the core challenges faced by cloud service providers is that the delivery of QoS is not made as declared due to the non-functional performance of web services correlated behavior with the invocation time. This is due to the reason that service status usually changes over time. This makes time aware personalized QoS prediction a very crucial research area for high-quality web service recommendation. To the best of our knowledge, none of the existing web service recommendation methods based on neighborhood-based CF considers the service invocation time. This is an important context factor affecting QoS Performance since web service performances usually are affected due to time-varying factors (e.g. service status, network condition, etc.)

To resolve this problem, this study utilized Long Short-Term Memory (LSTM) architecture with two deep learning neural network training options such as adaptive moment estimation [18] (ADAM) and root mean square propagation [26] (RMSProp). The RMSE, MAE, and MAPE are calculated at various Epoch through tuning of hyper parameters.

From a large number of real-world web services QoS values (response time and throughput), data is collected from 142 users who are accessing 4500 web services taken at 64-time slices where each time slice is of 15 minutes.

The experiments are conducted to find the best performing models for the prediction of response time and throughput and LSTM is found to be performing in the best manner. The current work is distributed into five sections. The first section introduces the problem and the related work is discussed in the second section. The third section discusses briefly about the data set and the experimentation. The fourth Section discusses the result and conclusion are given in the last section.

2 Related Work

2.1 Collaborative Filtering (CF)

It is extensively used in commercial recommender systems, such as Netflix and Amazon.com. In [34, 43, 50], CF based on user-item matrix concepts was used for recommendation systems. In [4, 3], the authors proposed a prediction method called empirical analysis for CF. They used several algorithms for predicting values based on correlation-coefficient, vector-based similarity calculation and statistical Bayesian methods. It has been shown that Bayesian networks perform better in terms of correlation-coefficient and vector-based similarity. They also used a ranking system for a product which was based on voting.

UPCC is a user based prediction algorithm using PCC. UPCC is employed to predict a QoS value y for the current time slot, it uses all other collected QoS values at the current time slot for prediction. In [39], the authors proposed a framework of imputation-boosted collaborative filtering (IBCF) using machine learning classifiers to fill-in the sparse user-item matrix, then it runs a traditional Pearson correlation coefficient based CF algorithm for this matrix to predict the rating. They did it by working with classifiers which are good in handling missing data, such as naïve Bayes, by use of predictive mean matching (PMM), with no content information.

They performed a comparative analysis of IBCF with 9 commonly-used machine learning classifiers and an ensemble classifier to impute the missing rating data, in terms of MAE.

2.2 Neighborhood-Based Collaborative Filtering Approach

These methods use correlation coefficients, vector-based similarity calculations, and statistical Bayesian methods. The drawback of Neighborhood-Based CF Approach is that it is vulnerable to data sparsity. In [3, 17] , User-Based methods that utilize historical QoS experience from a group of similar users were used to make QoS predictions. In [7, 32], Item Based methods that uses historical QoS information from similar services were utilized for QoS prediction. In [50, 49], the authors introduces a hybrid approach that combines the user based and service based methods which can achieve a higher QoS prediction accuracy. All the methods often uses the Pearson Correlation Coefficient (PCC) as their similarity models.

In [1], the authors discussed all other similarity computation methods e.g. cosine measure, adjusted cosine measure, constrained PCC, etc. of neighborhood-based CF approach. In [40, 14, 5], Model based CF approaches that incorporates training data to train a predefined model and then utilize that trained model were introduced to predict QoS. The most important model discussed are clustering models, aspect models, and latent factor models.

2.3 Matrix Factorization based Collaborative Filtering Approach

In [10], the authors considered neighborhood information for a Matrix Factorization (MF) based QoS predictor. In [23], the authors proposed an extended MF-based model by considering the location information in each historical QoS record. In [51], the authors proposed an MF-based model that integrates the time interval as an additional factor in an MF process and found that the prediction accuracy was found improving. In [43], the authors investigated the non-negative latent factor model to deal with the sparse QoS matrix subject to the non-negativity constraint. They also introduced Tikhonov regularization to obtain the regularized non-negative latent factor model. In [12], They proposed a method called location-based Hierarchical Matrix Factorization (HMF), which was used to provide service recommendations and makes a cluster of users and services on their location information. They used local matrix factorization and global matrix factorization to predict the missing QoS values. In [42], a general context-sensitive approach that took advantages of both implicit and explicit factors entailed in the QoS data through exploitation of contextual information was presented for collaborative QoS prediction. In [25, 35, 6], the authors discussed location as context information and it was deduced that geographically close users or services usually have similar experiences.

2.4 Service Recommendation System

In [22], the authors explored the use of two main machine learning approaches. Bayesian unsupervised method was utilized to cluster the data into classes and supervised decision tree-based classifier such as C4.5 to model the missing variables. In [8], the authors proposed a novel cloud service, selection through the missing value prediction and multi-attribute trustworthiness evaluation for user support. In [45], Non-negative Tensor Factorization (NTF) algorithm that works on three parameters like a user's-services-time model was used to predict the missing values of QoS performance and recommend Web services to the users. In [2], neural-network adapter models were used for Web Services response time prediction in cloud environments.

In [48, 53] , the authors combined item and user-based CF algorithms to recommend web services and also integrated Neighbourhood approach through MF. In [44], the authors presented an approach that integrates MF through decision tree learning to bootstrap service recommendation systems. In [6], the authors utilized a region-based CF algorithm web service recommendation. In [21], a comparative study of important seven machine learning algorithms was discussed in the prediction of QoS-values for Web service recommendations. Bagging and SMO-regression were performing well in the case of response time and throughput QoS datasets respectively. In [19], the authors applied an artificial neural network (ANN) approach for missing values imputation.

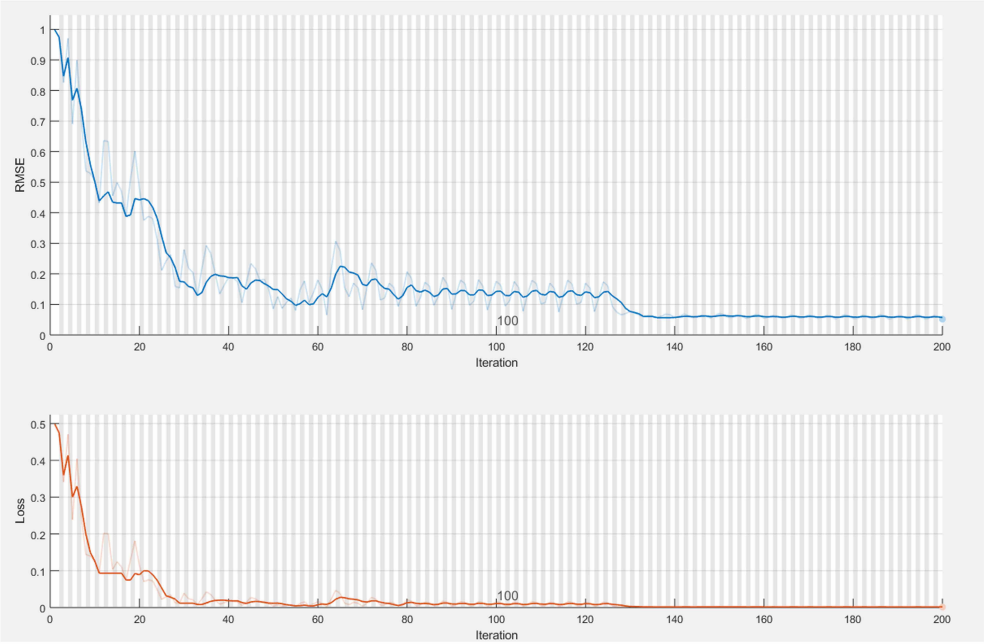

Fig. 1. Variation of Loss and RMSE Observed during Training at Epoch 200 for Response Time for training option ADAM

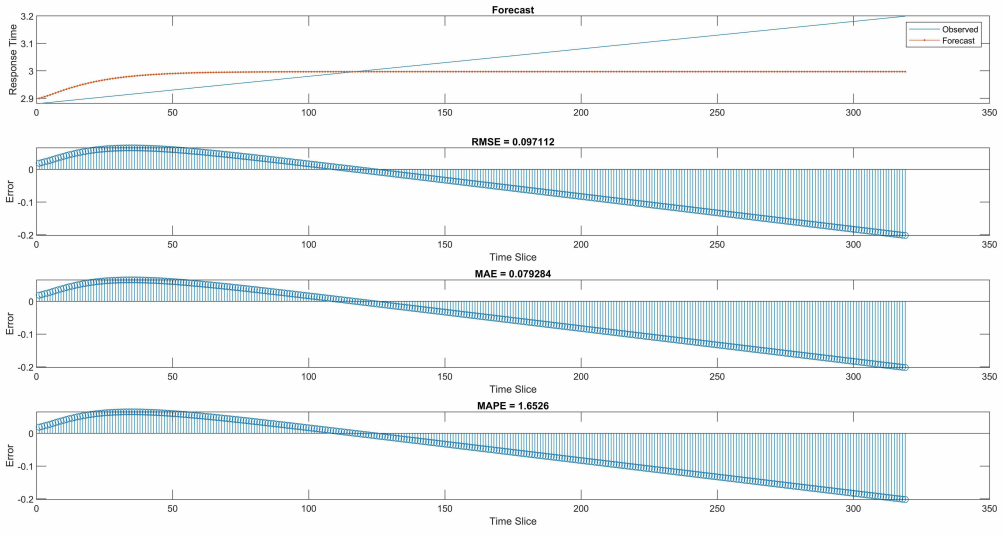

Fig. 2. Observed Vs Forecasted Cases at Epoch 200 for Response Time and associated RMSE, MAE, and MAPE for training option ADAM

Bayesian regularization (BR) was giving promising results for both datasets other than two important algorithms such as Levenberg Marquardt and scaled conjugate gradient. In [20], the authors have applied two major types of Meta learners, namely bagging and additive regression

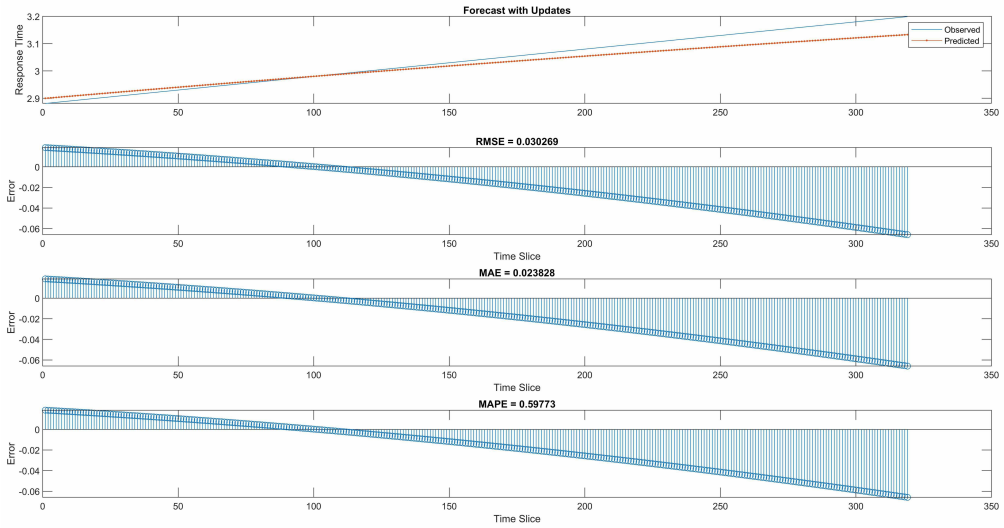

Fig. 3. Observed Vs Predicted Cases at Epoch 200 for Response Time and associated RMSE, MAE, and MAPE for training option ADAM

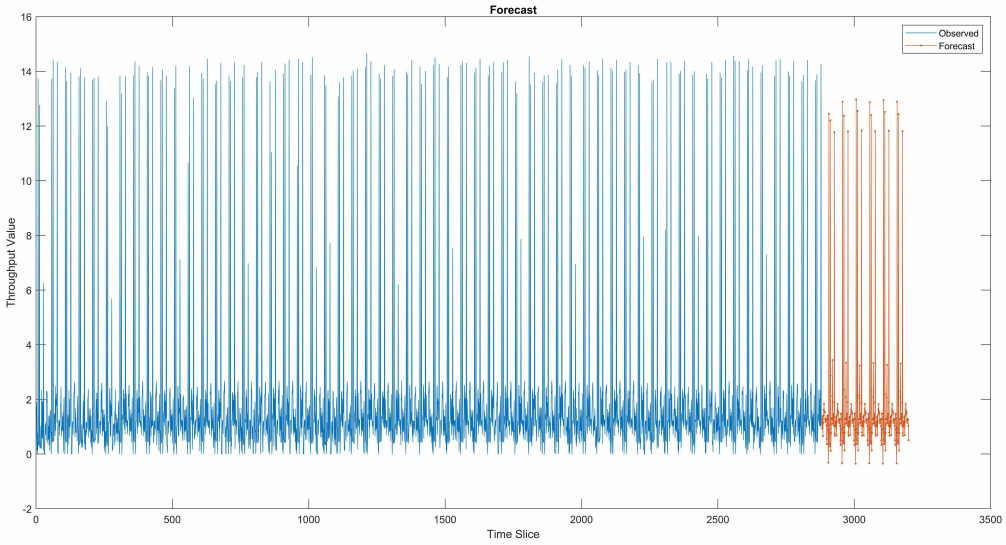

Fig. 4. Observed Vs Forecasted Throughput for training option ADAM

with a different combination of base learners for solving the problem of missing value imputation in QoS dataset consisting of response time and throughput values. Random forest algorithm was found performing relatively better than other base learners, for both bagging and additive regression.

2.5 Time Aware Web Service Recommendation System

In [15], the authors proposed a novel time aware approach for QoS Prediction that integrates time information into both the similarity measures. In [16], an improved time aware collaborative filtering

ISSN 2007-9737

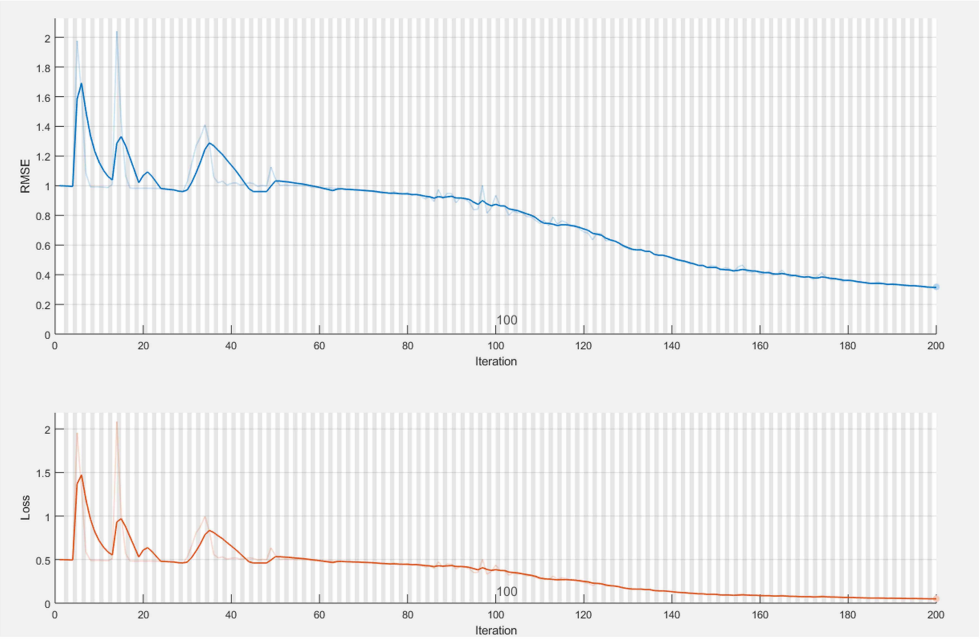

Fig. 5. Variation of Loss and RMSE Observed during Training at Epoch 200 for Throughput for training option ADAM

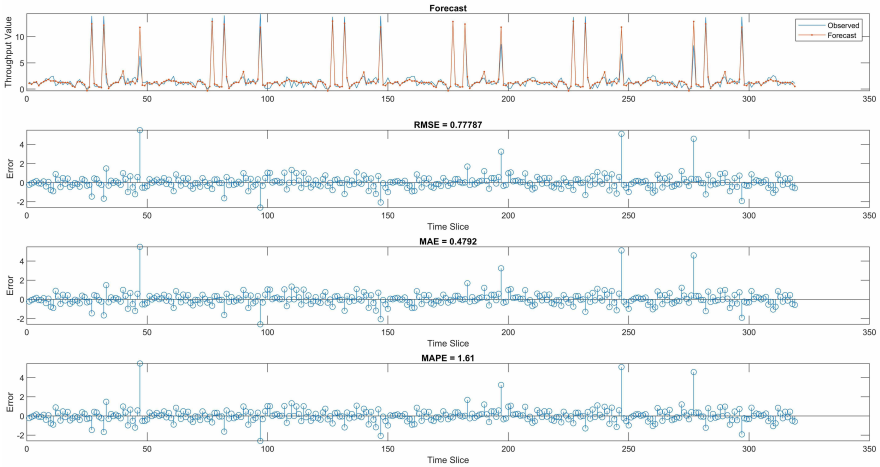

Fig. 6. Observed Vs Forecasted Cases at Epoch 200 for Throughput and associated RMSE, MAE, and MAPE for training option ADAM

approach that uses an applied hybrid personalized random walk to infer indirect user similarities and service similarities was proposed for high-quality QoS Prediction.

In [41], the authors proposed a novel spatial-temporal QoS prediction approach for time-aware Web Service recommendation by using ARIMA as the baseline method.

In [38], the authors used extraction of feature points of QoS sequences and dynamic time warping distance to compute the similarity instead of Euclidean distances for a personalized QoS

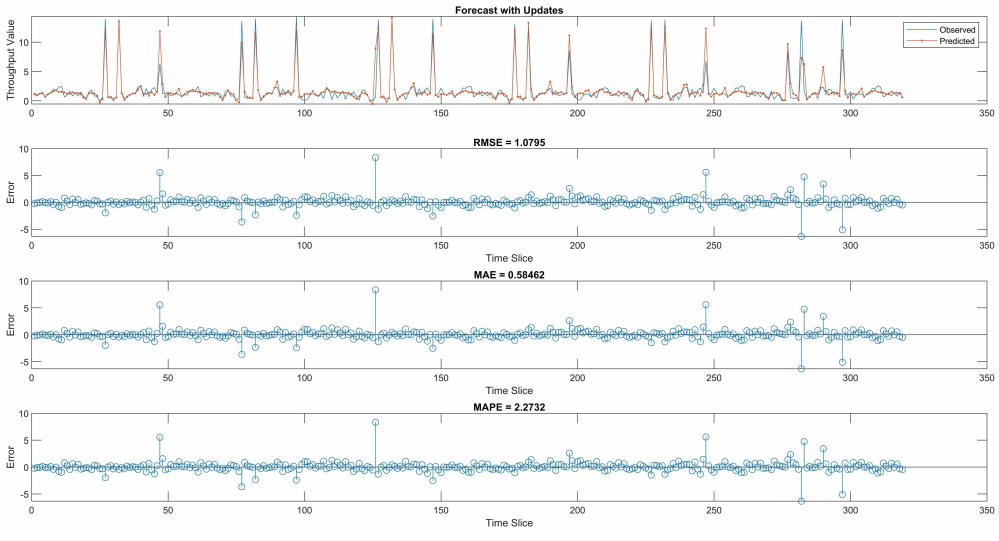

Fig. 7. Observed Vs Predicted Cases at Epoch 200 for Throughput and associated RMSE, MAE, and MAPE for training option ADAM

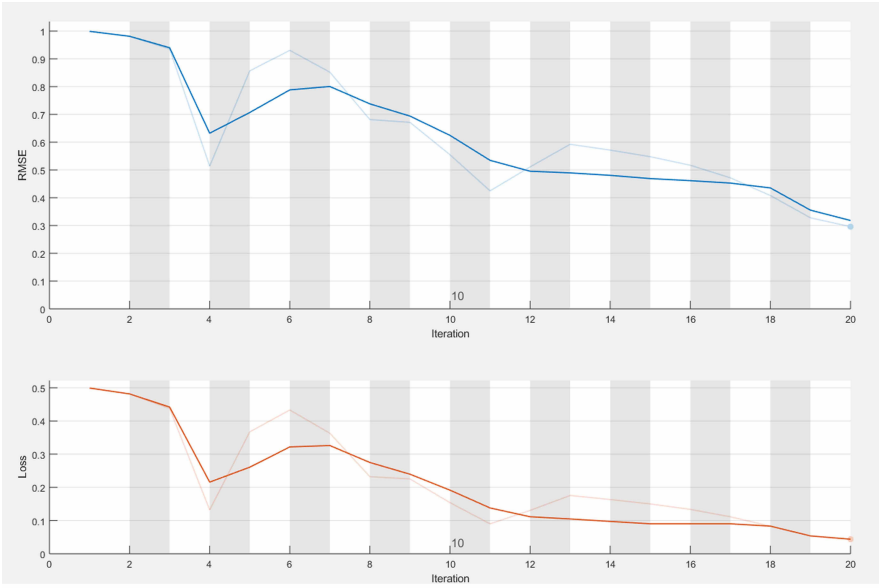

Fig. 8. Variation of Loss and RMSE Observed during Training at Epoch 200 for Response Time for training option RMSProp

prediction of dynamic web services. A web service QoS prediction framework WSPred was proposed to provide the time aware personalized QoS values prediction service for different service users in [47].

In [36, 37], authors experimented with econometrics model and wavelet enabled LSTM to explore the working in time aware recommendation systems.

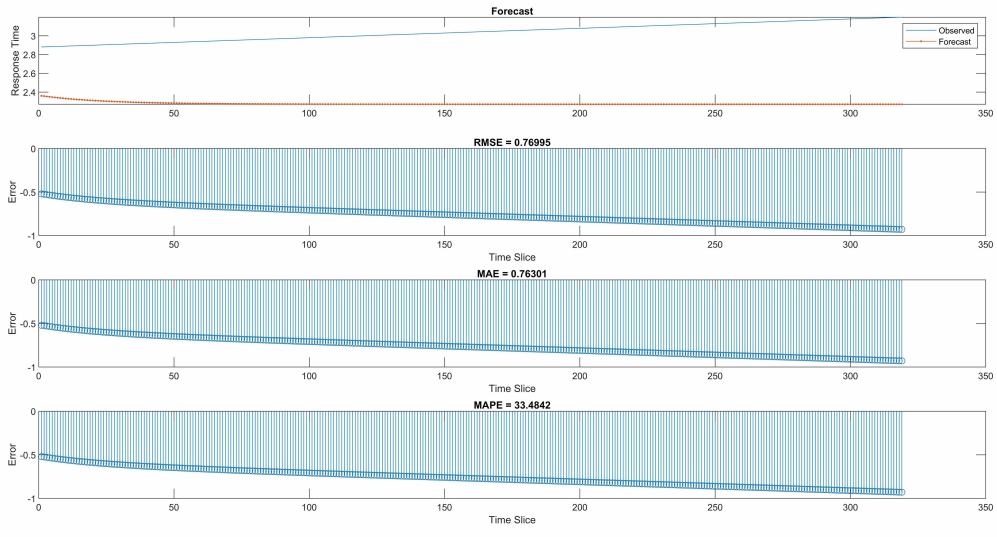

Fig. 9. Observed Vs Forecasted Cases at Epoch 200 for Response Time and associated RMSE, MAE, and MAPE for training option RMSProp

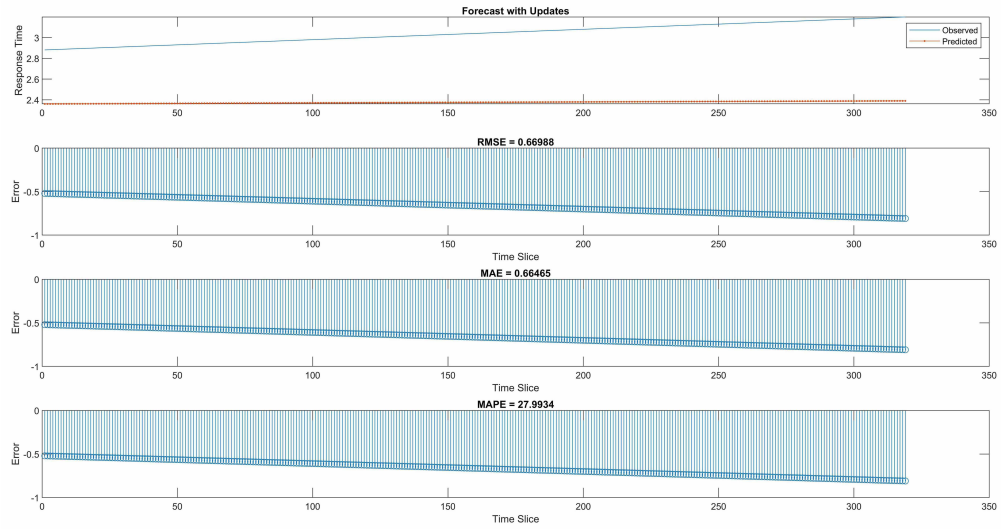

Fig. 10. Observed Vs Predicted Cases at Epoch 200 for Response Time and associated RMSE, MAE, and MAPE for training option RMSProp

3 Data Set and Experiments

The data set used in the current work is taken from the Web services linked dataset which was shaped by [2, 8, 46, 51, 52]. They set up a lab called PlanetLab[1] to collect these online invocations of Web services by active users in a cloud environment. The details about the dataset are given in Table 1. They have generated two matrices, which contains response time and

[1] http://www.planet-lab.org

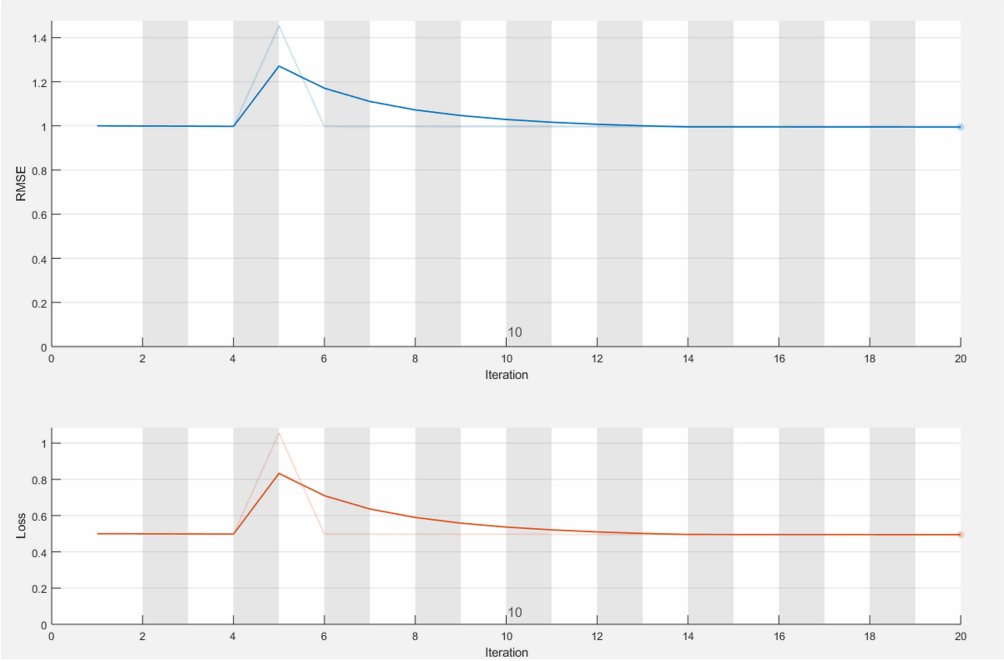

Fig. 11. Variation of Loss and RMSE Observed during Training at Epoch 200 for Throughput for training option RMSProp

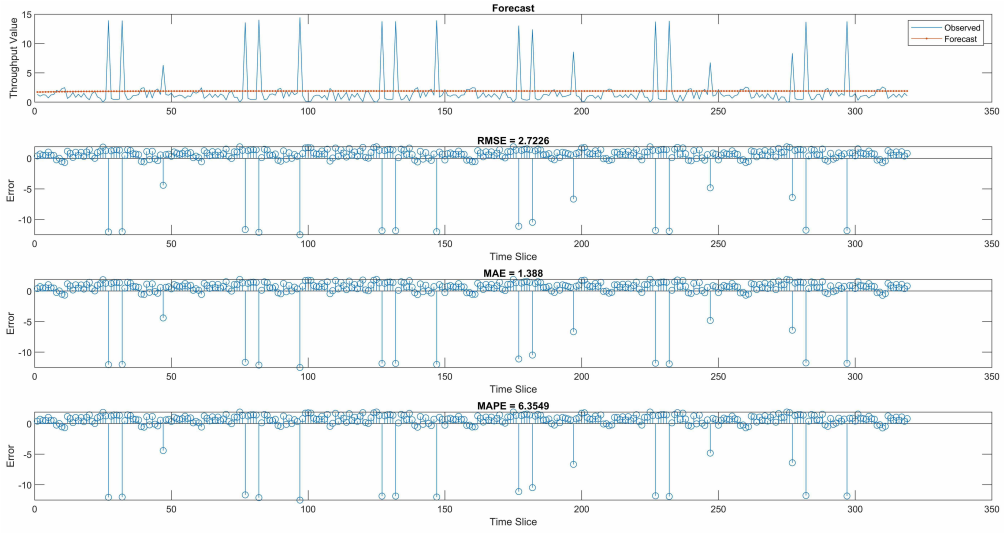

Fig. 12. Observed Vs Forecasted Cases at Epoch 200 for Throughput and associated RMSE, MAE, and MAPE for training option RMSProp

throughput values of different Web services for different users and are named as *rtmatrix* and *tpmatrix*. Here an element $RT_{i,j}$ in the *rtmatrix* indicates the value of response time of user i for Web service j and similarly the value of an element $TP_{i,j}$ in the *tpmatrix* indicates the throughput value of the user i for Web service j.

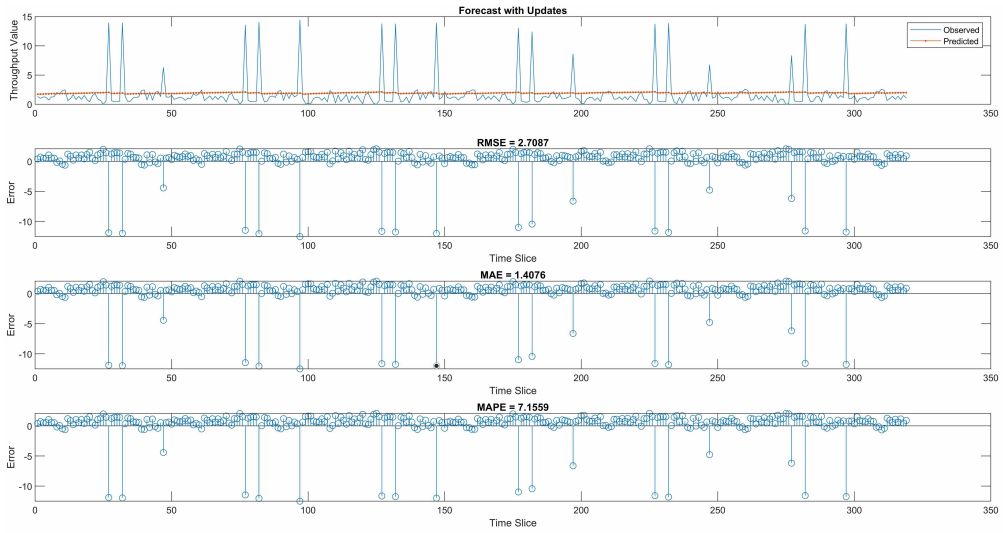

Fig. 13. Observed Vs Predicted Cases at Epoch 200 for Throughput and associated RMSE, MAE, and MAPE for training option RMSProp

Table 1. Descriptions of original dataset

Statistical parameters	Values
No. of Service Users	142
No. of Web-Services	4532
Time Slices	64(15 minutes)
No. of Web-Service Invocations	41186816
The range of Response time	0 – 20 seconds
Range of Throughput	0 – 1000 kbps

4 Results and Discussion

To access the performance of the proposed model, the following error measures are employed as Evaluation Criteria:

$$MAE = \frac{1}{N}\sum_{t=1}^{N}|d_t - y_t|, \tag{1}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(d_y - d_t)^2}, \tag{2}$$

$$MAPE = 100 \times \frac{1}{N}\sum_{t=1}^{N}\left|\frac{d_t - y_t}{d_t}\right|. \tag{3}$$

Table 2. Hyperparameters of the LSTM

Hyperparameters	Training Options
Hidden Units =200	Adaptive
Dropout layer =0.2	Moment
Gradient Threshold =1	Estimation
Initial Learn Rate =0.005	(ADAM)
Learn Rate Drop Factor = 0.2	
Learn Rate Drop Period =125	
Hidden Units =200	Root Mean
Dropout layer =0.2	Square
Initial Learn Rate =3e-4	Propagation
Squared Gradient Decay =0.99	(RMSProp)
Mini Batch Size =64	

In the above formula, d_t represents the original value of the t moment, y_t represents the predicted value of the t moment and N represents the total number of the samples. Smaller the value of *MAE*, *RMSE*, and *MAPE*, smaller is the deviation

Table 3. Prediction error of the model at 200 epoch for response time

Epoch	Model	MAE	RMSE	MAPE
200	ADAM LSTM (Observed Vs Forecast)	0.07928	0.09711	1.65260
200	ADAM LSTM (Observed Vs Predicted)	0.02382	**0.03026**	0.59773
200	RMS Optimizer LSTM (Observed Vs Forecast)	0.76301	0.76995	33.4842
200	RMS Optimizer LSTM (Observed Vs Predicted)	0.66465	**0.66988**	27.9934

between the predicted value and the original value. *MAPE* is being used as the main predictor because of its stability.

The tuning of hyperparameters is described in Table 2. The result of the experiment is given in the Table 3 and Table 4. It is quite clear from the results that LSTM produces best result for the prediction of response time as well as throughput as compared to the previous approaches as mentioned in Table 5.

The experiment is conducted with training options ADAM and RMSProp. The experiment is carried out by standardizing the data to avoid the data from diverging and then LSTM network architecture is created to train the LSTM network for forecasting the future time steps.

In case of ADAM as a training option, this is represented in Figure 1 and Figure 5 where the variation of Loss and RMSE can be observed. The Network state with Predicted value and observed values is represented in Figure 2, Figure 3 for response time and Figure 6, Figure 7 for throughput. Figure 4 represents the observed and forecasted throughput.

In case of RMSProp as a training option, LSTM architecture is represented in Figure 8 and Figure 11 where the variation of Loss and RMSE can be observed. The Network state with Predicted value and ob-served values is represented in Figure 9, Figure 10 for response time and Figure 12, Figure 13 for throughput.

The network state is updated with observed values and found that the predictions are more accurate when updating the network state with the observed values instead of the predicted values.

Table 4. Prediction error of the model at 200 epoch for throughput

Epoch	Model	MAE	RMSE	MAPE
200	ADAM LSTM (Observed Vs Forecast)	0.4792	**0.7778**	1.61
200	ADAM LSTM (Observed Vs Predicted)	0.58402	1.0795	2.2732
200	RMS Optimizer LSTM (Observed Vs Forecast)	1.388	2.7226	6.3549
200	RMS Optimizer LSTM (Observed Vs Predicted)	1.4076	**2.7087**	7.1559

In case of time series and sequence data, long term dependencies play a crucial role which makes an LSTM [13, 9, 11, 33] a wonderful choice in which layer learns through long-term dependencies in data. There is an additive interaction among the layers which improves the gradient flow very long sequences during training. The state of the layer comprises the hidden state also known as output state and the cell state. The cell state stores the information learned from the past time steps.

Figure 14 elucidates the flow of data at time step t. At every time step, the layer either adds the information to or removes the information from the cell state. The hidden state at time step t stores the output of the LSTM layer for that time step. These updates are controlled using gates. The roles of the gates are mentioned below:

Table 5. Comparison with the previous works

	MAE	RMSE	Reference
WSPred (RT)	2.1266	3.8943	[47]
WSPred (RT)	6.8558	36.572	[47]
AVG at 75 %	1.159	3.206	[49]
UPCC at 75 %	1.464	3.026	[4, 3]
UPCC* at 75 %	1.242	2.763	[41]
IPCC at 75 %	1.374	2.923	[32]
IPCC* at 75 %	1.202	2.717	[32]
WSRec at 75%	1.372	2.923	[49]
WSRec* at 75%	1.202	2.721	[41]
ARIMA	1.028	2.986	[41]
LS, K=1	1.006	2.774	[41]
	1.102	3.062	[41]
LASSO, K=1	0.997	2.735	[41]
	0.89	2.538	[41]
	RT-0.0635 at 35 %	RT-0.0692 at 35 %	[38]
	TP-4.4101 at 40 %	TP-5.1943 at 40 %	

Components	Purpose
Input gate (i)	Cell State Update is controlled through this gate.
Forget gate (f)	Cell State Reset or Forget is controlled here.
Cell candidate (g)	Information to the cell state is added here.
Output gate (o)	The level of cell state that is added to the hidden state is controlled here.

The weights that can be learned in an LSTM layer are the input weights W, the recurrent weights R and the bias b. The concatenated input weights are represented as matrices W, R and b. The state of the cell at a given time step t is provided as:

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t, \tag{4}$$

where $\odot$ designates the Hadamard product, which is element-wise multiplication of the vectors.

$$W = \begin{pmatrix} W_i \\ W_f \\ W_g \\ W_o, \end{pmatrix} \quad R = \begin{pmatrix} R_i \\ R_f \\ R_g \\ R_o, \end{pmatrix} \quad b = \begin{pmatrix} b_i \\ b_f \\ b_g \\ b_o. \end{pmatrix}$$

The state of the hidden at a given time step t is provided as:

$$h_t = o_t \odot \sigma_c \left(c_t \right), \tag{5}$$

here σ_c designates the state activation function. By default, the hyperbolic tangent function $(tanh)$ is used in the LSTM layer function to calculate the state activation function.

The component at a given time t is described as following:

Here, σ_g designates the gate activation function. The sigmoid function $\sigma(x) = (1 + e^{-x})^{-1}$ is used to calculate the date activation function.

5 Conclusions

In the current work, time information is incorporated into web service recommendation for the

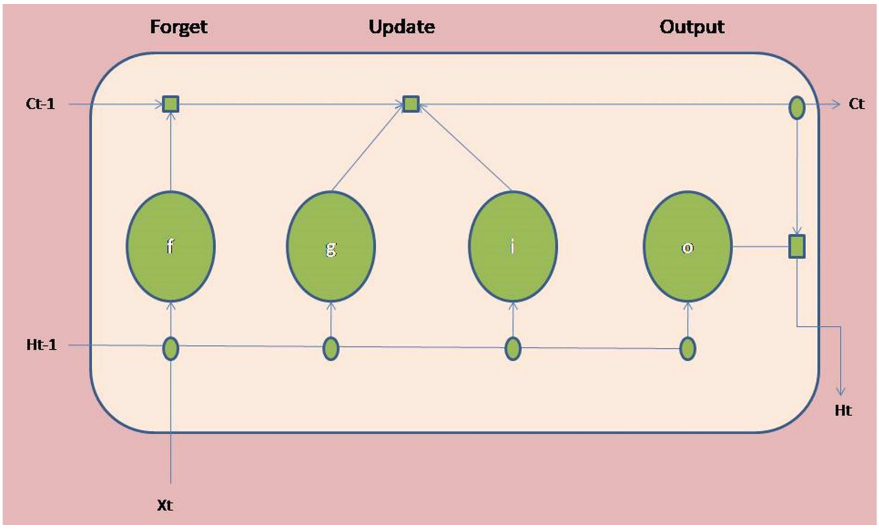

Fig. 14. The flow of the data in the LSTM network

Component	Formula
Input gate	$i_t = \sigma_g(W_i x_t + R_i h_{t-1} + b_i)$
Forget gate	$f_t = \sigma_g(W_f x_t + R_f h_{t-1} + b_f)$
Cell candidate	$g_t = \sigma_c(W_g x_t + R_g h_{t-1} + b_g)$
Output gate	$o_t = \sigma_g(W_o x_t + R_o h_{t-1} + b_o)$

prediction of QoS parameters namely, response time and throughput accurately. This forms a series which is a kind of nonlinear and non-stationary. Thus, efficient processing of the data is required. Effective preparation and processing strategies should take this nonlinear and non-stationary behavior into account. The effectiveness of the proposed approach is being validated by thorough experiments.

After standardizing the data to avoid the data from diverging, LSTM network architecture is created to train the LSTM network for forecasting the future time steps. The network state is updated with observed values and found that the predictions are more accurate when updating the network state with the observed values instead of the predicted values.

The LSTM gave the best RMSE in case of Response time as **0.030269** with ADAM as the training option and **0.66988** with RMSProp as the training option. In case of throughput, the LSTM gave the best RMSE value as **0.77787** with ADAM as the training option and **2.7087** with RMSProp as the training option.

In future work, context also needs to be explored in time aware web services recommendation through QoS prediction.

References

1. **Ahn, H. J. (2008).** A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Information Sciences*, Vol. 178, No. 1, pp. 37–51.
2. **Albu, R.-D., Felea, I., & Popentiu-Vladicescu, F. (2013).** On the best adaptive model for web services response time prediction. *2013 20th International Conference on Systems, Signals and Image Processing (IWSSIP)*, IEEE, pp. 39–42.
3. **Breese, J. S., Heckerman, D., & Kadie, C. (1998).** Empirical analysis of predictive algorithms for collaborative filtering. *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 43–52.
4. **Breese, J. S., Heckerman, D., & Kadie, C. (1998).** An experimental comparison of collaborative filtering methods. *Technical Report MSR-TR*.

5. **Canny, J. (2002).** Collaborative filtering with privacy via factor analysis. *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '02*, ACM Press, New York, New York, USA, pp. 238.
6. **Chen, X., Zheng, Z., Yu, Q., & Lyu, M. R. (2014).** Web service recommendation via exploiting location and QoS information. *IEEE Transactions on Parallel and Distributed Systems*, Vol. 25, No. 7, pp. 1913–1924.
7. **Deshpande, M. & Karypis, G. (2004).** Item-based top- n recommendation algorithms. *ACM Transactions on Information Systems*, Vol. 22, No. 1, pp. 143–177.
8. **Ding, S., Xia, C.-Y., Zhou, K.-L., Yang, S.-L., & Shang, J. S. (2014).** Decision support for personalized cloud service selection through multi-attribute trustworthiness evaluation. *PLoS ONE*, Vol. 9, No. 6, pp. e97762.
9. **Glorot, X. & Bengio, Y. (2010).** Understanding the difficulty of training deep feedforward neural networks. **Teh, Y. W. & Titterington, M.**, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, PMLR, Chia Laguna Resort, Sardinia, Italy, pp. 249–256.
10. **Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992).** Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, Vol. 35, No. 12, pp. 61–70.
11. **He, K., Zhang, X., Ren, S., & Sun, J. (2015).** Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *2015 IEEE International Conference on Computer Vision (ICCV)*, IEEE, pp. 1026–1034.
12. **He, P., Zhu, J., Zheng, Z., Xu, J., & Lyu, M. R. (2014).** Location-based hierarchical matrix factorization for web service recommendation. *2014 IEEE International Conference on Web Services*, IEEE, pp. 297–304.
13. **Hochreiter, S. & Schmidhuber, J. (1997).** Long short-term memory. *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780.
14. **Hofmann, T. (2004).** Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems*, Vol. 22, No. 1, pp. 89–115.
15. **Hu, Y., Peng, Q., & Hu, X. (2014).** A time-aware and data sparsity tolerant approach for web service recommendation. *2014 IEEE International Conference on Web Services*, IEEE, pp. 33–40.
16. **Hu, Y., Peng, Q., Hu, X., & Yang, R. (2015).** Time aware and data sparsity tolerant web service recommendation based on improved collaborative filtering. *IEEE Transactions on Services Computing*, Vol. 8, No. 5, pp. 782–794.
17. **Jin, R., Chai, J. Y., & Si, L. (2004).** An automatic weighting scheme for collaborative filtering. *Proceedings of the 27th annual international conference on Research and development in information retrieval - SIGIR '04*, ACM Press, New York, New York, USA, pp. 337.
18. **Kingma, D. P. & Ba, J. (2014).** Adam: A method for stochastic optimization. *arXiv 1412.6980*.
19. **Kumar, S., Pandey, M. K., Nath, A., & Subbiah, K. (2015).** Missing QoS-values predictions using neural networks for cloud computing environments. *2015 International Conference on Computing and Network Communications (CoCoNet)*, IEEE, pp. 414–419.
20. **Kumar, S., Pandey, M. K., Nath, A., & Subbiah, K. (2016).** Performance analysis of ensemble supervised machine learning algorithms for missing value imputation. *2016 2nd International Conference on Computational Intelligence and Networks (CINE)*, IEEE, pp. 160–165.
21. **Kumar, S., Pandey, M. K., Nath, A., Subbiah, K., & Singh, M. K. (2015).** Comparative study on machine learning techniques in predicting the QoS-values for web-services recommendations. *International Conference on Computing, Communication & Automation*, IEEE, pp. 161–167.
22. **Lakshminarayan, K., Harp, S. A., Goldman, R., & Samad, T. (1996).** Imputation of missing data using machine learning techniques. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, AAAI Press, pp. 140–145.
23. **Lo, W., Yin, J., Deng, S., Li, Y., & Wu, Z. (2012).** An extended matrix factorization approach for QoS prediction in service selection. *2012 IEEE Ninth International Conference on Services Computing*, IEEE, pp. 162–169.
24. **Menasce, D. (2002).** QoS issues in web services. *IEEE Internet Computing*, Vol. 6, No. 6, pp. 72–75.
25. **Mingdong Tang, Yechun Jiang, Jianxun Liu, & Xiaoqing Liu (2012).** Location-aware collaborative filtering for QoS-based service recommendation.

2012 IEEE 19th International Conference on Web Services, IEEE, pp. 202–209.

26. **Murphy, K. (2012).** *Machine Learning: A Probabilistic Perspective*. The MIT Press.
27. **Pandey, M., Kumar, S., & Subbiah, K. (2013).** Information security management system (isms) standards in cloud computing-a critical review. *International Conference on Control, Computing, Communication and Materials, Allahabad*.
28. **Pandey, M. & Subbiah, K. (2017).** Performance analysis of time series forecasting of ebola casualties using machine learning algorithms. *ITISE 201, Granada*, pp. 885–898.
29. **Pandey, M. K. & Subbiah, K. (2016).** A novel storage architecture for facilitating efficient analytics of health informatics big data in cloud. *2016 IEEE International Conference on Computer and Information Technology (CIT)*, IEEE, pp. 578–585.
30. **Pandey, M. K. & Subbiah, K. (2016).** Social networking and big data analytics assisted reliable recommendation system model for internet of vehicles. In *Internet of Vehicles – Technologies and Services*. Springer International Publishing, pp. 149–163.
31. **Pandey, M. K. & Subbiah, K. (2018).** Performance analysis of time series forecasting using machine learning algorithms for prediction of ebola casualties. In *Applications of Computing and Communication Technologies*. Springer Singapore, pp. 320–334.
32. **Sarwar, B., Karypis, G., Konstan, J., & Reidl, J. (2001).** Item-based collaborative filtering recommendation algorithms. *Proceedings of the tenth international conference on World Wide Web - WWW '01*, ACM Press, New York, New York, USA, pp. 285–295.
33. **Saxe, A. M., McClelland, J. L., & Ganguli, S. (2013).** Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv 1312.6120*.
34. **Shao, L., Zhang, J., Wei, Y., Zhao, J., Xie, B., & Mei, H. (2007).** Personalized QoS prediction forweb services via collaborative filtering. *IEEE International Conference on Web Services (ICWS 2007)*, IEEE, pp. 439–446.
35. **Shen, Y., Zhu, J., Wang, X., Cai, L., Yang, X., & Zhou, B. (2013).** Geographic location-based network-aware QoS prediction for service composition. *2013 IEEE 20th International Conference on Web Services*, IEEE, pp. 66–74.
36. **Singh, V. P., Pandey, M. K., Singh, P. S., & Subbiah, K. (2019).** An econometric time series forecasting framework for web services recommendation. *Procedia Computer Science*.
37. **Singh, V. P., Pandey, M. K., Singh, P. S., & Subbiah, K. (2019).** An empirical mode decomposition EMD enabled long short term memory LSTM based time series forecasting framework for web services recommendation. *Frontiers in Artificial Intelligence and Applications*, Vol. 320, pp. 715–723.
38. **Song, Y., Hu, L., & Yu, M. (2018).** A novel QoS-aware prediction approach for dynamic web services. *PLOS ONE*, Vol. 13, No. 8, pp. e0202669.
39. **Su, X., Khoshgoftaar, T. M., Zhu, X., & Greiner, R. (2008).** Imputation-boosted collaborative filtering using machine learning classifiers. *Proceedings of the 2008 ACM symposium on Applied computing - SAC '08*, ACM Press, New York, New York, USA, pp. 949.
40. **Tsai, C.-F. & Hung, C. (2012).** Cluster ensembles in collaborative filtering recommendation. *Applied Soft Computing*, Vol. 12, No. 4, pp. 1417–1425.
41. **Wang, X., Zhu, J., Zheng, Z., Song, W., Shen, Y., & Lyu, M. R. (2016).** A spatial-temporal QoS prediction approach for time-aware web service recommendation. *ACM Transactions on the Web*, Vol. 10, No. 1, pp. 1–25.
42. **Wu, H., Yue, K., Li, B., Zhang, B., & Hsu, C.-H. (2018).** Collaborative QoS prediction with context-sensitive matrix factorization. *Future Generation Computer Systems*, Vol. 82, pp. 669–678.
43. **Xin Luo, MengChu Zhou, YunNi Xia, & Qing-Sheng Zhu (2014).** Predicting web service QoS via matrix-factorization-based collaborative filtering under non-negativity constraint. *2014 23rd Wireless and Optical Communication Conference (WOCC)*, IEEE, pp. 1–6.
44. **Yu, T., Zhang, Y., & Lin, K.-J. (2007).** Efficient algorithms for web services selection with end-to-end QoS constraints. *ACM Transactions on the Web*, Vol. 1, No. 1, pp. 6–es.
45. **Zhang, W., Sun, H., Liu, X., & Guo, X. (2014).** Temporal QoS-aware web service recommendation via non-negative tensor factorization. *Proceedings of the 23rd international conference on World wide web - WWW '14*, ACM Press, New York, New York, USA, pp. 585–596.
46. **Zhang, Y., Zheng, Z., & Lyu, M. R. (2011).** Bftcloud: A byzantine fault tolerance framework

for voluntaryrresource cloud computing. *2011 IEEE 4th International Conference on Cloud Computing*, IEEE, pp. 444–451.

47. **Zhang, Y., Zheng, Z., & Lyu, M. R. (2011).** Wspred: A time-aware personalized QoS prediction framework for web services. *2011 IEEE 22nd International Symposium on Software Reliability Engineering*, IEEE, pp. 210–219.
48. **Zheng, Z. & Lyu, M. R. (2009).** A QoS-aware fault tolerant middleware for dependable service composition. *2009 IEEE/IFIP International Conference on Dependable Systems & Networks*, IEEE, pp. 239–248.
49. **Zheng, Z., Ma, H., Lyu, M. R., & King, I. (2009).** Wsrec: A collaborative filtering based web service recommender system. *2009 IEEE International Conference on Web Services*, IEEE, pp. 437–444.
50. **Zheng, Z., Ma, H., Lyu, M. R., & King, I. (2011).** QoS-aware web service recommendation by collaborative filtering. *IEEE Transactions on Services Computing*, Vol. 4, No. 2, pp. 140–152.
51. **Zheng, Z., Ma, H., Lyu, M. R., & King, I. (2013).** Collaborative web service QoS prediction via neighborhood integrated matrix factorization. *IEEE Transactions on Services Computing*, Vol. 6, No. 3, pp. 289–299.
52. **Zheng, Z., Zhang, Y., & Lyu, M. R. (2010).** Cloudrank: A QoS-driven component ranking framework for cloud computing. *2010 29th IEEE Symposium on Reliable Distributed Systems*, IEEE, pp. 184–193.
53. **Zheng, Z., Zhang, Y., & Lyu, M. R. (2010).** Distributed QoS evaluation for real-world web services. *2010 IEEE International Conference on Web Services*, IEEE, pp. 83–90.

Article received on 29/10/2019; accepted on 07/03/2020.
Corresponding author is Manish Kumar Pandey.