International Journal of Computer Application (2250-1797)
Volume 5– No. 6, October 2015

Performance Assessment of Web service: A Neural Network Approach

S.Uma[#1], D. Evangelin Geetha[#2], T.V. Suresh Kumar[#2]
#1 Department of Computer Applications
B.M.S. College of Engineering
Bangalore, India
9880253599
umasankartv@gmail.com
#2 Department of Computer Applications
MSRIT, Bangalore, India
9620293370
degeetha@msrit.edu
#2 Department of Computer Applications
MSRIT, Bangalore, India
9449435497
tvsureshkumar@msrit.edu

ABSTRACT

Performance is a key quality measure of a web service. The response time plays a key role in improving the performance of a web service. It is very important for the service providers to know in advance the response time of a web service before its actual deployment as it has direct impact in fulfilling the user request. It is always necessary for the service providers to predict the effects of possible performance barriers on servers before the arrival of actual service requests.To guarantee good performance in the system, the run-time critical response time variations have to be predicted in advance and hence proper corrective measure can be taken and the load of the system can be well balanced. This will prevent the system from major performance loss. In this paper, we have proposed an agent based prediction technique for predicting the service response time. The predictor agent uses Non Linear Auto Regressive with External Input (NARX) time series forecasting technique based on Neural Networks to predict web service response time. The result is validated by comparing the predicted values with the actual values using Mean Squared Error (MSE) method and Regression Value.

Key words: Web Services, Performance, Response Time, Non Linear Auto Regressive with External Input (NARX), Agent, Neural Networks

Corresponding Author: S. Uma

1. INTRODUCTION

The popularity of the web service to a great extent depends on the Quality of Service. It is very much essential to identify the bottlenecks affecting performance of Web services. The service response time is considered as one of the significant performance parameter for the system. In spite of meeting the functional requirements, many systems fail because the quality attribute such as response time is not met. Hence, the assessment of performance and corresponding prediction techniques prior to actual service request is important.

Effective planning for balancing the incoming requests between web servers is very much required to minimize the response time of service requests. Prediction of service response time based on past behaviour is considered as one of the important parameter for the load balancer. Understanding the possible future behaviour of the system is vital to prevent the system from damages. Multi agent system (MAS) offers a proficient computing environment for enhancing the QoS (Quality of Service) of web services. A predictor agent is used to predict the future response time for the service based on the previous response time values. If there is a significant increase in the average response time from the periodically measured earlier response time values, then corrective measure can be taken immediately before further performance degradation.

Neural network techniques are very much appropriate for Time Series prediction. The network is trained using multi-layer feed forward back propagation algorithm to test its performance. The backpropagation (BP) algorithm was introduced by Rumelhart [13].This algorithm is the popular technique for training a multilayer feed-forward artificial neural networks. It uses the gradient descent algorithm. In the basic BP algorithm the weights are adjusted in the steepest descent direction. By examining the past behaviour of time series, neural network model can make prediction precisely about the future. The main purpose of using neural networks for prediction is its ability to automatic learning of dependencies only from measured data without any need for further information. The neural network can be trained from the historical data and are capable of gathering hidden and strongly non-linear dependencies, even when there is a considerable noise in the training set[14].

In this work, Predictor Agent employ Non Linear Auto Regressive with External Input (NARX) time series technique for predicting the response time of web service as this is a suitable model well established for modeling nonlinear systems and in particular time series.

The context of our work related to the works of other researchers is provided in section 2. The proposed methodology is described in section 3. The algorithm for the proposed methodology is illustrated in Section 4. Section 5 discusses the simulation model. The simulated results with graphs are detailed in section 6. The paper concludes with a summary of the work and future research directions.

2. RELATED WORK

For Web services, retaining good QoS is a major challenge and this is one of the big issues for web services research community. The QoS Requirements for good quality Web Services are discussed in [16]. In [10], the authors have proposed a QoS prediction approach to predict the missing QoS values for QoS-based selection of web services. A-Cosine equation has been used to compute the service-based similarity and a data smoothing process is added to improve the prediction accuracy. In [11], a personalized QoS prediction approach is proposed which takes the similarity among consumer's experience into consideration and the consumers who have similar historical experiences on some services will have similar experience on other services. This approach may not be appropriate for all kind of users.

The key purpose of time series modeling is to carefully collect and thoroughly study the past observations of a time series to develop a suitable model which describes the intrinsic structure of the series. This model is then used to generate future values for the series, i.e. to make forecasts. The authors in [1] have described three important classes of time series models, viz. the stochastic, neural networks and SVM based models, together with their inherent forecasting strengths and weaknesses.

Non Linear Auto Regressive with External Input is a powerful class of models. It has been stated that they are appropriate for modelling nonlinear systems and time series [4]. Compared to the Non Linear Auto Regressive model(NAR), the Non Linear Auto Regressive with External Input produce more precise result. In NARX model, more than one input can be fed to the neural network. Some important qualities about NARX networks with gradient-descending learning gradient algorithm have been reported: (1) Learning is more effective in NARX networks than in other neural network (the gradient descent is better in NARX) and (2) these networks converge much faster and generalize better than other networks[15] [17].

Agents are software programs capable of analysing the requests and can determine how to fit it with a Web Service. A multi agent based framework for the provision of semantic web services is proposed in [6]. The selection of web services using software agents is discussed in [5,8,12].

In order to use a Web service a Service Level Agreement(SLA) is usually made between the service Provider and the Service Consumer and Response time is considered as one of the important quality attribute for a web service. The variation in response time between service requests have to be predicted in advance so that severe performance loss of the system can be avoided and the response time can be maintained within the acceptable range as specified in the SLA. From the literature, we understand that there is a need for an agent based web services QoS prediction technique especially for predicting the response time. Hence we have proposed a Neural Network based Approach for the Performance Assessment of Web service.

3. METHODOLOGY

In the present work, the past values of arrival time and response time are used to predict the future values of response time of a web service. Evaluation of input data is an important part of the validation of data assumption. The simulation tool Simulation of Multi Tier Queuing Applications (SMTQA) is used to validate the input data [3]. Non Linear Auto Regressive with External Input (NARX) dynamic neural networks technique is used for prediction. The input data is divided into three sets. The first set is used for training the network. The second set of input data is used for testing and the third set of data is used for validation. The network is trained by means of training set data. Since the actual data is available during training, the network is trained in Open loop form. During the training phase, the desired output error and the actual output error are measured. The aim of the training phase is to decrease the error by adjusting the weight. Hence, during training, the error should be minimized and the training will continue until the error becomes negligible. Next, using the test set, the prediction ability of the network is measured. After the prediction of future response time, the predicted response time values have to be validated. Mean Squared Error which is the average squared difference between predicted outputs and actual values is used for validating the predicted values of future response time. The Regression Value (R) is used to find the correlation between the predicted and actual values. The various graphs are plotted showing the Response time for training target and training response; Response time for validation target and validation response; Response time for testing target and testing response; Regression for training; Regression for validation; Regression for testing. The trained network is converted to closed loop form and the network is used for predicting the future response time values. The graphs are plotted with the actual and predicted response time.

4. ALGORITHM

For *each server*
{
Create a Nonlinear Autoregressive Network with past Arrival time and response time of web service as External Input using Predictor Agent;
Assign *x(t) = Past* **d** *arrival time of service requests;*
Assign *y(t) = Past* **d** *response time of service requests;*
Prepare the Data for Training and Simulation with random data division method for Training, Validation and Testing;
Define a NARX neural network with number of hidden neuron and number of delays;
Calculate y(t) = f(x(t-1), ...x(t-d),y(t-1)...y(t-d));
Train the Network using Levenberg-Marquardt back propagation method in open loop form;
if *(no improvement in generalization)*
Stop training;
Calculate Mean Squared Error[MSE] = Average [predicted outputs – actual targets]^2 ;
Measure the correlation R between predicted outputs and actual target;
If *!((MSE is closer to zero) && (R value is closer 1))*
{
Adjust the network weight;
retrain the network;
}
else
{
Plot Response time graph for training target and training response;
Plot Response time graph for validation target and validation response;
Plot Response time graph for testing target and testing response;
Plot regression graph for training;
Plot regression graph for validation;
Plot regression graph for testing;
}
Convert the network to closed form for future prediction of response time;
Predict the next value of y(t) from previous values of y(t) and x(t);
Plot the response time of actual and predicted values;
}

In the proposed model, we employ the Multi-layer perceptron (MLP) networks trained using BP algorithm [13] to predict the response time of web service. The MLP has input layer, hidden layer and output layer. The input layer corresponds to the problem input variables with one node for each input variable. In our model, we have used arrival time and response time as input variables. The hidden layer is used to find the non-linear relationships among the arrival time and response time. The output layer is used to provide predicted values of response time. Levenberg-Marquardt algorithm has been shown to be the fastest method for training feed-forward neural networks. The application of Levenberg-Marquardt to neural network training is described in [7,9]. In general, in function approximation problems, for networks that comprise up to a few hundred weights, the Levenberg-Marquardt algorithm has the fastest convergence. This quality is particularly useful when accurate training is required.

The network is trained using *trainlm* function in Matlab. Backpropagation is applied to calculate the Jacobian *jX* of performance with respect to the weight and bias variables *X*. Each variable is adjusted according to Levenberg-Marquardt equation,

$$jj = jX * jX$$

$$je = jX * E$$

$$dX = -(jj+I*mu) \backslash je \quad (1)$$

where *E* is all errors and *I* is the identity matrix. The value *mu* is improved until the change results in a reduced performance value [2].

5. SIMULATION

The simulation is carried out in two stages. In the first stage the input data is generated for the neural network. A java based simulation tool is developed for generating the input data. We consider an environment with 5 homogeneous servers S1, S2 … S5. The service requests arrive according to Poisson process with an inter arrival time $t_0 - t_n$. The average service time is considered as 10milliseconds. Each server is associated with a queue of infinite size where the users' request will be placed. When a new request arrives from a user, the dispatcher forwards the request to the queue of the server having less number of service requests and hence the service requests are distributed among five homogeneous servers. The input data used for the simulation is compared and validated and the correctness is proved using the simulation tool SMTQA [3]. A sample output screen is shown in Fig 1. Each server has served around 400 service requests and the overall average response time obtained from our simulation is 33.91milliseconds and that of SMTQA is 32.76milliseconds. Sample graph showing the Arrival and Waiting Time are shown for our simulation tool and SMTQA respectively in Fig 2 and Fig 3.

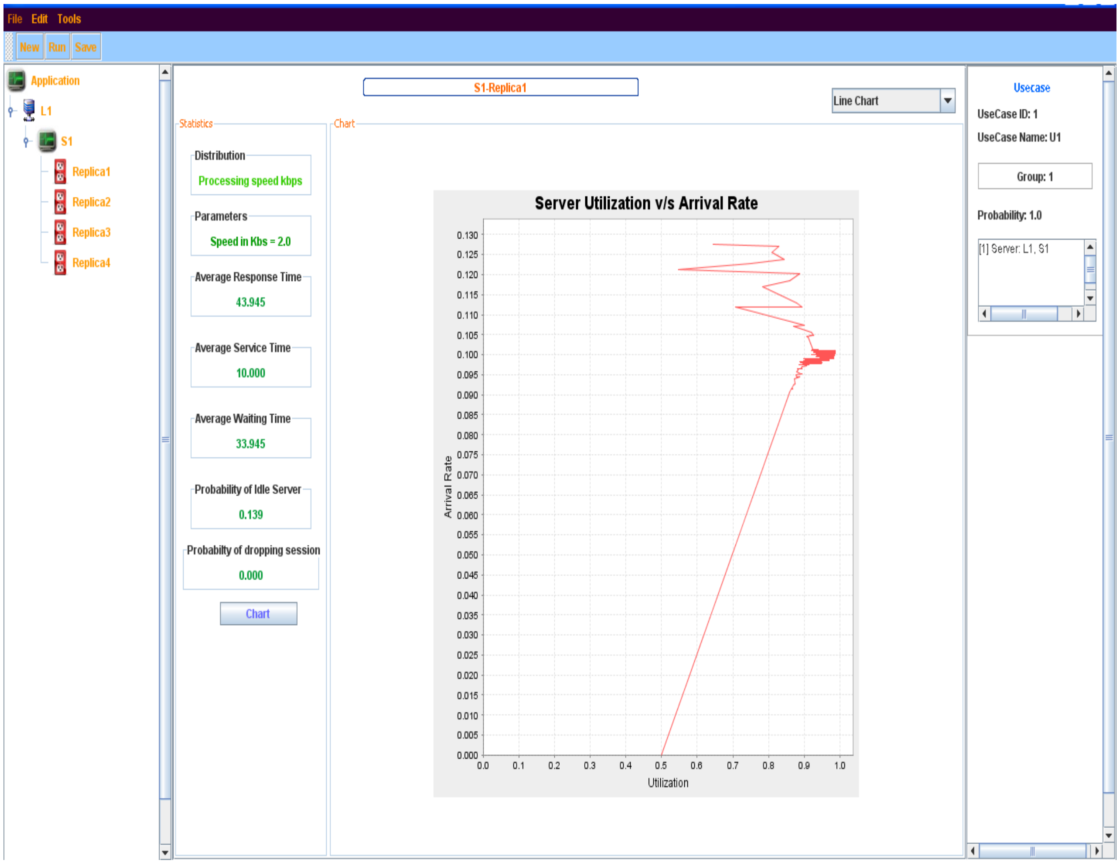

Fig 1: Sample output (SMTQA Tool)

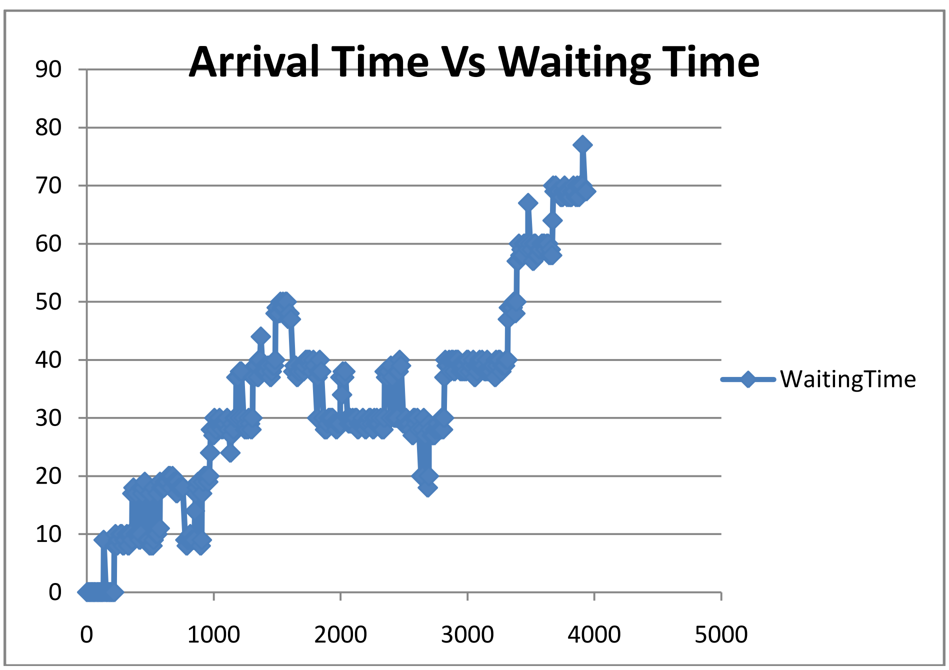

Fig 2: Arrival Time vs Waiting time for Server2

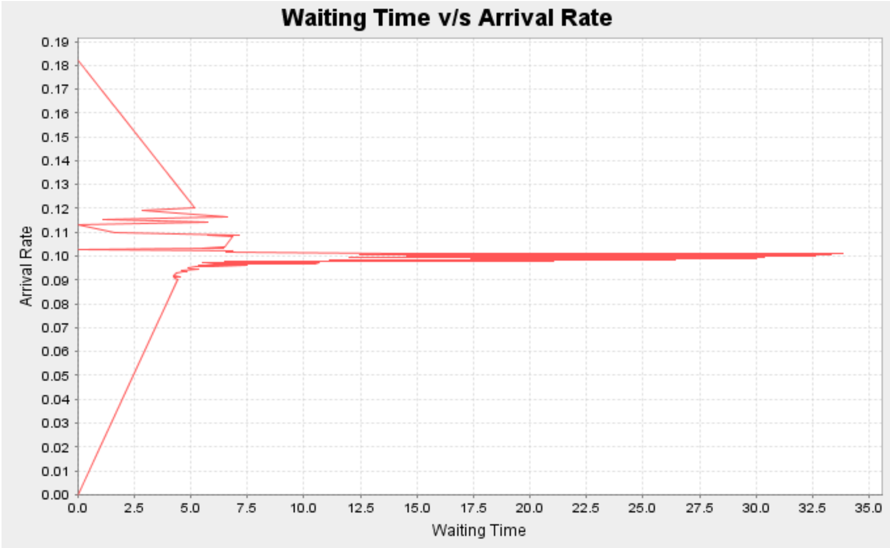

Fig 3: Waiting time vs Arrival Rate for Server2 in SMTQA

In the second stage, MATLAB tool is used for Simulation with NARX feed forward network. Simulation is carried out by considering the combinations of the arrival time of service requests and the corresponding response time of servers as input measures. In the present work, out of 5 servers 3 servers have been trained using neural network for predicting the future response time. The structure of NARX layered feed forward neural networks is shown in Fig 4. It computes the future response time y(t) as a function of previous response times. Arrival time(x[t]) and response time(y[t]) inputs of web service are connected to neurons in the first layer. This layer feeds input patterns into the rest of the network. The Neurons in input layer are connected to all neurons in the next layer which is a hidden layer. In the present design, the hidden layer consist of 10 neurons and has 2 delay lines to keep the previous values of x(t) and y(t). The last layer, which produces the output of the network, is called an output layer. In open loop architecture the network makes use of actual output instead of estimated output. Fig 4 shows the open loop architecture used for training. The true response time output is available during the training of the network and hence we employ the open-loop architecture. The open loop architecture is preferable for training the network because the input is more accurate. When the feedback loop is open on the NARX network, it performs one-step-ahead prediction. It predicts the next value of response time $y(t)$ from previous values of $y(t)$ and $x(t)$.

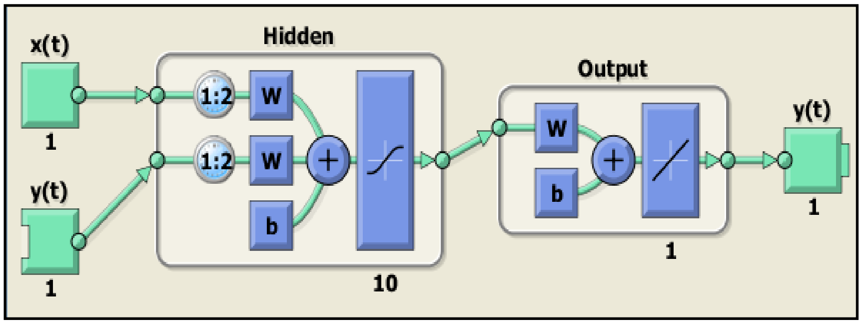

Fig 4: NARX Open loop network

Around 400 service requests are considered for each of the server for the simulation. Three networks are created with past outputs and trained to produce the correct current outputs. 70% of the past values of arrival time and response time are used for training the network. 15% of the data are used for testing purpose and remaining 15% of data are used for validation purpose.

The Matlab function *trainlm* is used for training. This is a network training function that updates weight and bias values according to Levenberg-Marquardt optimization. The networks are trained till an R value closer 1 is reached. After extensive training of network, the simulation showed promising results. During training it is important to assess the performance of the trained network. The correlation coefficient R, which is computed as the difference between the generated output and targets is used for assessing the performance of the networks. Regression line between actual and predicted values of response time for training, validation and testing data sets are plotted. In our work, the R value which is closer to 1 is obtained. Mean Squared Error (MSE) which is the average squared difference between predicted outputs and actual values is also used for validating the predicted values of future response time. MSE which is closer to 0 is obtained for all the three servers.

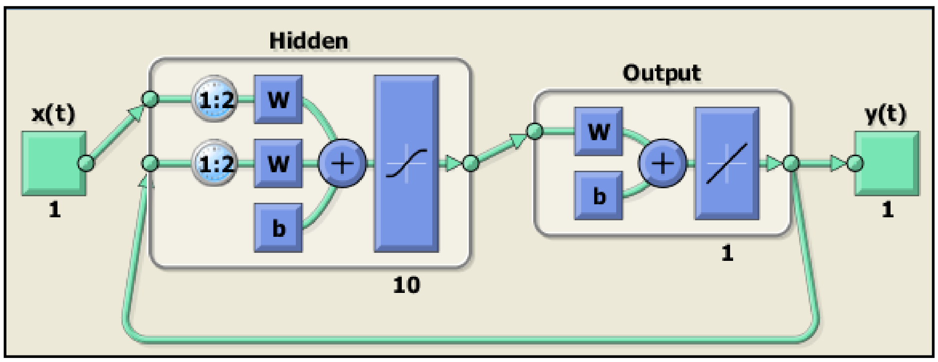

Fig **5**:Closed Loop network

Fig 5 shows the closed loop form of the NARX network. In this form of network, the output y(t) is fed back as one of the input to the network through the delay lines. After training, the network is converted into closed loop form for multistep ahead prediction of response time. The output of the NARX network, y(t), is given back to the input of the network in the form of delays. The prediction of y(t) is a function of y(t – 1), y(t – 2), ..., y(t – d). In closed loop form the estimated output is fed back as one of the input to the network as shown in Fig(5). Closed loop form of network is used for multi step ahead prediction of response time.

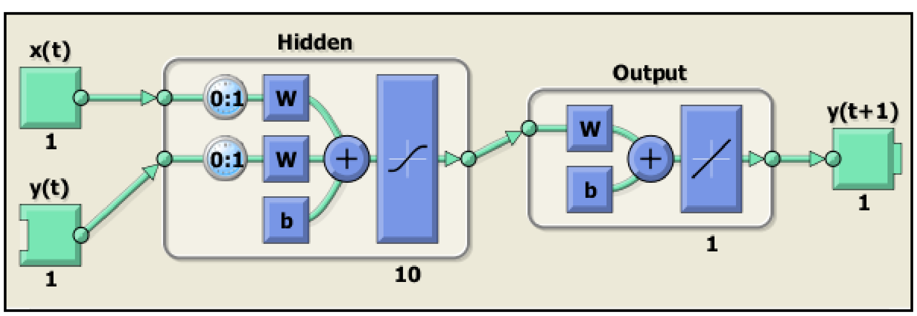

Fig 6: One step ahead prediction network

Fig 6 represents the One step ahead prediction network. In the simulation experiment conducted, one step ahead prediction is done initially and later the output of all three networks has been verified for multistep ahead prediction. In this form of network a delay is removed from the network to obtain the prediction one time step early. The minimal tap delay is made to 0 instead of 1. The calculated output of one step ahead prediction network is y(t + 1) instead of y(t). It would be useful to have predicted y(t+1) once y(t) is available, but before the actual y(t+1) occurs.

5. RESULT AND DISCUSSION

The simulation results obtained are discussed in this section. Fig 7 to Fig 9 shows the targets and outputs of training, testing and validations set data versus time for the response time of web server 1, web server 2 and web server 3 respectively. They also indicate the time points selected for training, testing and validation of response time. For all the three servers error between predicted and actual response time is within the range -10 to 10. As the number of service request increases the response time also increases. We observe from the graph that with the same proportion even the predicted response time value increases for training, validation and testing data set.

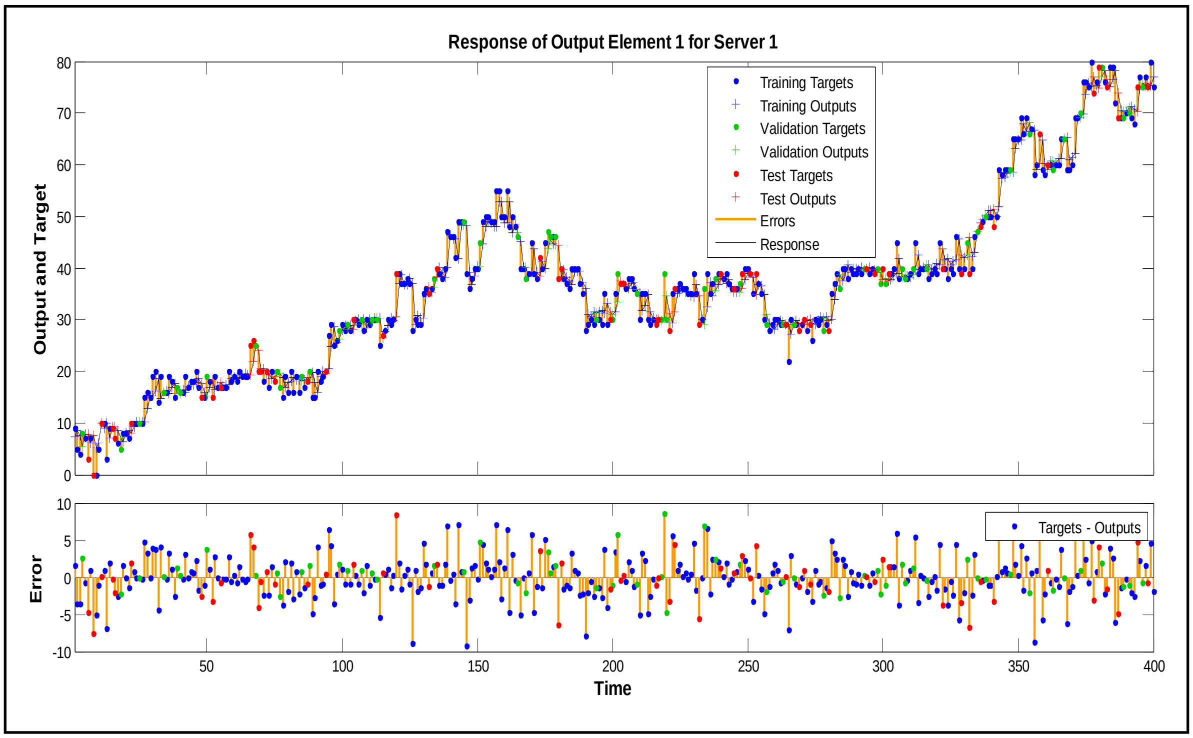

Fig 7: Actual and predicted response time of training, validation and testing data of server1

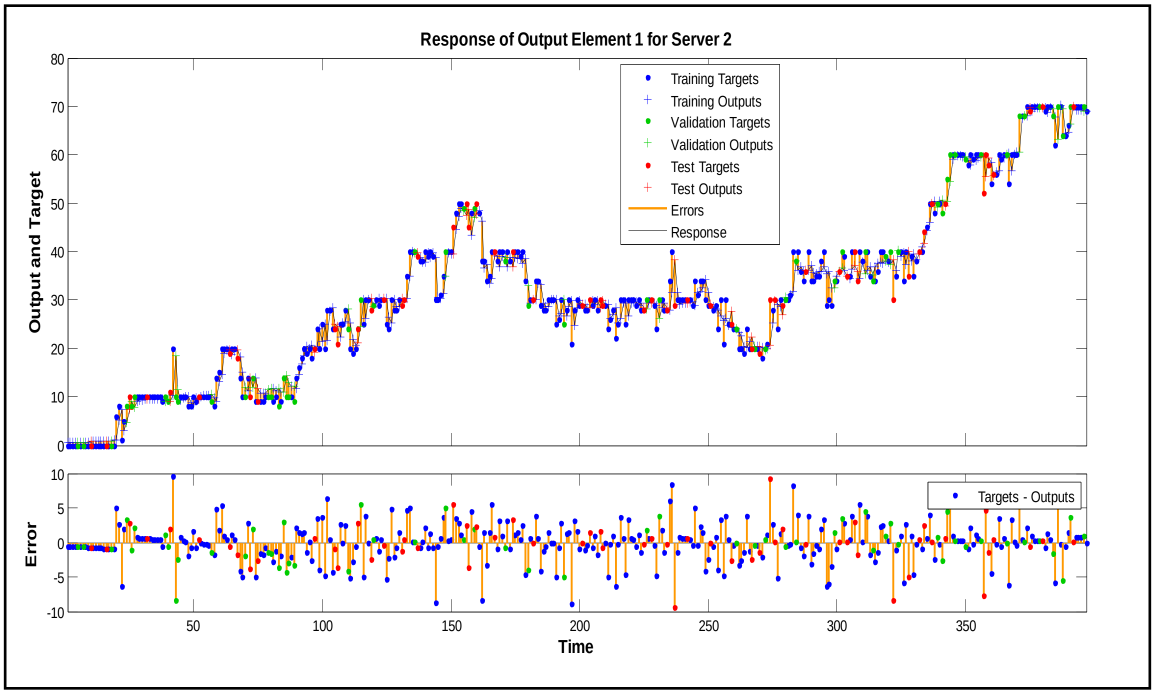

Fig 8: Actual and predicted response time of training, validation and testing data of server2

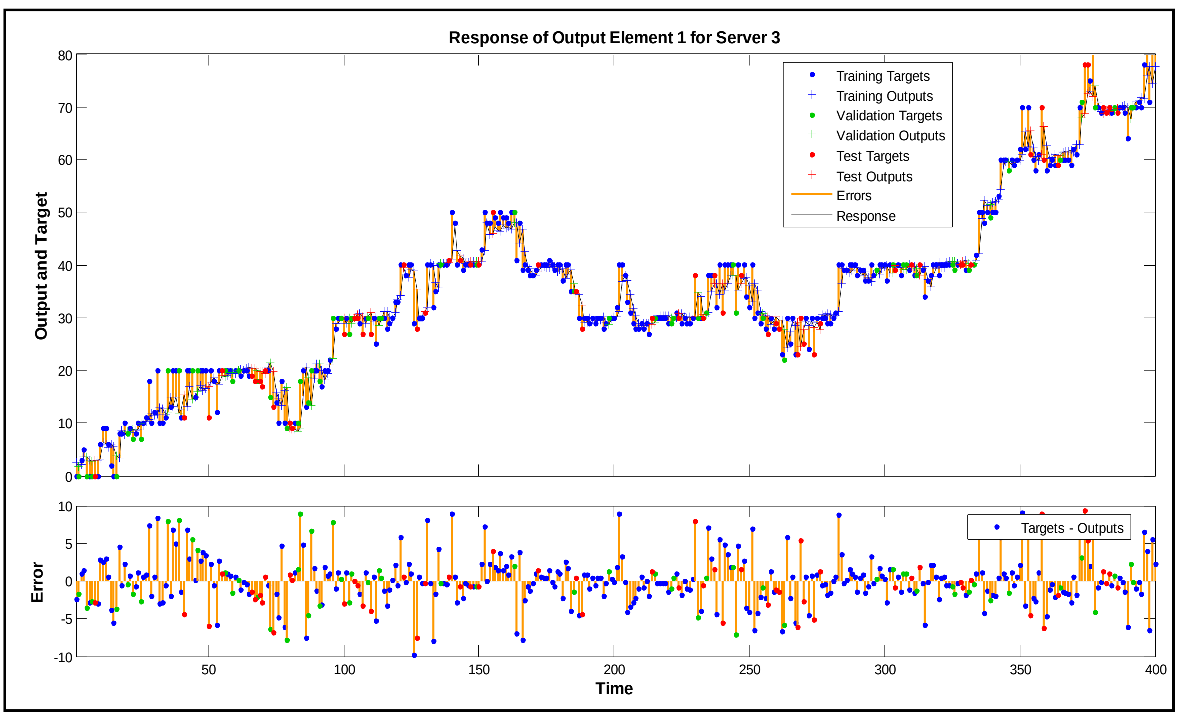

Fig 9: Actual and predicted response time of training, validation and testing data of server3

The regression plots for training, testing and validation of server 1, server 2 and server 3 are shown in Fig 10, Fig 11 and Fig 12 respectively. The R value is an indication of the relationship between the predicted outputs and actual targets of response time. The R value of training data is above 0.98 for all the three servers. Hence the training data indicates a good quality fit as the value of R is very close to 1. The validation and test results also show R values greater than 0.98 for all the three servers and hence the predictor agent present in the corresponding server can use the trained neural network model for predicting the future response time.

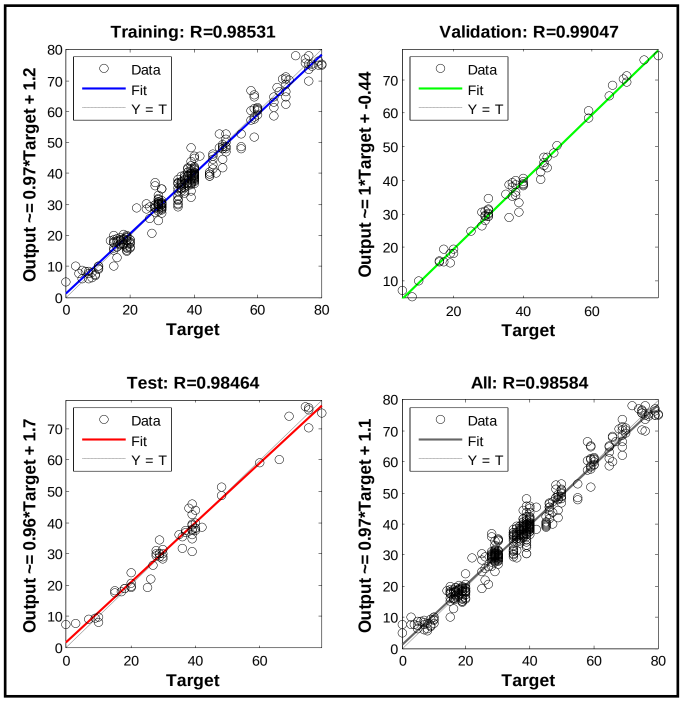

Fig 10: Regression Plot for traning, validation and tesing of server1

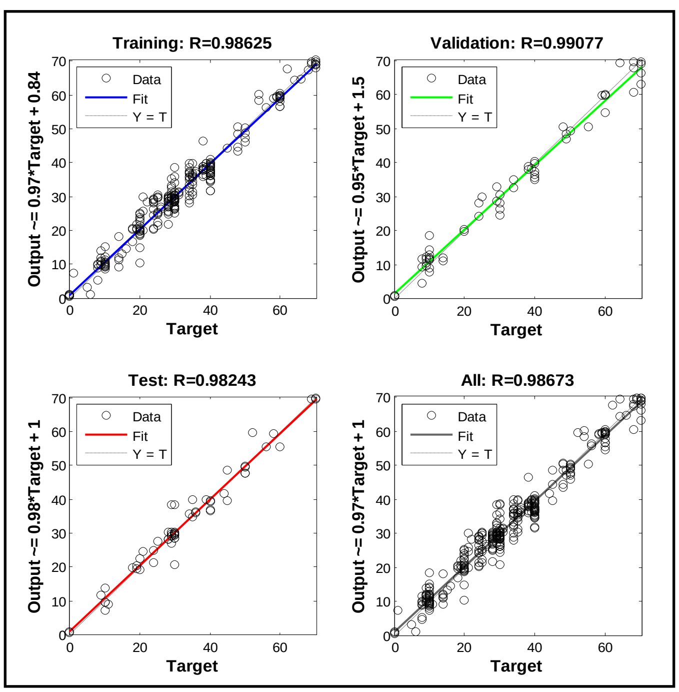

Fig 11: Regression Plot for traning, validation and tesing of server2

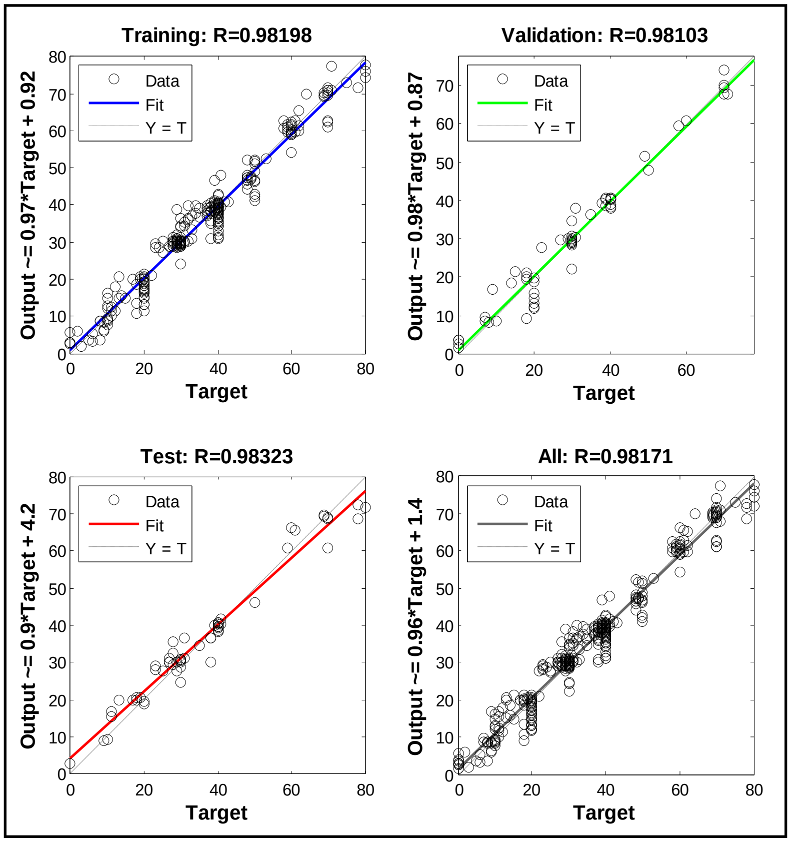

Fig 12: Regression Plot for traning, validation and tesing of server3

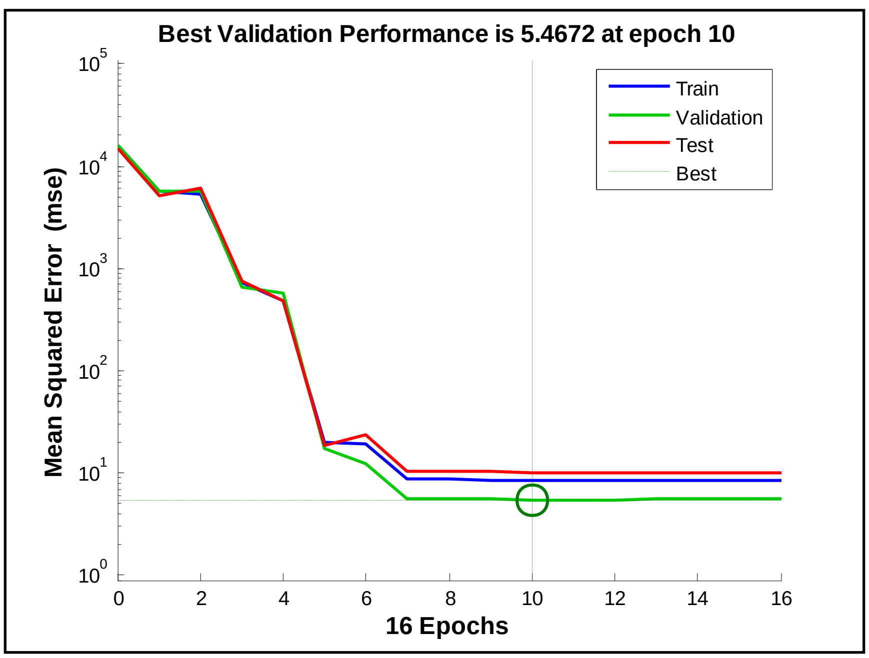

Fig 13: Neural network performance graph for server1

Fig 13, Fig 14 and Fig 15 represents the neural network training performance graph for server 1, server 2 and server 3 respectively. It is observed that for server1 the training, validation and testing errors have decreased until iteration 10. The overfitting has not occurred, since neither testing nor validation error increased before iteration 10. For server 2, the training, validation and testing errors have decreased until iteration 31. The overfitting has not occurred, since neither testing nor validation error increased before iteration 31. Similarly, for server 3, the training, validation and testing errors have decreased until iteration 7. The overfitting has not occurred, since neither testing nor validation error increased before iteration 7. The results imply that the Neural Network model with NARX prediction technique can perform good prediction with least error.

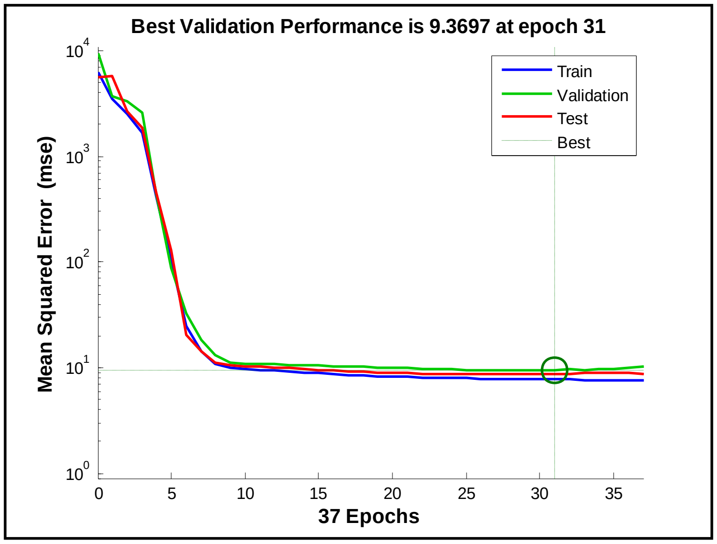

Fig 14: Neural network performance graph for server2

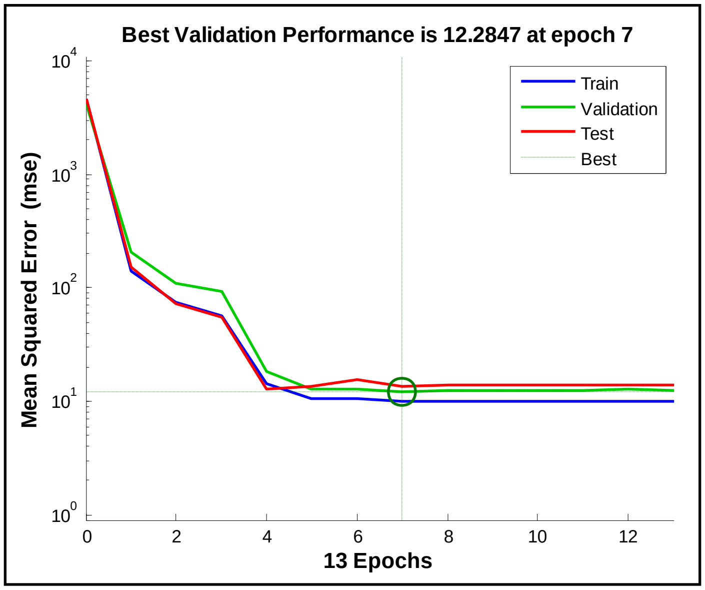

Fig 15: Neural network performance graph for server3

Fig 16, Fig 17 and Fig 18 shows the training state details of server 1, server 2 and server 3. The Mean square error obtained is 0.01 for server1 and server3 and 0.001 for server 2. This is an acceptable amount of error because for all the three servers MSE is closer to 0. It is observed that the best validation performance of 5.0566 at epoch 16 is obtained for server 1; 20.4235 is obtained at epoch 37 for server 2 and 22.8117 is obtained at epoch 13 for server 3. The experimental study indicates that Levernberg-Marquardt algorithm provides the best performance in the prediction of future response time of web service for all the three servers.

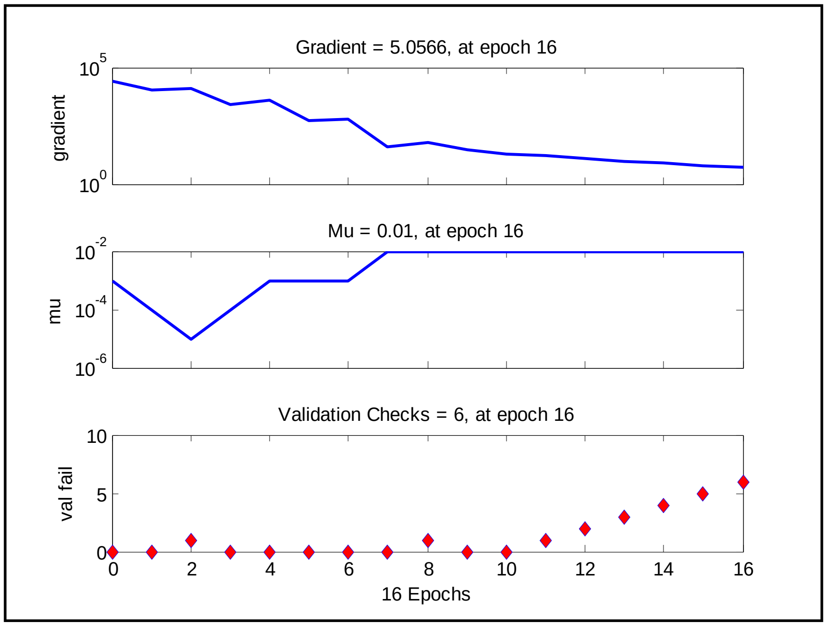

Fig 16: Neural network training state for server1

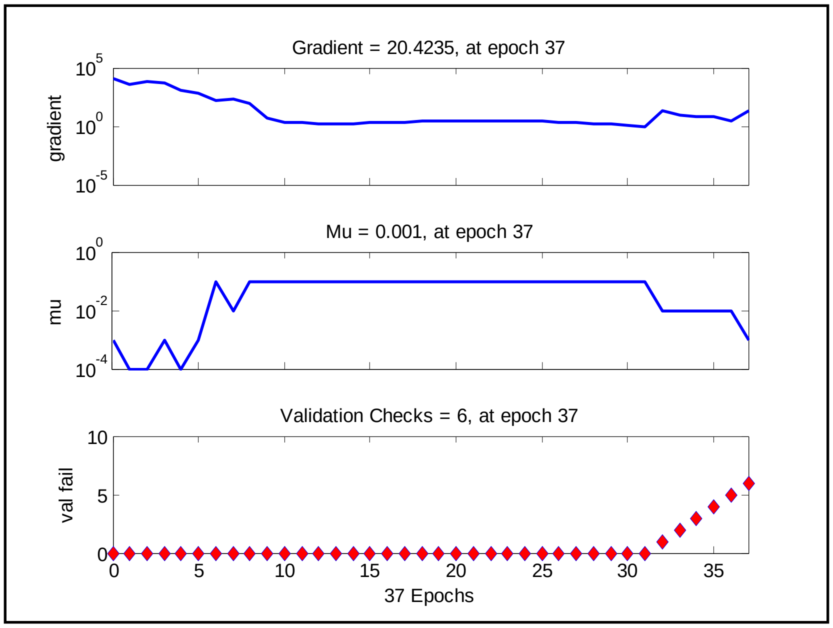

Fig 17: Neural network training state for server2

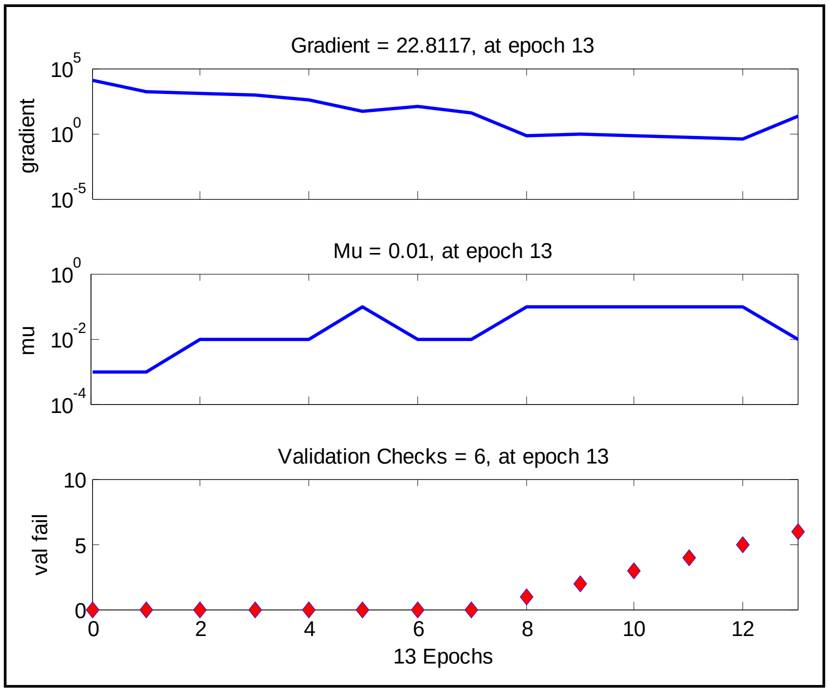

Fig 18: Neural network training state for server3

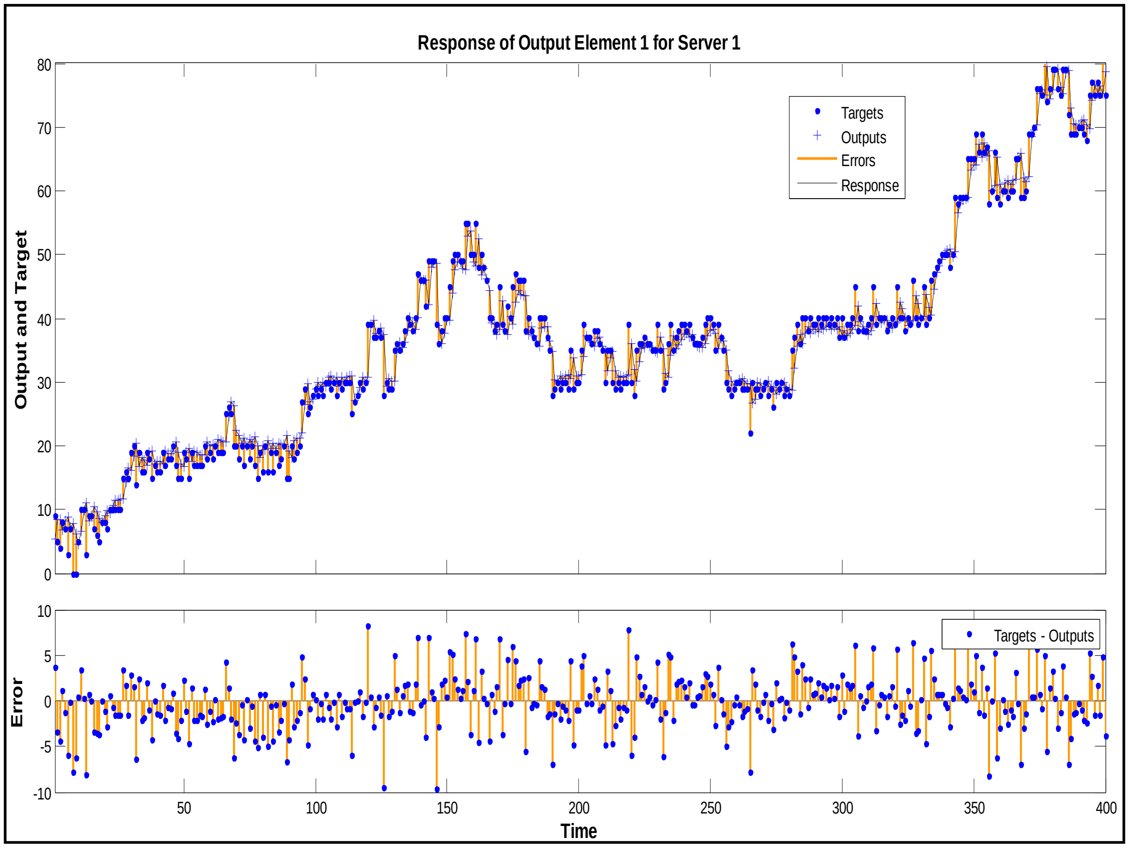

Fig 19: Actual and predicted response time of server1 (Multi step ahead prediction)

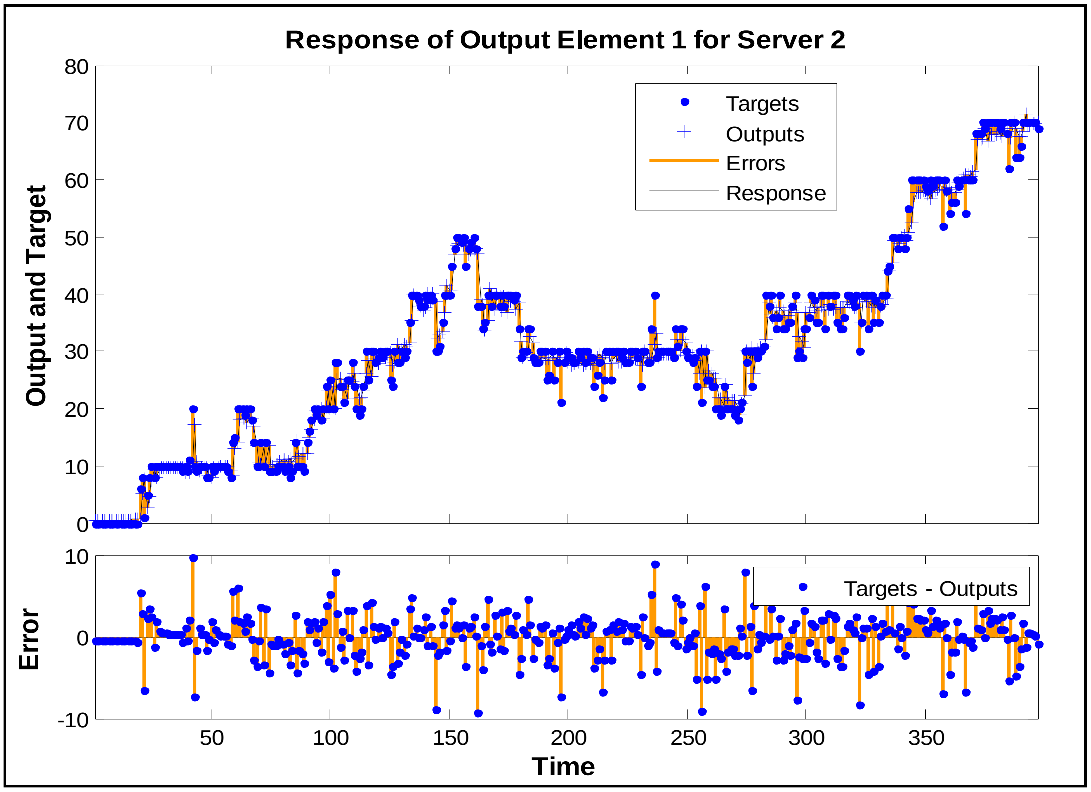

Fig 20: Actual and predicted response time of server2 (Multi step ahead prediction)

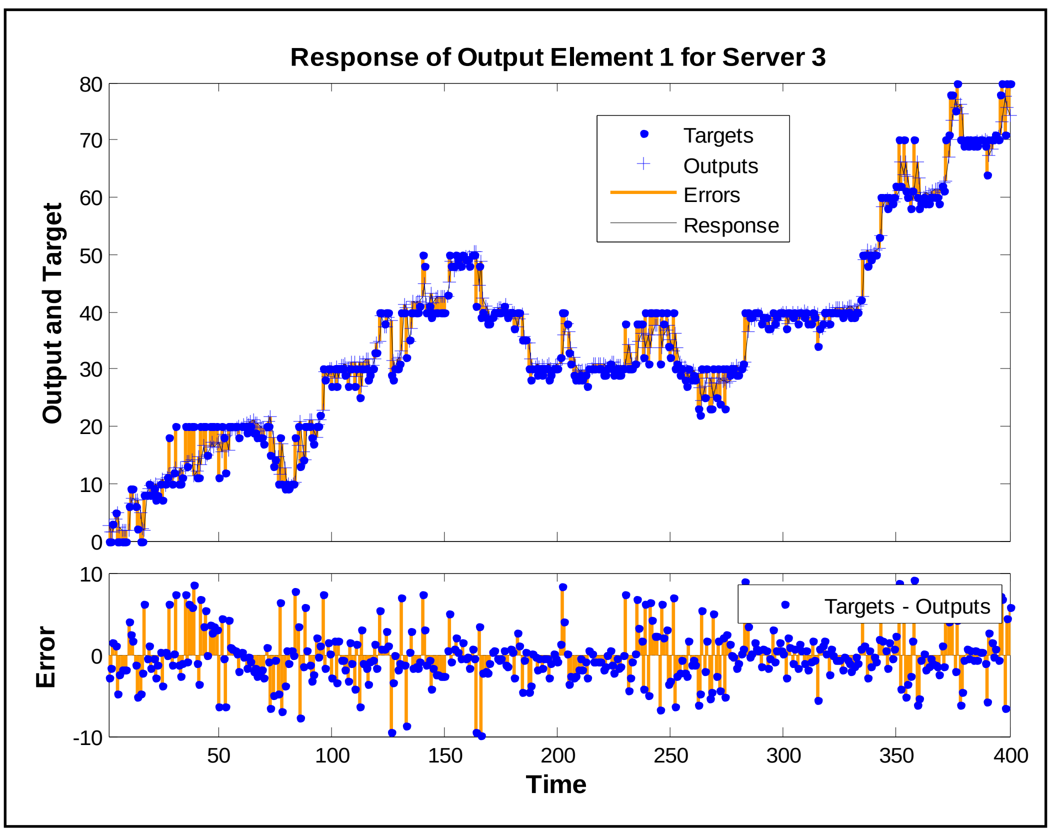

Fig 21: Actual and predicted response time of server3 (Multi step ahead prediction)

Fig 19, Fig 20 and Fig 21 illustrates the iterated response time prediction of three servers. The actual response time and the response time predicted by the NARX neural network are shown for server1, server2 and server3. The network is predicting 400 time steps ahead and the prediction is very close to actual response times and the error is very small. For the web server1, the average response time obtained by prediction using NARX technique is 36.50366 for 400 service requests and the actual average response time of the server 1 is 36.3425. For the web server2, the average response time obtained by prediction using NARX technique is 31.25108 for 400 service requests and the actual average response time of the server 2 is 31.27525. For the web server3, the average response time obtained by prediction using NARX technique is 35.15027 for 400 service requests and the actual average response time of the server 3 is 34.96491. The deviation between the actual and predicted average response time for all the three servers are very minimal.

Table1, Table 2 and Table 3 depict the actual response time and the simulation results of predicted response time for server 1, server 2 and server 3 respectively. A sample of 20 observations along with their time units is shown in the tables for all three servers.

For server 1, the expected average response time for the time period 98 to 107 is 28.7 and the predicted average response time obtained for the same time period is 29.30723. For the time period 270 to 279, the expected average response time is 28.8 and the predicted average response time obtained is 29.25629. From Table1, we observe that the forecasted response time closely resembles with the original response time using NARX time series forecasting method.

Table 1: Actual and predicted response time of server1 (Multi step ahead prediction)

Time	**Expected Response Time**	**Predicted Response Time**	**Time**	**Expected Response Time**	**Predicted Response Time**
98	26	26.90302	270	29	29.131
99	28	27.22603	271	30	29.25496
100	29	28.84344	272	28	30.06961
101	28	29.9198	273	29	29.3384
102	29	29.23922	274	26	29.15803
103	28	29.919	275	30	28.02752
104	30	29.24593	276	29	28.80519
105	30	30.70877	277	30	29.72594
106	29	30.94347	278	28	29.89638
107	30	30.12366	279	29	29.1559

For server 2, the expected average response time for the time period 29 to 38 is 9.9 and the predicted average response time obtained for the same time period is 9.62616. For the time period 220 to 229, the expected average response time is 29.4 and the predicted average response time obtained is 28.60778. From Table 2, we infer that there is no significant deviation between the actual and predicted values of response time.

Table 2: Actual and predicted response time of server2 (Multi step ahead prediction)

Time	Expected Response Time	Predicted Response Time	Time	Expected Response Time	Predicted Response Time
29	10	9.585377	220	30	29.29362
30	10	9.594727	221	30	28.09538
31	10	9.603949	222	29	28.17285
32	10	9.613053	223	30	28.3241
33	10	9.62205	224	28	28.55515
34	10	9.630953	225	28	28.51208
35	10	9.639776	226	30	28.748
36	10	9.648534	227	30	28.99213
37	10	9.657247	228	30	28.65315
38	9	9.665934	229	29	28.73135

For server 3, the expected average response time for the time period 55 to 64 is 19.7 and the predicted average response time obtained for the same time period is 19.95019. For the time period 285 to 294, the expected average response time is 38.8 and the predicted average response time is 38.82019. It can be seen from Table 3 that the best performance is obtained in predicting the future response time for the server by applying NARX time series forecasting method.

Table 3: Actual and predicted response time of server3 (Multi step ahead prediction)

Time	Expected Response Time	Predicted Response Time	Time	Expected Response Time	Predicted Response Time
55	20	19.02896	285	39	39.35634
56	20	19.31003	286	39	38.81662
57	20	19.56757	287	40	38.60129
58	20	19.80613	288	40	39.22441
59	18	20.02633	289	40	39.50277
60	20	19.63217	290	38	39.53792
61	20	20.36461	291	39	38.37745
62	19	20.58317	292	39	38.45428
63	20	20.37146	293	37	38.76289
64	20	20.81145	294	37	37.5679

5. CONCLUSION AND FUTURE WORK

In this work, an agent based performance prediction technique using time series prediction approach is proposed. The prediction of response time is very useful to maintain the service quality level of the web service. The prediction also helps in identifying any QoS Agreement violation in terms of response time. Back propagation learning algorithm using Levenberg-

Marquardt technique is applied by the Predictor agent to train the feed forward neural network in order to predict the response time of a web service. Three different servers have been used for simulation of the proposed technique. For each of the server, the networks were created with past arrival time and response time and trained to produce the correct future response time. The proposed algorithm based on neural network showed significant results in predicting the response time of web service. The average response time obtained for Server 1 by the proposed technique is 36.50366 for 400 service requests and the actual average response time of the server 1 is 36.3425. For the web server 2, the average response time predicted is 31.2510 for 400 service requests and the actual average response time of the server 2 is 31.2752. For the web server 3, the average response time predicted by our proposed techniques is 35.1502 for 400 service requests and the actual average response time of the server 3 is 34.9649. The results confirm that there is no considerable deviation between the predicted and the actual response time. It is evident from the results that the proposed prediction technique is perfectly capable of multi-step ahead response time prediction under dynamic situations. The response time values predicted by the Predictor agent are very promising. These predicted values can further be used for balancing the load of the web services. The prediction will also ease the maintenance of QoS for the web services.

5. REFERENCES

[1] Adhikari R and Agrawal R K, An introductory study on time series modelling and forecasting, Germany: LAP LAMBERT Academic Publishing, 2013.
[2] Charrier, C., Lebrun, G., & Lezoray, O. (2007), Selection of features by a machine learning expert to design a color image quality metrics. 3rd Int. Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM), pp.113-119, Scottsdale, Arizona.
[3] D.E. Geetha, T.V. Suresh Kumar, P. Mayank, K.Rajanikanth, (2010) "A Tool for Simulating Multi-tier Queuing Applications", Technical Report, Department of MCA, MSRIT, TRMCA 04.
[4] Dave Touretzky and Kornel Laskowsk (2006) Lecture Notes Neural Networks for Time Series Prediction 15-486/782: Artificial Neural Networks Fall 2006.
[5] Demian Antony D'Mello,V.S. Ananthanarayana and Santhi T., A QoS Broker Based Architecture for Dynamic Web Service Selection, IEEE Second Asia Internatonal Conference on Modeling & Simulation,2008.
[6] Francisco Garcı´a-Sa´nchez,Rafael Valencia-Garcı´a, Rodrigo Martı´nez-Be´jar, Jesualdo T.Ferna´ndez-Breis,"An ontology, intelligent agent based framework for the provision of semantic web services", Expert Systems with Applications ,Elsevier 2008.
[7] Hagan, M.T., & Menhaj, M. (1994). Training feed-forward networks with the Marquardt algorithm. IEEE Transactions on Neural Networks, Vol. 5, No. 6, 1999, pp. 989-99.
[8] Jaleh Shoshtarian Malak, Mehran Mohsenzadeh, Mir Ali Seyyedi, Multi Agent Based Service Evaluation and Selection Architecture ,2009 International Conference on Software Engineering Theory and Practice (SETP-09), Orlando, Florida, USA 2009.
[9] Kermani, B.G., Schiffman, S.S., & Nagle, H.G. (2005). Performance of the Levenberg–Marquardt neural network training method in electronic nose applications. Science Direct, Sensors and Actuators B: Chemical, Volume 110, Issue 1, pp. 13-22.
[10] Liang Chen, Yipeng Feng, Jian Wu, Zibin Zheng, An Enhanced QoS Prediction Approach for Service Selection, IEEE International Conference on Services Computing, 2011.

[11] Lingshuang Shao, Jing Zhang, Yong Wei, Junfeng Zhao, BingXie, and Hong Mei, Personalized QoS Prediction for Web Services via Collaborative Filtering, International Conference on Web Services, IEEE 2007.
[12] M.Adel Serhani, Rachida Dssouli, Abdelhakim Hafid, Houari Sahraoui," A QoS broker based architecture for efficient web services selection",Proceedings of the IEEE International Conference on Web Services , IEEE 2005.
[13] Rumelhart D.E., Hinton G.E., Williams R.J., Learning Representations by Back–Propagating Errors Nature, 323, 533-536. 1986.
[14] Simon Haykin, Neural Networks, Second Edition, Pearson Education, 1999.
[15] Tsungnan Lin, Bill G. Horne, Peter Tino, C. Lee Giles, Learning long-term dependencies in NARX recurrent neural networks, *IEEE Transactions on Neural Networks*, Vol. 7, No. 6, 1996, pp. 1329-1351
[16] Wan Nurhayati, Wan AB. Rahman and Dr Farid Meziane, Challenges to Describe QoS Requirements for Web Services Quality Prediction to Support Web Services Interoperability in Electronic Commerce, Communications of the IBIMA, Volume 4, 2008.
[17] Yang Gao, Meng Joo Er, NARMAX time series model prediction: feedforward and recurrent fuzzy neural network approaches, *Fuzzy Sets and Systems*, Vol. 150, No. 2, 2005, pp.331-350.