# DYNAMIC RESOURCE ALLOCATION THROUGH WORKLOAD PREDICTION FOR ENERGY EFFICIENT COMPUTING

**Adeel Ahmed[1], David J Brown[1] and Alexander Gegov[1]**

**Abstract** Rapid and continuous increase in online information exchange and data based services has led to an increase in enterprise data centres. Energy efficient computing is key to a cost effective operation for all such enterprise IT systems. In this paper we propose dynamic resource allocation in server based IT systems through workload prediction for energy efficient computing. We use CPU core as a dynamic resource that can be allocated and deallocated based on predicted workload. We use online workload prediction as opposed to offline statistical analysis of workload characteristics. We use online learning and workload prediction using neural network for online dynamic resource allocation for energy efficient computing. We also analyse the effect of dynamic resource allocation on clients by measuring the request response time to clients for variable number of cores in operation. We show that dynamic resource allocation through workload prediction in server based IT systems can provide a cost effective, energy efficient and reliable operation without effecting quality of experience for clients.

**Keywords:** Energy Efficient Computing, Dynamic Resource Allocation, Workload Prediction, NARX Neural Network

## 1 Introduction

Deployment of server based infrastructures such as sever farms and datacentre are now increasing rapidly as compared to a decade earlier. One reason for this increase is the explosion in amount of available data and information required by end users. The *Information Technology* (*IT*) systems in large enterprise organisations are increasingly using these server infrastructures. This high utilisation of servers in such *IT* systems is leading to higher energy consumption and [1] estimates that such *IT* system in U.S consume around 1.5 percent of total electricity consumption costing around 4.5 billion dollars annually [1]. Power

[1] Adeel Ahmed

Institute of Industrial Research, University of Portsmouth, 36 – 40 Middle Street, Portsmouth, PO5 4BP, United Kingdom, Email: adeel.ahmed@port.ac.uk

David Brown,

Institute of Industrial Research, University of Portsmouth, 36 – 40 Middle Street, Portsmouth, PO5 4BP, United Kingdom.

Alexander Gegov

School of Computing, University of Portsmouth, Buckingham Building, Lion Terrace, Portsmouth, PO1 3HE, United Kingdom.

management of such *IT* systems is very important to enterprises for cost effective operation of these *IT* systems. In [2] it has been estimated that for a 10MW data centre the exclusive cost of running computing devices is around eight million dollars. The incurring additional operational cost for such data centre is also estimated to be between four to eight million dollars for additional need of 0.5 to 1W of cooling system operation for each watt of power consumption in computing devices [2]. Hence, a reduction of 1W of computing power will also reduce 0.5 to 1W needed for operating cooling system. Although there is a massive cost associated to the operation of such enterprise *IT* systems but the computing resources of such *IT* systems are mostly not utilised to their full capacity. This underutilisation of computing resources is also one of the primary reasons for higher consumption of electricity [1, 3]. To use the computing resources efficiently and to optimise the resource utilisation in enterprise *IT* systems, an efficient workload prediction mechanism is required. In this paper we use neural networks for online workload prediction in server based *IT* environments. We also analyse the energy savings that could be achieved using dynamic resource allocation through this online workload prediction.

There are several prediction approaches that can be applied for workload prediction in IT systems. *Hidden Markov Models* (*HMM*) can be used for workload time series modelling but this approach suffers from complexity in convergence [4]. *Support Vector Machines* (*SVM*) have good prediction for small samples but the prediction performance of *SVM* depends on the specific kernel function used for *SVM* based workload prediction [5,6]. Auto regressive modelling approaches such as *Auto Regressive Integrated Moving Averages* (*ARIMA*), *Wavelet ARIMA* (*WARIMA*) [7] and *Seasonal ARIMA* (S*ARIMA*) [8-10] provide good prediction but these techniques take longer to train and converge [6]. *Artificial Neural Network* (ANN) are very efficient at learning from the historical data points and to predict the possible future behaviour of a variable. We use *Non-Linear Auto Regressive model with eXogenous input* (*NARX*) neural network to predict the server workload. *NARX* neural network have the capability to model the nonlinear dynamic systems with faster convergence to global minimum [11].

In this paper we have considered individual *CPU* core as a resource that can be allocated and deallocated dynamically. We accurately predict the server load one hundred and fifty time steps ahead and *CPU* cores on the server can be allocated and deallocated based on this load prediction. A brief overview of *NARX* neural network model is presented in next section followed by explanation of experimental system setup. Server load prediction performance is presented and discussed in results section followed by conclusion.

## 2 NARX Model Theory

NARX neural network based models are suitable for non-linear systems modelling and time series forecasting [12]. *NARX* neural network have evolved from various neural networks such as *Multilayer Feed-forward Neural Networks* (*MFNN*), *Recurrent Neural Network* (*RNN*) and *Time Delay Neural Network* (*TDNN*). NARX neural network are multilayer dynamic recurrent neural network with feedback connections to the input layer [13, 14] as shown in Fig. ***1***. In *NARX* neural network the prediction is performed through regression on the past values of the output signal along with past values of an independent signal. *NARX* neural network with one output and two inputs can be mathematically expressed as follows [15].

$$Y(t+1) = f\left\{b_0 + \sum_{h=1}^{N} w_{h0} f_h\left(b_h + \sum_{i_1=0}^{d_{u1}} w_{i_1 h} u_1(n-i_1) + \sum_{i_2=0}^{d_{u2}} w_{i_2 h} u_2(n-i_2) + \sum_{j=0}^{d_y} w_{jh} y(n-j)\right)\right\}$$

Where $i_x$= 1,2, 3, $\ldots\ldots, d_{u_x}$ and $w_{i_x h}$ are the weights of input units to the hidden layer.

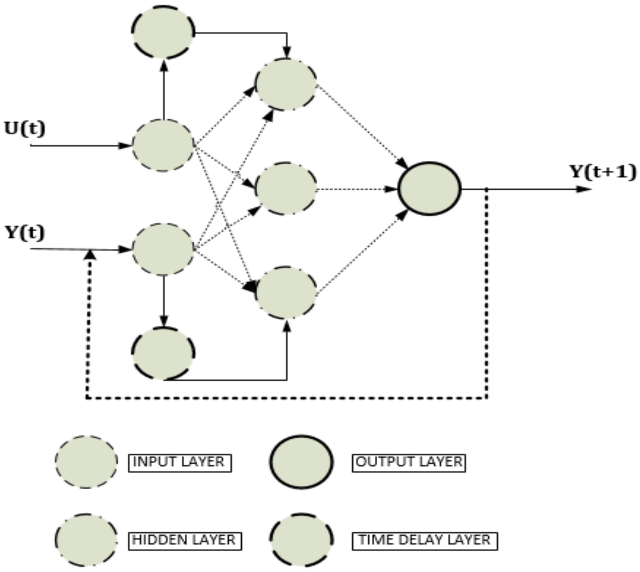


**Fig. 1. NARX Neural Network**

## 3 EXPERIMENTAL SYSTEM SETUP

For dynamic resource allocation based on server *CPU* load, we have used a *64-bit ARM* architecture based server with an *AMD A1100* series processor. The processor has eight *64-bit ARM* cortex *A57* cores. The server is a Linux based web server with support for standard GNU tool chain. We have used *Nginx* for web server setup and configuration. Following are the web server system specifications.

**Table 1** Server Specifications

| | |
|---|---|
| Server Architecture | 64-bit ARM |
| Processor | 1 AMD Opteron |
| Processor Cores | 8 |
| Memory | 32 GB |
| Storage | 1 TB |

Server is also configured to continuously measure the system resources i.e., *CPU* load at a rate of 0.1 KHz. The measured samples are averaged over one second time period. The *CPU* load is used for prediction and dynamic core allocation. The clients are configured to send requests to server using different configurations as shown in Table 2. Each configuration is classified as a separate category of workload and all the categories of workload generation from Table 2 are applied to the server. For each and every applied workload category, underlying processes on the server do not change for this experimental setup. In this experimental setup we have configured and used a web based server setup but there exist other type of servers such as data warehouse database servers, Online Transaction Processing (*OLTP*) database servers and mail servers. Although each type of server has its own characteristics and associated Quality of Service (*QoS*) requirements [16] but dynamic core allocation using predictive algorithms can be applied to all these server types for energy efficient computing, however this may be more feasible for one type of server than the other. Clients are also configured to measure the response time for every request sent by each process. Each workload is applied to the server for over twelve hours and measured *CPU* load is used for training and prediction using *NARX* neural network.

**Table 2** Client Configurations

| Workload Category | File Size Request Distribution | Number of Processes | Number of Requests Per Process | Inter-Request Delay (seconds) |
|---|---|---|---|---|
| A | 50%1K, 29%10K, 15%100K, 5%1000K, 1%10000K | 25 | 1000 | 0.1 |
| B | 50%1K, 29%10K, 15%100K, 5%1000K, 1%10000K | 25 | 1000 | 1 |
| C | 50%1K, 29%10K, 15%100K, 5%1000K, 1%10000K | 25 | 1000 | 10 |
| D | 20%1K, 20%10K, 20%100K, 20%1000K, 20%10000K | 25 | 1000 | 0.1 |
| E | 20%1K, 20%10K, 20%100K, 20%1000K, 20%10000K | 25 | 1000 | 1 |

## 4 RESULTS AND DISCUSSION

We have used server *CPU* load for dynamical core allocation based on *CPU* load prediction. The *CPU* load reflects the *CPU* utilisation and its value ranges from zero to the maximum number of cores. The server *CPU* load measurements for each category of workload are used for training the neural network. The trained neural network is then used to predict the future *CPU* load values. The training and prediction is performed for each category of workload and similar prediction performance is observed. We only present the training and prediction performance results with discussion for category '*A'* of the workloads. We use MATLAB to train and predict the *CPU* load using *NARX* neural network model. *NARX* neural network with two hidden layers with 10 neurons each has been used for training the network. Fig. ***2*** shows the server *CPU* load for category '*A'* of the workloads. Fig. ***2*** also shows that server *CPU* load shows high variation in short period of time samples.

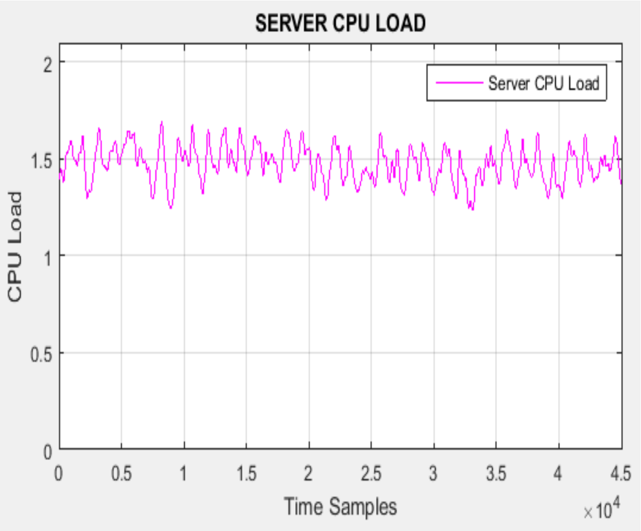


**Fig. 2.** Server CPU Load for Category 'A' Workload

We divide the server *CPU* load prediction into training frames of 1500 time samples and for each training frame the time samples within the training frame are used to train the *NARX* neural network and to predict the server *CPU* load for 150 time samples ahead. Fig. ***3*** shows a training frame consisting of 1500 time samples of *CPU* load. The frame is used for training, test and validation of *NARX* neural network.

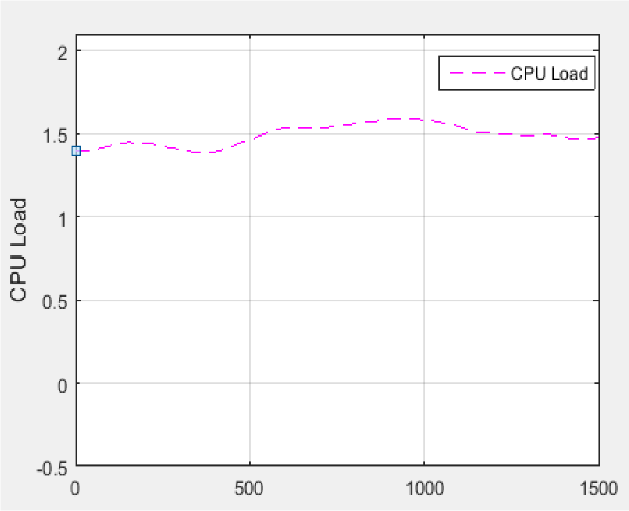


**Fig. 3.** A Training Frame of 1500 CPU Load Samples

1500 samples from the frame are used to train the NARX neural network. Training error, which is the difference between the target value and the predicted value, is nearly close to zero as shown in Figure 4. Error histogram in Figure 4 also shows test and validation errors of *NARX* for the training frame of Fig. ***3***. Fig. ***4*** shows that the difference between the expected output and the predicted output for NARX network is very small and the error is either very close to zero for majority of training, test and validation samples or the deviation from zero error is also very small.

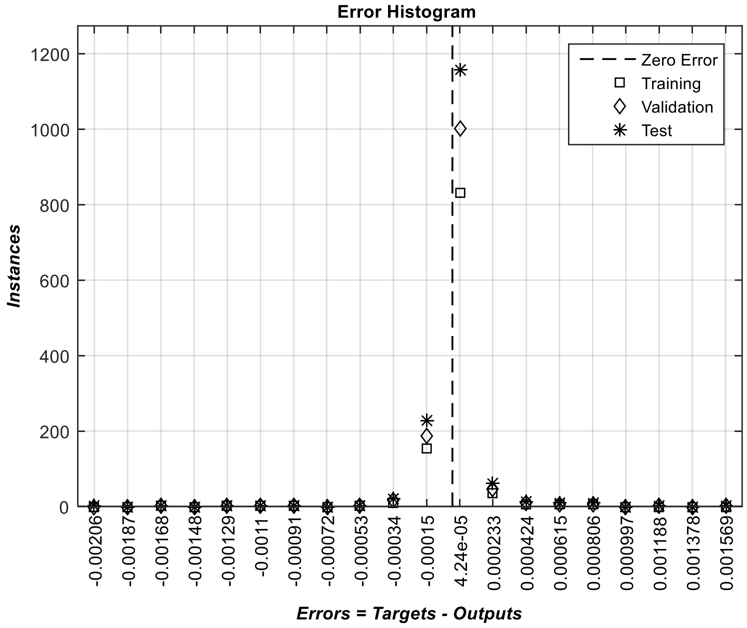


**Fig. 4**. Error Histogram for a Time Frame

The trained *NARX* neural network is used to predict the *CPU* load for 150 time samples and Fig. ***5*** shows the prediction performance of the trained *NARX* neural network. Fig. ***5*** shows that the trained *NARX* neural network has precisely predicted the *CPU* load for 150 time samples ahead with approximately zero prediction error.

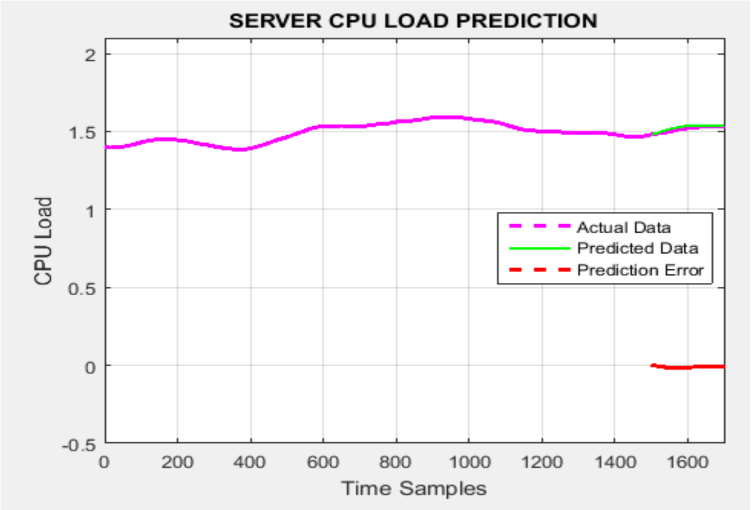


**Fig. 5.** Server CPU Load Prediction

Based on predicted workload as shown in Fig. ***5***, *CPU* cores can be allocated and deallocated dynamically. Fig. ***5*** also shows that for the predicted period the system does not need more than two cores for its operation and rest of the six cores can be turned off at least for the period of prediction. Hence, for a single *CPU* with eight cores, dynamic allocation of cores based on the predicted workload leads to 75% reduction in *CPU* energy consumption. This dynamic resource allocation for multi-server environment with multiple CPU can provide significant savings in terms of energy consumption.

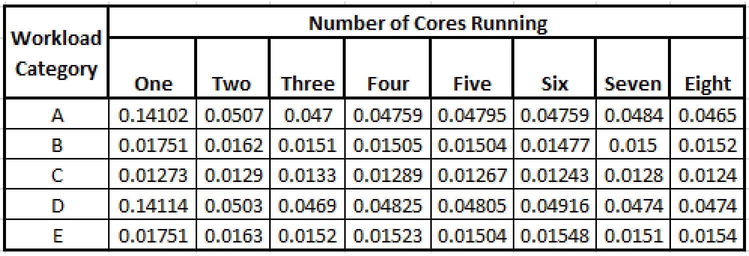
All the workload categories from Table 2 are applied to the server and aforementioned *NARX* neural network scheme is utilised for *CPU* load prediction for each category of workload. Table 3 shows the maximum value for dynamic core allocation found through workload prediction for each category of the workloads.

**Table 3** Optimal Resource Allocation for Each Workload Category

| Workload Category | Optimal Core Value |
|---|---|
| A | 2 |
| B | 1 |
| C | 1 |
| D | 2 |
| E | 1 |

In order to analyse at the effect of dynamic core allocation on the clients, we look at the effect of dynamic core allocation on clients in terms of any additional delay in response time to clients requests. Since the operation of dynamic core allocation is transparent to clients as all the other server processes are also transparent, the client's Quality of Experience (QoE) is measured in terms of response time to the client's request. Table 4 shows effect of dynamic core allocation on response time and it also shows that for each workload category the response time does not vary using dynamic core allocation maximum values from Table 3. Using the dynamic core allocation below these optimal core values will result in lower request handling rate at the server and hence resulting in high response time as shown in Table 4 and Figure 6 for workload categories A and D. Table 4 and Figure 6 also show that dynamic core allocation does not degrade the performance of the server if optimal predicted values for core allocation are used.

**Table 4** Response Time for Each Workload Category for Variable Cores



| Workload Category | Number of Cores Running | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | One | Two | Three | Four | Five | Six | Seven | Eight |
| A | 0.14102 | 0.0507 | 0.047 | 0.04759 | 0.04795 | 0.04759 | 0.0484 | 0.0465 |
| B | 0.01751 | 0.0162 | 0.0151 | 0.01505 | 0.01504 | 0.01477 | 0.015 | 0.0152 |
| C | 0.01273 | 0.0129 | 0.0133 | 0.01289 | 0.01267 | 0.01243 | 0.0128 | 0.0124 |
| D | 0.14114 | 0.0503 | 0.0469 | 0.04825 | 0.04805 | 0.04916 | 0.0474 | 0.0474 |
| E | 0.01751 | 0.0163 | 0.0152 | 0.01523 | 0.01504 | 0.01548 | 0.0151 | 0.0154 |

Fig. ***6*** is a bar graph representation of request response time for variable number of cores in operation and it clearly shows that dynamic core allocation can provide energy savings without affecting the response time to client's requests.

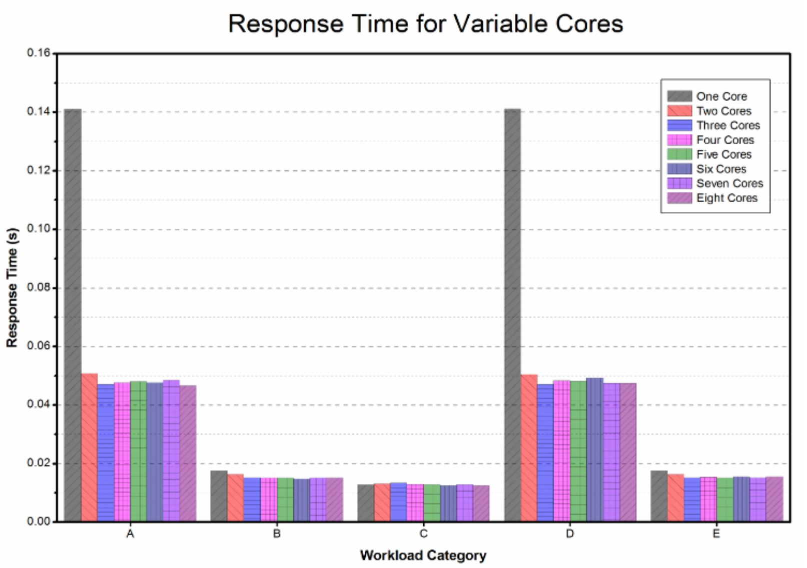


**Fig. 6.** Dynamic Core Allocation and Response Time

## 5 CONCLUSION

In this paper we have presented an energy efficient dynamic resource allocation scheme for server based IT infrastructures. The proposed scheme provides energy and cost savings without degrading the server performance, hence enabling a cost effective, green and reliable server operation in enterprise IT systems.

**References:**

[1] US EPA, “Report to Congress on server and data center energy efficiency,” in Public Law 109-431, U.S. Environmental Protection Agency ENERGY STAR Program, 2007.

[2] C. D. Patel, C. E. Bash, R. Sharma, and M. Beitelmal. Smart cooling of data centers, in IPACK, July 2003.

[3] P. Bohrer, E. N. Elnozahy, T. Keller, M. Kistler, C. Lefurgy,C. McDowell, and R. Rajamony, The case for power management in web servers. Norwell, MA, USA: Kluwer Academic Publishers, 2002, pp. 261–289.

[4] Kuo-Chin Fan; Sung-Jung Hsiao; Wen-Tsai Sung;"Developing a Web-based pattern recognition system for the pattern search of components database by a parallel computing," Parallel, Distributed and Network-Based Processing, 2003. Proceedings. Eleventh Euromicro Conference on , vol., no., pp.456-463, 5-7 Feb. 2003.

[5] L. Ni, X. Chen, Q. Huang, ARIMA Model for Traffic Flow Prediction Based on Wavelet Analysis, The 2nd International Conference on Information Science and Engineering [ICISE2010], Dec 2010, Hangzhou, China

[6] G. Tran, V. Debusschere and S. Bacha, "Neural networks for web server workload forecasting," *Industrial Technology (ICIT), 2013 IEEE International Conference on*, Cape Town, 2013, pp. 1152-1156.

[7] A. R. Syed, S.M. A. Burney and B. Sami, Forecasting Network Traffic Load Using Wavelet Filters and Seasonal Autoregressive Moving Average Model, International Journal of Computer and Electrical Engineering, Vol.2, No.6, December, 2010 pp.1793-8163.

[8] A. Tamimi, A.K. Jain, R.; C. So-In, "SAM: A Simplified Seasonal ARIMA Model for Mobile Video over Wireless Broadband Networks," Multimedia, 2008. ISM 2008. Tenth IEEE International Symposium on , vol., no., pp.178-183, 15-17 Dec. 2008.

[9] C. Chen; Q. Pei and N. Li, "Forecasting 802.11 Traffic Using Seasonal ARIMA Model," Computer Science-Technology and Applications, 2009. IFCSTA '09. International Forum on , vol.2, no., pp.347-350, 25-27 Dec. 2009.

[10] Y. Shu; M. Yu; J. Liu; O. Yang, "Wireless traffic modelling and prediction using seasonal ARIMA models," Communications, 2003. ICC '03. IEEE International Conference on , vol.3, no., pp. 1675- 1679 vol.3, 11-15 May 2003.

[11] Tsungnan Lin; Horne, B.G.; Tino, P.; Giles, C.L.; , "Learning long-term dependencies in NARX recurrent neural networks," Neural Networks, IEEE Transactions on , vol.7, no.6, pp.1329-1338, Nov 1996.

[12] I. J. Leontaritis, S. A. Billings, Input–output parametric models for nonlinear systems,part I : deterministic nonlinear systems, International Journal of Control, (1985) 303–328

[13] M. Norgaard, O. Ravn, N. K. Poulsen, L. K. Hansen, “Neural Networks for Modelling and Control of Dynamic Systems, Springer”, Berlin, 2000.

[14] Aida A. Ferreira, Teresa B. Ludermir, Ronaldo R. B. de Aquino, “Comparing Recurrent Networks for Time-Series Forecasting”, WCCI 2012 IEEE World Congress on Computational Intelligence - June, 10-15, 2012 - Brisbane, Australia.

[15] V. G. Tran, V. Debusschere and S. Bacha, "Neural networks for web server workload forecasting," *Industrial Technology (ICIT), 2013 IEEE International Conference on*, Cape Town,2013,pp.1152-1156.

[16] A.Oodan, K.Ward, C.Savolaine, M.Daneshmand, P.Hoath, “*Telecommunications Quality of Service Management from Legacy to Emerging Services*”, IET Telecommunication Series 48, 2002.