INT J COMPUT COMMUN, ISSN 1841-9836
9(2):131-138, April, 2014.

Input Projection Algorithms Influence in Prediction and Optimization of QoS Accuracy

R.D. Albu, I. Dzitac, F. Popentiu-Vladicescu, I.M. Naghiu

Răzvan-Daniel Albu*,
Florin Popentiu-Vlădicescu,**
Iuliana Maria Naghiu
1. University of Oradea,
Universitatii St., 1, 410610 Oradea, Romania
*Corresponding author: ralbu@uoradea.ro
**2. Academy of Scientist in Romania,
54, Splaiul Independentei, 0590094, Bucharest
E-mail: popentiu@imm.dtu.dk.

Ioan Dzitac
1. Aurel Vlaicu University of Arad
Elena Dragoi St., 2, 310330 Arad, Romania
ioan.dzitac@uav.ro
2. Agora University of Oradea
Piata Tineretului 8, 410526 Oradea, Romania
E-mail: idzitac@univagora.ro

Abstract: Regardless of new achievements in the research of prediction models, QoS is still a great issue for high quality web services and remains one of the key subjects that need to be studied. We believe that QoS should not only be measured, but have to be predicted in development and implementation phases. In this paper we assess how different input projection algorithms influence the prediction accuracy of a Multi-Layer Perceptron (MLP) trained with large datasets of web services QoS values.

Keywords: Quality of Service (QoS), adaptive models, web services, large/big data.

1 Introduction

The major web services QoS requirements are: availability, accessibility, integrity, performance, regulatory reliability and security. Models of prediction and/or optimization of QoS in web services are presented in many actual works: [19], [14], [16], [15], [20], [21], etc.

This research work continues the investigation on web services QoS criteria prediction and completes the results obtained in [1]- [5], works (co)authored by first author of this paper. We build in this work an adaptive model that offers good prediction results of QoS in web services. Since more data may lead to more accurate analyses and more precise analyses may lead to more confident decision making, the development of accurate and adaptive prediction models is both challenging and essential. Enhanced conclusions can cause superior operational efficiencies, cost reductions and lower risks.

Scott Zucker, Vice President of Business Services at Family Dollar, said: *"Small data is gone. Data is just going to get bigger and bigger and bigger, and people just have to think differently about how they manage it"* [8]. *Large data* is an expression used to define the exponential growth of data availability and size. Several recent technology improvements, that allow organizations to make the most of big data analytics, are: affordable bigger storage, faster and parallel processors, open source platforms, clustering, virtualization, grid computing, increased throughput and last, but not least, Cloud computing.

Another new term for the quantity of information generated by business, government, and science is: data deluge [10]. For instance, in 2010, the Large Hadron Collider (LHC) facility at CERN delivered 13 petabytes of data. Concurrently to this important growth, data are also becoming strongly interconnected. For example, Facebook is approximately fully connected. Presently, social networks interconnect people or groups who share similar interests, but soon, we expect they will also link software modules such as: Web-based services, or workflows. In recent happenings interrelated with the 2013 Boston Marathon violence, social networks of marathon competitors and high-performance computational systems were combined to group and analyze huge collections of photos and videos, finally leading to the identification of the terrorists. *Big data* is typically characterized by the "three V's": volume, velocity, and variety. In terms of volume, at the end of 2011, Facebook had 721 million individuals and 68.7 billion friendship edges [7]. In terms of velocity, Twitter generates 7 Tbytes of data daily, while Facebook produces 10 Tbytes. On 11 November 2012, a sales event at TaoBao, the largest online shopping marketplace in China, generated 100 million transactions and reached a peak transaction rate of 205,000 per minute [9]. In terms of variety, data today come from various sources, ranging from surveillance videos, to satellite images, to mobile tweets, to sensors and meters in the power grid [18]. A key difference between big data and large data is the rate at which the data can be collected and made accessible for analysis. Large data can also be handled by the traditional reporting and analysis tools. Big data is generally used to describe the massive amount of unstructured data, which costs a lot of time and money for analysis. However, large data may not have such special meaning, they just refer to the volume. We don't think there is any value in defining a threshold for what constitutes "big data." and what means "large data". Therefore, a flexible definition we use is "Big data is data that's an order of magnitude bigger than you're used to". The big data analysis affects the QoS because, if implemented efficiently, the big data infrastructure will permit carriers to preserve constantly optimized services and to set apart in the marketplace by offering QoS reports, diagnostic tools and other decision-support front-ends.

Cloud computing is a technology that exploded in the recent years and seems to be the ideal way to provide big and large data for mainstream uses, because it offers scale-out and on-demand computing resources in a pay-per-use style. For instance, Netflix stores movies and TV shows, while Dropbox saves clients' documents, both in Amazon's Simple Storage Service. In recent years, scale-out data stores, usually mentioned as NoSQL systems, were quickly gaining admiration as a possible solution for applications scaled at Internet level. These stores consist of technologies like: Amazon's DynamoDB, Google's BigTable and Yahoo's PNUTS. To address the "Big and Large data" challenge, NoSQL supporters limit ACID restraints, deliver completely scalable solutions and then gradually add back the Relational Database Management System (RDBMS) features like index or transaction support. A newly appeared paradigm named stream computing facilitates continuous queries over streaming data like social media feeds. Social networking on the cloud could empower sharing based on the social relationship between users. This would possibly make available technologies like volunteer computing. This is a distributed computing model in which associated users donate computing resources to a project. Two examples of volunteer computing are the following projects: Storage@home14 and Boinc15 [18]. In these scenarios, the resources are owned by individuals and they are shared in return for access to other resources. This could hypothetically transform the cloud's economics and raises doubts about the reliability and QoS warranties [11] [13].

2 Input Projection and Optimization Algorithms

Input Projection is the procedure that further reduces input dimensions by automatically mapping multiple pieces of information to single inputs. The main goal of input optimization is to make the network learn faster in the same time offering the same prediction performances or better. Input Optimization automatically determines the most informative inputs through genetic algorithms, greedy search, back-elimination and other techniques [6]. The four input projection algorithms we have investigated in this study are: PCA (Principal Component Analysis), MDS (Multi-dimensional Scaling), SOM (Self Organized Map) and LLE (Locally Linear Embed). Next, we will briefly describe each input projection algorithm.

Principal component analysis (PCA) also named Karhunen-Loeve transform of Singular Value Decomposition (SVD) finds an orthogonal set of directions in the input space and delivers a method of finding the projections into these directions in a well-ordered style. The first principal component is the one that has the largest projection (the shadow of our data cluster in each direction). The orthogonal directions are named the eigenvectors of the correlation matrix of the input vector, and the projections - the equivalent eigenvalues. Because PCA orders the projections, we can decrease the dimensionality by truncating the projections to a given order. The reconstruction error is equal to the sum of the projections left out. The features in the linear projection space become the eigenvalues. PCA networks are typically utilized for data compression, offering the best m linear features, but they can also be used for data reduction in conjunction with multilayer perceptron classifiers [17].

The SOM algorithm is as follows [12]:

a. Initialize the weights with small different random values for symmetry breaking;

b. For each input data find the winning PE using a minimum distance rule;

$$\bar{i}(x) = arg_j min \parallel \bar{x}(n) - w_j \parallel \tag{1}$$

c. For the winning PE, update its weights and those in its neighborhood by:

$$w_j(n+1) = w_j + \eta(n)[x(n) - w_j(n)] \tag{2}$$

Where $\eta(n)$ is the step size. In the beginning, the step size should be large, but decrease progressively to zero, according to:

$$\eta(n) = \frac{1}{a_\eta + b_\eta n} \tag{3}$$

Where a_η and b_η are problem dependent constants.

The purpose of these adaptive constants is to guarantee, in the early stages of learning, malleability and the formation of local neighborhoods as well as, in the later stages of learning, constancy and adjustment of the map. These problems are very difficult to study theoretically, so heuristics have to be involved in the designation of these values.

Multi-Dimensional Scaling (MDS) is a set of related statistical techniques used in information visualization for exploring similarities or dissimilarities in data. The MDS algorithm starts with a matrix of item-item similarities and then assigns a location to each item in a lower dimensional space say m, such that the original similarities or dissimilarities are represented by the relative position in the lower dimensional space. The implemented algorithm measures the similarity or distance between points by correlation coefficient and applies simulated annealing to find the mapping from n dimensional space to m dimensional space, where $n>=m$, such that the distances between points in the m dimensional space approximate the correlation coefficients between points in the n dimensional space [12].

Locally Linear Embedding (LLE) is an unsupervised learning algorithm that computes low dimensional, neighborhood preserving embedding of high dimensional data. LLE attempts to discover nonlinear structure in high dimensional data by exploiting the local symmetries of linear reconstructions. The LLE algorithm includes following three major steps [12]:

1. Based on desired number of inputs say m, get the neighbors of each point in the sense Euclidean distance.

2. Compute the weights W_{ij} that best reconstruct each data point X_i with dimension n from its k neighbors, minimizing the cost defined as:

$$\zeta(W) = \sum_{i=1}^{n} |X_i - \sum_{j=1}^{k} W_{ij} X_j| \tag{4}$$

The solution can be reached by first solving the linear system of equations involving local covariance matrix and then rescaling the weights so that they sum to one.

3. Compute the projected inputs Y_i with dimension m that is best reconstructed by the weights W_{ij} minimizing the quadratic form defined as:

$$\zeta(W) = \sum_{i=1}^{n} |Y_i - \sum_{j=1}^{k} W_{ij} Y_j| \tag{5}$$

This is found by extracting the appropriate bottom m eigenvectors of a Matrix derived from the cost function.

3 Experiments and Results

The adaptive models compared in this study have a name that respects the following syntax:
Topology_ name - number_ of_ hidden_ layers - learning_ rule - input projection_ algorithm

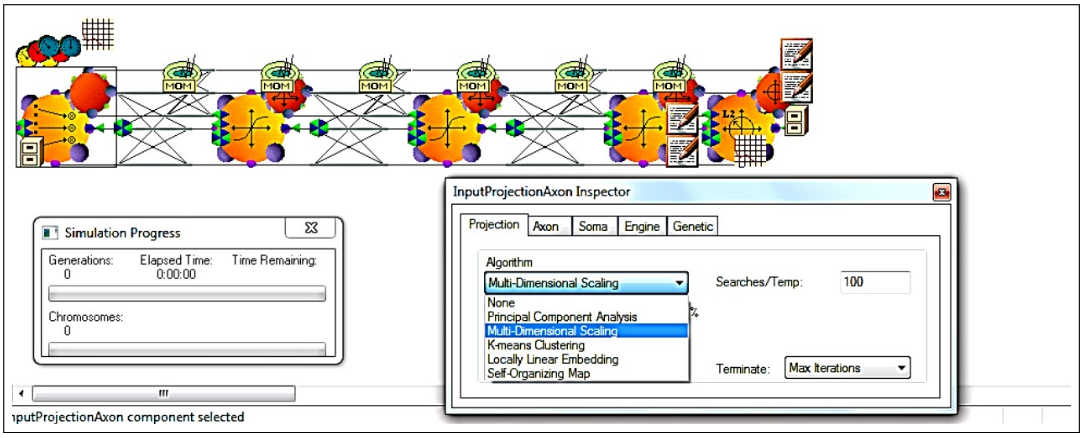
For example, the adaptive model named: MLP-2-CG-SOM is a Multi-Layer Perceptron with two hidden layers, trained using Conjugate Gradient learning rule and Self Organizing Map Input Projection algorithm.

In this study, in order to train the MLP, we made use of two large datasets: RTMATRIX and FPMATRIX. Each matrix has a size of 339 lines x 5825 columns. RTMATRIX consists of web services response time values while FPMATRIX stores similar web services throughput values. In our previous research works [1]- [5] we concluded that MLP-2-M and MLP-2-CG offered the most accurate prediction results for RTMATRIX and FPMATRIX, respectively. FPMATRIX and RTMATRIX were built by Z. Zheng, Y. Zhang and M.R. Lyu in [20] and [21].

To implement, train and test the adaptive models we utilized Neuro Solutions 6.21 development environment. As we can see in Figure 1, the main difference of this Neuro Solution implementation of MLP is the presence of InputProjectionAxon.

The InputProjectionAxon will apply either linear or non-linear transformation to convert n input data set into m input data set, where $n>=m$. The InputProjectionAxon Inspector allows us to select a different input projection algorithm and to set its parameters [12].

In this study, we have tested PCA in conjunction with MLP-2-M, but it works only on a subset of RTMATRIX/FPMATRIX since the separability of the classes is not always guaranteed. Another problem with linear PCA networks is outlying data points. Outliers will distort the estimation of the eigenvectors and create skewed data projections. RTMATRIX and FPMATRIX, having a huge number of values, have also a lot of outliers. Nonlinear networks are better able to handle this case. The importance of PCA analysis is that the number of inputs for the MLP classifier can be reduced a lot, which definitely influences the number of necessary training samples and the training time. And this was achieved when we made use of just a subset of RTMATRIX/FPMATRIX.



Simulation Progress

Generations: 0

Elapsed Time: 0:00:00

Time Remaining:

Chromosomes: 0

InputProjectionAxon Inspector

Algorithm: Multi-Dimensional Scaling

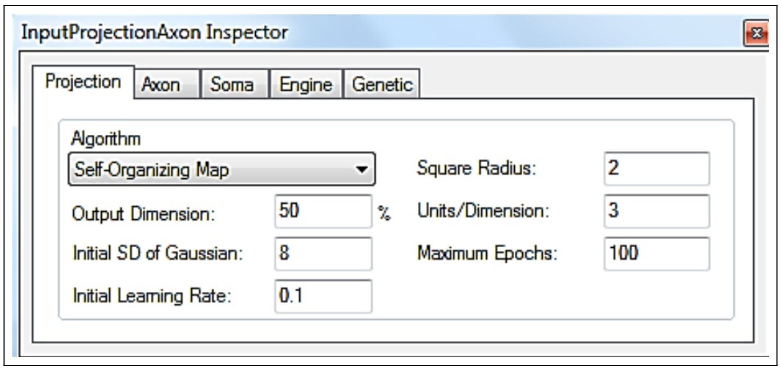
Searches/Temp: 100

Terminate: Max Iterations

ıputProjectionAxon component selected

Figure 1: MLP-2-M with InputProjectionAxon

SOM are an alternative to the PCA concept, and in our experiments, it works on both the entire RTMATRIX/FPMATRIX and a subset of them. By "works" we mean Neuro Solutions offer a results report. The parameters of SOM utilized in our simulation are presented in Figure 2.



InputProjectionAxon Inspector

Algorithm: Self-Organizing Map

Output Dimension: 50 %

Initial SD of Gaussian: 8

Initial Learning Rate: 0.1

Square Radius: 2

Units/Dimension: 3

Maximum Epochs: 100

Figure 2: SOM parameters

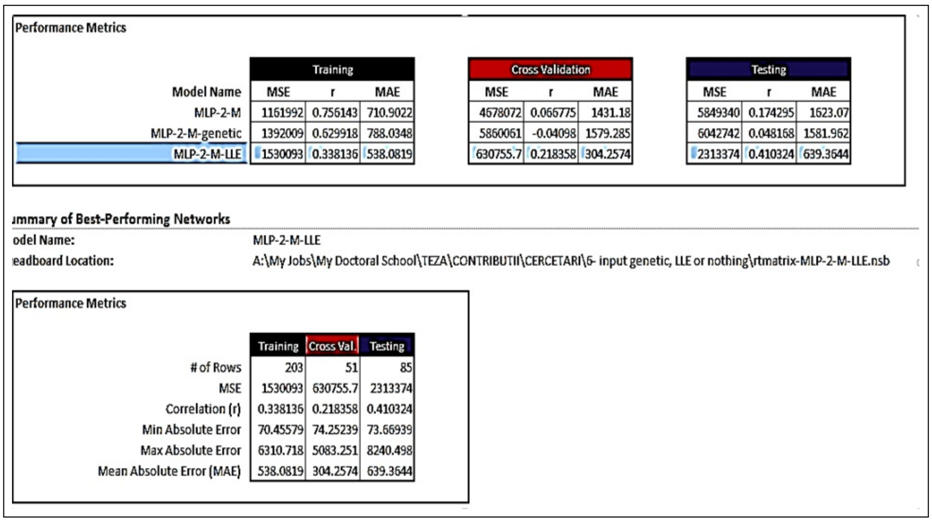
Output Dimension is used to specify the desired number of inputs as a percentage of number of inputs in the original data set. Initial SD of Gaussian is used to set the initial standard deviation for the Gaussian function that is used to define the neighborhood function. Initial Learning Rate is used to specify the starting learning rate and it is gradually reduced until the final learning rate is 0.01. Square Radius is used to define the neighbours. The shape of the neighborhood is defined as a square. Units/Dimension is used to define the smoothness of the projected space or grids. Maximum Epochs is used to specify the maximum number of epochs before termination of the computation.

MDS algorithm applied on the entire RTMATRIX/FPMATRIX determined Neuro Solutions to block indefinitely in a "not responding" state, but when we selected just a subset of them, it worked and offered some results.

LLE offers the best results, in comparison with the other four input projection algorithms, since it works on both FPMATRIX and RTMATRIX and provides the lowest prediction error.

In Neuro Solutions, the best input optimization algorithm is performed by the GeneticControl component. This component implements a genetic algorithm to optimize the inputs. Genetic Algorithms are search procedures based upon the principles of evolution witnessed in nature that combine selection, crossover, and mutation operators. They search for an optimal solution until a termination criterion is met. In Neuro Solutions the criteria used to evaluate the fitness of each potential solution is the lowest cost attained during the training run. The solution to a problem is called a chromosome and consists of a collection of genes, which are simply the inputs to be optimized. The genetic algorithm produces an initial population, evaluates this population by training a neural network for each chromosome and then evolves the population through multiple generations in the search for the best inputs.

In Figure 3 is presented the compared results between MLP-2-M with no input optimization or projection, MLP-2-M with only genetic optimization and MLP-2-M with both genetic and LLE input projection.



Performance Metrics

	Training			Cross Validation			Testing		
Model Name	MSE	r	MAE	MSE	r	MAE	MSE	r	MAE
MLP-2-M	1161992	0.756143	710.9022	4678072	0.066775	1431.18	5849340	0.174295	1623.07
MLP-2-M-genetic	1392009	0.629918	788.0348	5860061	-0.04098	1579.285	6042742	0.048168	1581.962
MLP-2-M-LLE	1530093	0.338136	538.0819	630755.7	0.218358	304.2574	2313374	0.410324	639.3644

ummary of Best-Performing Networks

odel Name: MLP-2-M-LLE

eadboard Location: A:\My Jobs\My Doctoral School\TEZA\CONTRIBUTII\CERCETARI\6- input genetic, LLE or nothing\rtmatrix-MLP-2-M-LLE.nsb

Performance Metrics

	Training	Cross Val.	Testing
# of Rows	203	51	85
MSE	1530093	630755.7	2313374
Correlation (r)	0.338136	0.218358	0.410324
Min Absolute Error	70.45579	74.25239	73.66939
Max Absolute Error	6310.718	5083.251	8240.498
Mean Absolute Error (MAE)	538.0819	304.2574	639.3644

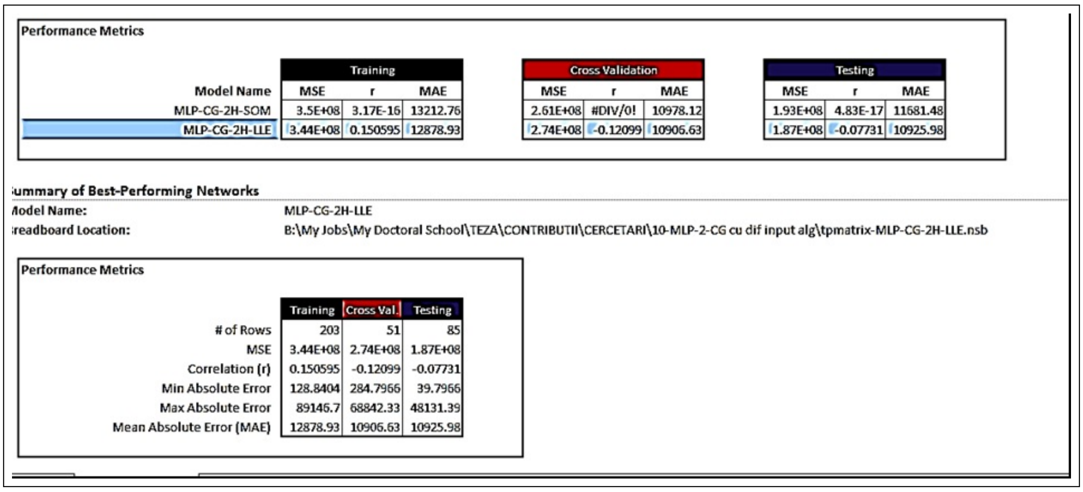
Figure 3: Effect of input optimization and projection on prediction accuracy

The comparison was performed on the entire RTMATRIX and the results show that MLP-2-M-LLE is the most accurate adaptive model for web services response time prediction. As we can observe, the dataset was divided in three subsets: training (50%), cross-validation (30%) and testing (20%). The training data set is obviously used for training, the cross-validation data set tests the model in the training phase and determines when the training is stopped, while the testing data set is utilized to investigate the prediction accuracy of the model when it receives new samples at the input.

Having the experience with the MLP trained on RTMATRIX, the subject of FPMATRIX investigation can be reduced at the comparison between MLP-2-CG-SOM and MLP-2-CG-LLE, both having as input optimization a genetic algorithm. We tested all input projection algorithms on FPMATRIX, each one separately and the only two that could find a solution were LLE and SOM. Consequently, we have labeled MDS and PCA input projection algorithms as not suitable for our prediction problem. The literature also recommends them for other types of problems like classification or clustering. The results of the comparison between LLE and SOM, when training a MLP with two hidden layers, Conjugate Gradient learning rule and Genetic input optimization, are shown in Figure 4.

The results reported in Figure 4 show that LLE is again the best input projection algorithm, since it offers the best prediction accuracy when training the MLP-2-CG on both RTMATRIX and FPMATRIX.

Neuro Solutions offers a fifth input projection algorithm, K-means clustering, but in our researches we have not used it, since it is not appropriate for prediction problems.



Performance Metrics

Model Name	Training			Cross Validation			Testing		
	MSE	r	MAE	MSE	r	MAE	MSE	r	MAE
MLP-CG-2H-SOM	3.5E+08	3.17E-16	13212.76	2.61E+08	#DIV/0!	10978.12	1.93E+08	4.83E-17	11681.48
MLP-CG-2H-LLE	3.44E+08	0.150595	12878.93	2.74E+08	-0.12099	10906.63	1.87E+08	-0.07731	10925.98

ummary of Best-Performing Networks

odel Name: MLP-CG-2H-LLE

readboard Location: B:\My Jobs\My Doctoral School\TEZA\CONTRIBUTII\CERCETARI\10-MLP-2-CG cu dif input alg\tpmatrix-MLP-CG-2H-LLE.nsb

Performance Metrics

	Training	Cross Val.	Testing
# of Rows	203	51	85
MSE	3.44E+08	2.74E+08	1.87E+08
Correlation (r)	0.150595	-0.12099	-0.07731
Min Absolute Error	128.8404	284.7966	39.7966
Max Absolute Error	89146.7	68842.33	48131.39
Mean Absolute Error (MAE)	12878.93	10906.63	10925.98

Figure 4: A comparison between MLP-2-CG-SOM and MLP-2-CG-LLE

4 Conclusions and Future Works

QoS is still essential for high quality web services and remains one of the subjects that raise researchers' interest. More and more authors believe that large data may be as significant to society as the Internet. Social networks can play an important role in large and big data analytics and the technology trends indicate that they soon will interconnect not just people, but software modules like web services.

Consequently, in this research work we have studied four input projections algorithms, in order to determine which one increases the prediction accuracy of a Multi-Layer Perceptron (MLP) with two hidden layers, trained with web services large data. The result reports, for both FPMATRIX and RTMATRIX, show the Locally Linear Embed (LLE) as the most accurate input projection algorithm.

Concluding, MLP with two hidden layer, having as input projection algorithm LLE and a genetic algorithm for input optimization, can provide more accurate prediction results, when it is trained with large datasets of web services QoS criteria values.

In future works we will investigate different prediction adaptive models in order to improve web services QoS criteria prediction accuracy.

Bibliography

[1] R.-D. Albu (2013), *Contributions regarding the quality and reliability of web services*, PhD Thesis, University of Oradea.

[2] R.-D. Albu (2013), Investigating the Effect of Hidden Layers Number on Web Services Response Time Prediction, *Nonconventional Technologies Review*, ISSN 1454-3087, 7(1):4-9.

[3] R.-D. Albu, I. Felea, F. Popentiu-Vlădicescu (2013), On the Best Adaptive Model for Web Services Response Time Prediction, *The 20th Int. Conference on Systems, Signals and Image Processing, IWSSIP 2013*, CD Edition, IEEE Catalog Number : CFP1355E-CDR, ISBN: 978-1-4799-0942-1,39-42.

[4] R.-D. Albu, F. Popentiu-Vlădicescu (2013), On the Best Learning Algorithm for Web Services Response Time Prediction, paper accepted at *ESREL "Annual Conference, Advances in Safety, Reliability and Risk Management.*

[5] R.-D. Albu, F. Popentiu-Vladicescu (2013), A Comparative Study For Web Services Response Time Prediction, *The 9th Int. Scientific Conference eLSE 2013 "eLearning and Software for Education"*, 1: 656-665, Bucharest, , ISSN 2006-026x, CD Edition.

[6] A. Klasnja-Milicevic, M. Ivanovic, A. Nanopoulos (2009), The Use of Nonlinear Manifold Learning in Recommender Systems, *4th Int. Conference On Information Technology*, http://www.zuj.edu.jo/conferences/ICIT09/PaperList/Papers/Aritificial Intelligence/525.pdf.

[7] http://arxiv.org/abs/1111.4503 (available 16.11.2013)

[8] http://blogs.sas.com/content/sascom/2012/04/11/will-big-data-and-high-performance-analytics-flatten-the-world/ (22.10.2013)

[9] http://tech.sina.com.cn/i/ 2012-11-12/00207788375.shtml (available 16.11.2013)

[10] http://www.datadeluge.com/ (available 12.11.2013)

[11] L. Aspirot, P. Belzarena, B. Bazzano, G. Perera (2005), End-to-end quality of service prediction based on functional regression, *Proc. of Third Int. Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks (HET-NETs 2005)*, Ilkley, UK, 1-8.

[12] Neuro Solutions help: http://www.aertia.com/docs/nd/neurosolutionshelp.pdf.

[13] P. Belzarena and L. Aspirot (2010), End-to-end quality of service seen by applications: A statistical learning approach, *Int. J. of Computer and Telecommunications Networking*, 54(17):3123-3143.

[14] Hu Y., Mu D., Gao A., Dai G.(2011), The Research of QoS Approach in Web Servers, *Int J Comput Commun*, ISSN 1841-9836, 6(4):636-647.

[15] Navarro M., Donoso Y. (2012), An IMS Architecture and Algorithm Proposal with QoS Parameters for Flexible Convergent Services with Dynamic Requirements, *Int J Comput Commun* ISSN 1841-9836, 7(1):123-134.

[16] Park E.-C. et al. (2011), Quality of Service Control for WLAN-based Converged Personal Network Service, *Int J Comput Commun*, ISSN 1841-9836, 6(4):716-733.

[17] P. Talebi Fard et al. (2013), Semantic Based Networking of Information in Vehicular Clouds Based on Dimensionality Reduction, *Proc. of the third ACM int. symposium on Design and analysis of intelligent vehicular networks and applications*, ACM, 69-76.

[18] Wei Tan, M. Brian Blake, Iman Saleh, Schahram Dustdar (2013), Social-Network-Sourced Big Data Analytics, Web-Scale Workflow, *Internet Computing, IEEE*, 17(5):62-69.

[19] Liang-Jie Zhang, Jia Zhang, Hong Cai (2007), *Services Computing*, Tsinghua University Press, Springer.

[20] Z. Zheng, Y. Zhang, M.R. Lyu (2010), Distributed QoS Evaluation for Real-World Web Services, *Proc. of the 8th Int. Conference on Web Services (ICWS2010)*, Miami, Florida, USA, 83-90.

[21] Z. Zheng, Y. Zhang, M.R. Lyu (2011), Exploring Latent Features for Memory-Based QoS Prediction in Cloud Computing, *Proc. of the 30th IEEE Symposium on Reliable Distributed Systems (SRDS 2011)*, Madrid, Spain, 1-7.