computing@computingonline.net
www.computingonline.net

ISSN 1727-6209
International Journal of Computing

SPOT PRICE PREDICTION FOR CLOUD COMPUTING USING NEURAL NETWORKS

Volodymyr Turchenko [1], Vladyslav Shults [1], Iryna Turchenko [1], Richard M. Wallace [2], Mehdi Sheikhalishahi [3], Jose Luis Vazquez-Poletti [2], Lucio Grandinetti [3]

[1] Dept. of Info. Comp. Sys. & Control, Ternopil National Economic University, Ternopil, Ukraine,
{vtu, vshu, itu}@tneu.edu.ua
[2] Dept. of Computer Arch. & Automation, Complutense University, Madrid, Spain
wallacerim@aol.com, jlvazquez@fdi.ucm.es
[3] Dept. of Electronics, Comp. Sci. & Sys., University of Calabria, Rende (CS), Italy
alishahi@unical.it, lugran@unical.it

Abstract: Advances in service-oriented architectures, virtualization, high-speed networks, and cloud computing has resulted in attractive pay-as-you-go services. Job scheduling on such systems results in commodity bidding for computing time. Amazon institutionalizes this bidding for its Elastic Cloud Computing (EC2) environment. Similar bidding methods exist for other cloud-computing vendors as well as multi–cloud and cluster computing brokers such as SpotCloud. Commodity bidding for computing has resulted in complex spot price models that have ad-hoc strategies to provide demand for excess capacity. In this paper we will discuss vendors who provide spot pricing and bidding and present the predictive models for future short-term and middle-term spot price prediction based on neural networks giving users a high confidence on future prices aiding bidding on commodity computing.

Keywords: Spot Market; Cloud Computing; Resource Management; Neural Networks; Prediction.

1. INTRODUCTION

Cloud computing is seen as a hyper-specialization of general-purpose information technology with computing attributes and characteristics making it an ideal information technology delivery model. We assert that the real definition of cloud computing is the convergence of essential ideal characteristics of various distributed computing technologies. A cloud/grid computing system is reusing known economic models, but has wide variances due to large sets of variables for both the operation of the system and the hosting and execution of client applications. These variances provide new business models and markets that provide profitability to the system owner, but must also be offered at a unit price that is attractive to users so that the commodity system is sufficiently used to provide profit. This price-point is dependent on time of use, number of users, and capitalization cost. With ubiquitous cloud computing and grid computing technology, it can become economically beneficial as it is available at any time for everyone. Economic benefits of cloud and grid adoption are the main drivers as shown in the study by Armbrust [1]. Initially, cloud providers had only a fixed price for their service offerings [2-3]. As cloud systems grow larger and are partitioned into more unique configurations, this fixed price method becomes inefficient when total demand is much lower than data center capacity leading to under-use of the system. Cloud providers need an incentive mechanism to encourage users to submit more jobs. When total demand rises over data center capacity, it is desirable to provide an incentive to users to reduce their demand through raising per–unit costs, decreasing performance, or decreasing system availability.

We illustrate the spot price (user cost) and spot instance (system instance at that cost) mechanism on the example of the Amazon Elastic Computing 2 Service (EC2). In 2009, Amazon introduced a new set of spot instances to sell its unused data center capacity based on a new market mechanism offering a variable pricing method. With this service, users are able to bid for unused capacity. The spot price mechanism for EC2 shares many similarities with the standard uniform price auction mechanism. The spot price charged for a request, may fluctuate depending on the supply of, and demand for, spot instance capacity. Spot prices are a tuple of {maximum price per hour the user wishes to pay for an instance type, the region desired, and the number of spot instances to run}. If the maximum price bid exceeds the current spot price, the job(s) will run until termination by the user or the spot price increases above the user set maximum price. The cost of spot instance hours are billed based on the spot price at the start of each hour an instance executes. If the user spot instance is interrupted in the middle of an hour of an instance use (because the spot price exceeded the user maximum bid price), the user is not billed for that partial hour of spot instance use. However, if the user terminates the spot instance a charge occurs for the partial hour of use.

Market driven resource allocation has been applied to grid computing environments [2-3]. Recently, it has also been adopted by cloud computing. The auction-based resource allocation mechanism in the cloud spot market causes the price of services to be dynamic. The auction-based mechanism tries to address the question of finding the best match for customer demanded services in terms of supply and price to maximize provider revenue and customer satisfaction. For the provider, we have revenue maximization, supply, and spot price; whereas for the customer, we have cost minimization, demand, and bid price.

Short term forecasting has been a key to economic optimization in the electric energy industry [4] and is essential for power systems planning and operation. An electricity costing model does not have a mechanism to store electricity as it can not store its service while a cloud system can, thus the floor of the electricity model can be much lower than that of a cloud system as electricity can not be stored in sufficient quantities to keep its floor higher. The alternative is to restrict generation and loose the currently produced power. In a cloud model, the system can be made idle, almost instantly, and await a price point when it would be profitable to operate. For both the cloud market and the electricity market, accurate forecasting is very important for both production and consumption of commodities like compute resources and electricity in order to optimize their buying and selling decisions.

In this paper, we demonstrate a neural network method to predict spot prices that can be useful to users of cloud computing for bidding on spot instances of cloud system providers.

2. RELATED WORK

2.1. SURVEY OF CLOUD AND GRID COMPUTING PROVIDERS

In an examination of the current literature available on more than 120 cloud IaaS and PaaS providers, over 98 percent of cloud and grid computing providers do not have a spot price and auction mechanism including Microsoft's Azure products and Google's Engine Products. The SpotCloud system is the only provider found that has a mechanism similar to the Amazon EC2, but it is devoid of many of the control, security, and ownership mechanisms that EC2 has. Also noteworthy is the OpenStack open source cloud infrastructure that many IaaS vendors are supporting as a competitor to Amazon EC2.

1) SpotCloud: SpotCloud is an IaaS cloud clearing-house from Enomaly Inc. SpotCloud brokers are buyers for, and sellers of, cloud infrastructure capacity. SpotCloud is a bidding exchange that establishes a "standard" computing unit across sellers for simplified user management allowing buyers to bid on a commodity product. Of note, the computing unit sold is "raw" as it does not offer service level agreements or value-added elements such as security or application restart and data backup.

Sellers of capacity join the exchange through the Enomaly Web site and installing its own cloud platform, Elastic Computing Platform (ECP), OpenStack, or other platforms. This is done via an API published by Enomaly. Sellers list the capacity, geography, and price requirements on the SpotCloud Web portal, where the information is presented to buyers. A blind-listing can be done if a seller feels that very low pricing on excess capacity may hurt its brand or impact direct sales channels otherwise the seller's name is listed.

Enomaly provides transaction monitoring, billing, buyer payment collection, and payment to sellers. Enomaly collects a percentage of the seller proceeds. In the SpotCloud model, much of the value that Enomaly brings to the complex cloud marketplace is convenience and simplification. Sellers have little administrative overhead, and their costs are aligned with revenue, since Enomaly's fee (a percentage of sales) covers marketing, sales, billing, collection, and other costs of doing business. Buyers are relieved of the burden of researching individual providers, and because the rates are posted, they can be assured that they are paying market-defined rates for the services they buy.

With only ten percent of U.S. businesses using IaaS, enterprises are cautious about entering the cloud due to concern over loss of control, security, performance of applications – factors that are not addressed in a commodity cloud which may lead to simple bid, commodity cloud services being only useful for a few cases relegating an IaaS like SpotCloud a small market of wholesalers.

2) OpenStack: OpenStack is a classic "mash-up" of the right technology and user needs being in the right place at the right time. The OpenStack Foundation (http://www.openstack.org/foundation) has many organizations and companies contributing and using this cloud system software. The primary commercial lead in this effort is Rackspace. Currently VMware does not have a mechanism for spot pricing or an auction mechanism. Through the vCloud product, part of their software suite, VMware lists 177 companies that offer private and public IaaS cloud configurations. VMware will also support OpenStack by extending support for ESX hypervisor in OpenStack.

IBM is using OpenStack as a central part of its future cloud strategy in the IBM SmartCloud Orchestrator service. IBM developed the SmartCloud platform before OpenStack was founded in 2009-2010, but now is replacing that core component with OpenStack which will be the foundation of the company's cloud strategy. The SmartCloud Orchestrator service provides configuration of the compute, storage and networking resources needed applications run on the IBM SmartCloud platform. SmartCloud is a pay-as-you-go public cloud offering with components for private cloud or dedicated hosted infrastructure as an IaaS or PaaS. OpenStack is becoming a central cloud component for IBM, HP, Dell, Cisco, Red Hat, and Rackspace have also announced major initiatives around the OpenStack project.

While an interesting competitor to Amazon EC2, the OpenStack project is only an enabler of IaaS and PaaS sites and has no auction or spot price mechanisms.

2.2. SPOT MARKET PREDICTION IN THE CLOUD

Spot price and spot market prediction have a key role in the economics of the electric energy industry and is essential for power systems planning and operations as discussed in [4-5]. Also in the literature, there are neural network based techniques to forecast electricity spot price. In [4], neural network techniques based on short-term load forecasting is presented to predict short-term spot price in the Australian national electricity market.

In [6], characteristics of Amazon spot instances have been explored and the authors have done their comprehensive analysis based on one-year price history in four data centers of Amazon's EC2. They analyzed different types of spot instances that Amazon offers in terms of spot price and the inter-price time (time between price changes) and determined the time dynamics for spot prices by hour-in-day and day-of-week. Moreover, they have proposed a statistical model that fits well these two data series. The statistical models based on the mixture of Gaussian distribution with three or four components are able to capture spot price dynamics as well as the inter-price time of each spot instance. Their model exhibits a good degree of accuracy under realistic working conditions.

Amazon provides the price history to help customers decide their bids. Figure 1 shows an example of a price history graph obtained from [7]. Currently, Amazon EC2 spot instance services are available for eight types of virtual machines. Each virtual machine type has different resource capacities for CPU, memory and disk. Amazon EC2 runs one spot market for each virtual machine type in each geographical availability zone [8]. All spot markets share the free data center capacity. This capacity is the remaining resources after serving all the guaranteed (i.e., contracted) instances.

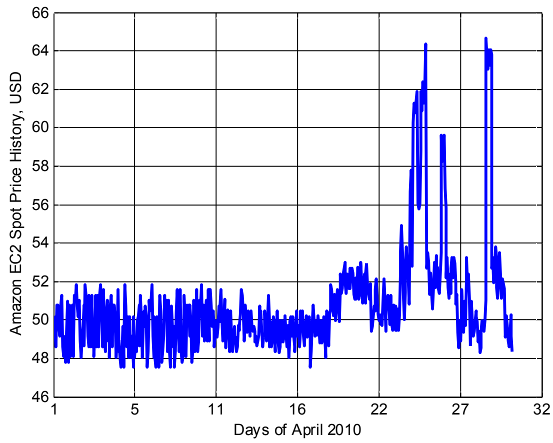

Fig. 1 – Amazon EC2 Spot Price History.

3. NEURAL-BASED PREDICTION METHOD

Over the past few decades, many different methodologies have been proposed for generating reliable predictions ranging from technical [9] and statistical analysis [10] to artificial intelligence techniques [11]. One of the artificial intelligence techniques, neural networks (NN), represent a promising alternative as the inherent learning ability allows effective capturing of the dynamic, nonlinear and complicated features of the predicted data. For example, a model of a feed-forward neural network, a multi-layer perceptron, showed excellent prediction results on many different financial examples [12–17]. Therefore, for the prediction of the spot prices we have used two standard models of

NNs: a multilayer perceptron (MLP), Fig. 2, and a recurrent neural network (RNN), Fig. 3 [11, 18]. These models are well researched and they are capable to fulfill approximation tasks with any required level of accuracy.

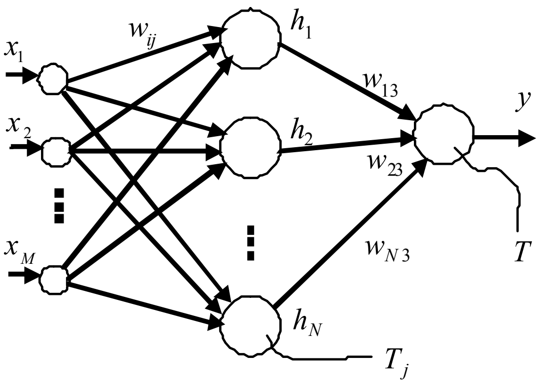

Fig. 2 – The structure of a three-layer MLP.

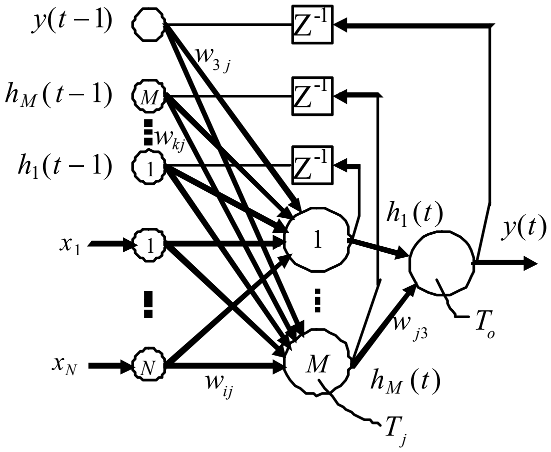

Fig. 3 – Structure of a recurrent neural network.

The output value of the three-layer perceptron can be formulated as:

$$y = F_3\left(\sum_{j=1}^{N} w_{j3}\left(F_2\left(\sum_{i=1}^{M} w_{ij}x_i - T_j\right)\right) - T\right), \quad (1)$$

where N is the number of neurons in the hidden layer, w_{j3} is the weight of the synapse from neuron j of the hidden layer to the output neuron, w_{ij} are the weights from the input neurons to neuron j in the hidden layer, x_i are the input values, T_j are the thresholds of the neurons of the hidden layer and T is the threshold of the output neuron [11, 18].

The output value of RNN can be formulated as:

$$y = F_3\left(\sum_{j=1}^{M} w_{j3}h_j - T_0\right), \quad (2)$$

$$y_j = F_2\left(\sum_{i=1}^{N} w_{ij}x_i + \sum_{k=1}^{M} w_{kj}h_k(t-1) + w_{3j}y(t-1) - T_j\right), (3)$$

where M is the number of neurons in the hidden layer, w_{j3} is the weight of the synapse from neuron j of the hidden layer to the output neuron, N is the number of input neurons, w_{ij} are the weights from the input neurons to neuron j in the hidden layer, x_i are the input values, w_{kj} is the synapse from k context neuron of the hidden layer to j neuron of the same layer, $h_k(t-1)$ is the output value of k context neuron of hidden layer in the previous moment of time $t-1$, w_{3j} is the synapse from context output neuron to j neuron of the hidden layer, $y(t-1)$ is the value of context output neuron in the previous moment of time $t-1$, T_j are the thresholds of the neurons of the hidden layer and T_o is the threshold of the output neuron [11, 18]. The logistic activation function $F(x) = 1/(1+e^{-x})$ is used for the neurons of the hidden (F_2) and output layer (F_3) for the both MLP and RNN models. The standard back-propagation training algorithm [11] with constant or adaptive learning rate [20] is used for the training for both NN models.

Amazon EC2 provides spot instances from small standard systems to extra-large multiprocessor systems (at about 88 cores) and GPU co-processing. We have used historical data about spot prices of the "medium" cloud instances based on Linux and Windows operation systems called m1.linux and m1.windows respectively. These data are available on the Amazon web site [7]. For our experiments, we used 3842 spot price data points for 7 months starting from December 2009 and ending June 2010, which is a period of 215 days. This averages to 17 records of spot price for each day. We have divided all the data on appropriate months in order to do the experiments and visualization in a more efficient way.

For the input data analysis, it is beneficial to apply a moving simulation mode [14] since it provides the use of last recent data in the time series avoiding the impact of the "old" historical data on the quality of the prediction. The successful usage of the moving simulation mode for the financial application [19] showed that it is not necessary to choose a large data "window" for the analysis since a larger window will include the "old" historical data that makes the NN re-training less efficient.

Spot price prediction is beneficial in fulfilling short-term (single step) and middle- or long-term (multiple steps) predictions. The short-term prediction mode may provide better prediction results since the preliminary analysis shows that the trend of data about spot price could change unpredictably fast and the short-term prediction could capture this change in an accurate manner. Since we have an average of 17 records about spot price per day, we have approximately 1.3 hours of time before new data arrives and, therefore, we can do the re-training of NN for each prediction step and improves prediction accuracy. On the other hand, the prediction interval of 1.3 hours is not suitable from the practical point of view since users of cloud resources want planning of their bidding strategies for several days ahead, with one to five days ahead being the most important.

4. EXPERIMENTAL RESULTS

We have formed the training set for our NN models using the Box-Jenkins method [10]. We have received the following results for the short-term [21] and middle-term prediction modes.

Short-Term Prediction Mode. Similar to [19], we have chosen 20 values as the size of the moving simulation window. The MLP architecture of 5-10-1 was chosen as the prediction structure in this mode. In particular, we chose 5 input neurons as it is sufficient within the 20 input data points of the moving simulation window with 10 neurons of the hidden layer being sufficient to provide a good generalization and prediction ability. We used the constant learning rate of 0.1 for both hidden and output layers of the MLP. The MLP is trained to reach the sum-squared training error of 10^{-5} with 2×10^{6} training epochs and then, on each step of the moving simulation mode, the MLP was re-trained using 7×10^{5} training epochs. One prediction step took about 12 seconds on a computer with an Intel Core 2 Duo processor at 2.4 GHz with 3 GB of RAM. The total computational time for the whole experiment in a short-term prediction mode was about 13 hours. According to the short-term prediction mode the real and predicted spot prices for both m1.linux and m1.windows cloud instances for each month from December 2009 through June 2010 are depicted in Figures 4 through 10. As shown in these figures, the MLP model in the proposed configuration provides a very good representation of the actual trending for both prediction cases. The numerical analysis of the predictions depicted in Table 2 shows the high accuracy of the proposed approach as the monthly average relative prediction errors do not exceed 5.6% for the m1.linux data and 6.4% for the m1.windows data. The average relative prediction errors for the whole testing period of six months are 3.3% and 3.7% respectively for m1.linux and m1.windows data. During the empirical analysis we have noticed that the amplitude of several data points is largely above or below of some average amplitude of signal change. In those points the prediction gives the result much different from this largely changed amplitude assuming that there should be the signal with much smaller amplitude. Therefore we have considered such prediction results as outliers. We have counted a prediction result as an outlier when its relative prediction error is more than 10%. The analysis of the prediction results shows that we have 155 (about 4.0% of the total results) and 188 (about 4.9% of the total results) outliers for the m1.linux and the m1.windows experiments respectively.

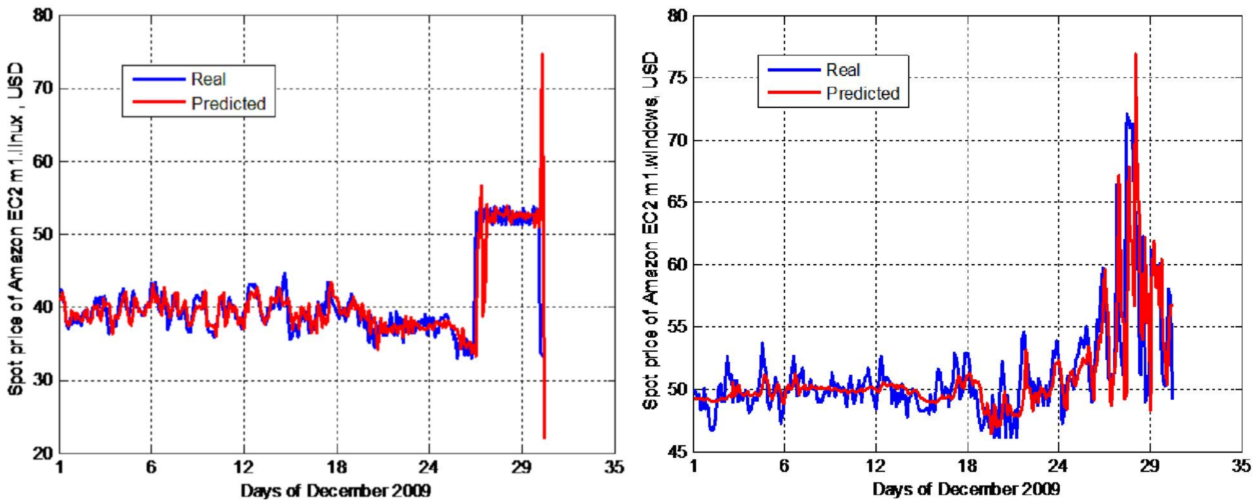

Fig. 4 – Short-term prediction results for m1.linux and m1.windows for Dec. 2009.

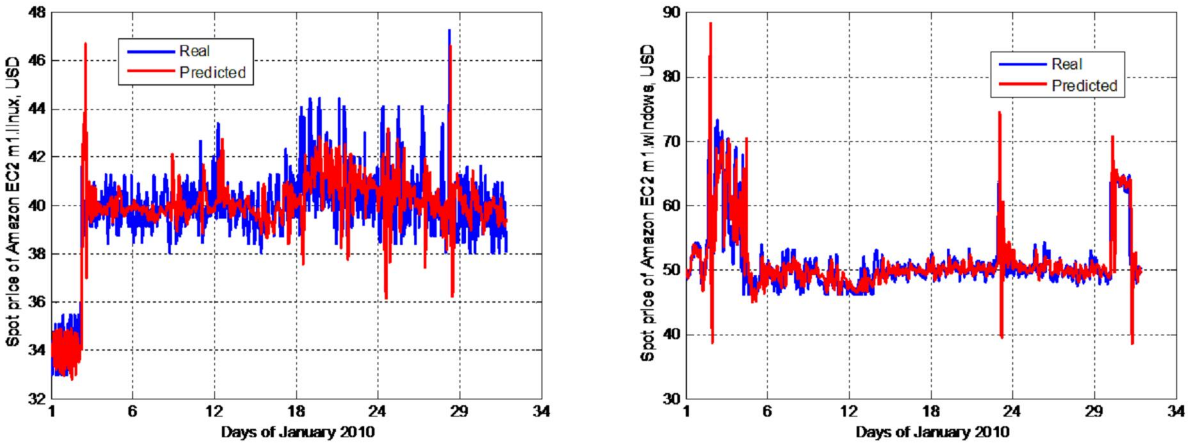

Fig. 5 – Short-term prediction results for m1.linux and m1.windows for Jan. 2010

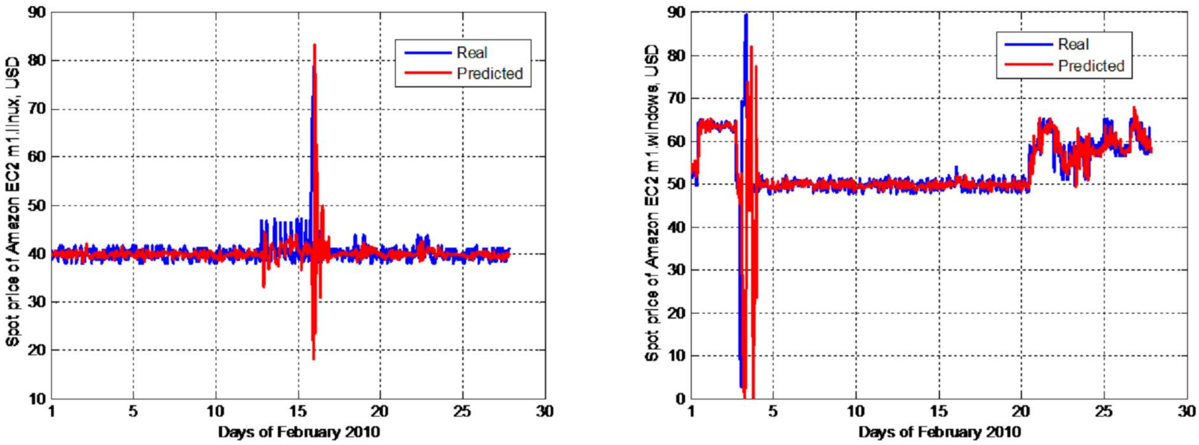

Fig. 6 –Short-term prediction results for m1.linux and m1.windows for Feb. 2010.

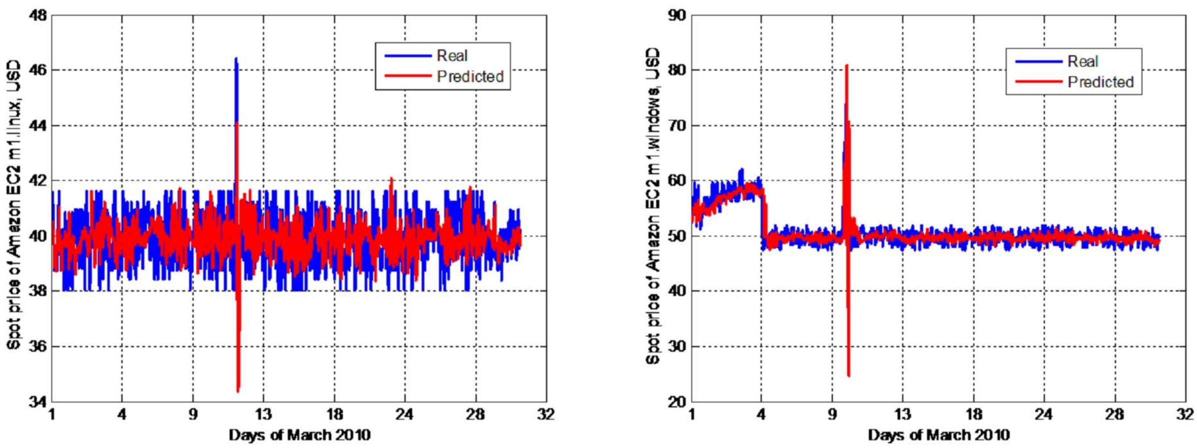

Fig. 7 – Short-term prediction results for m1.linux and m1.windows for Mar. 2010.

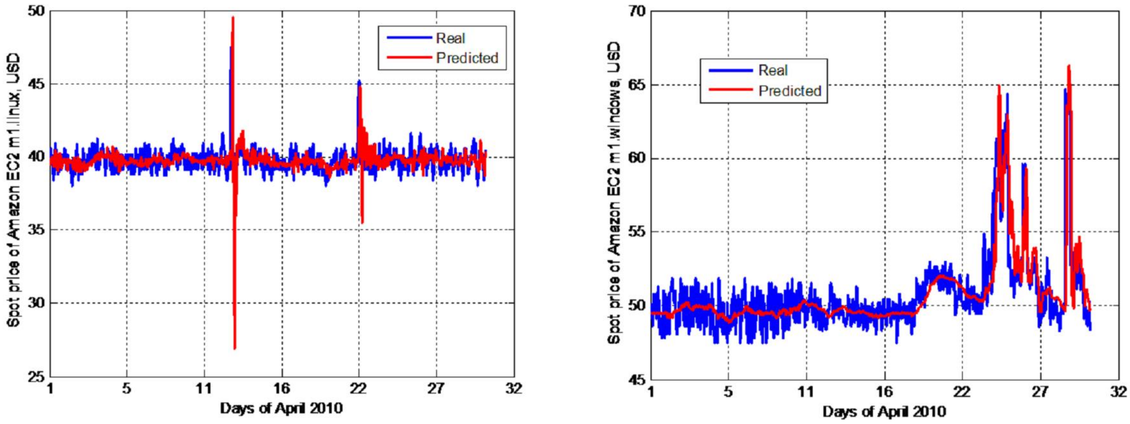

Fig. 8 – Short-term prediction results for m1.linux and m1.windows for Apr. 2010.

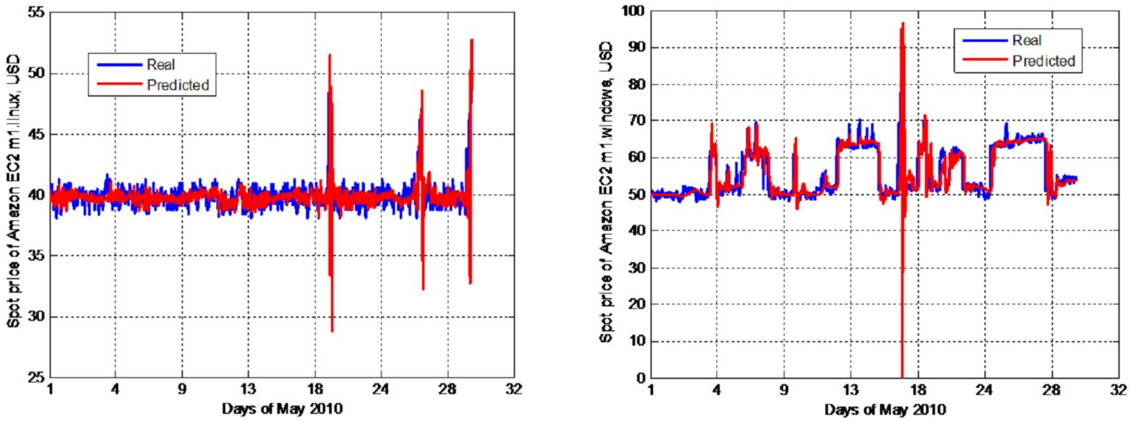

Fig. 9 – Short-term prediction results for m1.linux and m1.windows for May. 2010.

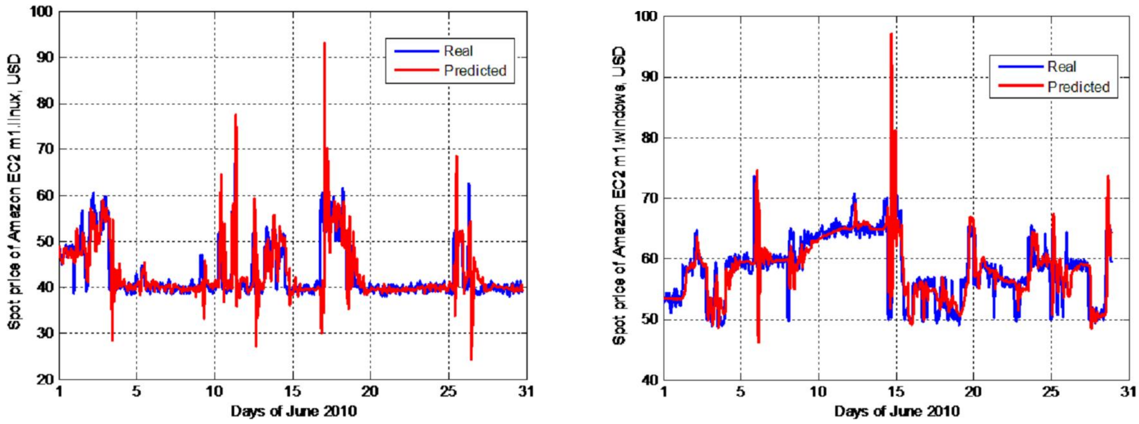

Fig. 10 – Short-term prediction results for m1.linux and m1.windows for Jun. 2010.

Table 1. Numerical results for short-term prediction mode.

Experiments	Avg. relative prediction error(%)		Num.& Percent of outliers (Rel. Predict. Err. >10%)	
	m1.linux	m1.windows	m1.linux	m1.windows
Dec.2009(266)	4.4	3.5	12 (4.5%)	16 (6.0%)
Jan.2010(556)	2.6	3.4	5 (0.9%)	25 (4.5%)
Feb.2010(556)	4.0	6.4	25 (4.5%)	28 (5.0%)
Mar.2010(663)	2.6	2.6	2 (0.3%)	10 (1.5%)
Apr.2010(564)	1.7	2.3	5 (0.9%)	7 (1.2%)
May 2010(637)	2.0	3.6	10 (1.6%)	49 (7.7%)
Jun.2010(595)	5.6	3.9	96 (16.1%)	53 (8.9%)
Average relative error/total (% number of) outliers:	3.3	3.7	155 (4.0%)	188 (4.9%)

Middle-Term Prediction Mode. Taking into account the long simulation time of the computational experiment above we have provided the middle-term prediction for the m1.linux data only. We have used 88 and 176 input data points from December 2009 to June 2010 as training data. We have used two NN models (MLP 5-10-1 and RNN 5-10-1) with reverse connections from both hidden and output layers. Both models use adaptive and constant learning rates. The constant learning rates were 0.5 and 0.5 for the hidden and output layers for the MLP model and 0.1 and 0.1 for the RNN model. Both models are trained to reach the sum-squared training error of 10^{-5} with 5×10^5 training epochs. The training time of one middle-term prediction experiment took about 30 seconds using MLP model and 45 seconds using RNN model for the case of 88 input data points and about 60 seconds using MLP model and 90 seconds using RNN model for 176 input data points. All middle-term prediction experiments were executed on a Phenom II x 4 956 processor 3.4 GHz and 4 GB of RAM. The total computational time for the whole experiment in a middle-term prediction mode was about 4 hours. According to the middle-term prediction mode, the average and maximum relative prediction errors for one to five days for the four NN models are presented in Table 2 using training data for 88 data points; Table 3 presents data for 176 data points. The lower-case index values indicate the following: 1) the MLP model with adaptive learning rate; 2) the MLP model with constant learning rate; 3) the RNN model with adaptive learning rate; 4) the RNN model with constant learning rate. The graphical representations of middle-term prediction results for each testing month are detailed in Figures 11 to 14.

As can be seen the MLP and RNN models provide accurate prediction results for the majority of cases. For both of the 88 and 176 input training data sets the prediction results are a bit less accurate for the December 2009 and the June 2010 time periods on the fifth prediction day. Therefore the obtained results showed us good prediction abilities of neural networks for the middle-term prediction of spot prices of cloud resources.

Table 2. Numerical results for middle-term prediction using 88 training data points for each month.

Month	Relative prediction errors, %									
	1 day		2 days		3 days		4 days		5 days	
	avr	max	avr	max	avr	max	avr	max	avr	max
Dec 2009	4.3_2	8.4_2	4.0_2	11.4_2	4.5_2	11.5_2	4.1_2	11.5_2	4.3_2	14.7_2
Jan 2010	1.7_2	4.3_2	1.6_2	4.3_2	1.5_2	4.3_2	1.7_2	5.4_2	1.7_2	5.4_2
Feb 2010	2.0_4	4.3_4	2.4_4	4.9_2	2.5_2	4.9_2	2.4_4	5.0_4	2.4_2	5.0_4
Mar 2010	2.2_2	4.6_2	2.2_2	4.7_2	2.3_2	4.8_2	2.4_2	4.9_2	2.4_2	5.0_2
Apr 2010	1.2_1	2.2_1	1.2_1	2.2_1	1.3_2	2.8_2	1.3_2	2.9_2	1.4_2	3.4_2
May 2010	1.4_3	3.8_3	1.5_3	3.8_3	1.5_2	4.2_2	1.6_2	4.2_2	2.0_2	4.3_2
Jun 2010	2.4_3	8.2_3	2.6_3	11.5_3	2.8_3	11.5_3	3.1_3	11.5_3	3.4_3	11.5_3
Total average error	**2.2**	**5.1**	**2.2**	**6.1**	**2.4**	**6.3**	**2.4**	**6.5**	**2.5**	**7.1**

Table 3. Numerical results for middle-term prediction using 176 training data points for each month.

Month	Relative prediction errors, %									
	1 day		2 days		3 days		4 days		5 days	
	avr	max	avr	max	avr	max	avr	max	avr	max
Dec 2009	2.6_1	6.0_1	2.8_1	6.0_1	5.4_1	30.6_1	11.6_2	28.1_2	14.2_2	28.1_2
Jan 2010	2.0_1	4.5_1	2.1_1	5.7_1	2.1_1	5.8_1	2.0_1	5.9_1	1.9_1	6.0_1
Feb 2010	2.2_2	4.5_2	2.3_2	4.6_2	2.3_2	4.7_2	2.2_2	4.8_2	2.6_1	15.7_1
Mar 2010	2.1_2	4.1_2	2.1_1	5.0_1	2.2_1	5.1_1	2.3_1	5.2_1	2.5_2	13.7_2
Apr 2010	1.1_2	2.7_2	1.2_3	2.9_3	1.5_4	3.5_4	2.0_4	16.3_4	2.1_4	16.3_4
May 2010	1.3_2	2.5_2	1.4_2	3.5_2	1.4_2	3.8_2	1.4_2	3.8_2	1.5_2	4.4_2
Jun 2010	3.7_1	14.3_1	5.0_1	22.8_1	6.0_1	40.1_1	6.1_1	40.1_1	7.2_1	40.1_1
Total average error	**2.2**	**5.5**	**2.4**	**7.2**	**3.0**	**13.4**	**4.0**	**14.9**	**4.6**	**17.8**

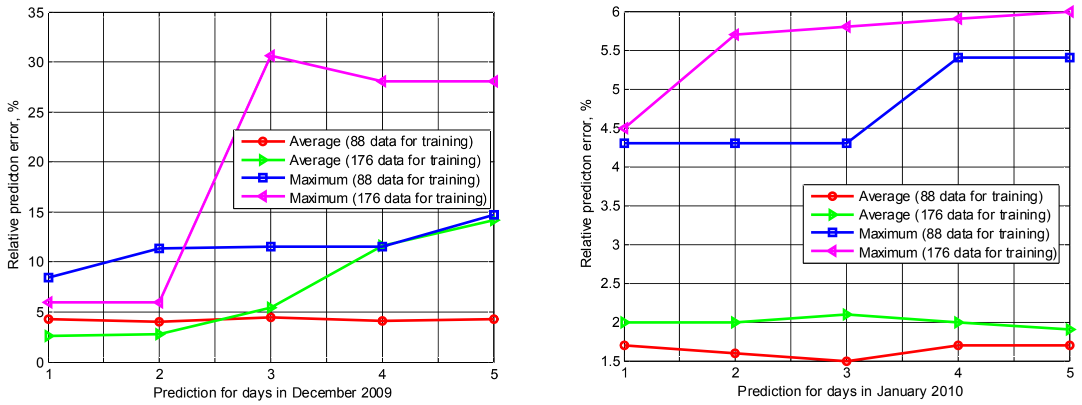

Fig. 11 – Middle-term prediction results for m1.linux for Dec. 2009 and Jan. 2010.

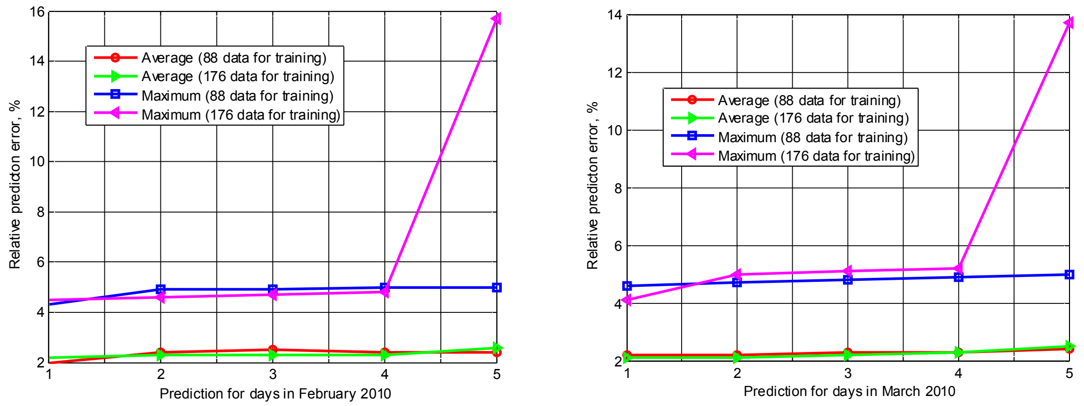

Fig. 12 – Middle-term prediction results for m1.linux for Feb. 2010 and Mar. 2010.

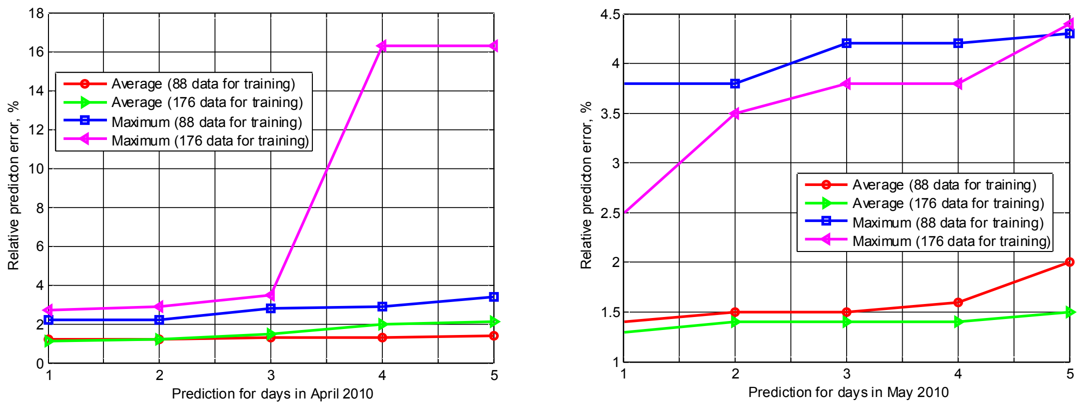

Fig. 13 – Middle-term prediction results for m1.linux for Apr. 2010 and May. 2010.

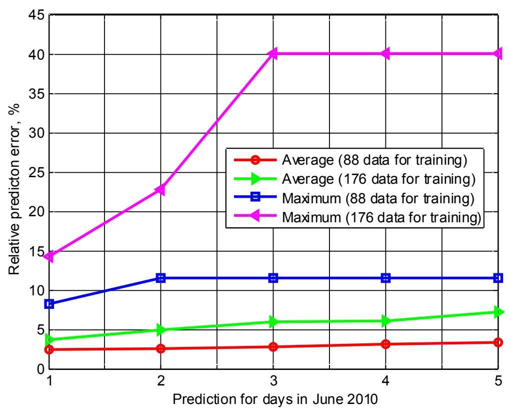

Fig. 14 – Middle-term prediction results for m1.linux for Jun. 2010.

5. CONCLUSIONS AND FUTURE WORK

Predictive models based on artificial neural networks for short-term and middle-term prediction of future spot prices for cloud computing are presented in this paper. Our models are based on standard multi-layer perceptron and recurrent neural network architectures. For prediction actions we used a moving simulation mode approach to remove old historical data for neural network re-training in order to improve a prediction accuracy of the model. The experimental results on the Amazon EC2 spot instances showed high prediction accuracy of the proposed approach. For the short-term prediction mode the average relative prediction error does not exceed 4% and the number of outliers (i.e., its relative prediction error is more than 10%) is not more than 5% for the total number of the prediction results. For the middle-term prediction mode, the average relative prediction error is in the range of 2.2 to 4.6% and the maximum relative prediction error is in the range of 5.1 to 17.8%. The obtained experimental results show that neural networks are well suited for such kind of prediction and could be very useful for users bidding on spot instance services.

Prediction of spot prices from other cloud service providers using neural networks will potentially be a future direction of our research.

REFERENCES

[1] Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia Above the clouds: A berkeley view of cloud computing, *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28*, Feb 2009. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html.

[2] B. N. Chun, P. Buonadonna, A. Auyoung, C. Ng, D. C. Parkes, J. Shneidman, A. C. Snoeren, and A. Vahdat, Mirage: A microeconomic resource allocation system for sensor net testbeds, in *Proceedings of the 2nd IEEE Workshop on Embedded Networked Sensors*, 2005.

[3] C. Weng, M. Li, X. Lu, and Q. Deng, An economic-based resource management framework in the grid context, in *Proceedings of the IEEE International Symposium on Cluster Computing and the CCGrid'2005*, Vol. 1, 2005, pp. 542–549.

[4] P. Doulai and W. Cahill, Short-term price forecasting in electric energy market, in *Proceedings of the International Power Engineering Conference*, 17-19 May 2001, Singapore, Vol. 21, No. 2, 2001, pp. 29–29.

[5] T. Lora, J. C. R. Santos, J. R. Santos, J. L. M. Ramos, and A. G. Exposito, Electricity market price forecasting: Neural networks versus weighted-distance k nearest neighbours, in *Proceedings of the 13th International Conference on Database and Expert Systems Applications, ser. DEXA'02*, Springer-Verlag, London, UK, 2002, pp. 321–330.

[6] B. Javadi, R. K. Thulasiram, and R. Buyya, Statistical modeling of spot instance prices in public cloud environments, in *Proceedings of the UCC, IEEE Computer Society*, 2011, pp. 219–228.

[7] Amazon ec2 spot price history, 2012, [Online]. http://aws.typepad.com/aws/2011/07/ ec2-spot-pricing-now-specific-to-each-availability-zone.html

[8] Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, A view of cloud computing, *Communications of ACM*, Vol. 53, No. 4, April 2010, pp. 50–58. [Online]. http://doi.acm.org/10.1145/1721654.1721672

[9] J. J. Murphy, *Technical Analysis of the Financial Markets: a Comprehensive Guide to Trading Methods and Applications*, New York, NY: New York Institute of Finance, 1999.

[10] G. Box and G. Jenkins, *Time Series Analysis: Forecasting and Control*, Hoboken, NJ: John Wiley and Sons, 1970.

[11] S. Haykin, *A Comprehensive Foundation*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 1998.

[12] E. Guresen, G. Kayakutlu, and T. U. Daim, Using artificial neural network models in stock market index prediction, *Expert Systems with Applications*, Vol. 38, No. 8, August 2011, pp. 10 389–10 397. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2011.02.068

[13] Lin, Z. Yang, and Y. Song, Short-term stock price prediction based on echo state networks, *Expert Systems with Applications*, Vol. 36, No. 3, Part 2, 2009, pp. 7313–7317, [Online]. http://www.sciencedirect.com/science/article/pii/S0957417408006519

[14] R. R. Lawrence, *Using Neural Networks to Forecast Stock Market Prices*, University of Manitoba, 1997.

[15] S. H. Kim and S. H. Chun, Graded forecasting using an array of bipolar predictions: application of probabilistic neural networks to a stock market index, *International Journal of Forecasting*, Vol. 14, No. 3, 1998, pp. 323–337. [Online]. Available: http:

//www.sciencedirect.com/science/article/pii/S016920709800003X

[16] X. Zhu, H. Wang, L. Xu, and H. Li, Predicting stock index increments b neural networks: The role of trading volume under different horizons, *Expert Systems with Applications*, Vol. 34, No. 4, 2008, pp. 3043–3054. [Online]. http://www.sciencedirect.com/science/article/pii/S0957417407002345

[17] Kara, M. Acar Boyacioglu, and O. K. Baykan, Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul stock exchange, *Expert Systems with Applications*, Vol. 38, No. 5, 2011, pp. 5311-5319. [Online]. http://dx.doi.org/10.1016/j.eswa.2010.10.027

[18] V. A. Golovko, *Neural networks: Training, models and applications*, Radiotechnika, 2011.

[19] V. Turchenko, P. Beraldi, F. De Simone, and L. Grandinetti, Short-term stock price prediction using mlp in moving simulation mode, in *Proceedings of the IEEE 6th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS)*, Vol. 2, 2011, pp. 666–671.

[20] Golovko V., Savitsky J., Laopoulos Th., Sachenko A., and Grandinetti L. Technique of Learning Rate Estimation for Efficient Training of MLP, *IEEE-INNS-ENNS International Joint Conference on Neural Networks - IJCNN'00*, Como, Italy, July 2000, pp. 323-328.

[21] Wallace R.M., Turchenko V., Sheikhalishahi M., Turchenko I., Shults V., Vazquez-Poletti J.L., Grandinetti L. Applications of Neural-based Spot Market Prediction for Cloud Computing, in *Proceedings of the 7th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems IDAACS'2013*, Berlin, Germany, 12-14 September 2013, Vol. 2, pp. 710-716.

Dr. Volodymyr Turchenko *received his Diploma in System Engineering from Brest Polytechnic Institute, Brest, Belarus (1995) and his Ph.D. degree in Computer Engineering from National University "Lviv Polytechnics", Lviv, Ukraine (2001). He is now an Associate Professor of the Information Computing System and Control Department of Ternopil National Economic University (TNEU), Ternopil, Ukraine. He is a leader of Neural Network and Parallel Computing Research Group of the Research Institute for Intelligent Computer Systems in TNEU. He is a recipient of INTAS YSF Postdoctoral Fellowship (2004-2006) and FP7 Marie Curie IIF Postdoctoral Fellowship (2009-2012) both at the University of Calabria, Italy and Fulbright Visiting Scholar grant at the University of Tennessee, USA (2012-2013). He has published more than 100 publications. His main research interests are theory and application of neural networks and parallel computing.*

Mr. Vladyslav Shults *received his M.S. degree in Computer Engineering from TNEU, Ternopil, Ukraine (2012). He is now a junior researcher of the Information Computing System and Control Department of TNEU, Ternopil, Ukraine. He is a member of Neural Network and Parallel Computing Research Group of the Research Institute for Intelligent Computer Systems in TNEU.*

Dr. Iryna Turchenko *received her M.S. degree in Management Information Systems from Ternopil Academy of National Economy, Ternopil, Ukraine (1997) and her Ph.D. degree in Computer Engineering from TNEU, Ukraine (2008), where she is now an Assistant Professor. She has more than 30 international publications. Her main research interests are: theory and application of neural networks, multiparameter sensors and intelligent and web-based instrumentations.*

Mr. Richard Wallace *received his BS from the University of Idaho, Moscow, Idaho (1980); his Masters in Computer Science from the University of Dayton, Dayton Ohio (1986), and is currently pursuing his PhD in Computer Architecture from Complutense University of Madrid, Madrid, Spain under Dr. José Luis Vázque-Poletti. He is an Advising Engineer for unmanned system ground-stations and works with multiple aerospace companies in the Dayton, Ohio area. His work is focused on real-time, secure, highly available, distributed, life-critical, information critical, and security critical systems. His prior company, Quantum Solutions, developed several control system innovations in the telecommunications industry for LDAP and distributed systems for MCI, WorldCom, and British Telecom. In his early career, he led development of VHDL (IEEE-STD-1076) and lately actively contributed to PSL (IEEE-STD-1850). He has been the author of over 30 technical reports and* *publications. His main research interests are the theory and application of large distributed event-based systems for HPC and Cloud computing with an emphasis on hybrid computing architectures.*

***Mehdi Sheikhalishahi** has 12 years of experience in Web technology, network security, and distributed computing technologies. Mehdi is currently a research fellow in Computer Science and Engineering department at University of Calabria in Italy after earning his PhD studies, and PhD thesis on energy efficient computing at the same department. In this position, he is doing research on cloud, green computing, and applications of cloud computing in scientific disciplines. Mehdi is co-author of several papers on grid, cloud computing, and energy efficiency. He was reviewer of several conferences and journals such as "Journal of Optimization Methods and Software", and "A Special Issue of IEEE Transactions on Parallel and Distributed Systems (TPDS) on Many Task Computing."*

***Dr. José Luis Vázquez-Poletti** received his M.E. in Computer Science from Universidad Pontificia de Comillas, Madrid, Spain (2004) and his Ph.D. in Computer Science from Universidad Complutense de Madrid, Madrid, Spain (2008), where he is now an Assistant Professor. He is a member of the Distributed Systems Architecture research group and has participated in 13 research projects funded by the European Commission and different Spanish ministries and agencies. He has more than 50 international publications. His main research interest is cloud computing and in particular, its application to high performance and high throughput applications.*

***Prof. Lucio Grandinetti** graduated from the University of Pisa, Italy and the University of California at Berkeley. He has been a post-doc fellow at University of Southern California, Los Angeles and Research Fellow at the University of Dundee, Scotland. Now he is Professor Emeritus at the Department of Computer Engineering, Electronics, and Systems of the University of Calabria, Italy. He was a member of the IEEE Committee on Parallel Processing, and European Editor of the book series of MIT Press on Advanced Computational Methods and Engineering. He is author of many research papers in well-established international journals and Editor (co-Editor) of several books on algorithms, software, applications in Parallel Computing, HPC, Grids and Clouds. He has been recipient and scientific leader of many European-Commission-Funded projects since 1993 (e.g. Molecular Dynamics Simulations by MPP Systems, EUROMED, HPC Finance, WADI, BEINGRID).*