Comparison of artificial neural networks using prediction benchmarking

DAVID SAMEK[1], DAVID MANAS[2]
Department of Production Engineering, Faculty of Technology
Tomas Bata University in Zlin
nam. T. G. Masaryka 5555, 76005 Zlin
Czech Republic
samek@ft.utb.cz[1], d.manas@ft.utb.cz[2] http://www.ft.utb.cz

Abstract: - Artificial neural networks are commonly used for prediction of various time series, linear and nonlinear systems. Nevertheless, the choice of proper type of artificial neural networks is difficult task, because each class of artificial neural networks has different features and abilities. Aim of this paper is to compare and benchmark four typical categories of artificial neural networks in artificial time series prediction and provide suggestions for this kind of applications.

Key-Words: - artificial neural network, prediction, time series, benchmark

1 Introduction

Artificial neural networks (ANNs) have become a standard tool for modeling and prediction of various types of processes in past few years. Their popularity comes from simple usage, scalability and broad range of software products that implement ANN algorithms. Artificial neural networks offer black-box modeling approach that does not necessarily require a priori knowledge of system dynamics. Moreover, ANNs can be easily utilized in simple signal prediction as well as in modeling of large scale multi-input multi-output systems. They are widely used in a variety of applications, such as weather forecasting [1], time series prediction of financial data [2, 3], biology and medicine [4, 5]. It is no wonder that ANNs are very extensively applied in all fields of industry, e.g. in power engineering [6] and in process control [7]. Despite the fact that in the process control area are in parallel developed progressive control methods, such as adaptive control [8] and model predictive control [9], artificial neural networks provide significant enhancement of control quality [7, 10, 18].

However, the selection of proper and usable artificial network might be difficult task. There are some works concerning prediction quality in various applications [11, 12, 13]. One of interesting way how to reveal the prediction ability is serious comparison or benchmarking. Benchmarks or contests might bring the key clues either to novices in ANN topic or experienced researchers. In this paper CATS (Competition on Artificial Time Series) benchmark [12, 13, 14] is introduced. From the large family of artificial neural networks there were chosen following types of ANN: multilayered feed-forward neural network, Elman neural network, radial basis function neural network, adaptive neural network.

2 CATS benchmark

The CATS benchmark originates from the Competition on Artificial Time Series [14, 15] organized on the IJCNN'04 conference in Budapest. Task of the predictor is to forecast five gaps in the artificial time series.

As can be seen from Figure 1, the whole time series has 5000 values with the 100 missing data. The missing data are divided into five blocks as follows: 981-1000, 1981-2000, 2981-3000, 3981-4000, 4981-5000.

The predictive error is described by two criterions: E_1 and E_2:

$$E_1 = \frac{\sum_{t=981}^{1000}(e_t - \hat{e}_t)^2}{100} + \frac{\sum_{t=1981}^{2000}(e_t - \hat{e}_t)^2}{100} + \\ + \frac{\sum_{t=2981}^{3000}(e_t - \hat{e}_t)^2}{100} + \frac{\sum_{t=3981}^{4000}(e_t - \hat{e}_t)^2}{100} + \\ + \frac{\sum_{t=4981}^{5000}(e_t - \hat{e}_t)^2}{100} \qquad (1)$$

$$E_2 = \frac{\sum_{t=981}^{1000}(e-\hat{e})^2}{80} + \frac{\sum_{t=1981}^{2000}(e-\hat{e})^2}{80} + \frac{\sum_{t=2981}^{3000}(e-\hat{e})^2}{80} + \frac{\sum_{t=3981}^{4000}(e-\hat{e})^2}{80} \quad (2)$$

Where e is the real value of the signal, $\hat{e}$ is the predicted value and t is the time step. The first criterion E_1 describes the prediction error for all 100 missing values, while the second criterion E_2 expresses the prediction error in the first four missing blocks of data (80 values). It is very important to distinguish these two criterions because some prediction methods could have problems to predict the last 20 values of the signal.

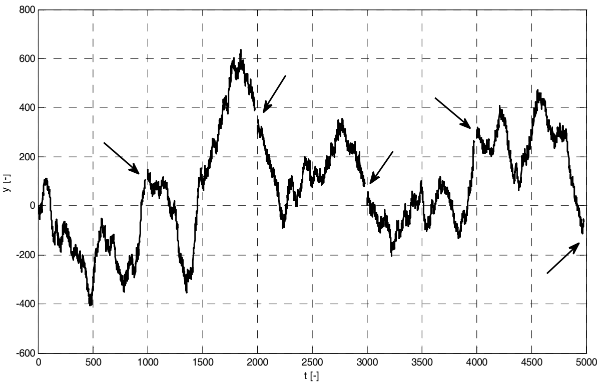

Fig. 1 – CATS time series data

3 Methodology

As was described earlier in this document, there were chosen four different types of artificial neural networks (multilayered feed-forward neural network, Elman neural network, radial basis function neural network, adaptive neural network) to cover whole ANN family.

Training of ANNs can be influenced by many parameters, such as number of layers, number of neurons, type of neurons (transfer function) and training algorithm settings. However, it can be usually found one the most influencing parameter that has key impact on the predictor quality for each single kind of ANN. In this contribution there is studied the influence of this key parameter for each benchmarked artificial neural network.

Multilayer feed-forward neural networks (MFFNNs) are very often called backpropagation networks because of the typical training algorithm. These neural networks are very often used for various type applications including modeling and prediction. As the key parameter of MFFNN was observed maximum numbers of training epochs value (MTE). In this paper two structures of multilayered feed-forward neural network are tested. Both tested structures used two layers (one hidden layer + output layer). The first structure has hyperbolic tangent sigmoid transfer function in the hidden layer and linear transfer function in the output layer. In the following text this structure will be denoted as *mffnntp*. The second configuration employs hyperbolic tangent sigmoid transfer function in the both layers (*mffnntt*).

Elman neural network (ENN) was chosen as the representative of recurrent artificial neural networks. It these ANNs data flows not only in forward direction (from inputs to outputs) but also in the backward direction. Typical Elman network has one hidden layer with delayed feedback. In this article the hidden layer contained neurons with hyperbolic tangent sigmoid transfer function and the output layer of the ENN used linear transfer function (below denoted as *enn*). The backpropagation algorithm was used for the *enn* training. Analogously to multilayered feed-forward neural networks the MTE parameter was identified as the key factor.

Artificial neural networks with radial basis function (RBF) have typically two layers. The hidden layer consists of radial basis transfer function, while the output layer uses linear transfer function. RBF networks are popular for their fast training and adaptation. However, these advantages bring some drawbacks too. The main disadvantage of RBF network is high memory requirement, because in the classic approach the number of neurons in the hidden layer is equal to the number of training data [16]. The key factor that was chosen for testing was spread parameter that defines the smoothness of the approximation function. RBF networks following this approach are further denoted as *rbf*. Nevertheless, there was developed improved design method that uses suboptimal solution of the function approximation using fewer RBF neurons in the hidden layer [17], where the training algorithm iteratively adds a RBF neuron to the hidden layer until the training error reaches the desired goal. Therefore, the goal parameter was selected as the driving factor for benchmarking. Such RBF networks will be in the following text symbolized as *rbfu*.

Adaptive linear networks have very simple structure. Nevertheless, these ANNs have a lot of applications even in the prediction of nonlinear systems. As the driving parameter was selected learning rate. The tested adaptive linear networks are in the following text denoted as *adaline*.

4 Simulations and results

For all simulations was used MATLAB with Neural Network Toolbox.

All artificial neural networks used five past values of the predicted signal as the input vector and all networks predicted only one step ahead. In other words, when it was needed the ANN repeatedly used its own predictions as inputs. Thus, in the input (zero) layer of all tested ANNs were five neurons and the output layer consisted one neuron.

Multilayered feed-forward neural networks (*mffnntp* and *mffnntt*) had thirty neurons in the hidden layer. This number was obtained by many experiments as “optimal” for this case. The structures of the MFFNN networks are illustrated in the figures 2 and 3.

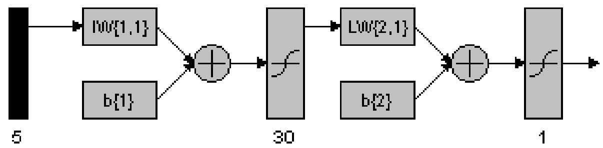

Fig. 2 – Scheme of *mffnntt*

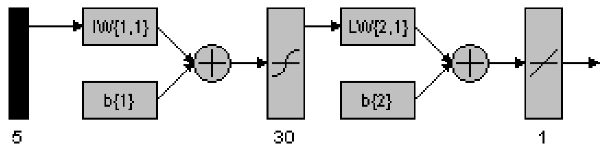

Fig. 3 – Scheme of *mffnntp*

In the case Elman neural network was used similar methodology and after lot of experiments with various structures it was found that “optimal” number of neurons in the hidden layer is ten. Simplified structure of *enn* is depicted in the Figure 4.

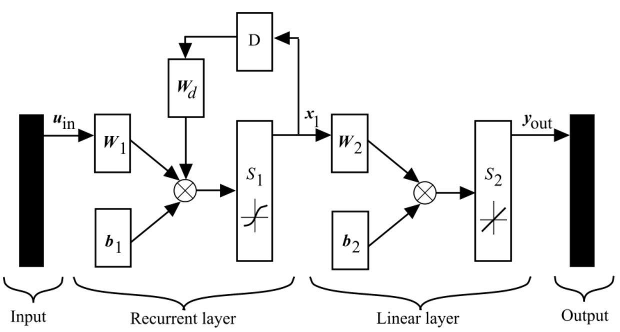

Fig. 4 – Scheme of *enn*

The structure of *rbf* comes from design method. The number of neurons in the hidden layer equals to number training data. Thus the structure of *rbf* looks like in the Figure 5. The structure of *rbfu* is similar, only the number of hidden neurons is lower.

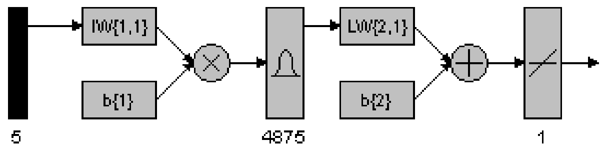

Fig. 5 – Scheme of *rbf*

The structure of *adaline* is very simple as can be seen from Figure 6.

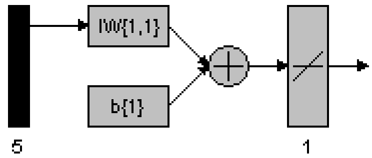

Fig. 6 – Scheme of *adaline*

The CATS prediction errors E_1, E_2, the time of prediction t_P and the time of training t_T have been observed for all types of benchmarked ANNs. Besides these general parameters, it was necessary to monitor other features for some artificial neural networks.

Table 1 – Results for *mffnntp*

MTE [1]	E_1 [E+04]	E_2 [E+04]	FGE [E-04]	Epochs [1]	t_P [s]	t_T [s]
25	31.4	31.2	31.8	25	0.59	2.46
50	5.31	5.80	12.3	50	0.59	3.99
75	4.42	4.67	9.42	75	0.59	5.93
100	1.60	1.41	7.08	100	0.59	8.11
125	1.48	1.45	6.15	125	0.59	10.2
150	1.49	1.29	5.59	150	0.59	12.3
175	1.44	1.30	5.26	173.3	0.59	14.3
200	1.58	1.43	5.36	198.9	0.59	16.5
225	1.43	1.30	4.99	220	0.59	18.2
250	5.11	5.85	5.13	201.6	0.59	16.6

In case of multilayered feed-forward neural networks (*mffnntp* and *mffnntt*) and Elman neural networks (*enn*) there were studied following parameters:

- FGE (Final Global Error) – shows Global Error of the training algorithm at the end of network training.
- Epochs – presents the real number of training epochs.

For radial basis neural networks there was observed real number of neurons in order to compare differences between *rbf* and *rbfu*.

There have been done 100 simulations for the each ANN settings. Then, the arithmetical means of

ISBN: 978-1-61804-004-6

simulation were computed and the results are presented in the tables 1 – 6.

Table 2 – Results for *mffnntt*

MTE [1]	E_1 [E+04]	E_2 [E+04]	FGE [E-04]	Epochs [1]	t_P [s]	t_T [s]
25	2.02	1.75	19.5	25	0.61	2.26
50	1.76	1.60	10.2	50	0.59	4.04
75	1.58	1.53	7.11	75	0.59	6.13
100	1.60	1.47	6.36	100	0.59	8.34
125	1.50	1.39	5.97	125	0.59	10.4
150	1.49	1.34	5.61	147.8	0.59	12.3
175	1.47	1.39	5.73	174.9	0.59	14.8
200	1.43	1.27	5.51	186.9	0.59	15.9
225	1.48	1.39	5.36	202.7	0.59	17.2
250	1.51	1.44	5.40	220.7	0.59	18.7

Table 3 – Results for *enn*

MTE [1]	E_1 [E+04]	E_2 [E+04]	FGE [E-04]	Epochs [1]	t_P [s]	t_T [s]
25	2.02	1.75	19.5	25	0.61	2.26
50	1.76	1.60	10.2	50	0.59	4.04
75	1.58	1.53	7.11	75	0.59	6.13
100	1.60	1.47	6.36	100	0.59	8.34
125	1.50	1.39	5.97	125	0.59	10.4
150	1.49	1.34	5.61	147.8	0.59	12.3
175	1.47	1.39	5.73	174.9	0.59	14.8
200	1.43	1.27	5.51	186.9	0.59	15.9
225	1.48	1.39	5.36	202.7	0.59	17.2
250	1.51	1.44	5.40	220.7	0.59	18.7

Table 4 – Results for *rbf*

spread [1]	E_1 [E+04]	E_2 [E+04]	Number of neurons	t_T [s]	t_P [s]
0.1	1.70E+8	1.71E+8	4875	82.14	0.71
0.5	1.56	1.44	4875	88.91	0.72
1	1.36	1.16	4875	84.91	0.70
5	1.36	1.21	4875	120.6	0.69
10	1.37	1.23	4875	68.00	0.69
50	1.37	1.19	4875	76.43	0.68
100	1.37	1.19	4875	70.72	0.69
500	1.36	1.20	4875	68.50	0.69
1000	1.36	1.20	4875	66.11	0.69
5000	1.36	1.20	4875	67.14	0.69

As can be seen from tables, it is difficult to find one absolute winner. From the point of view of computational requirements the *adaline* provides the best results, because the time of the prediction and time of training is definitely shortest. Conversely, the prediction quality of adaptive linear networks is under the average in this test.

Except *adaline*, all other tested ANN structures (*mffnntp*, *mffnntt*, *enn*, *rbf*, *rbfu*) performed good prediction quality. However, the lowest values of the prediction errors E_1 and E_2 were reached with improved design of radial basis network *rbfu*.

It is interesting that one of the most used type of artificial neural networks - MFFNN provided just average results as far as the prediction quality is concerned and relatively high computational demands (comparing both t_P and t_T).

Table 5 – Results for *rbfu*

spread [1]	E_1 [E+04]	E_2 [E+04]	Number of neurons	t_T [s]	t_P [s]
1.98	1.36	1.16	1902	1.43E+04	0.64
2	1.34	1.16	528	847.46	0.62
3	1.72	1.36	8	10.07	0.59
4	1.49	1.21	6	8.17	0.59
5	1.49	1.21	6	8.11	0.59
6	1.49	1.21	6	8.26	0.59
7	1.49	1.21	6	8.20	0.59
8	1.77	1.63	4	6.29	0.59
9	1.78	1.63	3	5.49	0.59
10	1.89	1.82	2	4.51	0.59

Table 6 – Results for *adaline*

learning rate [1]	E_1 [1]	E_2 [1]	t_P [s]	t_T [s]
1.00E-02	7.59E+42	8.97E+42	0.53	5.64E-02
1.00E-03	4.99E+13	5.75E+13	0.52	6.19E-03
1.00E-04	2.50E+04	2.66E+04	0.52	6.08E-03
1.00E-05	2.50E+04	2.46E+04	0.52	5.96E-03
1.00E-06	2.51E+04	2.45E+04	0.52	6.07E-03
1.00E-07	2.51E+04	2.45E+04	0.52	5.95E-03
1.00E-08	2.51E+04	2.45E+04	0.52	6.02E-03
1.00E-09	2.51E+04	2.45E+04	0.52	6.12E-03
1.00E-10	2.51E+04	2.45E+04	0.52	5.95E-03
1.00E-11	2.51E+04	2.45E+04	0.56	6.18E-03

5 Comparison and discussion

To obtain better assessment, it could be selected one best result of each tested type of ANN. Nevertheless, the selection of the best row from each table is not trivial, because for example *rbfu* has the prediction accuracy for the spread parameter=1.98, but the training time of this settings is incredibly long. Thus, the fifth row (spread=5) was selected instead. In other words, the choice of the selected representative involves both point of views – prediction accuracy (E_1 and E_2) and computational demands (t_P and t_T). Using this approach it was selected the seventh row from table 1 (*mffnntp*), the ninth row from table 2 (*mffnntt*), the fifth row from table 3 (*enn*), the eighth row from table 4 (*rbf*) and the fifth row from table 6 (*adaline*). Now these representatives could be compared in figures.

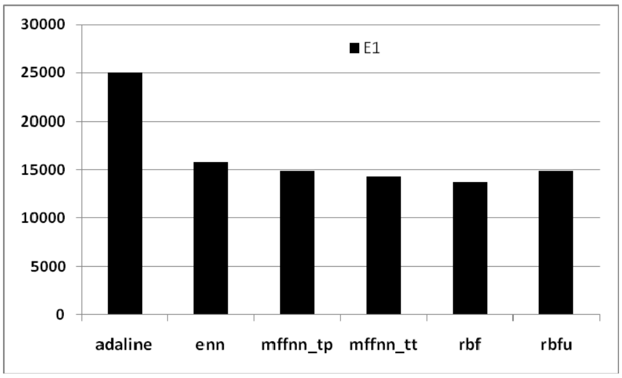

Fig. 7 – Comparison of the prediction error E_1

The Figure 7 illustrates the differences in the prediction of omitted gaps inside and outside the CATS signal. It can be assumed that the lowest value of E_1 was obtained by *rbf*. Though, the Figure 8 shows performance E_2 which describes inside prediction only. In this comparison *rbf* network wins again.

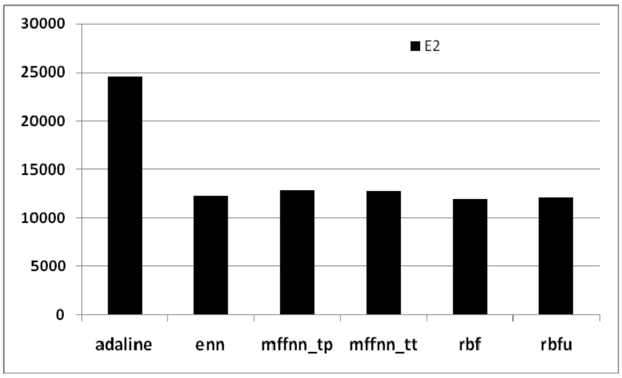

Fig. 8 – Comparison of the prediction error E_2

The figure 9 demonstrates time of prediction for each selected representative. As can be seen, the shortest time t_P can be obtained with *adaline*. The Figure 10 presents comparison of training time t_T. Here, the *adaline* gives the most impresive results. The time training of *adaline* was so short that the data in the graph had to be logarithmized.

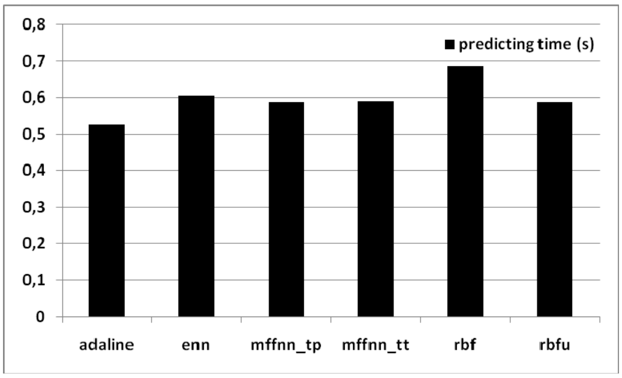

Fig. 9 – Comparison of the time of prediction t_P

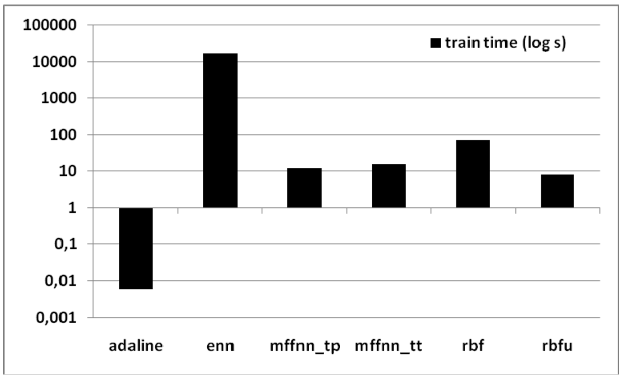

Fig. 10 – Comparison of the time of training t_T

It can be concluded that beside adaptive linear network all tested configurations have more or less comparable prediction accuracy. Predicting time was approximately same for all benchmarked artificial neural networks.

However, big differences lays in the Figure 10 (i.e. time of ANN training). Elman neural network suffers higher computational demands that probably originate from the complex structure (backward loops). Both configurations of MFFNN and radial basis network provide similar training times. Nevertheless, *adaline* showed the lowest computational demands without compare. This behavior is caused by very simple structure (one layer, linear transfer function).

Though, adaptive linear networks cannot be suggested for prediction of this kind of signals despite the fast training and prediction, because of the unsatisfactory prediction quality.

6 Conclusion

The paper presented comparison of artificial neural networks in prediction of artificial time series. The simulations proved that all tested ANNs can be used for prediction of such signals. There is only one exception – adaptive linear network. Although this network provides extremely short training and predicting times, the prediction errors were too high.

ISBN: 978-1-61804-004-6

The prediction benchmarking brings essential information about predictor abilities and its prediction accuracy. However, it has to be considered that all benchmarks (not only CATS prediction benchmark) are limited by the benchmarking method. In other words, the CATS benchmark provides information about prediction of artificial time series only. Therefore, the prediction performance for other types of signals could be different.

7 Acknowledgement

This article is financially supported by the Ministry of Education, Youth and Sports of the Czech Republic under the Research Plan No. MSM 7088352102 and by the European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089.

References:

[1] I. Maqsood, M. R. Khan, A. Abraham, An ensemble of neural networks for weather forecasting, *Neural Computing & Applications*, Vol.13, No.2, 2004, pp. 112-122.

[2] E. M. Azoff, *Neural network Time Series Forecasting of Financial Markets*, John Wiley & Sons, 1994.

[3] E. Diaconescu, The use of NARX Neural Networks to Predict Chaotic Time Series, *WSEAS Transactions on Computer Research*, Vol.3, No.3, 2008, pp. 182-191.

[4] G. Brion, Ch. Viswanathan, T. R. Neelakantan, S. Ligireddy, R. Girones, D. Lees, A. Allard, A. Vantarakis, Artificial Neural Network Prediction of Viruses in Shellfish, *Applied and Enviromental Microbiology*, Vol.71, No.9, 2005, pp. 5244-5253.

[5] E. Pasomsub, C. Sukasem, S. Sungkanuparph, B. Kijsirikul, W. Chantratita, The application of artificial neural networks for phenotypic drug resistance prediction: evaluation and comparison with other interpretation systems, *Japanese journal of infectious diseases*, Vol.63, No.2, 2010, pp. 87-94.

[6] A. D. Papalexopoulos, H. Shangyou, P. Tie-Mao, An implementation of a neural network based load forecasting model for the EMS, *IEEE Transactions on Power Systems*, Vol.9, No.4, 1994, pp. 1956-1962.

[7] D. Samek, P. Dostal, Artificial Neural Networks in Prediction and Predictive Control. In *Proceedings of the 22nd European Conference on Modelling and Simulation ECMS 2008*. European Council for Modelling and Simulation, 2008. pp. 525-530.

[8] F. Gazdos, P. Dostal, Adaptive control of a coupled drives apparatus using dual Youla-Kucera parametrization. In *Proceedings of 16th IFAC World Congress*. IFAC, 2005.

[9] P. Chalupa, V. Bobal, Modelling and Predictive Control of Inverted Pendulum, In *22nd European Conference on Modelling and Simulation*, European Council for Modelling and Simulation, 2008, pp. 531-537.

[10] E. F. Camacho, C. Bordons, *Model Predictive Control in the Process Industry*. Springer - Verlag, 2004.

[11] G. Zhang, B. E. Patuwo, M. Y. HuX, Forecasting with artificial neural networks: The state of the art, *International Journal of Forecasting*, Vol.14, No.1, 1998, pp. 35-62.

[12] D. Samek, Prediction benchmark of artificial neural networks. In *Annals of DAAAM for 2009 & Proceedings of the 20th International DAAAM Symposium "Inteligent Manufacturing & Automation: Focus on Theory, Practice and Education"*. DAAAM International Vienna, 2009, pp. 621-622.

[13] D. Samek, Artificial neural networks with radial basis function in prediction benchmark. In *Annals of DAAAM for 2010 and Proceedings of 21st International DAAAM Symposium: Intelligent Manufacturing & Automation: Focus on Interdisciplinary Solutions*. DAAAM International Vienna, 2010, pp. 581-582.

[14] A. Lendasse, E. Oja, O. Simula, Time Series Prediction Competition: The CATS Benchmark. In *Proc. of IEEE Int. Joint Conf. on Neural Networks*, IEEE, 2004, pp. 1615-1620.

[15] A. Lendasse, E. Oja, O. Simula, M. Verleysen, Time Series Prediction Competition: The CATS Benchmark, *Neurocomputing*, Vol.70, 2007, pp. 2325-2329.

[16] B. Yegnanarayana, *Artificial Neural Networks*, Prentice-Hall of India, 1999.

[17] M. H. Beale, M. T. Hagan, H. B. Demuth, *Neural Network Toolbox 7*, Mathworks, 2010.

[18] L. Macku, D. Samek, Two step, PID and model predictive control using artificial neural network applied on semi-batch reactor. *WSEAS TRANSACTIONS on SYSTEMS*. Vol.9, No.10, 2010, pp. 1039-1049.