Artificial neural networks in artificial time series prediction benchmark

David Samek, David Manas

***Abstract*—**The work is aimed to research of predicting abilities of artificial neural networks. The characteristic samples of artificial neural network types were selected to be compared in numerous simulations, while influences of key parameters are studied. The tested artificial networks are as follows: multilayered feed-forward neural network, recurrent Elman neural network, adaptive linear network and radial basis function neural network.

***Keywords*—**Artificial neural network, benchmark, prediction, time series.

I. 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], [22]. Despite the fact that in the process control area are in parallel developed progressive control methods, such as adaptive control [8]-[10] and model predictive control [11],[21], artificial neural networks provide significant enhancement of control quality [7], [12], [19].

Despite the minor skeptic opinions [28], artificial neural networks are successfully utilized in prediction applications. For example an extensive survey of the forecasting with artificial neural networks can be found in [13]. However, the selection of proper and usable artificial network might be difficult task. There are some works concerning prediction quality in various applications [13]-[15]. One of interesting ways 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, because they can compare own predictor results to competitive methods using given objective criterions. There have been published a few of such comparison methods. For example The EUNITE network (EUropean Network on Intelligent TEchnologies for Smart Adaptive Systems) organized two competitions during 2001 and 2002. The first one was focused on the forecasting of maximum daily electrical load based on electrical load values and additional data [23]. The second EUNITE competition's target was to model the Customer Intelligence in the Bank [24]. Short survey of benchmarking methods and prediction contests is introduced in [17]. Authors mention the so called Santa Fe competition, which is described in [25], another mentioned competition, which was presented in The International Workshop on Advanced Black-Box: Techniques for Nonlinear Modeling, is published in [26]. The 2010 Time Series Forecasting Grand Competition for Computational Intelligence [29] is aimed to empirical time series prediction. Same author presents good but a little out-of-date survey to neural network forecasting competitions in [30]. Interesting prediction benchmark was used by the ASHRAE (American Society of Heating, Refrigerating, and Air-Conditioning Engineers). In the "The Great Energy Predictor Shootout"-competition [32], [33] four environmental parameters (ambient temperature, absolute humidity ratio, solar radiation, and wind speed) were predicted. This competition was followed by the second benchmark "Great energy predictor shootout II" two years later [34]. In this paper the CATS (Competition on Artificial Time Series) benchmark [14]-[16] is chosen, because it is widely used as "first choice" benchmark and all data including the testing data were available.

Lot of types of ANNs can be used for prediction. The most versatile type is multilayered feed-forward neural network (MFFNN). Almost all variations of the MFFNN, even the simple adaptive linear network (ADALINE), are capable to model and predict various systems. When the feedback connections are added to the ANNs, the recurrent neural networks are created. These networks can model temporally/sequentially extended dependencies over

Manuscript received June 27, 2011: Revised version received June 30, 2011. This work 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.

D. Samek is with the Department of Production Engineering, Faculty of technology, Tomas Bata University in Zlin, nam. T. G. Masaryka 5555, 760 01 Zlin Czech Republic (phone: +420-576-035-157; fax: +420-576-035-176; e-mail: samek@ft.utb.cz).

D. Manas is with the Department of Production Engineering, Faculty of technology, Tomas Bata University in Zlin, nam. T. G. Masaryka 5555, 760 01 Zlin Czech Republic (e-mail: dmanas@ft.utb.cz).

unspecified (and potentially infinite) intervals [31]. There are other special categories of artificial neural networks that are used for modeling/prediction; e.g. radial basis function neural networks [35], functional networks [36], Kohonen networks [37], [38], probabilistic fuzzy neural network [39], etc.

In this paper, there were chosen following types of ANN to be tested: multilayered feed-forward neural network, because of its wide usage, Elman neural network as the representative of the recurrent neural networks, radial basis function neural network, because it provides simple training with good prediction performance and adaptive neural network due to its simplicity. The paper is organized as follows: in the next chapter CATS benchmark is explained, then methodology of simulations is described, furthermore the structures of the tested ANNs is introduced, following part of the article shows results of simulations, their description and discussion, and finally the paper is closed by short concluding remarks.

II. CATS Benchmark

The CATS benchmark originates from the Competition on Artificial Time Series [16], [17] organized on the IJCNN'04 conference in Budapest. Task of the predictor is to forecast five gaps in the artificial time series.

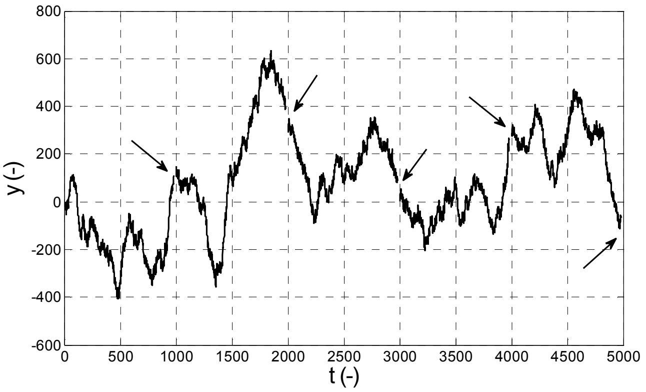

Fig. 1 CATS time series data

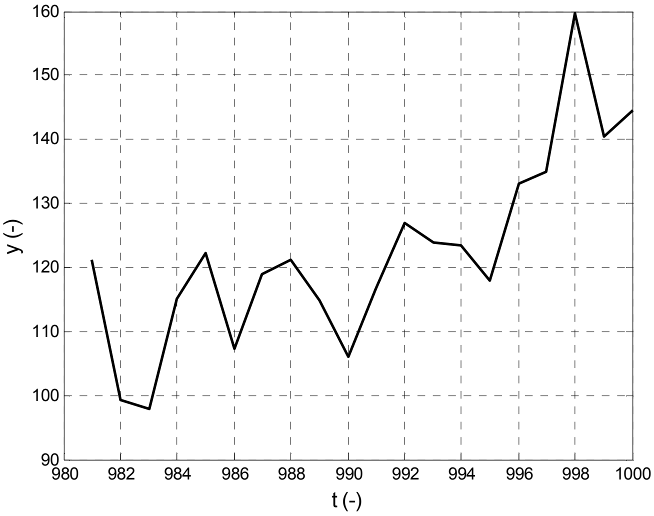

Fig. 2 Missing data to predicted, gap 1 (980-1000)

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 missing gaps in the signal are marked in the Fig. 1 by the black arrows. The goal data from missing gaps are presented in the Fig. 2-6.

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

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

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

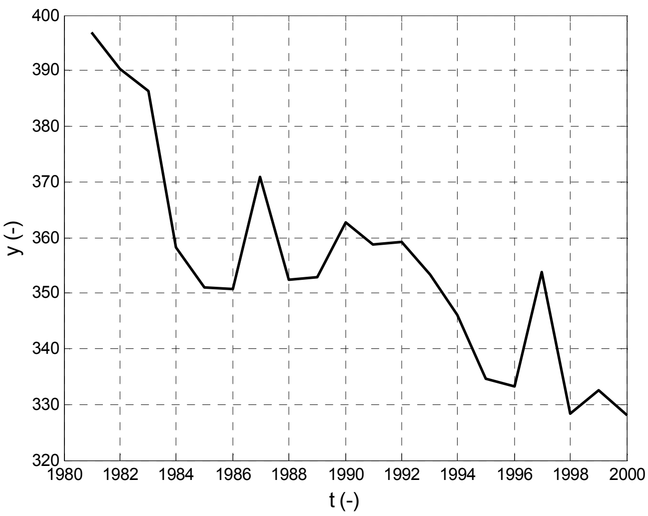

Fig. 3 Missing data to predicted, gap 2 (1981-2000)

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.

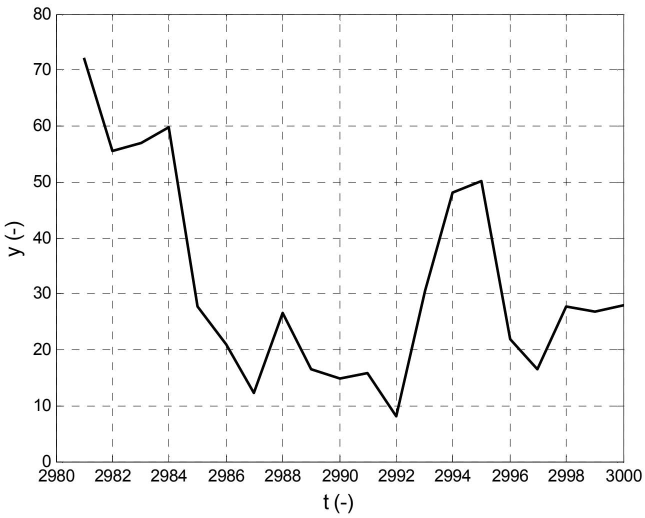

Fig. 4 Missing data to predicted, gap 3 (2981-3000)

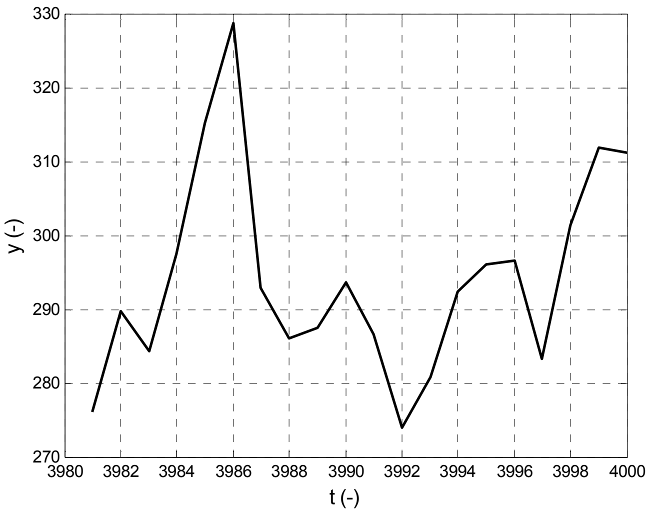

Fig. 5 Missing data to predicted, gap 4 (3981-4000)

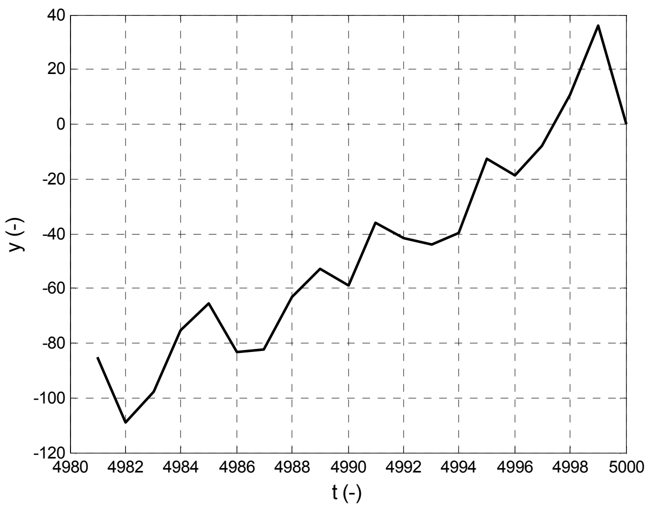

Fig. 6 Missing data to predicted, gap 5 (4981-5000)

III. 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 and easy 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 [18]. 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 [19], 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*.

In order to obtain comparable results we tried to keep same conditions for all tested networks as much as it was possible. For example all tested neural networks used five last values of the signal for one future value prediction, as is depicted in the Fig. 7. Furthermore, the same number of layers and same number of neurons in the layers was used where it was possible. Of course each ANN has specific features and limits. Thus, for example in case of one-layered *adaline* it was not possible to use one hidden layer as in the MFFNNs.

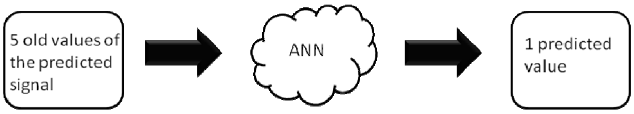

Fig. 7 One-step-ahead prediction from the five last values

IV. Simulations and Results

For all simulations MATLAB with Neural Network Toolbox was used.

As was mentioned hereinbefore, all artificial neural networks used five past values of the predicted signal since 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. Therefore, five neurons were in the input (zero) layer of all tested ANNs and the output layer consisted of 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 Fig. 8 and 9.

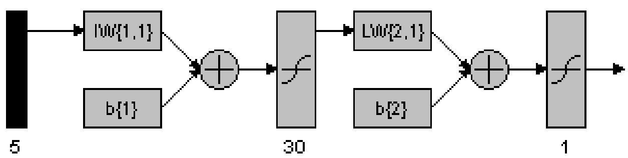

Fig. 8 Scheme of *mffnntt*

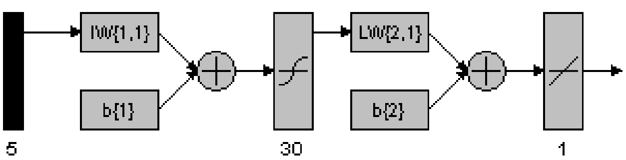

Fig. 9 Scheme of *mffnntp*

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 Fig. 10.

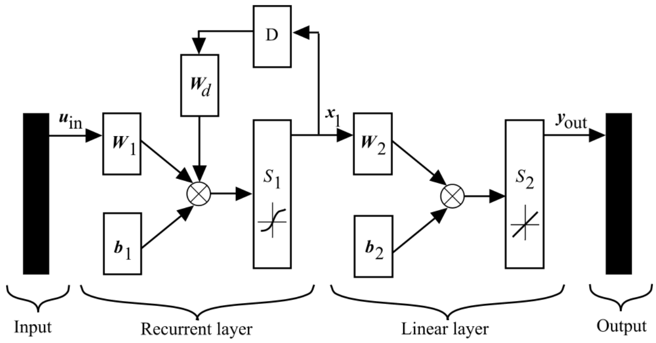

Fig. 10 Scheme of *enn*

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 Fig. 11. The structure of *rbfu* is similar, only the number of neurons in the hidden layer is lower.

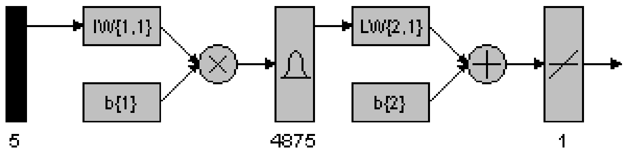

Fig. 11 Scheme of *rbf*

The structure of *adaline* is very simple as can be seen from Fig. 12.

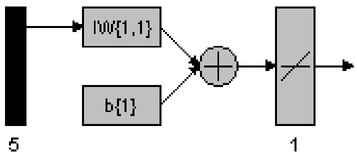

Fig. 12 Scheme of *adaline*

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 that were specific for each tested artificial neural network.

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.

Table I. 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

Table II. 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 III. Results for *enn*

MTE (1)	E_1 (E+04)	E_2 (E+04)	FGE (E-04)	Epochs (1)	t_P (s)	t_T (E+04 s)
150	2,36	1,60	12,2	150	0,62	0,86
200	17,8	16,4	10,5	200	0,60	1,16
250	2,43	1,91	11,3	248,1	0,60	1,42
300	1,58	1,23	9,56	298,3	0,60	1,70
350	1,68	1,27	11,5	318,1	0,60	1,84
400	1,92	1,40	9,05	332,5	0,60	1,91
450	2,24	1,54	10,0	347,8	0,64	2,02
500	2,24	1,54	9,91	372,8	0,62	2,15
550	17,7	16,3	8,56	421,8	0,61	2,45
600	1,66	1,23	8,20	517,5	0,61	2,98

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 simulation were computed and the results are presented in the Tables I – VI.

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.

Table IV. Results for *rbf*

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

Table V. Results for *rbfu*

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

Table VI. 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

It is interesting that one of the most used types 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).

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*. The absolutely best (lowest) prediction errors were obtained for the goal = 2 (the second row in the Table V).

Relatively misleading could be finding the worst prediction errors, because inaccurate predictions can be easily achieved with all artificial neural networks by inferior setting only. While the influence of the chosen key parameter was studied, some results, especially in the limits of the studied parameter range, can be strongly imprecise.

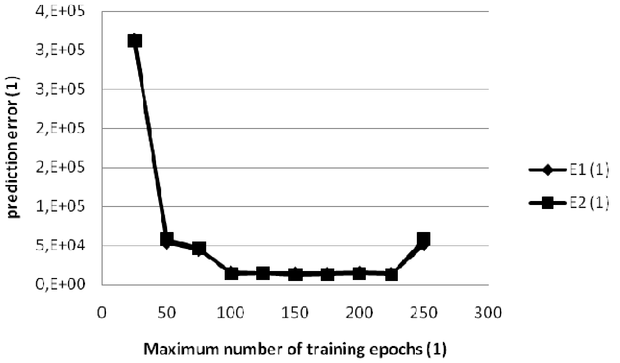

Fig. 13 Influence of the MTE to $E1$ and $E2$ for *mffnntp*

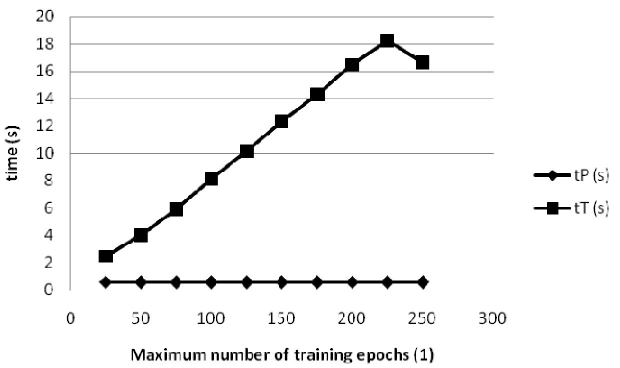

Fig. 14 Influence of the MTE to t_P and t_T for *mffnntp*

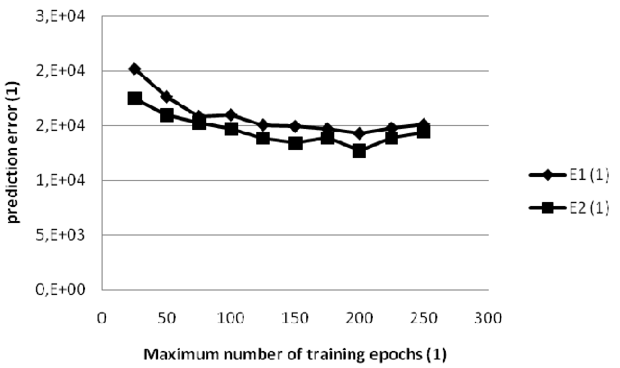

Fig. 15 Influence of the MTE to $E1$ and $E2$ for *mffnntt*

As far as the key parameter is concerned, the maximum number of training epochs influences the prediction error for multilayered feed-forward neural network as can be seen from Fig. 13 and 15. First, while the MTE rises, the E_1 and E_2 fall down. Then, at a certain level (approx. MTE = 100) prediction errors starts stagnate. And finally, when maximum number of training epoch reaches approximately 225, the prediction errors go up. Furthermore, it can be deduced that prediction time t_P is not significantly influenced by this parameter, but time for .training t_T is directly proportional to the MTE, as is depicted in the Fig. 14 and 16.

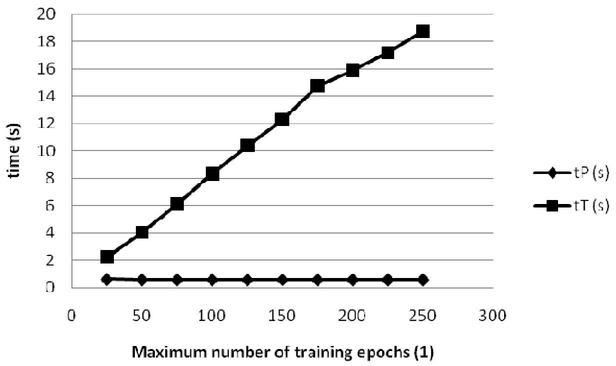

Fig. 16 Influence of the MTE to t_P and t_T for *mffnntt*

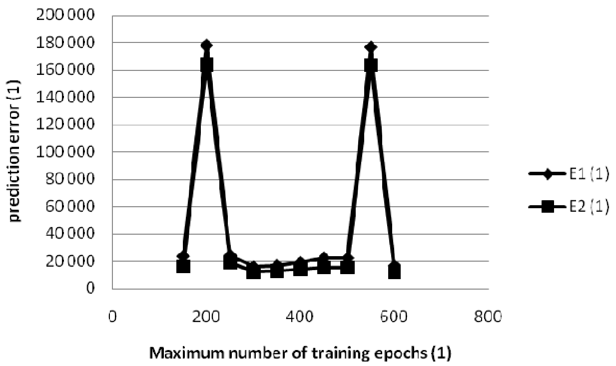

Fig. 17 Influence of the MTE to $E1$ and $E2$ for *enn*

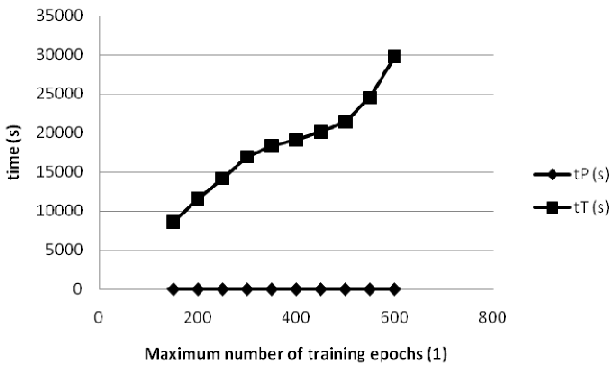

Fig. 18 Influence of the MTE to t_P and t_T for *enn*

Very similar behavior was achieved for Elman neural network, where the maximum number of training epochs was observed too. The training time t_T is directly proportional to the maximum number of training epochs, while the prediction time t_P remains almost the same, as is illustrated in the Fig. 18. On the other hand, the prediction errors E_1 and E_2 show interesting dependency on the MTE. As can be seen from the Fig. 17, there are two peaks at the limits of the observed range.

Between limits is flat valley with almost constant values of the prediction errors.

From the Table IV and the Fig. 19 it can be concluded that when spread parameter reaches value 1 the prediction errors become steady. Additionally, it can be seen from Fig. 20 that time t_P is not notably influenced by the spread parameter and the time of training is mostly decreasing.

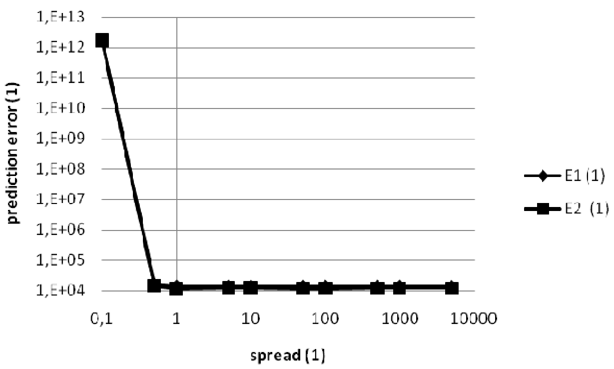

Fig. 19 Influence of the spread to $E1$ and $E2$ for *rbf*

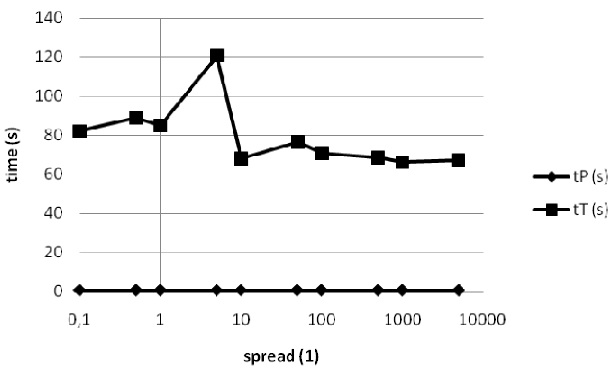

Fig. 20 Influence of the spread to t_P and t_T for *rbf*

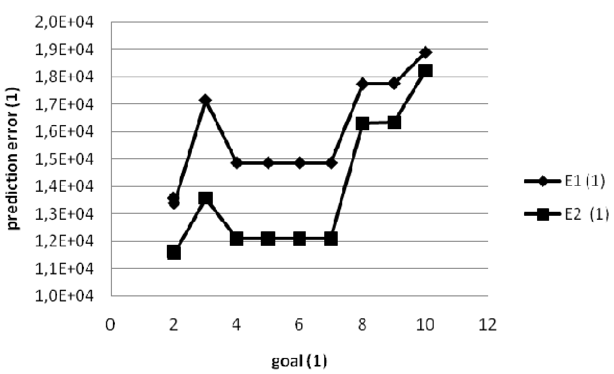

Fig. 21 Influence of the goal to $E1$ and $E2$ for *rbfu*

Fig. 21 shows again ambiguous course of the prediction errors E_1 and E_2 similarly to the Fig. 17. Generally it can be assumed that the best prediction accuracy for *rbfu* is obtained in the range goal=(4, 7). The fact that the time of prediction t_P reaches minimum for the goal=3 results from the Fig. 22. In addition it can be concluded that time of the training t_T decreases with the increase of the goal parameter.

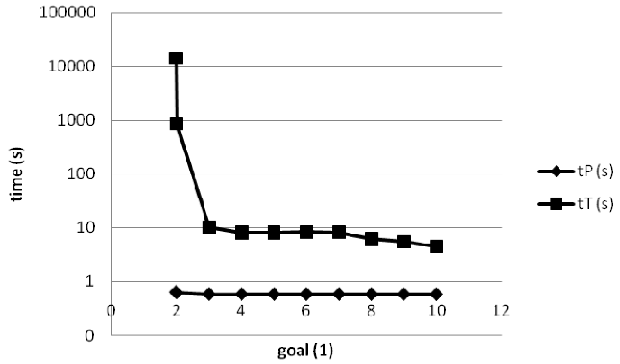

Fig. 22 Influence of the goal to t_P and t_T for *rbfu*

Fig. 23 proves that prediction errors E_1 and E_2 are the highest from the tested ANN. Even the increase of the learning rate cannot improve this result – the prediction accuracy remain at the same level after reaching saturation around $2{,}50 \cdot 10^4$. In other words, adaptive linear network is not able to train this kind of signal effectively. Same conclusion can done with computational times. As can be seen from Fig. 24, the learning rate does not notably change the time of prediction t_P and the time of training t_T.

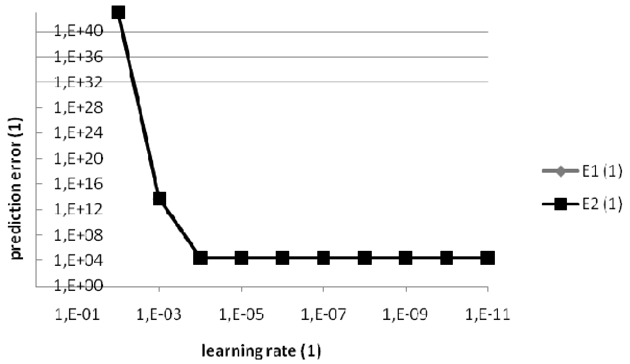

Fig. 23 Influence of the learning rate to $E1$ and $E2$ for *adaline*

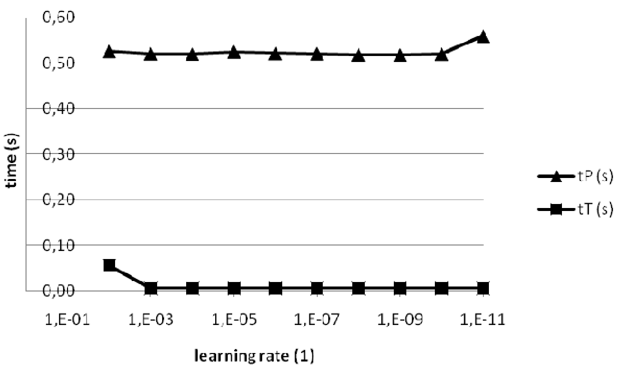

Fig. 24 Influence of the learning rate to t_P and t_T for *adaline*

V. 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 (goal=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 (time t_P and t_T).

Using this approach it was selected the seventh row from Table I (*mffnntp*), the ninth row from Table II (*mffnntt*), the fifth row from Table III (*enn*), the eighth row from Table IV (*rbf*) and the fifth row from Table VI (*adaline*). Now these representatives could be compared in bar charts.

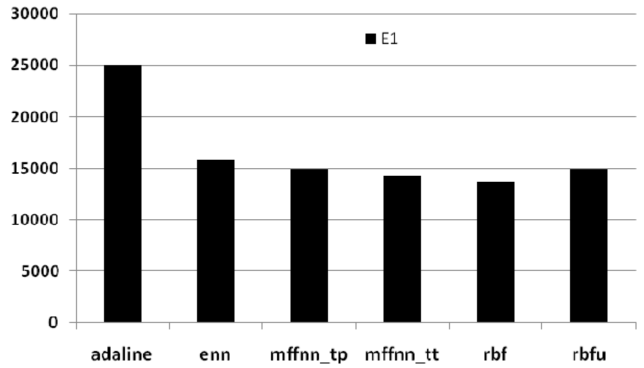

Fig. 25 Comparison of the prediction error E_1

The Fig. 25 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 Fig. 26 shows performance E_2 which describes internal prediction only. In this comparison *rbf* network wins again.

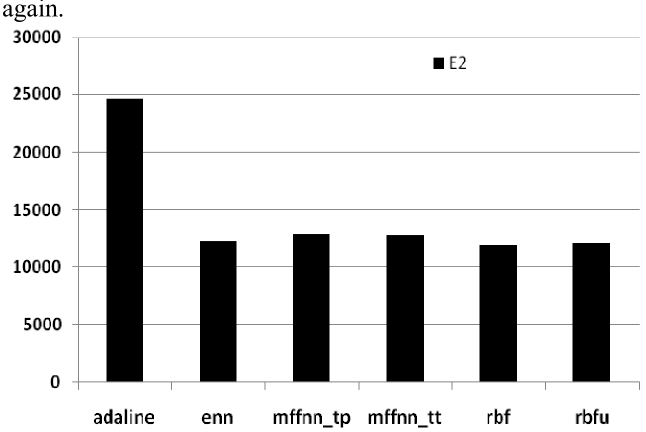

Fig. 26 Comparison of the prediction error E_2

The Fig. 27 demonstrates time of prediction for each selected representative. As can be seen, the shortest time t_P can be obtained with *adaline*. The Fig. 28 presents comparison of training time t_T. Here, the *adaline* gives the most impresive results. The training time of *adaline* was so short that the data in the graph had to be logarithmized.

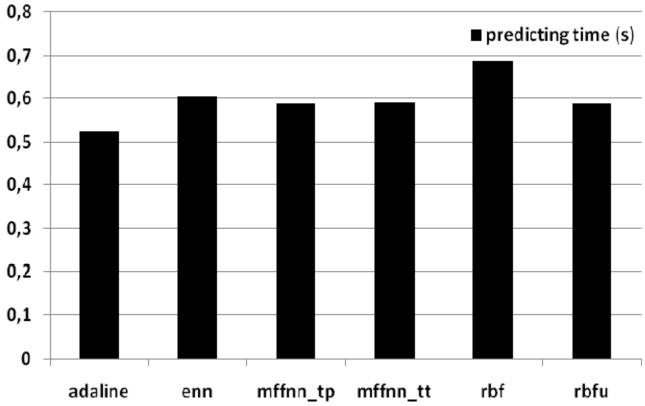

Fig. 27 Comparison of the time of prediction t_P

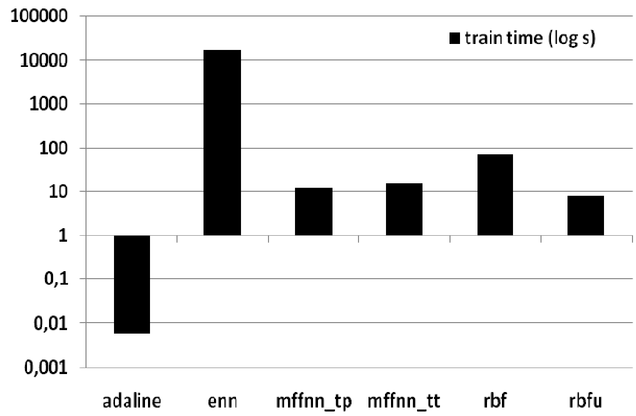

Fig. 28 Comparison of the time of training t_T

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 Fig. 28 (i.e. time of ANN training). Elman neural network suffers higher computational demands that probably originate from the more 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.

VI. Conslusion

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.

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.

REFERENCES

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

[2] E. M. Azoff, *Neural network Time Series Forecasting of Financial Markets*, New York: 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, pp. 182-191, Mar. 2008.

[4] G. Brion, et al., "Artificial neural network prediction of viruses in shellfish," *Applied and Enviromental Microbiology*, vol.71, no.9, pp. 5244-5253, Sep. 2005.

[5] E. Pasomsub, C. Sukasem, S. Sungkanuparph, B. Kijsirikul and 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, pp. 87-94, Mar. 2010.

[6] A. D. Papalexopoulos, H. Shangyou and 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, pp. 1956-1962, Nov. 1994.

[7] D. Samek and P. Dostal, "Artificial neural networks in prediction and predictive control," in *Proceedings of the 22nd European Conference on Modelling and Simulation ECMS 2008*, Nicosia, 2008, pp. 525-530.

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

[9] J. Garus, "Adaptive track-keeping control of underwater robotic vehicle," *International Journal of Mathematical Models and Methods in Applied Sciences*. vol.1, no.4, pp. 217-222, 2010.

[10] P. Dostalek, J. Dolinay, V. Vasek and L. Pekar, "Self-tuning digital PID controller implemented on 8-bit freescale microcontroller," *International Journal of Mathematical Models and Methods in Applied Sciences*. vol. 4, no. 4, 2010

[11] P. Chalupa and 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.

[12] E. F. Camacho and C. Bordons, *Model Predictive Control in the Process Industry*, London: Springer - Verlag, 2004, ch. 9.

[13] G. Zhang, B. E. Patuwo and M. Y. HuX, "Forecasting with artificial neural networks: The state of the art," *International Journal of Forecasting*, vol.14, no.1, pp. 35-62, July 1998.

[14] 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"*, Vienna, 2009, pp. 621-622.

[15] 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*, Zadar, 2010, pp. 581-582.

[16] A. Lendasse, E. Oja and O. Simula, "Time series prediction competition: The CATS benchmark," in *Proc. of IEEE Int. Joint Conf. on Neural Networks*, Budapest, 2004, pp. 1615-1620.

[17] A. Lendasse, E. Oja, O. Simula and M. Verleysen, "Time series prediction competition: The CATS benchmark", *Neurocomputing*, vol.70, pp. 2325-2329, 2007.

[18] B. Yegnanarayana, *Artificial Neural Networks*, New Delhi: Prentice-Hall of India, 1999, ch. 7.

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

[20] L. Macku and 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, pp. 1039-1049, Oct. 2010.

[21] A. Bemporad and D. M. de la Pena, "Multiobjective model predictive control, "*Automatica*. vol. 45, no. 12, pp. 2823-2830, Dec 2009.

[22] P. Pivonka and V. Veleba, "Adaptive controllers by using neural network based identification for short sampling period," *International Journal of Circuits, Systems and Signal Processing*. vol. 1, no. 1, 2007.

[23] Electricity Load Forecast using Intelligent Adaptive Technology [Online], EUNITE, 2001. Available: http://neuron.tuke.sk/competition/index.php

[24] Modeling the Bank's Client behavior using Intelligent Technologies [Online], EUNITE, 2001. Available: http://neuron.tuke.sk/competition2/

[25] A. Weigend and N. Gershenfeld, *Times Series Prediction: Forecasting the Future and Understanding the Past*, Reading, MA: Addison-Wesley, 1994.

[26] J. Suykens and J. Vandewalle, "The K.U. Leuven time-series prediction competition," in *International Workshop on Advanced Black-Box Techniques for Nonlinear Modeling - Theory and Applications*, Louvain, 1998, pp. 241–253.

[27] J. Suykens and J. Vandewalle, "The K.U. Leuven time-series prediction competition," in *International Workshop on Advanced Black-Box Techniques for Nonlinear Modeling - Theory and Applications*, Louvain, 1998, pp. 241–253.

[28] E. de Bodt, J. Rynkiewicz and M. Cottrell, "Some known facts about financial data," in *European Symposium on Artificial Neural Networks*, Bruges, 2001, pp.223-236.

[29] S. F. Crone, Time Series Forecasting Competition for Neural Networks and Computational Intelligence [Online], 2010. Available: http://www.neural-forecasting-competition.com/

[30] S. F. Crone, Portal on forecasting with artificial neural networks [Online], 2005. Available: http://www.neural-forecasting.com/

[31] M. Boden. (2001, November, 13). A guide to recurrent neural networks and backpropagation [Online], *The Dallas project, SICS technical report*. Available: http://itee.uq.edu.au/~mikael/papers/rn_dallas.pdf

[32] J. F. Kreider and J. S. Haberl, "Predicting hourly building energy usage," *ASHRAE Journal*, vol. 36, no. 6, pp. 72-81, June 1994.

[33] J. F. Kreider and J. S. Haberl, "Predicting hourly building energy use: The great energy predictor shootout – Overview and discussion of results," *ASHRAE Transactions*", vol. 100, no. 2, pp. 1104–1118, June 1994.

[34] J. Haberl and S. Thamilseran, "Predicting hourly building energy use: The great energy predictor shootout II: Measuring retrofit savings," *ASHRAE Journal*, vol. 40, no. 1, pp. 49 – 56, January 1998.

[35] R. Zemouri, D. Racoceanu and N. Zerhouni, "Recurrent radial basis function network for time-series prediction," *Engineering Applications of Artificial Intelligence*, vol. 16, no. 5-6, pp. 453-463, August-September 2003.

[36] E. Castillo, B. Guijarro and A. Alonso, Electricity load forecast using functional networks [Online], *Report for EUNITE 2001 Competition*, 2001. Available: http://neuron.tuke.sk/competition/reports/BerthaGuijarro.pdf

[37] F. Ortega, et al. An Hybrid Approach to Prediction of Electric Load with MARS and Kohonen Maps [Online], *Report for EUNITE 2001 Competition*, 2001. Available: http://neuron.tuke.sk/competition/reports/FranciscoOrtega.pdf

[38] F. J. Marin, F. Garcia-Lagos, G. Joya and F. Sandoval, Peak Load Forecasting using Kohonen Classification and Intervention Analysis [Online], *Report for EUNITE 2001 Competition*, 2001. Available: http://neuron.tuke.sk/competition/reports/JavierMarin.pdf

[39] S. Hengjiea, Ch. Miaoa, Z. Shena, W. Roelb, M. D'Hondtb and C. Francky, "A probabilistic fuzzy approach to modeling nonlinear systems," *Neurocomputing*, vol. 74, no. 6, pp. 1008-1025, February 2011