BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 61, No. 3, 2013
DOI: 10.2478/bpasts-2013-0060

Prediction of corrections for the Polish time scale UTC(PL) using artificial neural networks

M. LUZAR[1*], Ł. SOBOLEWSKI[2], W. MICZULSKI[2], and J. KORBICZ[1]

[1] Institute of Control and Computation Engineering, University of Zielona Góra, 50 Podgórna St., 65-246 Zielona Góra, Poland
[2] Institute of Electrical Metrology, Faculty of Electrical Engineering, Computer Science and Telecommunications, University of Zielona Góra, 50 Podgórna St., 65-246 Zielona Góra, Poland

Abstract. In this paper, the effectiveness of using Artificial Neural Networks (ANNs) for predicting the corrections of the Polish time scale UTC(PL) (Universal Coordinated Time) is presented. In particular, prediction results for the different types of neural networks, i.e., the MLP (MultiLayer Perceprton), the RBF (Radial Basis Function) and the GMDH (Group Method of Data Handling) are shown. The main advantages and disadvantages of using such types of neural networks are discussed. The prediction of corrections is performed using two methods: the time series analysis method and the regression method. The input data were prepared suitable for the above mentioned methods, based on two time series, *ts1* and *ts2*. The designation of prediction errors for specified days and the influence of data quantity for the prediction error are considered. The paper consists of five sections. After Introduction, in Sec. 2, the theoretical background for different types of neural networks is presented. Section 3 shows data preparation for the appropriate type of neural network. The experimental results are presented in Sec. 4. Finally, Sec. 5 concludes the paper.

Key words: neural network, prediction methods, national timescale, atomic clock.

1. Introduction

With the increasing amount of accurate devices based on precise time, a coordination method for the universal time clock is needed. The Polish time scale UTC(PL) is a local realization of Universal Coordinated Time (UTC). It is the basis for the dissemination of standard frequency and time signals in Poland. UTC(PL) is implemented by the Central Office of Measures (Główny Urząd Miar, GUM) with a type 5071A caesium-beam atomic clock (called Cs2) and a control device (Microstepper Austron 2055). Depending on the control device, it is possible to make corrections, to be sure that the difference between UTC and UTC(PL) is minimal (Fig. 1, Station 1).

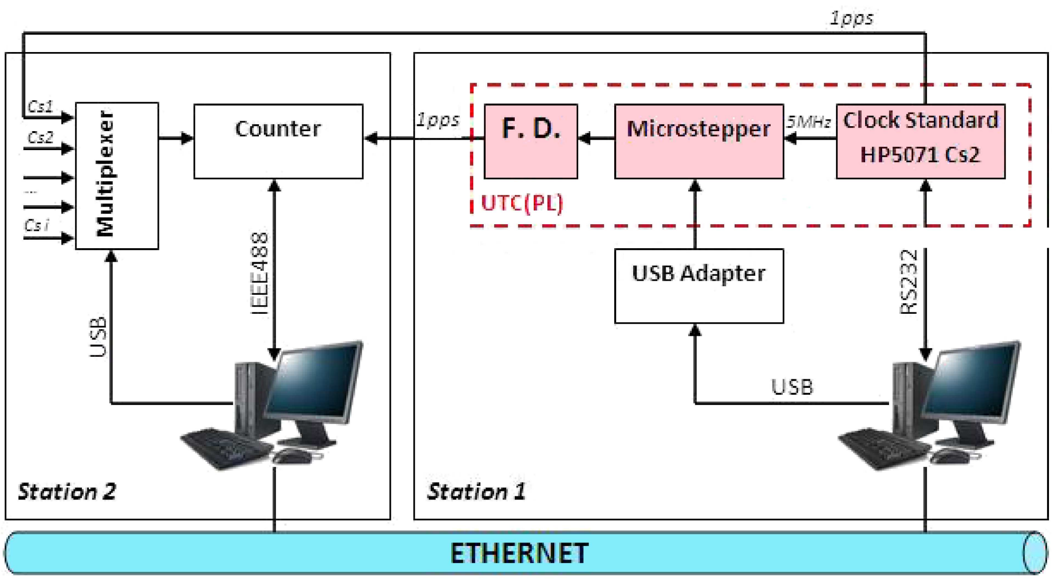

Fig. 1. Realization of the UTC(PL) scale and its comparison to the 1pps signal generated by the atomic clock

*e-mail: M.Luzar@issi.uz.zgora.pl

The quality of national timescales UTC(k), including UTC(PL), is determined by BIPM (Bureau International des Poids et Mesures). Each month, for individual UTC(k), BIPM designates the corrections defining the divergence of timescales in relation to UTC. For individual UTC(k), BIPM designates the corrections which define the difference between national timescales and UTC. The values of these corrections are published by BIPM in the *Circular T* bulletin, more or less on the 10th day of the subsequent month. UTC(k) for countries whose corrections do not exceed the value of ± 10 ns are the best group of timescales. The other timescales groups are defined with correction values of ± 20 ns and ± 50 ns. The main task of GUM is to keep the UTC(PL) level, in relation to UTC, below ± 10 ns. When there is a significant delay in the publication of *Circular T*, between 8 and 12 days, the problem of keeping the best compatibility between UTC(PL) and UTC might be solved by predicting the corrections. The prediction method for UTC(k) is used by several national metrology laboratories [1].

GUM uses a laborious procedure of predicting the corrections based on the analytical linear regression method [2]. It requires the results of the previous work of the clocks in the laboratory. Moreover, during the prediction process, it is necessary to arbitrarily choose which correction are the most reliable and which are useless in the particular month out of all the obtained ones. Other prediction methods are based on linear regression with stochastic differential equations [3], Allan deviations [4] or the Kalman filter [5].

An increase in the use of artificial neural networks in different fields has recently been observed, particularly for prediction tasks, i.e., fault prediction [6] or systems response prediction [7]. Moreover, special types of neural networks are used for adaptive synthesis of the wavelet transform [8], image segmentations [9] and systems identification and diagnosis [10, 11], which proves that ANNs can be used in many technical areas. Based on this fact, for the prediction of corrections, the method based on artificial neural networks is examined in detail in this paper.

The main objective of this work is to compare the results of predicted corrections obtained with different types of artificial neural networks. The difference is in their structure. There are three types under consideration:

1. MultiLayer Perceptron (MLP),
2. Radial Basis Function (RBF),
3. Group Method of Data Handling (GMDH).

The results of the work [12] prove that the use of the MLP and RBF neural networks gives a good prediction quality for UTC(PL). The main disadvantage of using such networks is the considerable amount of time needed to obtain a result [13]. The reason is the need to fit the specific structure of the network and the number of neurons to the nature of the input data during the training process. This disadvantage might be eliminated using GMDH neural networks, which are self-organizing networks [14].

2. Artificial neural networks

The main advantage of artificial neural networks is the ability to generalize knowledge to new, previously unknown information, which is not presented during training. An essential component of an artificial neural network is the neuron, whose model (Fig. 2) was developed in 1943 by McCulloch and Pitts and was based on the construction of a nerve cell.

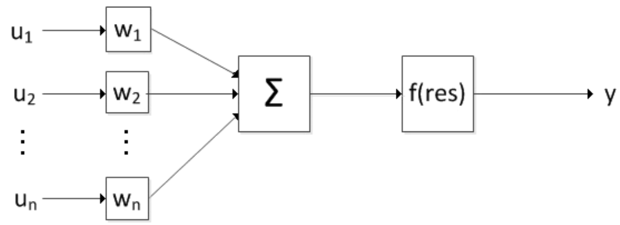

Fig. 2. Artificial neuron model

In Fig. 2 the artificial neuron model is presented, where $u_1 ... u_n$, $n \in \Re$, are the input data, $w_1 ... w_n$, $n \in \Re$, denotes the neuron weights, $res = \sum w \cdot u$ stand for membrane potential and $f(\cdot)$ is the activation function, whose result is the y output from the neuron. Also it is possible to add the bias value, which receives the response of the neuron. The activation function is selected according to the problem considered. In the task of predicting the corrections, the sigmoidal function is applied. The task of neural network training is to determine the weight values [15]. After the training process, the neural network output value should be consistent with the expected value, called the training set.

2.1. Multilayer perceptron. The multilayer perceptron is a typical example of the feed-forward neural network. It consists of multiple layers of neurons, where simultaneously the output of the previous layer is the input in the next layer. The main objective of a single neuron is to classify the input data and to set the appropriate output values, according to the inputs. Before using the perceptron, it is necessary to train it by providing sample input data, and modify the input weights and the connections weights between the layers of neurons so that the output value is the expected value.

2.2. Neural network with the radial basis function. Neural networks with radial activation functions are commonly used for nonlinear approximation of numerical variables. They are used in prediction tasks as well.

The neural architecture of such a network consists mostly of only three layers, i.e., the input layer, one hidden layer with radial neurons and the output layer with a linear neuron (Fig. 3). The task of the linear neuron is weight summation of the signals from the hidden layer. The information pieces in an RBF neural network is propagated forward and do not refer to each other between the neurons of the same layer. The neurons in a hidden layer of an RBF neural network perform a special function. This function changes radially around a chosen center c and has nonzero values only in the close area of the center. The task of each neuron is to project the neuron radial space around a given point or around the cluster, which is the group of points. Next, using the output neuron, the superposition for all signals from the hidden layer is made. Thanks to that it is possible to project the whole multidimensional space.

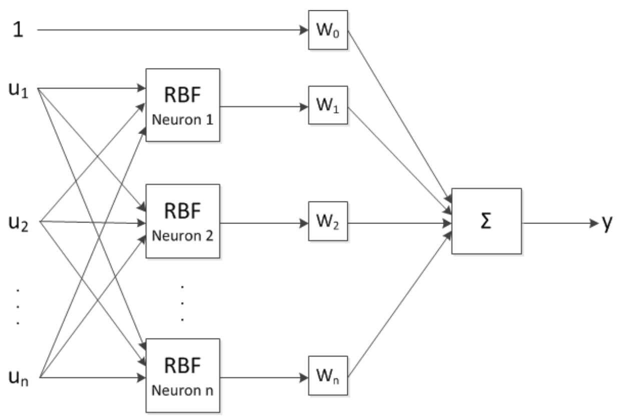

Fig. 3. Structure of the RBF neural network

2.3. GMDH neural networks. For the correction prediction problem, the GMDH neural network can be applied [16]. Such networks are used in many areas, mainly related to data collection, prediction, systems modeling and optimization. This method was proposed in 1968 by A.G. Ivakhnenko from the Institute of Cybernetics in Kiev. GMDH algorithms allow us to automatically find correlation in the data. The idea of the GMDH can be usefully applied in the design of artificial neural networks.

The main problem in designing artificial neural networks is the selection of the optimum structure of the network so as to give the best results. This is a great challenge, because the structure of the network consists of a number of neurons in different layers of the network, a number of layers, and a number of inputs and outputs. Another problem is the preparation of the training and testing data sets. The final stage in solving the prediction problem using neural network is to train it.

The application of the GMDH in the design of artificial neural networks allows predicting the next sample and overcoming the main problem mentioned above. This is because the network structure is generated automatically, based on previously prepared training and testing sets. That is why it is very important to prepare it in a good way, which is discussed in the next section in detail.

The main idea of GMDH neural networks is to replace the single, comprehensive model for a hierarchical structure, consisting of polynomial partial models [17]. The synthesis of a GMDH model is based on alternating estimation of partial models parameters and combining these models using selection methods in such a way that the network structure evolves to the form in which the output signals are the best approximation to the expected values, in the sense of the some criterion.

Partial models parameters are estimated separately, before their inclusion in the neural network structure. Each of these models is trained in such a way that only one model can generate the response close to the expected output value as much as possible. Then each model must pass the quality test, during which some of the models are eliminated from the network structure, based on certain conditions. This process is called selection (Fig. 4).

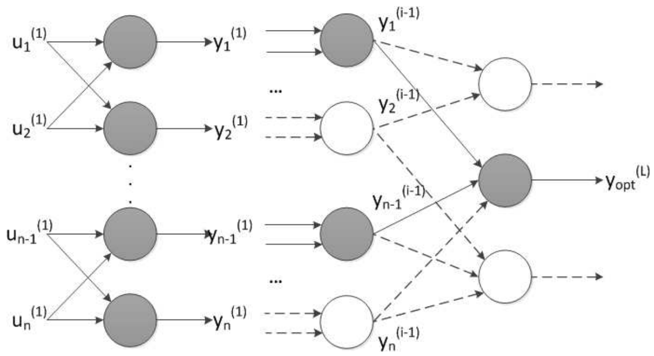

Fig. 4. Selection phase during GMDH model synthesis (light grey circles stand for eliminated neurons)

There are several selection methods, for example, evolutionary methods such as the decreasing population method or the optimum population method. However, the most popular is the constant population method, due to implementation simplicity. Using this method, partial models with the biggest processing error are eliminated and not taken into account during further synthesis of the neural network.

In order to determine the error processing value for each partial model, it is necessary to define the evaluation criterion. To be sure that the model has ability to generalize, the criterion is calculated using the testing data set. There are several criterions. Choosing the appropriate one for the prediction task is a challenging problem. For predicting the corrections, the criterion without data split is used. In such a criterion, for parameter estimation and partial models evaluation, the complete data set is used. Models obtained in such a way have good prediction properties, which is crucial for the present research. An important condition for using such a criterion is that the measurement data used for training and model evaluation must have similar characteristics, i.e., a similar mean value, trend and variance.

GMDH model synthesis continues until the network reaches the optimal structure, according to an appropriately defined criterion. The criterion is to determine the processing error $Q(\widehat{y}_n^{(l)})$ for every n partial models from the l^{th} layer,

$$Q_{\min}^{(l)} = \min_{n=1,\dots y} Q(\widehat{y}_n^{(l)}), \tag{1}$$

where $Q(\widehat{y}_n^{(l)})$can be obtained based on previously defined evaluate criterions used in the selection phase. Values $Q_{\min}^{(l)}$are determined for each model layer. The network structure, obtained during the synthesis process, is optimal if the following condition is satisfied:

$$Q_{opt}^{(L)} = \min_{i=1,\dots L} Q_{\min}^{(l)}, \tag{2}$$

where L stands for the number of all layers. The process of adding additional layers of the network stops when the processing error of the best partial model in the layer begins to grow. The value Q_{opt} represents the processing error of the best partial model in the whole network, which as the same time is the output of the network.

When the final neural network structure is determined, it is possible to improve the final prediction results by training the network again, but this time as a whole structure, not individual partial models [18].

3. Data preparation for GMDH neural networks

3.1. General characteristics of the data. Predicting the corrections for UTC(PL) based on neural networks requires a training process. It depends on the number of input training data and the data preparation method [19]. Data preparation for the GMDH neural network was based on the historical results of measurements of the phase time between UTC(PL) and Cs2 clock (Fig. 1, Station 2), defined for each day according to the relation

$$x_a(t) = UTC(PL) - clock. \tag{3}$$

The data were collected for the period from 53736 MJD (Modified Julian Date) (January 1st, 2006) to 54617 MJD (end of May 2008). For the same period of time, the values of corrections for UTC(PL) relative to UTC were available, with the relation

$$x_b(t) = UTC - UTC(PL) \tag{4}$$

defined at the interval of five days and published in the *Circular T* bulletin. Using the PCHIP (Piecewise Cubic Hermite Interpolating Polynomial) interpolation function available in MATLAB for the data set from BIPM, a mathematical model was determined, which extends the training data set by calculating the values of corrections for UTC(PL) relative to UTC for each day of the analyzed period of time.

The final input data set for the GMDH neural network was calculated as follows:

$$x(t) = x_a(t) + x_b(t) = UTC - clock. \tag{5}$$

3.2. Input data with and without trend elimination. The final data set $x(t)$is the time series $ts1$ which reflect the time instability of the Cs2 clock for each day with reference to UTC. In the time series $ts1$ there is a linear trend $x_r(t)$ and a variable component (Fig. 5). A GMDH neural network training process and prediction of the corrections for UTC(PL) were performed based on $ts1$, which was the first data set.

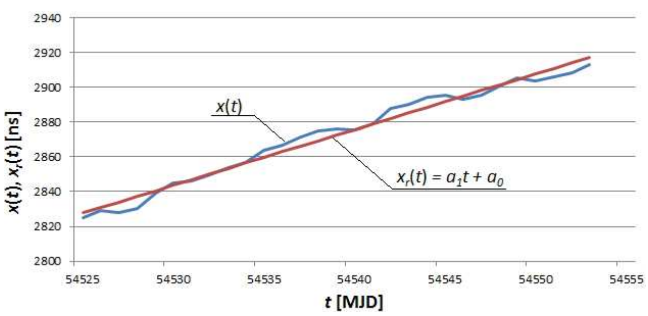

Fig. 5. Sample set of the phase time $x(t)$ and the trend $x_r(t)$ for one month period

Moreover, the additional time series $ts2$ was prepared. Its final value is obtained by eliminating the long-term trend of the phase-time changes $x_r(t)$ described by the linear regression equation (Fig. 5). All time series $ts2$ values, which are deviations of the trend $xd(t)$, were calculated as follows:

$$xd(t) = x(t) - x_r(t). \tag{6}$$

Such a method was used because of the small values of the trend deviations (±9ns, which is 0.5% $x_r(t)$). Such a situation may cause the neural network to take the trend as an important information in the training process, and this may affect the deterioration of the prediction results of UTC(PL). However, when the data series $ts2$ is added to network input, a model which describes the deviation from the trend is taken into account during the neural network training process. This method of input data preparation for a neural network is recommended and gives better prediction results for UTC(PL).

4. Experiment

4.1. Experiment conditions. The prediction of corrections is performed using two methods: the time series analysis method and the regression method. The prediction was made on the 15th day of five consecutive months of 2008 (from January to May). The reason for taking into account such a period of time is that then there was the largest trend deviation for the Cs2 clock in relation to UTC, which may influence the value of predicted corrections. The GMDH neural network training process and predicting the corrections may be performed with a free GMDH application such as GMDH Toolbox [20, 21], or using a commercial one. It this case, the GMDH Shell 2.2. tool is used. Before the experiment, the following assumptions were taken into account:

1. For the time series analysis, the size of the training data frame is set as maximal on every day of prediction, i.e., for day MJD 54479 the frame contains 713 elements and for day MJD 54599 it contains 833 elements.
2. The validation method in both methods of predicting the corrections is "whole data testing", which means that the input data are split into two parts, a training part and a validation part, but for testing the whole set of data is used.
3. The ratio parameter defines the relationship in which the data set is divided into training data and test data. The ratio of training data to the test data was changed every one step from the value of 70/30 to 80/20.
4. The transfer function of the neuron in the GMDH Shell tool is a polynomial whose degree was changed from the value of 1 to 4.
5. The polynomial degree of the transfer function of the neuron and the ratio of training data to test data were selected individually for each day of the prediction.

As input data, the time series $ts1$ and $ts2$ were taken. The predicted value for day t_0+m is the network output, where t_0 is the last day of the month preceding the prediction. Based on it, calculating the correction for UTC(PL) is possible. The first step is to calculate the prediction value for the time series $ts1$:

$$x_{pred}(t_0+m) = (UTC - clock)_{pred}. \tag{7}$$

Based on this value, the prediction value $(UTC(PL) - UTC)_{pred}$ is calculated. Such a prediction value is the correction value, which can be used during UTC(PL) correction in order to ensure the best compatibility of UTC(PL) with UTC. The next step for the time series $ts2$ was to calculate the prediction value of the trend deviation

$$xd_{pred}(t_0+m) = (x(t_0+m) - x_r(t_0+m))_{pred}, \tag{8}$$

which was added to the trend prediction value obtained using regression equations. The result is the prediction value $(UTC - clock)_{pred}$. Further calculations aimed at achieving the prediction correction value were carried out on the time series $ts1$ in the same way.

In order to compare the values of predicted corrections for UTC(PL) obtained for the time series $ts1$ and $ts2$ (GMDH neural networks) and the predictions made by GUM using the analytical linear regression method, the prediction error Δ_{pred} was calculated according to (9),

$$\begin{aligned}\Delta_{pred} &= (UTC(PL) - UTC)_{pred} \\ &-(UTC(PL) - UTC)_{cirt}.\end{aligned} \tag{9}$$

It defines the difference between the predicted value $(UTC(PL) - UTC)_{pred}$ and the value $(UTC(PL) - UTC)_{cirt}$ read from the *Circular T* bulletin for the same day of prediction.

4.2. Experimental results. As has already been mentioned, the prediction of the corrections for UTC(PL) was made on the 15th day for each of five consecutive months. The last prediction value is obtained for the month in which the data set was ended. In Fig. 6 the prediction errors are presented.

In the case of determining the corrections prediction for UTC(PL) using the time series analysis method (Fig. 6a), it can be noted that the preparation way of both time series ($ts1$ and $ts2$) has no significant influence on predicting quality. For the time series $ts1$, the obtained prediction errors in the analyzed period of five months do not exceed the value of ± 3.1 ns, and for the time series $ts2$, ± 0.4 ns. Prediction errors obtained for the same period by GUM, using the analytical linear regression method to predict the corrections for UTC(PL), exceed the value of ± 12 ns. In the case of the regression method (Fig. 6a), the obtained prediction errors for the time series $ts1$ are in the range of ± 6 ns. For the time series $ts2$ and predictions made by GUM, prediction errors exceed the value of ± 12 ns.

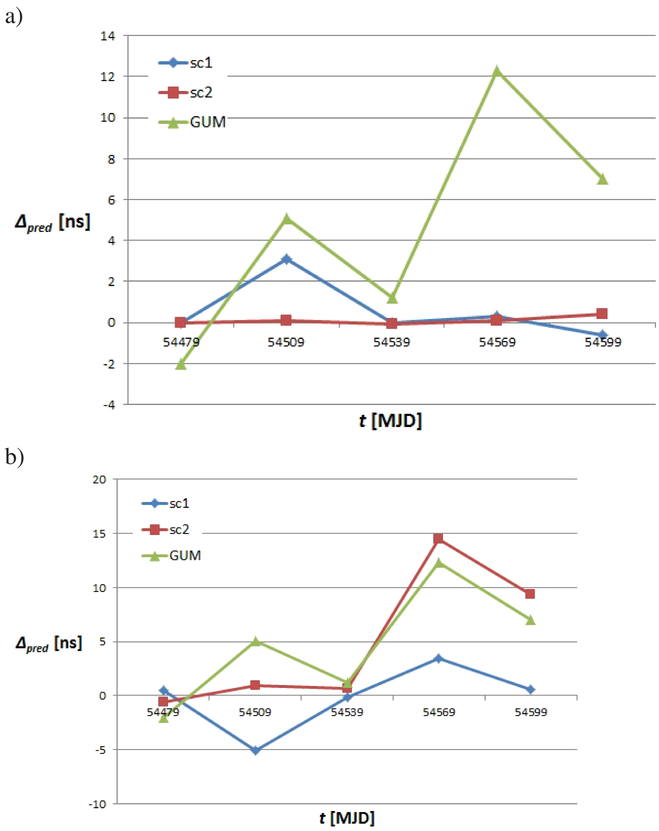

Fig. 6. Prediction errors on the 15th day of five consecutive months using (a) time series analysis method, (b) regression method

Table 1 presents prediction errors obtained using several methods of predicting the corrections for UTC(PL). The results obtained using the analytical linear regression method used in GUM, MLP, RBF and GMDH neural networks were brought together. In the case of GMDH neural networks, to predict the corrections, two methods were used: time series analysis and the regression method. The smallest prediction error was obtained for GMDH networks and time series analysis methods for the time series $ts2$. The maximum error of prediction for the time series $ts2$ does not exceed the value of ± 0.4 ns and was much smaller than the maximum errors obtained with other methods.

Table 1
Prediction errors using prediction with the GMDH neural network, the MLP, the RBF and the linear regression method used in GUM

	Δ_{pred} [ns]							
	GMDH				MLP		RBF	GUM
	time series analysis method		regression method		regression method		regression method	analytical linear regression method
MJD	ts1	ts2	ts1	ts2	ts1	ts2	ts2	
54479	−0.02	0.00	0.51	−0.59	−0.40	0.00	0.20	−2.00
54509	3.08	0.09	−5.11	0.92	−1.70	−3.40	−0.70	5.10
54539	0.00	−0.07	−0.18	0.63	26.30	−1.20	3.10	1.20
54569	0.30	0.07	3.45	14.51	2.10	3.90	3.30	12.30
54599	−0.60	0.40	0.60	9.34	−3.70	5.40	0.80	7.00

5. Conclusions

The objective of this paper was to compare the results of corrections prediction for UTC(PL) using different types of artificial neural networks. The proposed neural networks include: the group method of data handling, the multilayer perceptron and the radial basis function. The obtained results prove, that it is possible to predict the corrections for UTC(PL) using neural networks in effective way. The best results were obtained using the GMDH neural network and the time series analysis method. Moreover, an important advantage of using GMDH neural networks is their automatic adjustment of the structure and the number of neurons to the nature of the time series changes applied to the input of the network in the training process. This results in a very small amount of time needed to receive the predicted correction for UTC(PL) (tens of seconds). In the case of predicting the corrections for UTC(PL), based on MLP and RBF neural networks, the person leading the training process adjusts its structure and the number of neurons in the hidden layer, which greatly extends the time needed to receive the final result of the predicted correction (up to several hours).

REFERENCES

[1] W. Miczulski and Ł. Sobolewski, "Influence of the GMDH neural network data preparation method on UTC(PL) correction prediction results", *Metrol. Meas. Syst.* 19 (1), 123–132 (2012).

[2] A. Czubla, J. Konopka, and J. Nawrocki, "Realization of atomic SI second definition in context UTC(PL) and TA(PL)", *Metrol. Meas. Syst.* (2), 149–159 (2006).

[3] G. Panfilo and P. Tavella, "Atomic clock prediction based on stochastic differential equations", *Metrology* 45, 108–116 (2008).

[4] L.G. Bernier, "Use of the Allan deviation and linear prediction for the determination of the uncertainty on time calibrations against predicted timescales", *IEEE Trans. on Instrumentation and Measurement* 52 (2), 483–486 (2003).

[5] J.A. Davis, S.L. Shemar, and P.B. Whibberley, "A Kalman filter UTC(k) prediction and steering algorithm", *Proc. Joint Conf. IEEE Int. Frequency Control and the Eur. Frequency and Time Forum (FCS)* 1, CD-ROM (2011).

[6] M. Luzar, A. Czajkowski, M. Witczak, and J. Korbicz, "Actuators and sensors fault diagnosis with dynamic, state-space neural networks", *Proc. 17th Int. Conf. Methods and Models in Automation and Robotics* 1, CD-ROM (2012).

[7] S. Koziel, S. Ogusrtov, and S. Szczepański, "Rapid atenna design optimization using shape-preserving response prediction", *Bull. Pol. Ac.: Tech.* 60 (1), 143–149 (2012).

[8] J. Stolarek, "Adaptive synthesis of a wavelet transform using fast neural network", *Bull. Pol. Ac.: Tech.* 59 (1), 9–13 (2011).

[9] J. Gocławski, J. Sekulska-Nalewajko, and E. Kuźniak , "Neural network segmentation of images from stained cucurbits leaves with colour symptoms of biotic and abiotic stresses", *Int. J. Applied Mathematics and Computer Science* 22 (3), 669–684 (2012).

[10] J. Tan, R. Dong, H. Chen, and H. He, "Neural network based identification of hysteresis in human meridian systems", *Int. J. Applied Mathematics and Computer Science* 22 (3), 685–694 (2012).

[11] M. Mrugalski, "An unscented Kalman filter in designing dynamic GMDH neural networks for robust fault detection", *Int. J. Applied Mathematics and Computer Science* 23 (1), 157–169 (2013).

[12] W. Miczulski and M. Cepowski, "Influence of type of neural network and selection of data pre-processing method on UTC-UTC(PL) prediction result", *Measurements, Automation and Monitoring* 11, 1330–1332 (2010).

[13] K. Siwek, S. Osowski and R. Szupiluk, "Ensemble neural network approach for accurate load forecasting in a power system", *Int. J. Applied Mathematics and Computer Science* 19 (2), 303–315 (2009).

[14] S.J. Farlow, *Self-organizing Methods in Modelling: GMDH-type Algorithms*, CRC Press, London, 1984.

[15] M. Huk, "Backpropagation generalized delta rule for the selective attention Sigma-if artificial neural network", *Int. J. Applied Mathematics and Computer Science* 22 (2), 449–459 (2012).

[16] A.G. Ivakhenko and J.A. Mueller, "Self-organizing of nets of active neurons", *System Analysis Modeling Simulation* 20, 93–106 (1995).

[17] M. Mrugalski, E. Arinton, and J. Korbicz, "Dynamic GMDH type neural-networks", *Proc. 6th Conf. Neural Networks and Soft Computing: NNSC'03* 1, 698–703 (2003).

[18] W. Duch, J. Korbicz, L. Rutkowski, and R. Tadeusiewicz, *Biocybernetics and Biomedical Engineering: Neural Networks*, EXIT, Warsaw, 2000, (in Polish).

[19] T. Masters, *Practical Neural Networks Recipes in C++,* Academic Press, New York, 1993.

[20] M. Luzar, "GMDH Toolbox for Matlab" (in Polish), *Proc. 12th Int. PhD Workshop – OWD* 28, 85–90 (2010).

[21] M. Luzar, M. Witczak, "A GMDH toolbox for neural network-based modeling", *Proc. 8th Int. Workshop on Advanced Control and Diagnosis – ACD* 1, 202–206 (2010).