

Ensembles of Neural Networks for Forecasting of Time Series of Spacecraft Telemetry

E. E. Marushko* and A. A. Doudkin**

United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Minsk

**e-mail: marushko@lsi.bas-net.by*

***e-mail: doudkin@newman.bas-net.by*

Received July 22, 2016; in final form, January 11, 2017

Abstract—We analyzed different approaches to developing ensembles of neural networks in respect to their forecasting accuracy. We describe a two level model of ensembles of neural networks for forecasting of telemetry time series of spacecraft's subsystems. A possibility of additional training of these ensembles of neural networks is examined. Our results show that use of ensembles of neural networks with dynamic weighing allows us to reduce the forecasting error.

Keywords: ensemble of neural networks, multidimensional time series, telemetry data, forecasting, training

DOI: 10.3103/S1060992X17010064

INTRODUCTION

The space telemetry is a set of technics allowing us to collect distantly information about the state of on-board subsystems of a spacecraft. They inspect their functioning by means of analysis of readings of monitors distributed over the subsystems. Thus, the telemetry data is a multidimensional time series. One of the problems of the analysis is to forecast the given series.

In the general form, a forecasting problem for a multidimensional time series is formulated as follows [1, 2]: using the known value of the sequence $y(k)$ and its history $y(k-1), y(k-2), \dots, y(k-m)$ it is necessary to estimate the next value of the sequence $y(k+1)$. In the instant k each sequence member is a set of numbers. The length of the history m we use for forecasting is called a time-window.

Since neural network (NN) approaches allow on to simulate dependences between variables by means of training by examples, they are widely used now when solving forecasting problems. At that, their efficiency depends on architectural solutions and training methods [1, 3]. Consequently, the experiments have to be repeatedly performed. There are examples of neural networks use in on-board intellectual decision support systems [4, 5].

In the present publication, we analyze a possibility of short-term forecast of telemetry data using ensembles of neural networks (ENN) [6, 7]. They are sets of NN making decision by averaging the results of individual NNs.

There is a drift in forecasting analysis and machine learning meaning that statistical properties of a target variable the model tries to forecast changes unpredictable with time [8] increasing the forecasting error. Consequently, we can increase the efficiency of a neural network model with the aid of incremental learning methods of ENN [6, 8, 9].

CONSTRUCTION OF ENSEMBLES FOR TELEMETRY DATA PROCESSING

We solved the forecasting problem of telemetry data for three subsystems of the Belarusian spacecraft. They are the power system (PS), the corrective propulsion system (CPS) and the target equipment (TE). Consequently, we generated three ENN for the telemetric data (TD) processing (Fig. 1).

Preprocessed TD and the identifier of the subsystem in question are fed to the ENN input, which is delivered to the supervisor of the ensembles with the aid of the communication block. The supervisor generates a signal for choosing of ENN for the given subsystem and initiates the procedure of its additional

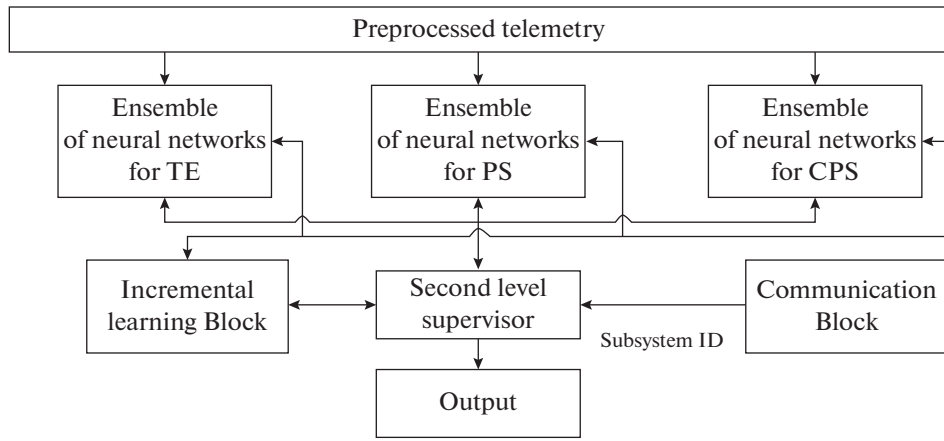


Fig. 1. General scheme of telemetry data processing using ENN.

training. The incremental block of additional training is responsible for preparation of the training data set and training of new elements of ENN.

We use a multilayer perceptron with one hidden layer and the nonlinear activation function in the form of the *hyperbolic tangent* as the basic element of ENN.

We define the size of the input layer N_I for NN of single ENN as the product of the number of the monitors of the subsystem for which we generate this ENN and the time-window we use in the forecast.

We define the size of the hidden layer N_H for NN of single ENN during the experiment using the procedure of searching of the suboptimal size of the hidden layer of single NN.

The size of the output layer N_O for NN of single ENN is defined by the number of monitors of subsystem for which this ENN is generated.

The algorithm RPROP [10] trains individual neural networks.

We use a number of different approaches when generating the output value ENN. They are:

(1) The output value of ENN is generated as a sum of outputs of individual NNs. In the case of one output neuron we calculate it according the formula

$$y = \frac{1}{n} \sum_{i=1}^n y_i \rightarrow, \quad (1)$$

where n is the number of models of NN, and y_i – is the output of the i -th NN;

(2) The output value of ENN has the form of the weighted sum of outputs of individual NNs. In the case of one output neuron it is calculated by the formula

$$y = \sum_{i=1}^n y_i w_i, \quad (2)$$

where n is the number of models, y_i is the output of the i -th NN, w_i is the weight of the i -th NN calculated as

$$w_i = \frac{mse_i}{\sum_{i=1}^n mse_i}, \quad (3)$$

and mse_i is the MSE-error of the i -th NN on the validation set.

(3) The output value of ENN is the weighted sum of outputs of individual NNs (Eqs. (2) and (3)). At that, we repeat the weighting after specified intervals of the processed sets (time readings) with an estimate on this set (dynamically weighted ENN).

Fig. 2 illustrates this approach.

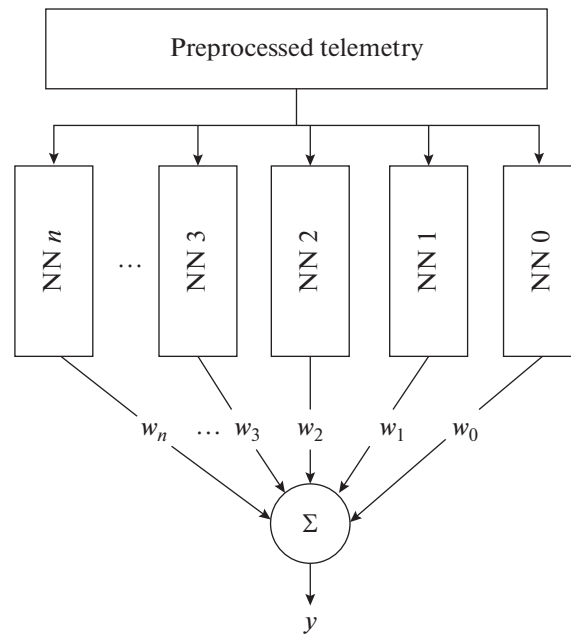


Fig. 2. Output value of ENN is generated as the weighted sum of outputs of individual NNs.

The concept of *drift of values* refers to changes of the defined value with time and, consequently, to changes in the distribution of the given value. The medium from which these values were obtained is not stationary. A shift of the probability may indicate that definitions of the events may also change.

An ensemble of experts trained step-by-step on the input data (without access to the previous data) combined with a form of weighted voting for obtaining the final solution is the common of the algorithms of the drift detecting [8, 9].

Incremental training of ENN means estimating of accuracy of all models and their ranging by accuracy at each forecasting iteration. When the error of ENN increases, the drift of the target variable is detected and a new element trained at the relevant data is added the ensemble. In this approach, we retain the model put in during the initial training and add new parameters without the problem of “forgetting”. This is the way of additional training of ENN.

The algorithm of the incremental training of ENN includes the following steps:

1. The algorithm estimates the accuracy of ENN comparing the errors in the previous and the current steps of its functioning.
2. If the error does not change or the change is in a predetermined range the work stops.
3. Otherwise the training data set including all the stored data after the last additional training is formed.
4. Generation and training of new NN takes place.
5. Generated NN is added to the ensemble.
6. For all NN in the ensemble recalculation of the weight coefficients according Eq. (3) takes place.

Consequently, shown in Fig. 1 the two-level model of ENN forecasting data of subsystems of the spacecraft is generated and works according the following algorithm.

1. Introduction of the sizes N_{PS} ENN_{PS} for processing of TD of PS, N_{CPS} ENN_{CPS} for processing of TD of CPS, and N_{TE} ENN_{TE} for processing of TD of TE
2. Training of N_{PS} individual NN using the set of TD of PS.
3. Training N_{CPS} individual NN using the set of TD of CPS.
4. Training of N_{TE} individual NN using the set of TD of TE.
5. Calculation of the weighted coefficients for individual NNs for each of the ensembles ENN_{PS} , ENN_{CPS} , and ENN_{TE} .
6. Receiving of TD for processing.

Table 1. Telemetric data of CPS monitors

Name of set	Discretization time, s	Dimension of time series	Number of readings
Dt_set_s01	1	24	57501
Dt_set_s05	0.5	24	12245
Dt_set_s1	0.1	24	6613

7. Defining of the processed subsystem (PS, CPS, or TE).
8. Transmission of TD to the corresponding ENN.
9. Generation of the result according Eqs. (1) or (2).
10. Incremental additional training of the chosen ENN.
11. If there are data for processing, repeat item 6.

EXPERIMENTAL EXAMINATION OF EFFICIENCY OF ENSEMBLES

In Table 1, we show the parameters of data sets used in our experiment.

Each of the set is a list of telemetric data generated by the monitors of CPS subsystem of the Belarusian spacecraft characterizing the functioning of the given subsystem. They are the temperature parameters, pressure level inside the xenon delivery unit, and electrical parameters of the flow regulator, and parameters of the anode and cathode of the engines. The values of the monitored parameters depend on the operational condition of CPS, which in turn is defined by executive instructions transmitted from Earth. The readings of the CPS monitors correlate and this allows us to expect a satisfactory forecast.

Since usually we deal with a supervised training, it is necessary to generate a training sample using pairs “input vector, output vector”. We generate the training pairs basing on the principle of sliding window [2]: we extract a segment from the time series; some data of monitoring from this segment we treat as an input vector. The value of the desired output in the training example is the next in order observation. Then the window is shifted one position in direction of the increasing time and we repeat the process of generation of the next training sample.

Consequently, if the dimension of the time series of data is equal to N and the size of the window is W , NN has to accept an input sample of the size $N \times W$. For $W = 20$ the set Dt_set_s1 is transformed into the input set of the size 480×6589 and the target set of the size 28×6589 .

When preparing a universal set of telemetric data from monitors of the spacecraft, resampling and scaling are performed.

We perform resampling in order to transform the source data that are sequences of tickmarks of important events into a form with fixed discretization time. The scaling is necessary to transfer the data into acceptable range $[-1, 1]$. The outputs of the neural network are also scaled.

The aim of our simulations was to define influence of characteristics of an individual NN and architecture of ENN on the results of the forecast with the given forecasting window equal to 20 readings.

To estimate the trained NN and ENN as well as to compare different architectures of ENN we used the following quantities:

—the mean square error (MSE):

$$\text{MSE} = \frac{1}{m} \sum_{i=1}^m e_i^2, \quad (4)$$

—the mean absolute error (MAE):

$$\text{MAE} = \frac{1}{m} \sum_{i=1}^m |e_i|, \quad (5)$$

where $e_i = y_i - t_i$, y_i and t_i are the obtained and the desired signals at the output of i -th neuron of the output layer, respectively, and m is the size of the output layer of NN.

In our simulations, we divided the output sample into training and final sets in a ratio 9 : 1. The total training sample was divided at random into the validation set (15%), the test set (15%) and the training set (70%). We used them for training, estimate and search for the best NN architecture.

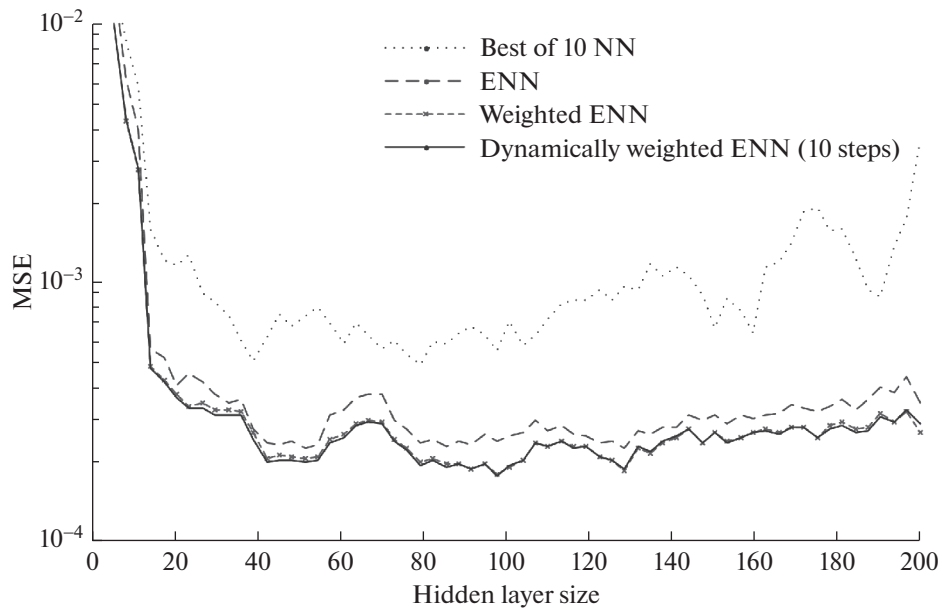


Fig. 3. Estimate of different architectures depending on size of hidden layer of basic NN.

We used the resulting test set calculating the final estimates of the obtained NN.

Estimate of Hidden Layer of NN

We estimated the ability to find a suboptimal size of the hidden layer of the individual NN using the following algorithm.

1. Definition of a search interval of the size of the hidden layer.
2. Training of 10 NN with the size of the hidden layer equal to the current value inside the interval; generation of the weighted ensemble from 10 NN.
3. Estimate of accuracy of ENN.
4. Transition to the next element of the interval until the end of the search interval is reached.
5. Selection of ENN with the minimal MSE on the search interval, the size of the hidden layer of the element of ENN is a suitable solution.

In Table 2, we show the sizes of hidden layers for the sets of data of CPS system.

It worth to be mentioned that this algorithm increases substantially the time of generation of the architecture of NN.

Analysis of Approaches to Generation of Output Value of ENN

Figure 3 shows the dependence of MSE on the size of the hidden layer of the basic NN.

In Table 3, we present estimates of different architectures on the test set Dt_set_s1 with the size of the hidden layer of the individual neural network given in Table 2.

Table 2. Estimate of size of hidden layer of neural network

Data set	Size of hidden layer
Dt_set_s1	28
Dt_set_s05	33
Dt_set_s01	70

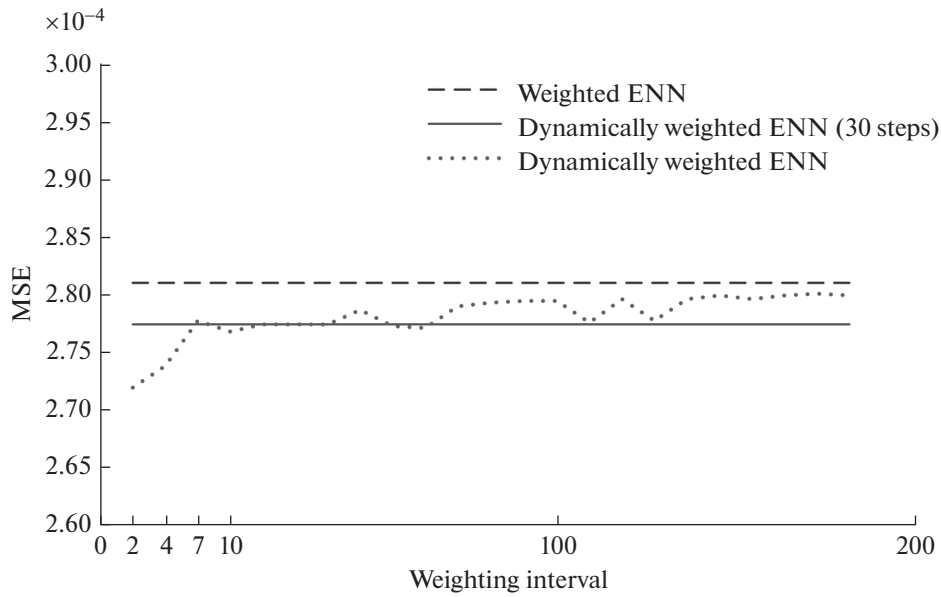


Fig. 4. Estimate of dynamically weighted ENN as function of weighting step.

From data of Table 3, we see that the minimal error corresponds to the dynamically weighted ENN. At that, the differences in the estimated parameters of the dynamically weighted ENN and of singly weighted ENN are very small.

Estimate of Dynamically Weighted ENN

We estimated the weighting step for dynamically weighted ENN. In Fig. 4, we show the MSE for different ENN.

The estimate procedure consists of the following steps.

1. Training with the suboptimal size of the hidden layer.
2. Determination of the search interval for the weighting step
3. Estimate of MSE of ENN, weighted ENN, dynamically weighted ENN with fixed step, dynamically weighted ENN with the step equal to the current value from the interval.
4. Until the end of the search interval is reached move to the next element from the interval.

Figure 4 show that the minimal error corresponds to the dynamically weighted ENN with a small step (less than 10 tick marks).

Estimate of Architecture of NN in Presence of Drift of Target Variable

In the course of this simulation, some modifications were artificially introduced to the analyzed data. We added a linearly increasing trend and simulated a sinusoidal signal as a periodical component.

In Fig. 5, we show a modification of the data set Dt_set_s1 for estimate of additional training of ENN. First 1319 readings were used without the modification.

Table 3. Estimate of different architectures using test set Dt_set_s1

Architecture	MSE, 10^{-4}	MAE, 10^{-3}
Single neural network	3.51	0.115
Ensemble of neural networks	3.66	0.114
Weighted ensemble of neural networks	2.76	9.65
Dynamically weighted ensemble of neural networks, step = 10	2.75	9.59

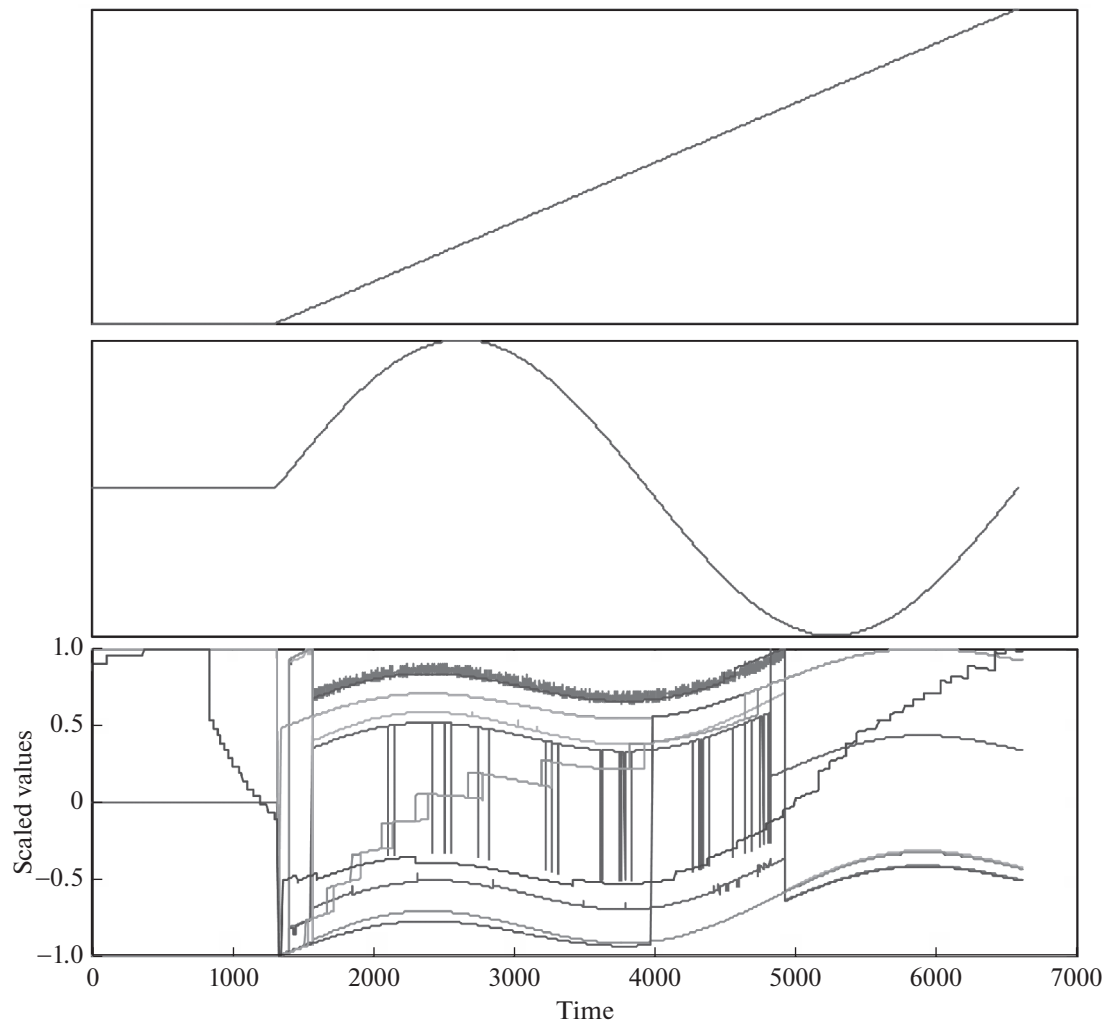


Fig. 5. Modification of data set when estimating additional training of ENN: (1) increasing trend, (2) periodic component and (3) modified multidimensional signal.

In Table 4, we present estimates of different architectures for the test set `Dt_set_s1` with trend and seasonal deviations.

The estimation procedure consists of the following steps:

1. ENN training.
2. Modification of the date set for processing by adding a trend or seasonal component.
3. Specification of the threshold for the algorithm of additional training.
4. Specification of the minimal volume of the additional training set.

Table 4. Estimate of different architectures on the test set with trend and seasonal deviation

Architecture	MSE	MAE
Single neural network	0.6035	0.48428
Ensemble of neural networks	0.030388	0.12742
Weighted ensemble of neural networks	0.028874	0.12115
Dynamically weighted ensemble of neural networks, step = 10	0.02661	0.11703
Dynamically weighted ensemble of neural networks with additional training, step = 10	0.015286	0.08767

5. Estimation of the accuracies of the best of the ten NNs, of ENN, of weighted ENN, of the dynamically weighted ENN, of the dynamically weighted ENN with a fixed step, and of the dynamically weighted ENN with additional training (additional training is performed only when the required data volume is stored).

All the models including ENN with additional training showed a substantial decrease of their accuracies at the modified set due to the interval of storing of the minimal volume of the additional training set.

CONCLUSIONS

Comparing ENN with individual NN, we see that the forecast error decreases substantially.

The best accuracy has the dynamically weighted ENN with small weighting step (less than 10 time readings). At that, the difference in the estimable parameters between the dynamically and singly weighted ENN is insignificant.

Analyzing the telemetric data generated by the monitors of CPS subsystem we obtain the minimal mean square error equal to 2.75×10^{-4} when using the dynamically weighted ENN with the weighting step equal to 10.

For modified data sets with trend and periodical deviation, all the models showed a considerable decrease of accuracy. In the same time, a decrease of accuracy showed ENN with additional training because of necessity of training of additional NN on the suitable number of readings of the additional training set. However, in the case of additional training the resulting accuracy was higher.

REFERENCES

1. Hao Quan et al., Short-Term Load and Wind Power Forecasting Using Neural Network-Based Prediction Intervals, *IEEE Trans. Neural Networks and Learning Systems*, 2013, vol. 25, no. 2, pp. 303–315, ISSN: 2162-237X.
2. Lysayak, A.S., Prediction of multidimensional time series, Lysayak, A.S. and Ryabko, B.Y., Ed., *Vestnik Sib-GUTI*, 2014, no. 4, pp. 75–88 (in Russian).
3. Valipour, M. et al., Comparison of the ARMA, ARIMA, and the autoregressive artificial neural network models in forecasting the monthly inflow of Dez dam reservoir, *J. Hydrol.*, 2013, vol. 476, pp. 433–441.
4. Khachumov, V.M., Review of Standards and the concept of monitoring, control and diagnostics of the spacecraft tools building, *Software Systems: Theory and Applications*, Khachumov, V.M., et al., 2015, vol. 6, no. 3 (26), pp. 21–43 (in Russian).
5. Emelyanov, Yu.G., Neural orientation angles and distance of the spacecraft sensor control system, *Software Systems: Theory and Applications*, Emelyanov, Yu.G., Konstantinov, K.A., Pogodin, S.V., et al., 2010, no. 1 (1), pp. 45–59 (in Russian).
6. Marushko, Y., Using ensembles of neural networks with different scales of input data for the analysis of telemetry data, *Proc. of the XV Intern. PhD Workshop OWD 2013* (Wisla, 2013), Gliwice: Silesian University of Technology, 2013, pp. 386–391.
7. Kourentzes, Nikolaos et al., Neural network ensemble operators for time series forecasting, *Expert Systems Appl.*, 2014, vol.41, no. 9, pp. 4235–4244, ISSN: 0957-4174.
8. Elwell, R., *Incremental learning of variable rate concept drift*, Elwell, *Lecture Notes in Computer Science*, vol. 5519, Elwell, R. and Polikar, R., Ed., MCS, 2009, pp. 142–151.
9. Parikh, D., An ensemble-based incremental learning approach to data fusion, *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, Parikh, D. and Polikar, R., Ed., 2007, vol. 37, no. 2, pp. 437–450.
10. Riedmiller, M., A direct adaptive method for faster backpropagation learning: The RPROP algorithm, Riedmiller, M. and Braun, H., Ed., *Proceedings of the IEEE International Conference on Neural Networks (ICNN)*, San Francisco, 1993, pp. 586–591.

SPELL: 1. ok