Medium-term Electric Energy Demand Forecasting using Generalized Regression Neural Network

Paweł Pełka and Grzegorz Dudek[0000-0002-2285-0327]

Department of Electrical Engineering, Czestochowa University of Technology,
Al. Armii Krajowej 17, 42-200 Czestochowa, Poland
dudek@el.pcz.czest.pl

Abstract. Medium-term electric energy demand forecasting is becoming an essential tool for energy management, maintenance scheduling, power system planning and operation. In this work we propose Generalized Regression Neural Network as a model for monthly electricity demand forecasting. This is a memory-based, fast learned and easy tuned type of neural network which is able to generate forecasts for many subsequent time-points in the same time. Time series preprocessing applied in this study filters out a trend and unifies input and output variables. Output variables are encoded using coding variables describing the process. The coding variables are determined on historical data or predicted. In application examples the proposed model is applied to forecasting monthly energy demand for four European countries. The model performance is compared to performance of alternative models such as ARIMA, exponential smoothing, Nadaraya-Watson regression and neuro-fuzzy system. The results show high accuracy of the model and its competitiveness to other forecasting models.

Keywords: Generalized Regression Neural Network, Medium-term Load Forecasting, Pattern-based Forecasting.

1 Introduction

Medium-term load forecasting (MTLF) provides useful information for energy management, maintenance scheduling, power system planning and operation. It includes forecasts from one month to several years. In competitive markets, where energy is traded, the accurate forecast of monthly, quarterly and yearly energy demands can provide an advantage in negotiations and concluding contracts for medium term generation, transmission and distribution.

The mid-term electric load as a function of time has a complex nonlinear behavior. It expresses a trend following the economic and technological development of a country, yearly seasonality corresponding to climatic factors and weather variations and random component disturbing the time series.

In literature MTLF methods can be categorized into two general groups [1]. The first one includes the conditional modeling approach and focuses on economic analysis, management and long term planning energy load and energy policies. As input information are considered: historical load data, weather factors, economic indicators and

Pełka P., Dudek G.: **Medium-Term Electric Energy Demand Forecasting Using Generalized Regression Neural Network**. In: Świątek J., Borzemski L., Wilimowska Z. (eds) Information Systems Architecture and Technology: Proceedings of 39th International Conference on Information Systems Architecture and Technology – ISAT 2018. Advances in Intelligent Systems and Computing, vol 853, pp. 218-227. Springer, Cham 2018.

electrical infrastructure measures. A MTLF model of this type can be found in [2], where macroeconomic indicators, such as the consumer price index, average salary earning and currency exchange rate are taken into account as inputs.

The second group includes the autonomous modeling approach, which requires a smaller set of inputs: primarily past loads and weather variables. This approach is more suited for stable economies. The forecasting methods applied in this approach are classical methods such as ARIMA or linear regression [3], and computational intelligence methods, such as neural networks [4].

Neural networks have many attractive features, such as: universal approximation property, learning capabilities, massive parallelism, robustness in the presence of noise, and fault tolerance. They are often use to modeling of complex, nonlinear problems such as MTLF [1], [2]. In this work we propose MTLF model based on Generalized Regression Neural Network (GRNN). This is a memory-based, fast learned and easy tuned type of neural network which is able to generate forecasts for many subsequent time-points in the same time. Time series preprocessing applied in this study filters out the trend and unifies input and output variables. Output variables are encoded using coding variables describing the process. ARIMA and exponential smoothing models are applied for prediction of coding variables.

The rest of this paper is organized as follows. In Section 2 we define a forecasting model based on GRNN describing network architecture and learning, and data preprocessing methods. In Section 3 we test the model on real load data. We compare results of the proposed methods to other MTLF methods. Finally, Section 4 concludes the paper.

2 Forecasting Model based on GRNN

2.1 GRNN

GRNN is a type of supervised neural network with radial basis activation functions. It was introduced by Specht in 1991 [5] as a memory-based network that provides estimates of continuous variables. In comparison of other NN types, where data are propagated forward and backward many times until an acceptable error is found, in GRNN data only needs to propagate forward once. Thus, the training of GRNN is very fast. Other advantages of GRNN are: easy tuning, highly parallel structure and smooth approximation of a target function even with sparse data in a multidimensional space.

The GRNN architecture in Fig. 1 is shown. The network is composed of four layers: input, pattern (radial basis layer), summation and output. The input layer distributes inputs x_j without processing to the next layer. In the pattern layer nonlinear transformation is applied to the inputs. Each neuron of this layer uses a radial basis function which is commonly taken to be Gaussian:

$$G_i(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{s_i^2}\right) \tag{1}$$

where: $\mathbf{x}_i$ is the i-th learning sample which is a center vector of the Gaussian function, s_i is a smoothing parameter and $\|.\|$ is a Euclidean norm.

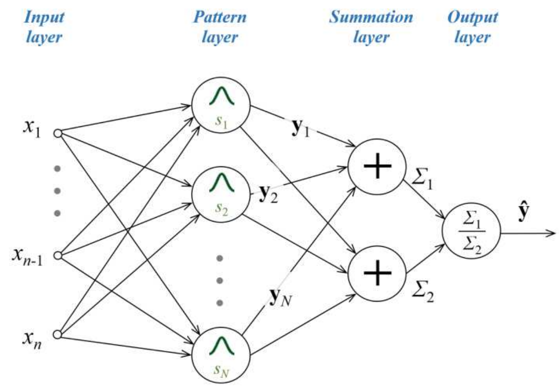

Fig. 1. GRNN architecture.

Each neuron represents individual training vector. Its output expresses the similarity between the input vector $\mathbf{x}$ and the i-th training vector. So the pattern layer maps the n-dimensional input space into N-dimensional space of similarity, where N is the number of training vectors.

The summation layer contains two neurons. The first one calculates the sum Σ_1 of the target patterns $\mathbf{y}_i$ weighted by the neuron outputs, whiles the second one calculates the arithmetic sum Σ_2 of the pattern layer outputs.

The GRNN output calculated by the output layer neuron expresses the weighted sum of the target patterns $\mathbf{y}_i$:

$$\hat{\mathbf{y}} = g(\mathbf{x}) = \frac{\sum_{i=1}^{N} G_i(\mathbf{x})\mathbf{y}_i}{\sum_{i=1}^{N} G_i(\mathbf{x})} \tag{2}$$

Note that the lower distance between $\mathbf{x}$ and $\mathbf{x}_i$ entails the higher i-th neuron output and consequently the higher contribution of the target pattern $\mathbf{y}_i$ to the sum (2).

A smoothing parameter s is the only parameter to estimate. It determines the smoothness of the fitted function and generalization performance of the model. When s becomes larger, the neuron output increases (weights for $\mathbf{y}_i$ in (2) are bigger), with the result that the fitted function becomes smoother. Smoothing parameter s can be the same for all neurons or individually adjusted for each neuron. Finding the optimal smoothing parameter value is a key issue in GRNN learning. In [6] for adjusting s, the same for all neurons, simple enumerative method was used. In [7] for searching N-dimensional space of smoothing parameters a differential evolution algorithm was applied. In this study we assume the same s for all neurons calculated as $s = 0.02 \cdot l \cdot \mathrm{median}(\mathbf{x})$, where median($\mathbf{x}$) is the median of pairwise distances between learning x-patterns and l is tuned by enumerating.

2.2 Time Series Preprocessing

Vector **x** called an input pattern represents predictors, and vector **y** called an output pattern represents the forecasted time series fragment. The input pattern is an n-component vector representing a time series fragment preceding the forecasted fragment. Let us denote the forecasted fragment by $Y_i = \{E_{i+1}\ E_{i+2}\ \ldots\ E_{i+m}\}$, and the preceding fragment by $X_i = \{E_{i-n+1}\ E_{i-n+2}\ \ldots\ E_i\}$, where E_k is the monthly energy consumption and k is the time index. An input pattern $\mathbf{x}_i = [x_{i,1}\ x_{i,2}\ \ldots\ x_{i,n}]^T$ represents the fragment X_i. Components of this vector are preprocessed points of the sequence X_i. Different preprocessing methods are considered [8]:

$$x_{i,t} = E_{i-n+t} \tag{3}$$

$$x_{i,t} = \frac{E_{i-n+t}}{\overline{E}_i} \tag{4}$$

$$x_{i,t} = E_{i-n+t} - \overline{E}_i \tag{5}$$

$$x_{i,t} = \frac{E_{i-n+t} - \overline{E}_i}{D_i} \tag{6}$$

where $t = 1, 2, ..., n$, $\overline{E}_i$ is the mean value of the sequence X_i, and $D_i = \sqrt{\sum_{j=1}^{n}(E_{i-n+j} - \overline{E}_i)^2}$ is a measure of their dispersion.

Pattern components defined using (3) are the same as elements of the sequence X_i. Pattern components defined using (4) are the points of the sequence X_i divided by the mean value of this sequence. Patterns (5) are composed of the differences between points and the mean sequence value. Pattern (6) is the normalized vector $[E_{i-n+1}\ E_{i-n+2}\ \ldots\ E_i]^T$. All patterns defined using (6) have the unity length, mean value equal to zero and the same variance.

Similarly to input patterns, output patterns $\mathbf{y}_i = [y_{i,1}\ y_{i,2}\ \ldots\ y_{i,m}]^T$ representing the forecasted sequence Y_i, are defined as follows:

$$y_{i,t} = E_{i+t} \tag{7}$$

$$y_{i,t} = \frac{E_{i+t}}{\overline{E}_i} \tag{8}$$

$$y_{i,t} = E_{i+t} - \overline{E}_i \tag{9}$$

$$y_{i,t} = \frac{E_{i+t} - \overline{E}_i}{D_i} \tag{10}$$

To calculate the forecast of the monthly energy consumption E_{i+t} on the basis of the forecasted y-pattern generated by the GRNN model we use transformed equations (7)-(10). For example, in the case of (10) the forecasted energy consumption for the horizon t is calculated as follows:

$$\widehat{E}_{i+t} = \widehat{y}_{i,t} D_i + \overline{E}_i \tag{11}$$

where $\widehat{y}_{i,t}$ is the t-th component of the pattern $\widehat{\mathbf{y}}$ predicted by GRNN (2).

In the above formulas (7)-(11), the coding variables $\overline{E}_i$ and D_i are determined in three ways [9]:

C1. In the first approach they are calculated from the sequence X_i. So, $\overline{E}_i$ and D_i for Y_i are the same as for X_i. This enables us to calculate the forecast substituting in (11) coding variables for Y_i, which are unknown at the moment of forecasting, by known coding variables determined for X_i.

C2. In the second approach $\overline{E}_i$ and D_i in (7)-(10) are determined from the sequence Y_i. Note, that in this case coding variables are not available for the forecasted sequence Y_i at the time of making the forecast. Thus, they should be forecasted. We use ARIMA and exponential smoothing (ETS) for this purpose. The forecasted coding variables are inserted into (11) to calculate the forecasted energy consumption.

C3. In the third approach, which is used only for one-step ahead forecasts (variant B in the experimental part of the work), the coding variables $\overline{E}_i$ and D_i are determined from the annual period including time series fragments $\{E_{i-n+2}, E_{i-n+3}, \ldots, E_{i+1}\}$. In this case when using (11) the coding variables cannot be calculated from time series elements because the value of E_{i+1} is not known. Thus, $\overline{E}_i$ and D_i should be predicted. Just like in the case of C2, we use for this ARIMA and ETS.

3 Application Examples

In this section the proposed GRNN model is applied to model and forecast the electricity load demand for four European countries: Poland (PL), Germany (DE), Spain (ES) and France (FR). The data including monthly electricity demand time series were obtained from the ENTSO-E repository (www.entsoe.eu). Data for PL cover the period from 1998 to 2014 and data for the other countries cover the period from 1991 to 2014. The forecasts are made for data from 2014, using data from previous years to GRNN learning. The forecasts were prepared in two variants:

A. for all 12 months 2014 simultaneously (GRNN generates output pattern **y** representing the sequence $Y_i = \{E_{i+1}\ E_{i+2} \ldots E_{i+12}\}$),

B. individually for 12 consecutive months of 2014 (12 GRNN models are created each of which generates a forecast for one month from the period January 2014 – December 2014).

In variant A the training set contains pairs ($\mathbf{x}_i$, $\mathbf{y}_i$), which are historical for the forecasted sequence. The y-pattern having 12 components (m = 12) represents 12 months from January to December. The x-pattern represents n months directly preceding the forecasted sequence. In variant B y-pattern having only one component (m = 1) represents one month of the year. The x-pattern represents n months directly preceding the forecasted month. In variant A the y-patterns are encoded using C1 or C2 approach, whilst in variant B they are encoded using C1 or C3 approach. In variants C2 and C3 the coding variables are predicted using ARIMA and ETS. In Fig. 2 results of forecasting the coding variables in variant C2 are shown.

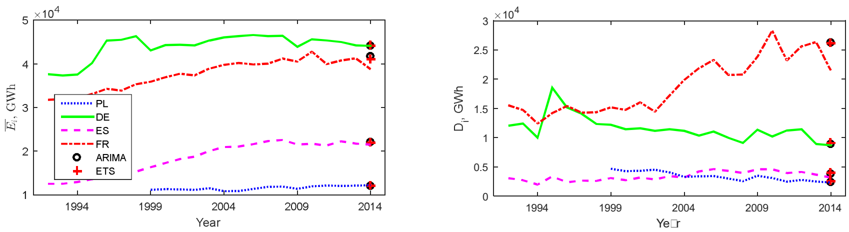

Fig. 2. Forecasts of coding variables in variant C2.

There are two parameters to estimate in GRNN model: the input pattern length n and the smoothing parameter s which is tuned by enumerating variable l (see Section 2.1). The model parameters were selected using grid search in leave-one-out procedure, where n was searched in the range from 3 to 24, and l was searched in the range from 1 to 10.

Figures 3 and 4 show forecast errors for 2014 depending on the model variant and definition of patterns. In most cases the best results were achieved for C1 variant, which does not need additional forecasting of the coding variables. Only for DE data in B variant a little better results were obtained when using C3-ETS. In five out of eight considered variants the lowest errors were achieved when patterns were defined by normalization (6)-(10). In two cases definitions (5)-(9) gave better results, and in one case, for FR data and variant A, the model without time series preprocessing turned out to be the most accurate.

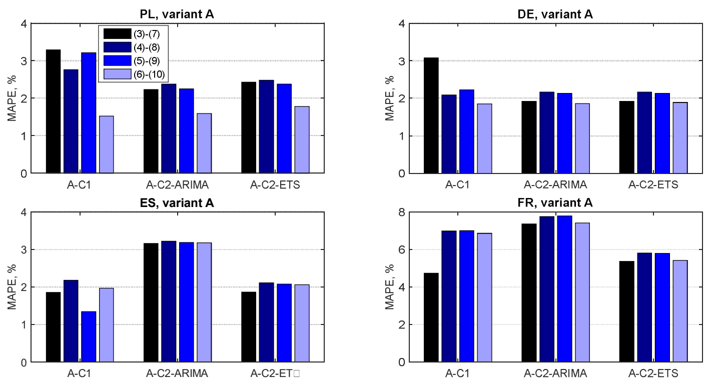

Fig. 3. Errors for different variants of coding variables determination and pattern definitions, variant A.

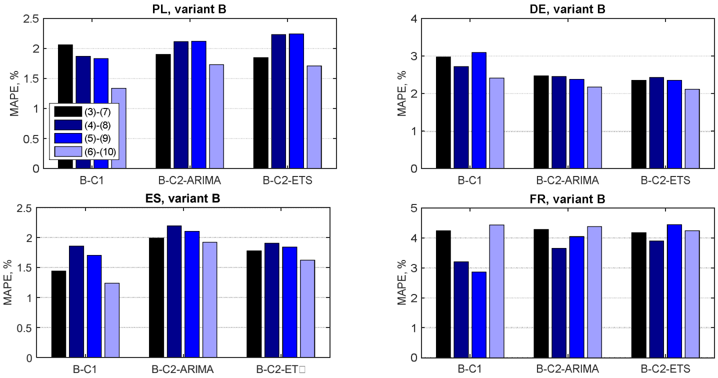

Fig. 4. Errors for different variants of coding variables determination and pattern definitions, variant B.

The real and forecasted monthly demand are presented in Figs. 5 and 6, and errors for each month of 2014 in Figs. 7 and 8. Forecast errors for validation and test samples for best variants of pattern definitions in Tables 1 and 2 are presented. In these tables the results of comparative models are also shown: ARIMA, ETS, Nadaraya-Watson estimator (N-WE) [8] and neuro-fuzzy system (N-FS) [9]. Best results are shown in bold. As you can see from these tables the proposed GRNN model looks quite good against the comparative models. In all cases it outperformed the classical models such as ARIMA and ETS and was competitive in accuracy with state-of-the-art models.

Variant B which generates one-step ahead forecasts, usually provides better results than variant A. An exception is DE, where higher errors in variant B are observed. It is difficult to draw conclusions from Figs. 7 and 8, where errors for successive months are very diverse and there is no regularity here.

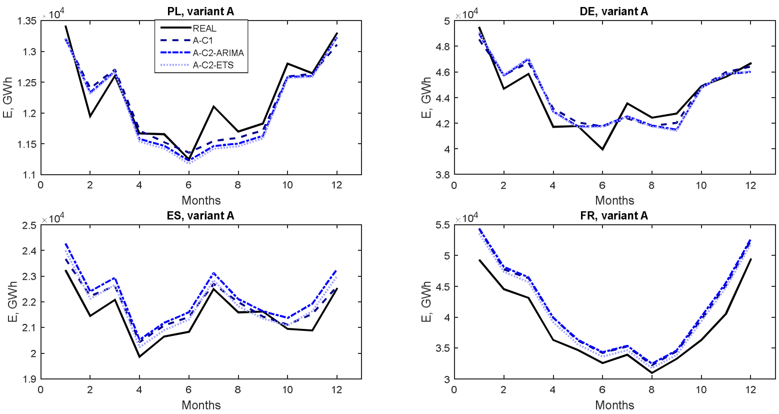

Fig. 5. Real and forecasted monthly demand for 2014, variant A.

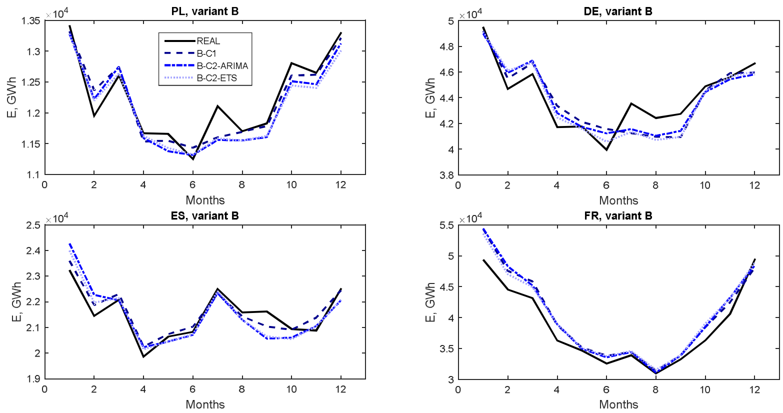

Fig. 6. Real and forecasted monthly demand for 2014, variant B.

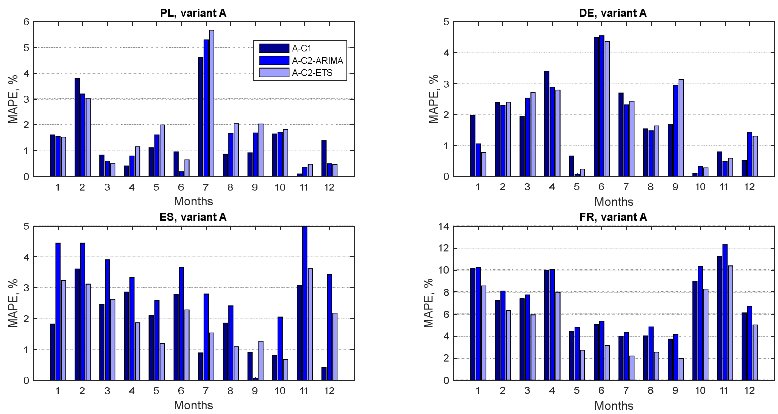

Fig. 7. Errors for consecutive months of 2014, variant A.

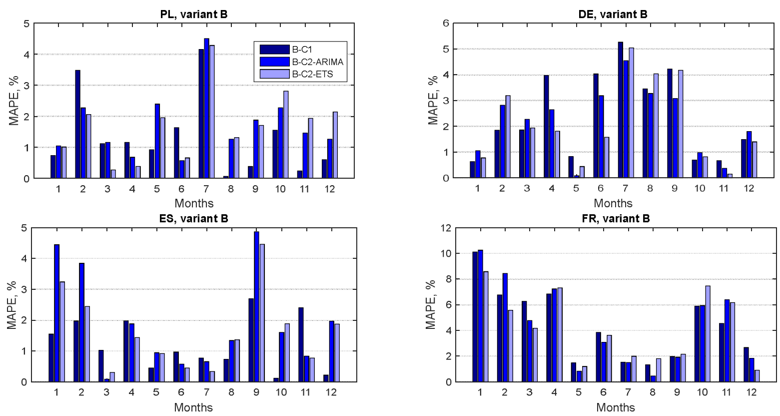

Fig. 8. Errors for consecutive months of 2014, variant B.

Table 1. Forecast errors, variant A.

	PL		DE		ES		FR	
	$MAPE_{val}$	$MAPE_{tst}$	$MAPE_{val}$	$MAPE_{tst}$	$MAPE_{val}$	$MAPE_{tst}$	$MAPE_{val}$	$MAPE_{tst}$
A-C1	2.95	**1.52**	3.25	1.85	2.90	**1.34**	3.20	4.73
A-C2-ARIMA	1.61	1.60	2.03	1.86	2.47	3.16	2.50	7.36
A-C2-ES	1.61	1.78	2.03	1.89	2.47	1.86	2.50	5.37
ARIMA	-	3.25	-	4.36	-	1.93	-	10.76
ETS	-	6.42	-	2.82	-	2.36	-	6.77
N-WE	-	1.53	-	**1.80**	-	1.49	-	4.71
N-FS	-	1.57	-	4.94	-	1.67	-	**3.34**

Table 2. Forecast errors, variant B.

	PL		DE		ES		FR	
	$MAPE_{val}$	$MAPE_{tst}$	$MAPE_{val}$	$MAPE_{tst}$	$MAPE_{val}$	$MAPE_{tst}$	$MAPE_{val}$	$MAPE_{tst}$
B-C1	2.04	1.34	2.52	2.41	2.44	1.24	3.05	2.86
B-C3-ARIMA	1.97	1.73	2.33	2.18	2.10	1.92	3.01	4.04
B-C3-ES	1.97	1.71	2.33	**2.11**	2.10	1.62	2.80	3.90
ARIMA	-	1.75	-	2.33	-	1.43	-	4.10
ETS	-	2.28	-	2.64	-	2.85	-	3.66
N-WE	-	1.30	-	2.47	-	1.16	-	**2.83**
N-FS	-	**1.06**	-	2.87	-	**0.95**	-	5.85

4 Conclusion

In this work we present GRNN model for medium-term load forecasting. In this approach the forecast is derived from the neighborhood of the query pattern using locally weighted regression. The model works on preprocessed time series sequences to filter out a trend and unify input and output patterns. Four methods of preprocessing are considered. Output variables are encoded using coding variables calculated from historical data or forecasted using classical methods: ARIMA or ETS. In most cases forecasting the coding variables does not improve model accuracy compared to calculating them from history.

The model has only two parameters: the smoothing parameter of radial activation functions and the input pattern length. They are searched in a simple grid search procedure. Fast one pass learning and easy tuning are the biggest advantages of the GRNN. In the light of the experimental study, it can be concluded that GRNN has been proven to be useful in medium-term load forecasting. It outperformed the classical models such as ARIMA and ETS and was competitive in accuracy with state-of-the-art models.

References

1. Ghiassi, M, Zimbra, D.K., Saidane, H.: Medium term system load forecasting with a dynamic artificial neural network model. Electric Power Systems Research, vol. 76, pp. 302-316 (2006).
2. Gavrilas, M., Ciutea, I., Tanasa, C.: Medium-term load forecasting with artificial neural network models. In: 16th International Conference and Exhibition on Electricity Distribution, IET, Amsterdam (2001).
3. Hor, C.L., Watson, S., Majithia, S.: Analyzing the impact of weather variables on monthly electricity demand. IEEE Transactions On Power Systems, vol. 20, pp. 2078-2085 (2005).
4. González-Romera, E., Jaramillo-Morán, M., Carmona-Fernández, D.: Monthly electric energy demand forecasting based on trend extraction. IEEE Transactions On Power Systems, vol. 21, pp. 1946-1953 (2006).
5. Specht, D.F.: A General Regression Neural Network. IEEE Transactions on Neural Networks 2(6), 568–576 (1991).
6. Dudek, G.: Neural networks for pattern-based short-term load forecasting: A comparative study, Neurocomputing, vol. 2015, pp. 64- 74 (2016).
7. Dudek, G.: Generalized regression neural network for forecasting time series with multiple seasonal cycles. In: Filev D. et al. (eds.): Intelligent Systems'2014, Advances in Intelligent Systems and Computing 323, pp. 839-846 (2015).
8. Dudek, G., Pełka, P.: Medium-term electric energy demand forecasting using Nadaraya-Watson estimator. In: Rusek S., Gono R. (eds) Proceedings of 18th International Scientific Conference on Electric Power Engineering (EPE), pp. 300-305, IEEE, New York (2017).
9. Pełka, P., Dudek, G.: Neuro-Fuzzy System for Medium-term Electric Energy Demand Forecasting. In: Borzemski L., Świątek J., Wilimowska Z. (eds) Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017, Advances in Intelligent Systems and Computing, vol. 655, pp. 38-47, Springer, Cham (2018).