Applied Soft Computing 37 (2015) 277–287

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Pattern similarity-based methods for short-term load forecasting – Part 1: Principles

Grzegorz Dudek*

Department of Electrical Engineering, Czestochowa University of Technology, Al. Armii Krajowej 17, 42-200 Czestochowa, Poland

ARTICLE INFO

Article history:
Received 9 December 2014
Received in revised form 31 July 2015
Accepted 15 August 2015
Available online 29 August 2015

Keywords:
Patterns of the seasonal cycles
Short-term load forecasting
Similarity-based methods of forecasting
Time series with multiple seasonal cycles

ABSTRACT

Load forecasting is an integral problem in the power system operation, planning and maintenance. The article presents the principles of the pattern similarity-based methods for short-term load forecasting. A common feature of these methods is learning from the data and using similarities between patterns of the seasonal cycles of the load time series. These series are non-stationary in mean and variance, contain long run trend, many cycles of seasonal fluctuations and random noise. The new approach based on the pattern similarity and local nonparametric regression simplifies the forecasting problem and enables us to develop effective forecasting models. Several functions mapping daily cycles of the load time series into input and output patterns are defined. The assumption underlying the pattern similarity-based methods of forecasting and the way of its verification are presented. Some indicators of the strength and stability of the relationship between patterns are described. In the experimental part of the work pattern definitions and the validity of the assumption were verified using Polish power system data. The data analysis was performed specific for load time series. The results show that pattern similarity-based methods can be very useful for forecasting time series with multiple seasonal cycles.

1. Introduction

Short-term load forecasting (STLF) is an integral part of power system control and scheduling. By STLF we usually mean forecasts of hourly, half-hourly or quarter-hourly load in the range from one hour to seven days ahead. STLF is extremely important for energy suppliers, system operators, financial institutions, and other participants in electric energy generation, transmission, distribution, and markets. Precise load forecasts are necessary for electric companies to make important decisions connected with electric power production and transmission planning, such as unit commitment, generation dispatch, hydro scheduling, hydro-thermal coordination, spinning reserve allocation and interchange evaluation. Load forecasts are also used as inputs to the power analysis procedures such as load flow and contingency analysis. Short-term load forecasts are essential for the functioning of electricity markets because the load behavior is the basic driver of electricity prices. Thus the forecast accuracy translates to financial performance of energy companies and other market participants. A conservative estimate is that a 1% reduction in forecasting error for a 10 GW utility can save up to $1.6 million annually [1].

Owing to this importance various STLF models have been reported, that includes conventional methods and new computational intelligence, machine learning and pattern recognition methods. The task is not easy because the load time series is non-stationary in mean and variance, with trend and multiple seasonal cycles. Several factors affect the load behavior. These include weather (temperature, wind speed, cloud cover, humidity, precipitation), time, demography, economy, electricity prices, and other factors such as geographical conditions, consumer types and their habits.

Fig. 1 shows a periodical time series representing hourly electrical loads of the Polish power system, where we can observe annual, weekly and daily variations. From this figure it can be seen that the daily cycles differ in shape. Usually shapes for Tuesday through Friday from the same period of the year are similar, and those for Monday, Saturday and Sunday are distinct. Moreover the shapes are dependent on the year period and can vary over the years. We observe that the underlying levels of the daily cycles changes during the year according to the annual cycle and may change also from one week to the next. The level ratio of the neighboring days may change as well even from week to week. In a few year time horizon the trend is observed connected to the economic activity and economy-wide fluctuations in production and trade. The load time series shown in Fig. 1 is analyzed in the experimental part of this work (Section 5).

* Tel.: +48 343250896; fax: +48 343250803.
E-mail address: dudek@el.pcz.czest.pl

http://dx.doi.org/10.1016/j.asoc.2015.08.040
1568-4946/

Dudek G.: *Pattern Similarity-based Methods for Short-term Load Forecasting – Part 1: Principles*. Applied Soft Computing, vol. 37, pp. 277-287, 2015.

http://dx.doi.org/10.1016/j.asoc.2015.08.040
http://www.gdudek.el.pcz.pl/files/SBFM_Principles_15.pdf

Pattern Similarity-based Methods for Short-Term Load Forecasting – Part 1: Principles

Grzegorz Dudek

***Abstract*—Load forecasting is an integral problem in the power system operation, planning and maintenance. The article presents the principles of the pattern similarity-based methods for short-term load forecasting. A common feature of these methods is learning from the data and using similarities between patterns of the seasonal cycles of the load time series. These series are non-stationary in mean and variance, contain long run trend, many cycles of seasonal fluctuations and random noise. The new approach based on the pattern similarity and local nonparametric regression simplifies the forecasting problem and enables us to develop effective forecasting models. Several functions mapping daily cycles of the load time series into input and output patterns are defined. The assumption underlying the pattern similarity-based methods of forecasting and the way of its verification are presented. Some indicators of the strength and stability of the relationship between patterns are described. In the experimental part of the work pattern definitions and the validity of the assumption were verified using Polish power system data. The data analysis was performed specific for load time series. The results show that pattern similarity-based methods can be very useful for forecasting time series with multiple seasonal cycles.**

***Index Terms*— Patterns of the Seasonal Cycles, Short-Term Load Forecasting, Similarity-based Methods of Forecasting, Time Series with Multiple Seasonal Cycles**

I. Introduction

SHORT-term load forecasting (STLF) is an integral part of power system control and scheduling. By STLF we usually mean forecasts of hourly, half-hourly or quarter-hourly load in the range from one hour to seven days ahead. STLF is extremely important for energy suppliers, system operators, financial institutions, and other participants in electric energy generation, transmission, distribution, and markets. Precise load forecasts are necessary for electric companies to make important decisions connected with electric power production and transmission planning, such as unit commitment, generation dispatch, hydro scheduling, hydro-thermal coordination, spinning reserve allocation and interchange evaluation. Load forecasts are also used as inputs to the power analysis procedures such as load flow and contingency analysis. Short-term load forecasts are essential for the functioning of electricity markets because the load behavior is the basic driver of electricity prices. Thus the forecast accuracy translates to financial performance of energy companies and other market participants. A conservative estimate is that a 1% reduction in forecasting error for a 10 GW utility can save up to $1.6 million annually [1].

Owing to this importance various STLF models have been reported, that includes conventional methods and new computational intelligence, machine learning and pattern recognition methods. The task is not easy because the load time series is non-stationary in mean and variance, with trend and multiple seasonal cycles. Several factors affect the load behavior. These include weather (temperature, wind speed, cloud cover, humidity, precipitation), time, demography, economy, electricity prices, and other factors such as geographical conditions, consumer types and their habits.

Fig. 1 shows a periodical time series representing hourly electrical loads of the Polish power system, where we can observe annual, weekly and daily variations. From this figure it can be seen that the daily cycles differ in shape. Usually shapes for Tuesday through Friday from the same period of the year are similar, and those for Monday, Saturday and Sunday are distinct. Moreover the shapes are dependent on the year period and can vary over the years. We observe that the underlying levels of the daily cycles changes during the year according to the annual cycle and may change also from one week to the next. The level ratio of the neighboring days may change as well even from week to week. In a few year time horizon the trend is observed connected to the economic activity and economy-wide fluctuations in production and trade. The load time series shown in Fig. 1 is analyzed in the experimental part of this work (Section V).

The most commonly employed conventional approaches to modeling time series with seasonal cycles are the Holt-Winters exponential smoothing and the autoregressive integrated moving average (ARIMA) models. In exponential smoothing the time series is decomposed into a trend component (combination of a level term and growth term) and a seasonal component. These components can be combined additively or multiplicatively giving a total of 15 variants of exponential smoothing models. For each of these variants there are two possible state space models, one corresponding to a model with additive errors and the other to a model with multiplicative errors [2]. The exponential smoothing state space models are all non-stationary. One advantage of the exponential smoothing models is that they can be nonlinear. So time series that exhibit nonlinear characteristics including heteroscedasticity may be modeled using exponential smoothing state space models. A disadvantage of exponential smoothing is that the exogenous variables cannot be

The study was supported by the Research Project N N516 415338 financed by the Polish Ministry of Science and Higher Education.

G. Dudek is with the Department of Electrical Engineering, Czestochowa University of Technology, 42-200 Czestochowa, Al. Armii Krajowej 17, Poland (e-mail: dudek@el.pcz.czest.pl).

introduced into the model. Other important weaknesses are overparameterization and a large number of starting values, which for the hourly load time series with daily and weekly cycles is 7·24 = 168. Taylor generalized basic Holt-Winters model by introducing two seasonal components [3]. STLF application examples showed that the double seasonal Holt-Winters model outperforms one seasonal Holt-Winters models as well as double seasonal ARIMA model. Five recently developed exponentially weighted methods in application to STLF were presented in [4]. A new exponential smoothing formulation [5], useful for modeling load time series with daily and weekly cycles, involves smoothing a different intraday cycle for each distinct type of day of a week. Similar days are allocated identical intraday cycles. Improved version of this approach, called parsimonious seasonal exponential smoothing, that has the flexibility to allow parts of different days of a week to be treated as identical was proposed in [6]. For electricity load data this new method compares well with a range of alternatives such as double seasonal Holt-Winters model, double seasonal ARMA model and artificial neural network. More flexible parsimonious approach with the trigonometric representation of the seasonal components based on Fourier series was lately proposed in [7].

The stochastic nature of load demand as a function of time has been frequently modeled with seasonal ARIMA models. An attractive feature of these approaches is that ARIMA processes are a very rich class of possible models and we can find a process which describes accurately enough our data. Introducing exogenous variables is not a problem in ARIMA models as well as modeling multiple seasonal cycles. A disadvantage of ARIMA models is that they can represent only linear relationships which entails their limited accuracy when the variables are related in the nonlinear manner such as weather factors and load. A common obstacle in using ARIMA models is that the order selection process is usually considered subjective and difficult to apply. Due to the combinatorial problem of selecting orders of model, which takes considerable size for models with multiple seasonal cycles, the time series is often decomposed and components, showing less complexity than the original series, are predicted independently. Such an approach is used for example in [8], where the series is decomposed into a trend, seasonal components and irregular component using the moving average technique and smoothing. The deseasonalized series is modeled by an ARMA process with hyperbolic noise. Another type of decomposition using the lifting scheme is proposed in [9]. Based on the results of wavelet multi-resolution analysis, the lifting scheme decomposes the time series into subseries at different resolution levels. The subseries are then forecasted using ARIMA models. Finally, forecasting results at different levels are reconstructed to generate an original load prediction by the inverse lifting scheme.

Another classical approach to the multiple seasonal time series forecasting are the structural time series models formulated in terms of unobserved components [10]. Within these models the time series is seen as the sum of unobserved components: trend, seasonal and irregular components. The

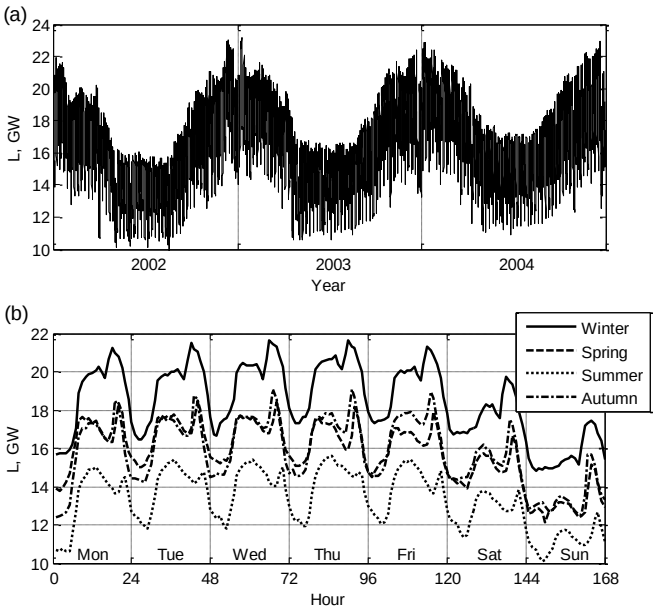

Fig. 1. The load time series of the Polish power system in three-year (a) and one-week (b) intervals.

periodic components are expressed as a mixture of sine and cosine waves. A model is a regression a time trend and a set of seasonal dummies. The explanatory variables are functions of time. To achieve the necessary flexibility the regression coefficients are allowed to change over time. These models are expressed in a state space form with the state representing the unobserved components. The model hyperparameters are estimated using the Kalman filter. One of the advantages of that model is that it allows data irregularities (such as missing observations and observations obtained at mixed frequencies) to be handled. An example of the unobserved components model to the electricity load demand forecasting is presented in [11]. This model is based on modulated periodic components and its advantages are the ability to exhibit several periodic components and less parameters than in a standard unobserved components model.

A typical procedure for the multiple seasonal time series forecasting in the case of conventional models described above and also in the case of models based on the new methods of computational intelligence and machine learning is simplifying the problem by the time series deseasonality or decomposition. After decomposition the components showing less complexity than the original time series can be predicted using simpler models. The time series is usually decomposed on seasonal, trend and stochastic components. Besides decomposition methods mentioned above a useful tool for this purpose is STL filtering procedure based on LOESS smoother (locally weighted polynomial smoother) [12]. STL consists of a sequence of smoothing operations employing the locally weighted regression. The method can be robust to outliers. It has several parameters which allows to control the smoothness and rate of change over time of the seasonal component (this is not allowed in the classical decomposition methods, where it is assumed that the seasonal component repeats from one period to another. In the electrical load time series the seasonal

pattern is not fixed but changes over the year, see Fig. 1).

Another popular technique for seasonal time series decomposition is a wavelet transform. Wavelets produce the local representation of the load signal in both time and frequency domains. The multi-resolution analysis using the wavelet transform splits up the load time series into some subseries in the wavelet domain. One of them is low-frequency representation called an approximation and others are high-frequency representations called details. An approximation expresses a trend whilst details depict components of different higher frequencies. Each subseries is predicted independently in the wavelet domain and the final forecast is obtained using inverse wavelet transform.

Maybe the simplest way to deal with multiple seasonal time series forecasting is decomposition into as many subseries as there are elements in one seasonal cycle of the shortest length. In our load time series example 24 subseries are formed. Each of them is composed of loads at the same hour of a day. In this way a daily seasonality is removed. Each subseries can be forecasted independently using simpler model with only weekly seasonality (in STLF the annual season can be ignored when the model is learned on the period much shorter than one year). To eliminate both daily and weekly seasonalities the time series can be decomposed into 24·7 subseries composed of loads at the same hour of a week, e.g. loads at hour 12 in the successive Mondays.

In conclusion of the above discussion, it can be seen that the forecasting models for multiple seasonal time series generally require deseasonality or decomposition. Sometimes these procedures are built-in in the model structure like in the case of the exponential smoothing. In other cases they are independent of the model. Modeling the non-stationary time series with trend and multiple seasonal cycles usually requires many parameters of the model including decomposition method parameters (tens, hundreds or even thousands of parameters). Thus the searching of the model space to find the globally optimal solution is not a simple task, usually time consuming and of high computational complexity. Other disadvantages of the complex models are their worse generalization ability, unclear structure and uninterpretable parameters, which translates into lack of confidence to the forecasts.

The pattern similarity-based univariate STLF methods proposed in this work are characterized by simplicity. The number of parameters here is small, which implies a simple procedure of model optimization. The principles of the model operation are also simple and clear. Defining patterns of the daily cycles of the load time series we can simplify the forecasting problem which is difficult because of many seasonal cycles, trend and non-stationarity of the time series. A typical procedure for the seasonal time series: deseasonality or decomposition is not necessary here. The functions defining patterns described in this work filter the time series removing the trend and seasonal variations of periods longer than the daily one. Non-stationarity of the time series is also not a problem when using appropriate pattern definitions.

This paper is organized as follows: in Section II the forecasting methods using pattern similarities are outlined. In Section III the patterns of the time series seasonal cycles are defined. Pattern similarity is analyzed in Section IV. In Section V we perform simulation studies of the proposed forecasting approach on the load time series of the Polish power system. Finally, the work is concluded in Section VI. Models of STLF based on the pattern-similarity approach are described in the second part of this work [13]. The study was supported by the research habilitation project and presented in the book in Polish [14]. In these two-piece work we summarize the habilitation thesis.

II. Outline of the Pattern Similarity-based Forecasting Methods

Similarity-based methods are a generalization of the minimal distance methods which are the basis of many pattern recognition and machine learning algorithms [15]. In the time series analysis and forecasting similarity-based methods use analogies between fragments of the time series with seasonal cycles. A time series behavior in the future can be deduced from the behavior of this time series in similar conditions in the past or from the behavior of other time series with similar changes in time. In the first stage of this approach the time series is divided into sequences of length n, which usually contain one seasonal cycle. Our task is to forecast the time series elements in the daily period, so we divide the time series into sequences including 24 successive hours of the daily periods.

In order to eliminate trend and seasonal variations of periods longer than n (weekly and annual variations in our case), the sequence elements are preprocessed to obtain their patterns. The pattern is a vector with components that are functions of actual time series elements. The input and output (forecast) patterns are defined: $\mathbf{x} = [x_1\ x_2\ \dots\ x_n]^T$ and $\mathbf{y} = [y_1\ y_2\ \dots\ y_n]^T$, respectively. The patterns are paired $(\mathbf{x}_i, \mathbf{y}_i)$, where $\mathbf{y}_i$ is a pattern of the time series sequence succeeding the sequence represented by $\mathbf{x}_i$ and the interval between these sequences (forecast horizon τ) is constant. The pattern similarity-based forecasting methods (PSBFMs) are based on the following assumption:

Assumption: If the pattern $\mathbf{x}_a$ representing a period preceding the forecasted period is similar to the pattern $\mathbf{x}_b$ from the history of the process, then the forecast pattern $\mathbf{y}_a$ is similar to the forecast pattern $\mathbf{y}_b$.

Patterns $\mathbf{x}_a$, $\mathbf{x}_b$ and $\mathbf{y}_b$ are determined from the history of the process. The above assumption allows us to forecast the $\mathbf{y}_a$-pattern on the basis of known patterns $\mathbf{x}_a$, $\mathbf{x}_b$ and $\mathbf{y}_b$. Usually we select several patterns $\mathbf{x}_b$ and aggregate patterns $\mathbf{y}_b$ paired with them to get the forecast pattern $\mathbf{y}_a$ paired with the query pattern $\mathbf{x}_a$.

The idea of PSBFMs in Fig. 2 is presented and summarized in the following steps:

1. Mapping the original time series elements (usually seasonal cycles) to patterns $\mathbf{x}$ and $\mathbf{y}$.
2. Selection of similar x-patterns to the query pattern $\mathbf{x}^*$.
3. Aggregation of the y-patterns paired with the similar x-

patterns to get the forecasted pattern $\hat{\mathbf{y}}$ paired with $\mathbf{x}^*$.

4. Decoding the pattern $\hat{\mathbf{y}}$ to get the forecast of the seasonal cycle $\hat{\mathbf{z}}$.

We assumed above that x- and y-patterns represent the seasonal cycles of length n. But in general these patterns can represent any fragments of the time series and these fragments can be different for x- and y-patterns. They can contain several cycles or the part of one cycle. They can be shifted in time, e.g. when we need to prepare the forecast at the time t of the current day we can define x-patterns representing the system load in the period from time t+1 of the previous day to the time t of the current day. Moreover the pattern does not have to represent the contiguous sequence of the time series. The time series elements can be selected to the input pattern. We can use also the feature extraction methods to create new pattern components from the original time series. In further part of the work for the sake of simplicity we focus on patterns representing continuous sequences of the time series which coincide with the daily cycles.

The analyzed PSBFMs belong to the class of nonparametric regression methods. A general regression model has the form [16], [17]:

$$y = m(\mathbf{x}) + \xi\,, \tag{1}$$

where y is the response (target) variable, $\mathbf{x}$ is a vector of explanatory variables (predictors), ξ is a normal distributed random error with zero mean and variance σ^2, and $m(\mathbf{x}) = E(Y|\mathbf{X} = \mathbf{x})$ is the regression curve.

In the nonparametric regression case the regression curve is not predefined, but constructed on the basis of information contained in the data. Regression function is estimated directly, instead of estimate its parameters as in the parametric approach. Most methods assume that the function $m(.)$ is continuous and smooth. The most popular nonparametric regression methods include: kernel estimators, local linear and polynomial regression, smoothing splines, LOESS and LOWESS.

The forecasting models based on pattern similarities are described in the second part of this work [13]. The regression curve estimated by them (or the aggregation method mentioned above in step 3 of the algorithm) is of the simple form:

$$m(\mathbf{x}) = \sum_{i=1}^{N} w(\mathbf{x}_i)\mathbf{y}_i\,, \tag{2}$$

where N is the number of observations and $w(.)$ is the weighting function.

It is worth noting that the variables can be linked in a nonlinear way, if the weighting function maps $\mathbf{x}$ nonlinearly. The function $m(.)$ is a vector-valued function, although we can decompose it to n one-valued functions for each component of vector $\mathbf{y}_i$.

III. Patterns of the Time Series Seasonal Cycles

Definitions of the functions transforming the time series elements into patterns $\mathbf{x}$ and $\mathbf{y}$ are dependent on the specificity of the series (periodicities, trend, variance), the forecast period and horizon. These functions should be defined in such a way that the patterns pass on the most important information about

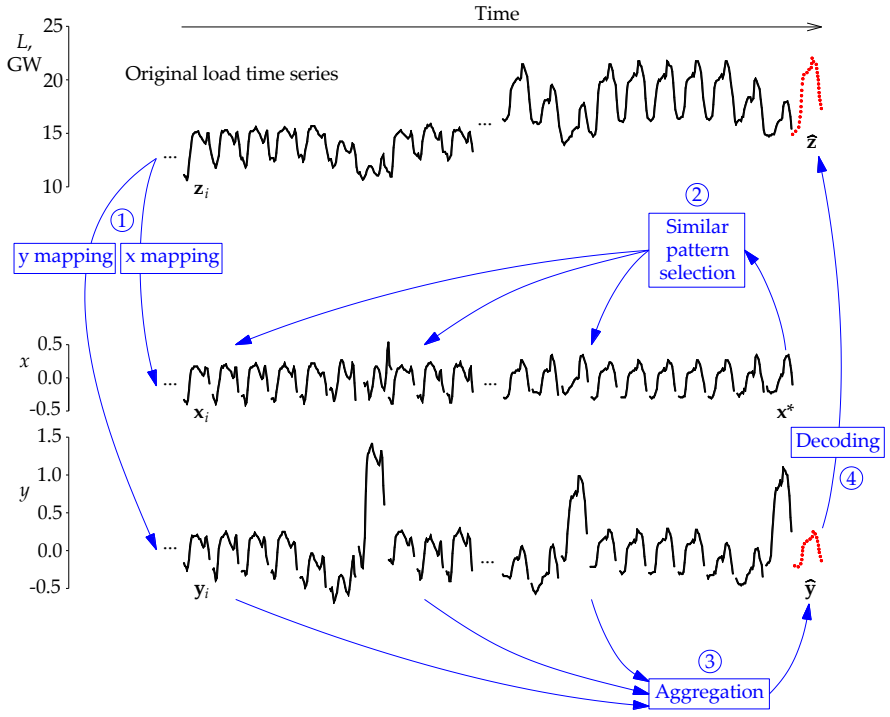

Fig. 2. The PSBFM flowchart.

the process and the quality of the forecasting model was the highest. In addition, functions mapping the time series elements into patterns **y** should provide the opposite transformation (decoding): from the forecasted pattern **y** to the forecasted actual time series elements.

The forecast pattern $\mathbf{y}_i = [y_{i,1}\ y_{i,2}\ \dots\ y_{i,n}]$ represents the sequence of the actual time series elements z in the forecast period $i+\tau$: $\mathbf{z}_{i+\tau} = [z_{i+\tau,1}\ z_{i+\tau,2}\ \dots\ z_{i+\tau,n}]$, and the corresponding input pattern $\mathbf{x}_i = [x_{i,1}\ x_{i,2}\ \dots\ x_{i,n}]$ represents the time series elements in the period i preceding the forecast period: $\mathbf{z}_i = [z_{i,1}\ z_{i,2}\ \dots\ z_{i,n}]$. Sequences $\mathbf{z}_{i+\tau}$ are mapped to patterns $\mathbf{y}_i$ using variables φ_i describing a process in the nearest past, which allows us to take into account current variability of the process and ensures possibility of decoding.

For our load time series with daily, weekly and annual seasonal cycles we define some functions mapping the original space Z into the pattern spaces X and Y, i.e. $f_x : Z \to X$ and $f_y : Z \to Y$. These functions are shown in Table I and II. In Fig. 3 examples of daily load curves for different days of the week are shown and their x- and y-patterns defined using some functions from Table I and II.

Definition X3.1 expresses normalization of vectors $\mathbf{z}_i$. After normalization they have the unity length, zero mean and the same variance. Using the standard deviation of the $\mathbf{z}_i$ components in the denominator of X3.1, we receive vector $\mathbf{x}_i$ with the unity variance and zero mean. Note that after normalization the nonstationary time series is represented by patterns **x** having the same mean and variance. Trend and additional seasonal variations are filtered. This is very important because now we do not need to decompose the time series to built the forecasting model.

Definitions X1 and X2 express the change between z-value and its past values (from the previous moment, day or week) or indices/differences of z-value and the mean value of the daily or weekly periods. There is the relationship between corresponding definitions of X1 and X2: $\mathbf{x}_i^{(2)} = \varphi_i \mathbf{x}_i^{(1)} - \varphi_i$, where $\mathbf{x}_i^{(1)}$ is a pattern from the group X1 and $\mathbf{x}_i^{(2)}$ is a corresponding pattern from the group X2.

The components of x-pattern defined using X4 are angles between line passing through two neighboring time series points and Ox axis. The parameter a allows to control the shape of the pattern. For small a the components have only two values: $\pi/2$ for positive angels and $-\pi/2$ for negative ones. For large a the transformation comes down to the transformation for X2.5 pattern.

Definitions of f_y are analogous to f_x but variables φ_i express time series elements or characteristics determined from the process history. These parameters have to be known at the forecast moment (before the forecast period $i+\tau$). This allows us to calculate the forecast $\hat{\mathbf{z}}$ having the forecasted pattern $\hat{\mathbf{y}}$. To do this we use the inverse function $f_y^{-1}(y_{i,t}, \varphi_i)$. For example the inverse function for Y3.1 is:

TABLE I
FUNCTIONS DEFINING X-PATTERNS

Pattern symbol	$f_x(z_{i,t}, \varphi_i)$	φ_i
X0	$z_{i,t}$	–
X1.1	$\frac{z_{i,t}}{\varphi_i}$	$\bar{z}_i$
X1.2		$\bar{z}_{i-6:i}$
X1.3		$z_{i-1,t}$
X1.4		$z_{i-7,t}$
X1.5		$z_{i,t-1}$
X2.1	$z_{i,t} - \varphi_i$	$\bar{z}_i$
X2.2		$\bar{z}_{i-6:i}$
X2.3		$z_{i-1,t}$
X2.4		$z_{i-7,t}$
X2.5		$z_{i,t-1}$
X3.1	$\frac{z_{i,t} - \varphi_i}{\sqrt{\sum_{j=1}^{n}(z_{i,j} - \varphi_i)^2}}$	$\bar{z}_i$
X3.2		$\bar{z}_{i-6:i}$
X4	$\arctan\left(\frac{z_{i,t} - z_{i,t-1}}{a}\right)$	–

TABLE II
FUNCTIONS DEFINING Y-PATTERNS

Pattern symbol	$f_y(z_{i+\tau,t}, \varphi_i)$	φ_i
Y0	$z_{i+\tau,t}$	–
Y1.1	$\frac{z_{i+\tau,t}}{\varphi_i}$	$\bar{z}_i$
Y1.2		$\bar{z}_{i-6:i}$
Y1.3		$z_{i,t}$
Y1.4		$z_{i+\tau-7,t},\ \tau \le 7$
Y2.1	$z_{i+\tau,t} - \varphi_i$	$\bar{z}_i$
Y2.2		$\bar{z}_{i-6:i}$
Y2.3		$z_{i,t}$
Y2.4		$z_{i+\tau-7,t},\ \tau \le 7$
Y3.1	$\frac{z_{i+\tau,t} - \varphi_i}{\sqrt{\sum_{j=1}^{n}(z_{i,j} - \varphi_i)^2}}$	$\bar{z}_i$
Y3.2		$\bar{z}_{i-6:i}$

Symbols in Table I and II: $i = 1, 2, \dots, N$ – the daily period number, $t = 1, 2, \dots, n$ – the time series element number in the period i, τ – the forecast horizon in the daily periods, $a > 0$ – parameter, $z_{i,t}$ – the t-th time series element in the period i, $\bar{z}_i$ – the mean value of elements in period i, $\bar{z}_{i-6:i}$ – the mean value of elements in the weekly period: from i–6 to i.

$$f_y^{-1}(y_{i,t}, \varphi_i) = y_{i,t} \sqrt{\sum_{j=1}^{n}(z_{i,j} - \bar{z}_i)^2} + \bar{z}_i . \qquad (3)$$

We can reduce the number of x-pattern components and compress information if in the functions f_x we use the mean values of several subsequent time series elements instead of

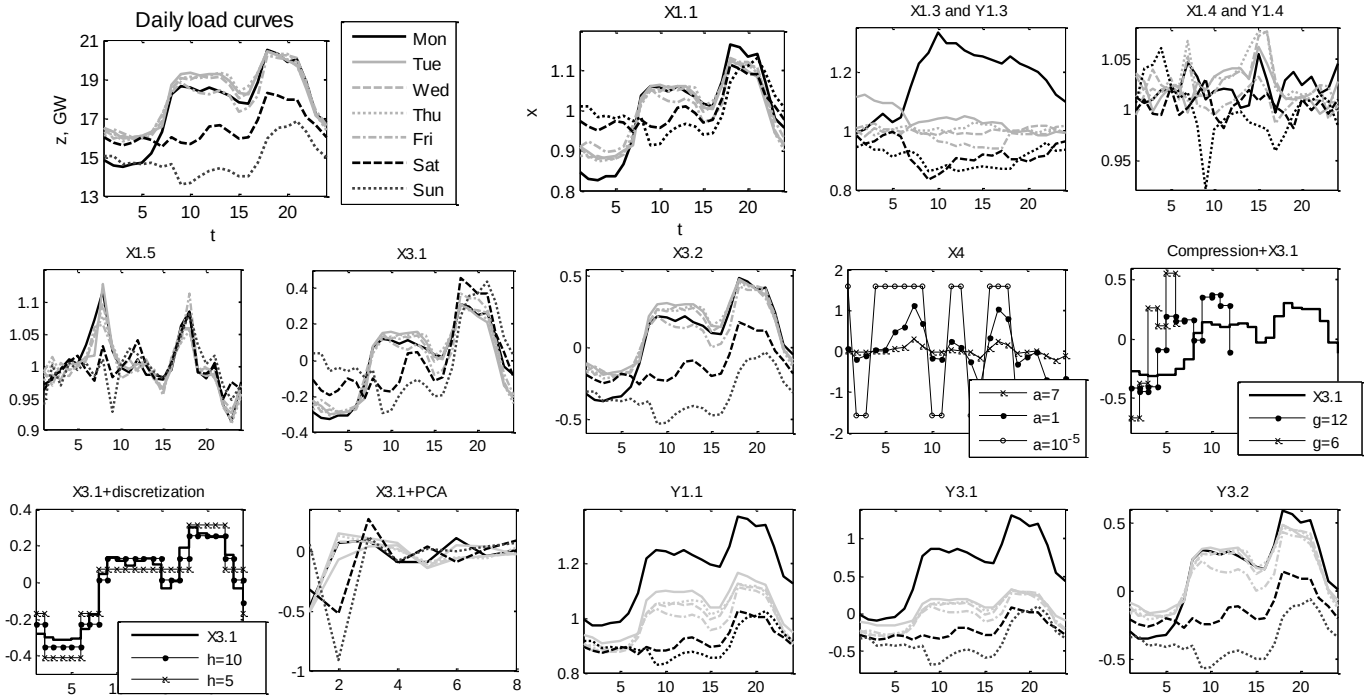

Fig. 3. Examples of x- and y-patterns of the daily load curves.

the original z-values. In this case the period i is divided into g equal intervals (for n = 24, g = 2, 3, 4, 6, 8 or 12). Then the mean values of elements in these intervals are calculated and substituted to f_x in place of z. An example of compression is shown in Fig. 3 (pattern X3.1 and its two compressed versions for g = 6 and 12 are shown).

X-patterns can be further transformed: scaled, discretized or orthogonalized. Scaling can be linear:

$$x'_{i,t} = a_t x_{i,t} + b_t, \tag{4}$$

or nonlinear:

$$x'_{i,t} = \begin{cases} (x_{i,t})^{c_t}, & \text{for } x_{i,t} \ge 0, \\ -(-x_{i,t})^{c_t}, & \text{for } x_{i,t} < 0, \end{cases} \tag{5}$$

where a_t, b_t and $c_t > 0$ are parameters.
The coefficient a_t and exponent c_t determine the degree of smoothing (if $0 < a_t$, $0 < c_t < 1$) or sharpening (if a_t, $c_t > 1$).

The patterns of zero mean can be nonlinearly mapped to interval [0, 1] using sigmoid function:

$$x'_{i,t} = \frac{1}{1+\exp(-dx_{i,t})}, \tag{6}$$

where d is the slope of the sigmoid function.
Using (6) the x-pattern components which are distant form 0 are exponential smoothed.

The discretization consists in dividing the range of variability of all x-pattern components for h usually equal intervals and assigning to $x'_{i,t}$ the middle value of the range which includes $x_{i,t}$. The ranges can be specified for each component separately. Discretization involves the loss of information, but it can reduce model construction time, simplify the algorithm and give good results for noisy data. The discretized X3.1 patterns are shown in Fig. 3.

An example of the orthogonalization method is the Principal Component Analysis (PCA). This method transform linearly correlated x-pattern components into a set of linearly uncorrelated principal components which form the x'-patterns. The number of principal components is less than or equal to the number of original components. The patterns created by orthogonalization of X3.1 patterns are shown in Fig. 3.

In general the x-pattern may represent other variables (exogenous) affecting the forecasted variable. In our load forecasting example such exogenous variables can include weather variables: temperature, wind speed, cloud cover, humidity and precipitation. The y-pattern in particular can represent only one time series element or a characteristic that describes a time series, e.g. the mean, variance or extreme value of the time series in a certain period.

The selection of the best pattern definitions for a given load time series requires testing the forecasting model on different pairs of x- and y-pattern definitions. This is performed usually in cross-validation procedure.

IV. Pattern Similarity Analysis

The most popular similarity measures between two real-valued vectors include [18]: the inner product (for normalized vectors) or closely related to it cosine similarity measure, Pearson's correlation coefficient or Tanimoto measure. The similarity measures are often defined on the base of the distance measures, e.g. using linear mapping: $s(\mathbf{x}_a,\mathbf{x}_b) = c - d(\mathbf{x}_a,\mathbf{x}_b)$ or nonlinear mapping: $s(\mathbf{x}_a,\mathbf{x}_b) = 1/(1+ d(\mathbf{x}_a,\mathbf{x}_b))$,

where: $s(.,.)$ is a similarity, $d(.,.)$ is a distance and c is a constant greater than the highest value of the distance. The most often used distance measures are: Euclidean, Manhattan or Canberra distances. If the components of the vectors are expressed in different units or change in different ranges, in order to offset their impact on the distance, their weighting is recommended. The Mahalanobis distance allows us to take into account the correlations between components.

To confirm the validity of the assumption underlying the PSBFMs, we analyze for a given time series the relationship between similarities of x-patterns and paired with them y-patterns. The analysis is performed on the sample consisting of the pattern pairs:

$$((\mathbf{x}_i, \mathbf{y}_i), (\mathbf{x}_j, \mathbf{y}_j)), \tag{7}$$

where $i, j = 1, 2, ..., N, i \neq j$.
The distances between $\mathbf{x}_i$ and $\mathbf{x}_j$: $d(\mathbf{x}_i, \mathbf{x}_j)$, and between $\mathbf{y}_i$ and $\mathbf{y}_j$: $d(\mathbf{x}_i, \mathbf{x}_j)$ are random variables. We define the realization set of pairs of these random variables of the form:

$$\{[d(\mathbf{x}_i, \mathbf{x}_j), d(\mathbf{y}_i, \mathbf{y}_j)]\} = \{[d(\mathbf{x}_1, \mathbf{x}_2), d(\mathbf{y}_1, \mathbf{y}_2)], [d(\mathbf{x}_1, \mathbf{x}_3), d(\mathbf{y}_1, \mathbf{y}_3)], ..., [d(\mathbf{x}_n, \mathbf{x}_{n-1}), d(\mathbf{y}_n, \mathbf{y}_{n-1})]\}, \tag{8}$$

In order to show the stochastic interdependence of the random variables $d(\mathbf{x}_i, \mathbf{x}_j)$ and $d(\mathbf{y}_i, \mathbf{y}_j)$ we formulate the null hypothesis H_0: The observed differences in numbers of occurrence of the sample elements in the specified categories of the values of random variables $d(\mathbf{x}_i, \mathbf{x}_j)$ and $d(\mathbf{y}_i, \mathbf{y}_j)$ are caused by a random nature of the sample. To verify this hypothesis the chi-square test is used. This test is based on a contingency table which shows joint empirical distribution of random variables $d(\mathbf{x}_i, \mathbf{x}_j)$ and $d(\mathbf{y}_i, \mathbf{y}_j)$. The number of categories of $d(\mathbf{x}_i, \mathbf{x}_j)$ is g and that of $d(\mathbf{y}_i, \mathbf{y}_j)$ is h. The boundaries of the categories for $d(\mathbf{x}_i, \mathbf{x}_j)$ are quantiles of order 0, $1/g$, $2/g$, …, 1 and that for $d(\mathbf{y}_i, \mathbf{y}_j)$ are quantiles of order 0, $1/h$, $2/h$, …, 1. The high value of chi-square statistic, greater than its critical value for the assumed significance level and number of degrees of freedom, justifies the rejection of the null hypothesis and entitles us to create and use PSBFMs.

The relationship strength between $d(\mathbf{x}_i, \mathbf{x}_j)$ and $d(\mathbf{y}_i, \mathbf{y}_j)$ is measured using the Cramer's contingency coefficient V and the Pearson's correlation coefficient ρ:

$$V = \sqrt{\frac{\chi^2}{m[\min(g,h)-1]}}, \tag{9}$$

$$\rho = \frac{\sum_{i=1}^{n}\sum_{j=1, j\neq i}^{n}[d(\mathbf{x}_i, \mathbf{x}_j) - \bar{d}_x][d(\mathbf{y}_i, \mathbf{y}_j) - \bar{d}_y]}{(m-1)s_x s_y}, \tag{10}$$

where: m – the number of elements in set (8), $\bar{d}_x, \bar{d}_y$ – the mean values of $d(\mathbf{x}_i, \mathbf{x}_j)$ and $d(\mathbf{y}_i, \mathbf{y}_j)$, respectively, s_x, s_y – the standard deviations of $d(\mathbf{x}_i, \mathbf{x}_j)$ and $d(\mathbf{y}_i, \mathbf{y}_j)$, respectively.

The relationship between $d(\mathbf{x}_i, \mathbf{x}_j)$ and $d(\mathbf{y}_i, \mathbf{y}_j)$ is strengthened when analyzing the similarity of pairs representing the same days of a week. The statistical sample in this case is of the form (7), but now j is the index of pattern representing the same day of a week as the pattern i. It is recommended to built forecasting models using training data representing the same days of weeks as the test data. For example when a query pattern $\mathbf{x}^*$ represents the Monday load curve and we forecast next daily cycle for Tuesday ($\tau = 1$), the training set contains x-patterns representing Mondays and y-patterns representing Tuesdays.

In the PSBFMs the analysis of relationship between x- and y-pattern similarities or distances can be limited to the nearest neighbors of the x-pattern. Now the index j in (7) corresponds to the k nearest neighbors of $\mathbf{x}_i$. The null hypothesis and the way of its verification are the same as above.

The y-pattern paired with the nearest neighbor of the input pattern $\mathbf{x}_i$ not always is the nearest neighbor of the forecast pattern $\mathbf{y}_i$ paired with $\mathbf{x}_i$. We define the ratio of distances: the distance between patterns $\mathbf{y}_i$ and $\mathbf{y}_{i^*}$ paired with the nearest neighbor of $\mathbf{x}_i$ to the distance between $\mathbf{y}_i$ and its nearest pattern $\mathbf{y}_{i'}$:

$$r_d = \frac{d(\mathbf{y}_i, \mathbf{y}_{i^*})}{d(\mathbf{y}_i, \mathbf{y}_{i'})} - 1, \tag{11}$$

The forecasting model has the largest accuracy if $\mathbf{y}_{i^*}$ and $\mathbf{y}_{i'}$ are the same patterns. Then $r_d = 0$. When $\mathbf{y}_{i^*}$ and $\mathbf{y}_{i'}$ vary significantly, it means weak relationship between variables and poor model accuracy.

The similarity of y-patterns is dependent on the relationship stability between loads in the periods encoded in the paired x- and y-patterns. If the loads of days i and $i+\tau$ are represented by the daily mean loads $\bar{z}_i$ and $\bar{z}_{i+\tau}$, respectively, the measure of stability of this relationship is the indicator:

$$r_s = \frac{\bar{z}_i}{\bar{z}_{i+\tau}}. \tag{12}$$

It is desirable that the indicators r_s for the nearest neighbors of the query pattern $\mathbf{x}^*$ were similar, i.e. show the smallest dispersion. One way to reduce this dispersion is to searching for the nearest neighbors among patterns representing the same type of a week day as the query pattern. This eliminates for example the cases where the query pattern representing Friday has the nearest neighbors among Tuesdays, Wednesdays and Thursdays, which have similar x-patterns to Friday but y-patterns paired with them are completely different from the expected y-pattern of Saturday (for $\tau = 1$). The second way to reduce the r_s dispersion is elimination or replacement of outliers. The simplest method is to replace untypical daily period i by the averaged daily periods i–7 and i+7. Another way to reduce dispersion is to create the synthetic training set having the representative features of the original set but without outliers, misleading, distorted and noisy data.

V. Simulation Studies

The simulation studies were carried out using the time series of the hourly electrical load of the Polish power system from the period of 2002-2004 presented in Fig. 1 (these data can be downloaded from the website http://gdudek.el.pcz.pl/varia/stlf-data). This time series exhibits trend approximated by the linear equation $3.885 \cdot 10^{-5} t + 15.518$ and tree seasonal variations: annual, weekly and daily. The dispersion of the loads measured using weekly moving standard deviation has annual periodic fluctuations following the load fluctuations and the trend: $1.221 \cdot 10^{-5} t + 1.594$. The time series is non-stationary. The Durbin-Watson statistics is equal to 0.054 which is much smaller than the threshold indicating no autocorrelation, i.e. the value of 2. The autocorrelation and partial autocorrelation functions are shown in Fig. 4. The autocorrelation between successive hour loads are very strong and are strengthened for the daily, weekly and annual periods. The Ljung-Box test indicated the significant autocorrelations for all tested lags (the significance level α = 0.01).

The seasonal variations can be identified using the harmonic analysis. The time series is decomposed into the Fourier series. The squares of amplitudes (R^2) of the harmonics are presented in the periodogram (Fig. 5). The annual period is dominant. Its amplitude is greater more than twice than the second largest amplitude. There is large daily amplitude and less 12-hour amplitude which comes from the presence of two valleys and two peaks in the daily load curve. The variation in load between weekends and workdays is expressed by the high weekly and half-weekly amplitudes.

Based on the Parseval's theorem the variance of a time series s^2 can be expressed by the sum of squares of the harmonic amplitudes. The contribution of the i-th harmonic to the variance can be expressed by the ratio $R^2/(2\ s^2)$. For the analyzed time series the contributions of the most significant harmonics were: 8760 h – 0.43, 24 h – 0.16, 168 h – 0.06, 12 h – 0.05 and 84 h – 0.04.

To evaluate the dispersion of the load time series in different intervals we use the variation coefficients: $c = 100 \cdot s / \bar{z}$, where $\bar{z}$ and s are respectively:

- mean and standard deviation of a daily load curve for daily variation coefficients c_d,
- weekly mean and standard deviation of daily means for weekly variation coefficients c_w,
- annual mean and standard deviation of weekly means for annual variation coefficients c_a.

The variation coefficients inform about the contribution of the standard deviation in mean. Fig. 6a shows the average values of the variation coefficients for the Polish power system between years 1997-2009. It can be seen the decreasing tendency for the annual variation coefficient and the increasing tendency for the daily variation coefficient. In Fig. 6b the mean annual values of the daily variation coefficients are shown for each day of the week. It can be seen from this figure the strong differentiation of the daily variation depending on the day type. The largest variation for Mondays is due to the transition from the low-loaded Sunday to workday. Other workdays have similar daily variation to each other. The daily variation coefficients is dependent also on the season of the year (not shown in the figures). In the early months of the year the value of c_d has the decreasing tendency, while in the following months c_d gradually increases.

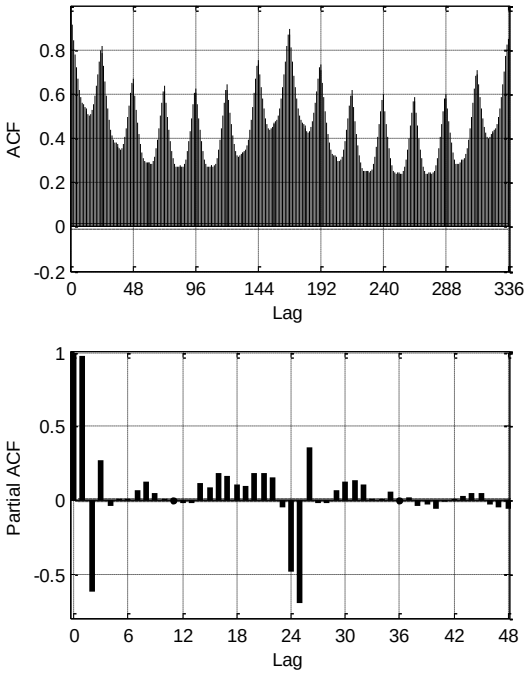

Fig. 4. The autocorrelation and partial autocorrelation functions of the load time series of the Polish power system.

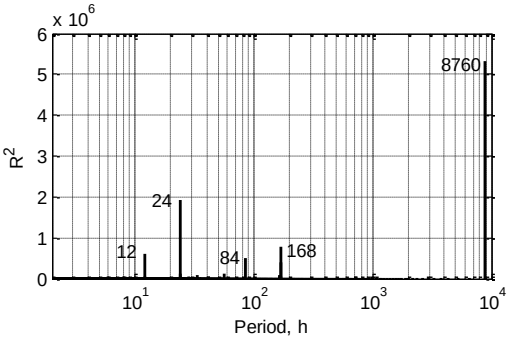

Fig. 5. The periodogram of the load time series of the Polish power system.

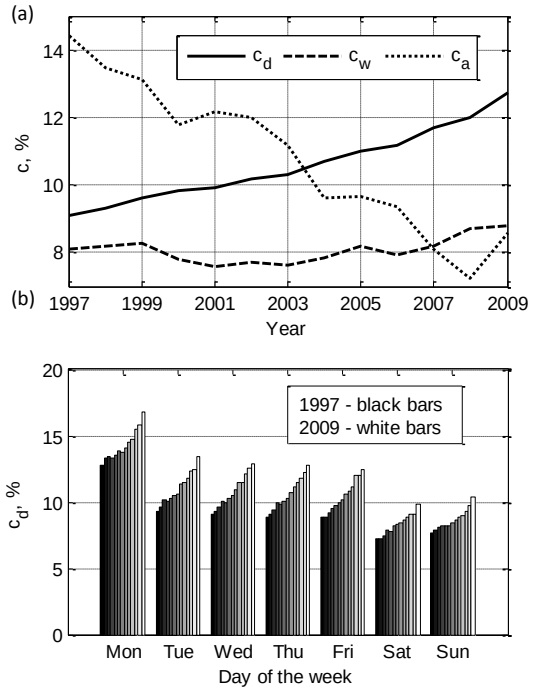

Fig. 6. The mean values of the variation coefficients (a) and the daily variation depending on the day of a week (b).

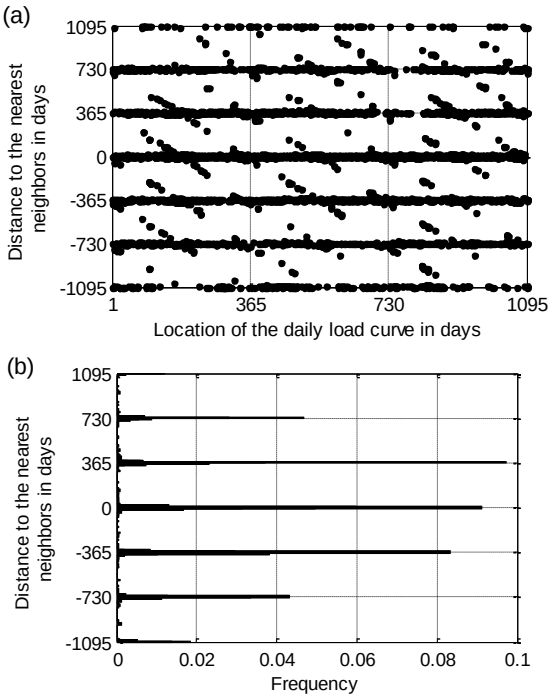

Fig. 7. The distances in time between shapes of the daily load curves and their five nearest neighbors (a) and the histogram of these distances (b).

Let us analyze the similarity between shapes of the daily load profile. It is very important in daily load curve forecasting models. The simplest characteristic of the daily shape is pattern X1.1. The similarity analysis of the shapes shows that the shapes most similar to each other lie in the same periods of the year which are limited to several weeks (Fig. 7). The points arranged in diagonal lines in Fig. 7 shows that the nearest neighbors can also lie symmetrically about the middle days of the year. In the case of shapes of Mondays their nearest neighbors belong to the same category of day type (see Fig. 8). The same applies to Saturdays and Sundays. The nearest neighbors of Tuesday, Wednesday, Thursday and Friday shapes form common category, although the Friday nearest neighbors represent Fridays more frequently than other days of a week (this was confirmed by a statistical test).

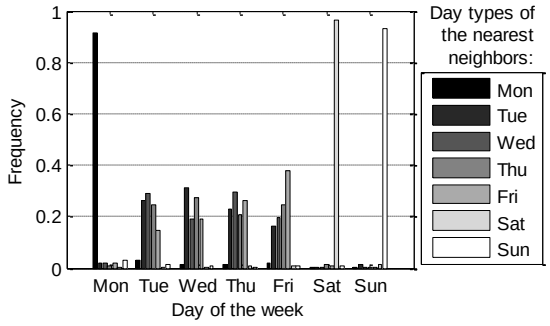

Fig. 8. The day types of the daily load curves and their nearest neighbors in shapes.

The shapes of the daily curves for the same week day category and from the corresponding periods of the year can change in time. This change is indicated by the distance between shapes increasing with the distance in time. This can be seen in Fig. 9, where we can see the average Euclidean distance between shapes from the year 2009 and shapes representing the same week days from the previous years. The shape similarity decreases in time: the average growth rate of the distance is 9.6% per year. The greatest similarity in shapes in the long-term horizon is for Saturdays, and the smallest similarity is for Sundays. The change in shapes can be affected by the change in daily load variance (see Fig. 6). This is illustrated in Fig. 10.

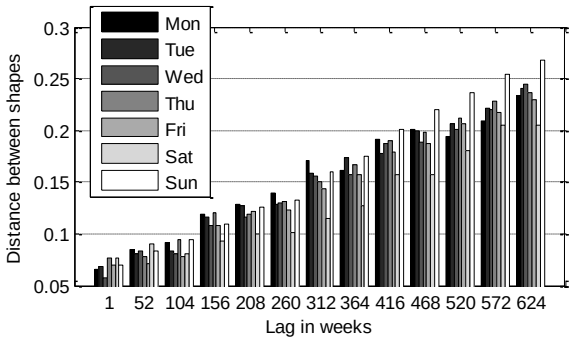

Fig. 9. The average distances between shapes form the year 2009 and shapes representing the same week days from the previous years.

The measure of the nonlinearity of the daily load curve can be expressed by the smallest degree of polynomial approximating this curve with the average error δ. In Fig. 11 the polynomial degrees for the daily curves of the Polish power system in 2009 are shown, where $\delta = MAPE = 1\%$. Average degrees were: 10.6 for workdays, 9.8 for Saturdays and 9.9 for Sundays. The smallest degrees are observed in the

summer and winter seasons, and the highest in the spring season.

Let us focus now on the assumption underlying the PSBFMs presented in Section II. To confirm the validity of this assumption we formulated the null hypothesis (see Section IV) for sample (7). We use the chi-square test to verify H_0 for our load time series. The contingency table having $g = h = 8$ categories for the random variables $d(\mathbf{x}_i, \mathbf{x}_j)$ and $d(\mathbf{y}_i, \mathbf{y}_j)$ is created and the values of statistics χ^2, V and ρ are calculated. Table III shows these statistic values for three types of patterns and two distance measures: Euclidean and Manhattan. The critical value of χ^2 statistic is 66.34 ($\alpha = 0.05$) and its calculated values lie in all cases in the critical region. This results in the rejection of the null hypothesis and means that the relationships between the random variables $d(\mathbf{x}_i, \mathbf{x}_j)$ and $d(\mathbf{y}_i, \mathbf{y}_j)$ are not caused by random nature of the sample. The values of V and ρ coefficients indicate significant, moderately strong and positive correlation between $d(\mathbf{x}_i, \mathbf{x}_j)$ and $d(\mathbf{y}_i, \mathbf{y}_j)$. The largest values of these coefficients are observed for patterns X3.1-Y3.1. Both metrics gave similar results.

Table IV shows the above statistics for the case where we limit the analysis to $k = 5$ nearest neighbors of the x-pattern. Two variants are considered: V2, where the nearest neighbors represent the same day of a week as $\mathbf{x}_i$, and V1, where there is not such a restriction. The results for both Euclidean and Manhattan metrics were similar. In Table IV the former case is presented. In this table the average distances between x-patterns and their five nearest neighbors $\bar{d}_{x,nn}$ and distances between paired with them y-patterns $\bar{d}_{y,nn}$ are also shown. The strongest relationship between random variables for variant V2 and patterns X3.1-Y3.1 is observed. The lower mean distances $\bar{d}_{y,nn}$ for V2 compared to V1 imply lower forecast errors.

The distance ratio r_d for the analyzed load time series, patterns X3.1-Y3.1, Euclidean distance and variant V1 was equal to zero in 11% of cases. Its mean value was 3.07 and the standard deviation was 3.10. For variant V2 these statistics were: 16%, 1.79 and 1.34, respectively. The empirical cumulative distribution functions of r_d are shown in Fig. 12. As we can see again, the variant V2 provides a stronger relationship between random variables.

The average values and standard deviations of the indices r_s expressing the stability of the relationship between loads represented by the x- and y-patterns ($\tau = 1$) in Table V are shown. The $\bar{r}_s$ value is calculated as a mean of r_s values for the nearest neighbors of $\mathbf{x}_i$ representing particular day of a week. In variant V2 the nearest neighbors are selected among x-patterns of the same day type as $\mathbf{x}_i$, whilst in V1 they are selected among all x-patterns. In variant V2 compared to V1 the significant reduction of r_s dispersion can be noted when x- and y-patterns represent days: Tue-Wed, Wed-Thu, Thu-Fri, Fri-Sat and Sat-Sun.

TABLE III
STATISTIC VALUES FOR SAMPLE (7)

Patterns	Distance	χ^2	V	ρ
X3.1–Y3.1	Euclidean	$1.04 \cdot 10^6$	0.35	0.67
	Manhattan	$9.88 \cdot 10^5$	0.35	0.65
X1.1–Y1.1	Euclidean	$7.25 \cdot 10^5$	0.30	0.49
	Manhattan	$6.87 \cdot 10^5$	0.29	0.50
X1.3–Y1.3	Euclidean	$8.32 \cdot 10^5$	0.32	0.24
	Manhattan	$7.97 \cdot 10^5$	0.31	0.25

TABLE IV
STATISTIC VALUES FOR SAMPLE INCLUDING THE NEAREST NEIGHBORS OF X-PATTERN

Patterns	Variant	χ^2	V	ρ	$\bar{d}_{x,nn}$	$\bar{d}_{y,nn}$
X3.1–Y3.1	V1	1583	0.20	0.34	0.13	0.53
	V2	2425	0.25	0.62	0.16	0.38
X1.1–Y1.1	V1	588	0.12	0.21	0.05	0.19
	V2	1163	0.18	0.53	0.06	0.13
X1.3–Y1.3	V1	353	0.10	0.20	0.06	0.18
	V2	904	0.15	0.37	0.07	0.14

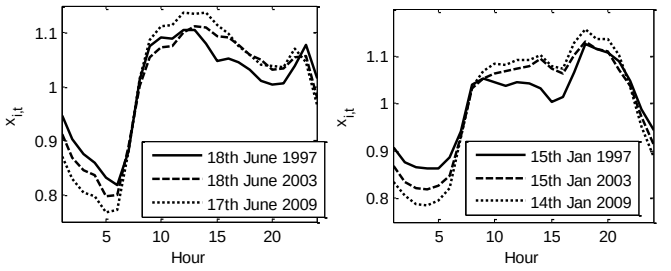

Fig. 10. The shapes of the daily load curves of Wednesdays in summer and winter seasons.

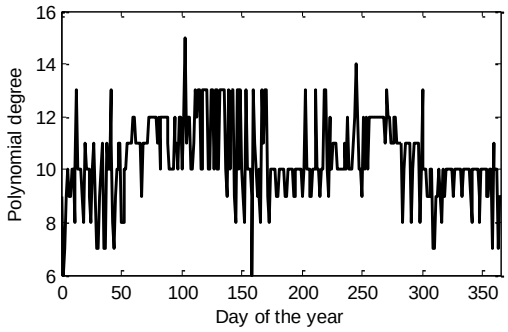

Fig. 11. The degrees of polynomials approximating the shapes of the load curves in 2009.

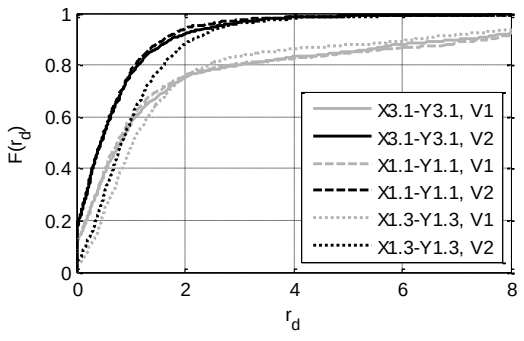

Fig. 12. The cumulative distribution functions of r_d.

TABLE V
MEANS AND STANDARD DEVIATIONS OF r_s

Patterns	Variant		Day type represented by pattern $\mathbf{x}_i$ – Day type represented by pattern $\mathbf{y}_i$ Sun–Mon	Mon–Tue	Tue–Wed	Wed–Thu	Thu–Fri	Fri–Sat	Sat–Sun
X3.1–Y3.1	V1	$\bar{r}_s$	1.180	1.033	0.987	0.988	0.993	0.978	0.897
		s_{rs}	0.042	0.046	0.041	0.041	0.043	0.048	0.050
	V2	$\bar{r}_s$	1.183	1.035	1.003	0.999	1.002	0.927	0.890
		s_{rs}	0.040	0.047	0.021	0.032	0.027	0.028	0.019
X1.1–Y1.1	V1	$\bar{r}_s$	1.182	1.033	0.984	0.988	0.993	0.978	0.895
		s_{rs}	0.041	0.048	0.042	0.040	0.042	0.050	0.043
	V2	$\bar{r}_s$	1.183	1.037	1.002	0.998	1.003	0.925	0.888
		s_{rs}	0.040	0.052	0.020	0.036	0.025	0.025	0.018
X1.3–Y1.3	V1	$\bar{r}_s$	1.185	1.033	1.004	0.987	0.982	0.963	0.893
		s_{rs}	0.045	0.045	0.021	0.044	0.046	0.049	0.028
	V2	$\bar{r}_s$	1.187	1.036	1.005	0.995	1.000	0.927	0.889
		s_{rs}	0.036	0.052	0.017	0.036	0.031	0.025	0.019

TABLE VI
ERRORS (MAPE) FOR DIFFERENT PATTERN DEFINITIONS

	Y0	Y1.1	Y1.2	Y1.3	Y1.4	Y2.1	Y2.2	Y2.3	Y2.4	Y3.1	Y3.2
X0	2.21	2.16	3.21	2.12	4.63	2.16	3.22	2.13	4.49	2.23	3.24
X1.1	4.17	**1.90**	2.95	**1.93**	4.62	**1.94**	2.96	**1.95**	4.45	**1.93**	2.99
X1.2	4.36	2.02	2.13	2.00	3.09	2.06	2.18	2.02	3.04	2.07	2.17
X1.3	7.46	2.70	3.29	**1.98**	4.46	2.79	3.36	**1.99**	4.36	2.72	3.32
X1.4	9.44	2.94	3.33	2.19	2.81	3.07	3.47	2.20	2.82	3.09	3.33
X1.5	4.01	2.05	3.08	2.09	4.60	2.10	3.10	2.12	4.48	2.13	3.11
X2.1	3.85	**1.91**	2.93	**1.93**	4.62	**1.91**	2.91	**1.94**	4.45	**1.92**	2.96
X2.2	4.16	2.04	2.16	2.02	3.11	2.05	2.18	2.02	3.06	2.07	2.18
X2.3	7.18	2.71	3.32	2.01	4.46	2.77	3.34	2.01	4.31	2.78	3.36
X2.4	8.84	2.88	3.28	2.18	2.87	2.99	3.37	2.19	2.82	3.05	3.30
X2.5	3.90	2.08	3.10	2.11	4.62	2.10	3.11	2.13	4.50	2.14	3.12
X3.1	4.14	**1.92**	2.99	**1.91**	4.67	**1.96**	3.00	**1.93**	4.49	**1.92**	3.01
X3.2	4.39	2.05	2.18	2.00	3.08	2.09	2.24	2.01	3.03	2.09	2.17
X4	3.90	2.07	3.10	2.11	4.62	2.09	3.11	2.12	4.50	2.14	3.11

Now we verify the pattern definitions in the task of the next day load curve forecasting for the Polish power system. The simple forecasting model is based on the k nearest neighbor method. The training set contains pairs of patterns $(\mathbf{x}_i, \mathbf{y}_i)$ defined using functions from Tables I and II. The forecasting procedure is as follows: for the query pattern $\mathbf{x}^*$ the k nearest neighbors representing the same day of a week are selected using the Euclidean distance. The forecasted y-pattern paired with $\mathbf{x}^*$ is calculated from the formula:

$$\hat{\mathbf{y}} = \frac{1}{k} \sum_{j \in \Theta(\mathbf{x}^*)} \mathbf{y}_j \,, \tag{13}$$

where $\Theta(\mathbf{x}^*)$ is a set of indices of k x-patterns nearest to $\mathbf{x}^*$ and representing the same day of a week as $\mathbf{x}^*$.

The number of nearest neighbors k was equal to 5. The forecasting errors (MAPE) estimated using the leave-one-out cross-validation method are presented in Table VI. The best results ($MAPE \le 2.00\%$) are bolded.

The lowest errors for patterns X4 was achieved at high values of parameter a (linear scaling). The reduction of the number of x-pattern components using the method described in Section III leads to the error increasing (Fig. 13a). The nonlinear mapping of patterns X3.1 using sigmoid function (6) and discretization of the pattern components did not improve results as well. The PCA method allows to keep the same error levels at the reduction of the principal components up to 10 (Fig. 13b).

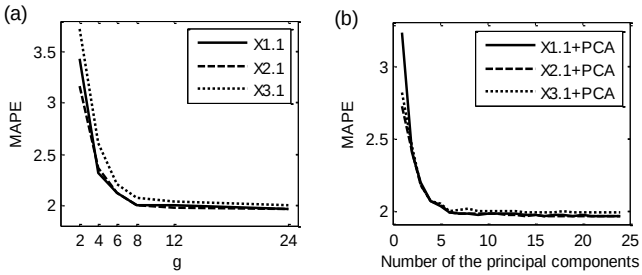

Fig. 13. The forecasting errors depending on the level of compression (a) and number of the principal components (b).

Taking into account the further information about the process history in functions defining patterns (i.e. loads of days i–1 and i–7 or mean loads of the weeks preceding the day i) we did not give such good results as taking into account current informations: loads of the i-th days (patterns X1.1, X2.1, X3.1, X1.3, Y1.1, Y1.3, Y2.1, Y2.3 and Y3.1).

The diagrams of errors for each day of a week and hour of a day, when patterns are defined using different functions, in Fig. 14 are shown. Each diagram has seven rows corresponding to the week days (top row represents Monday, bottom row represents Sunday) and 24 columns corresponding

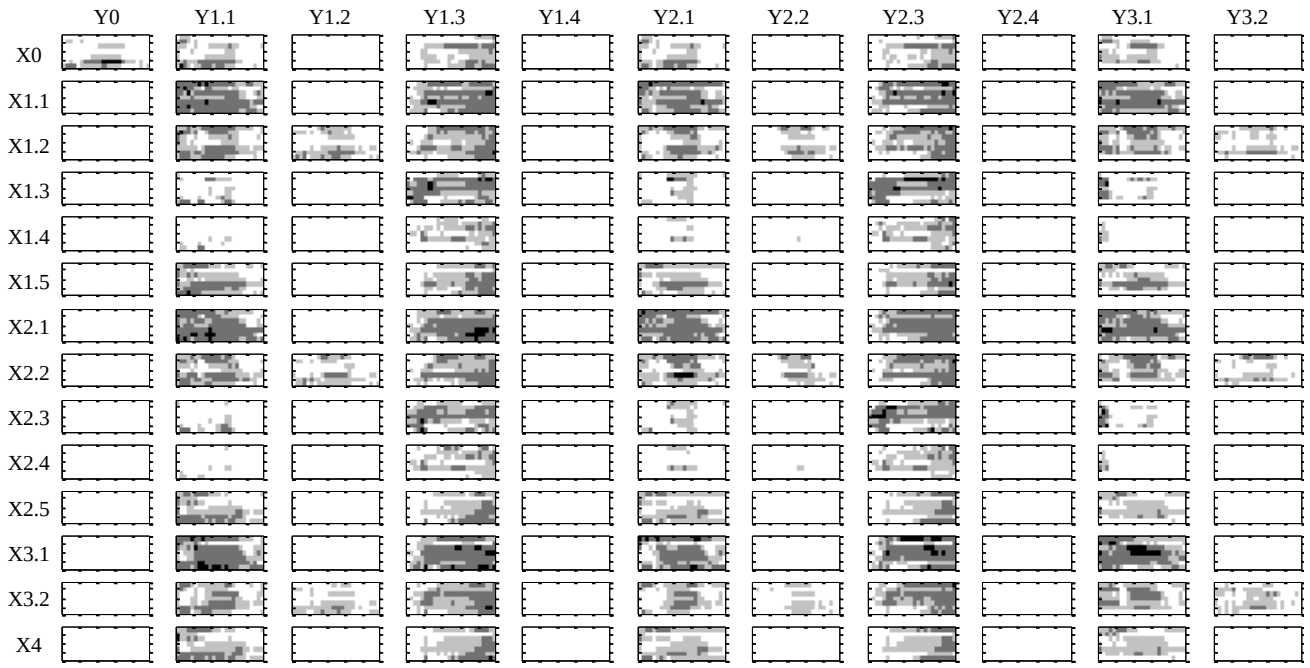

Fig. 14. The diagrams showing the error levels for each day of a week and hour of a day.

to the hours of a day. The black elements in the diagrams indicate the lowest errors. For example, if we want to know which patterns allow us to forecast the load at hour 1 on Monday with the lowest error, we find the diagram having the black element in the position: row = 1 and column = 1. In our case this diagram corresponds to patterns X3.1 and Y2.1 (it lies at the intersection of patterns X3.1 and Y2.1). The dark gray elements in the diagrams indicate errors greater than the lowest one for this day and hour by no more than 5%, and light gray elements indicate errors greater than the lowest one at least 5% and no more than 10%. White elements indicate errors that exceed the minimum error more than 10%. For example, patterns X2.1 and Y3.1 give good forecasts at hour 1 on Monday (diagram for these patterns has dark gray element in the position 1, 1), patterns X4 and Y1.1 give worse forecasts for this case (light gray element in the position 1, 1), and patterns X2.3 and Y2.1 give errors higher than 10% than the best model for this case (white element in the position 1, 1).

From Fig. 14 it can be seen which are the best pattern definitions for the load forecasting at given day of a week and hour of a day. For example, the lowest errors for the noon and afternoon hours on Saturdays were achieved using patterns X0 and Y0. The lowest errors for the same hours on Wednesdays and Thursdays were achieved using patterns X3.1 and Y3.1. For last hours on Sundays patterns X3.1 and Y1.3 have brought best results. Models using patterns Y1.4 and Y2.4 generated the largest errors for each day type and hour regardless of the x-pattern definition (note only white elements in the diagrams corresponding to these y-patterns).

VI. Conclusions

STLF is an important problem in the daily operation of power systems. Accurate forecasts lead to improve the power system security and cost savings as well. For these reasons, many researchers are making efforts to construct forecasting models which are more and more accurate. This is not a simple task given that the load time series exhibit many seasonal variations as well as trend.

In this article we describe principles of the pattern similarity-based methods of STLF. The patterns carry information about the shape of the daily load curve and allow us to simplify the forecasting problem reducing or eliminating non-stationarity and filtering trend and seasonal cycles longer than the daily cycle. Several pattern definitions are proposed. The best definitions for the given time series depend on its specificity, forecast period and horizon. In the experimental part of the work we test pattern definitions in the STLF problem using Polish power system data. The lowest errors were achieved when using patterns defined with variables describing process in the nearest past. In this case the current variability of the process is included into patterns. The way in which the output (forecast) patterns are defined provides the ability of decoding: from the forecasted pattern to the forecasted actual loads.

The PSBFMs are based on the assumption that if the input patterns representing the time series elements in the periods preceding the forecast periods are similar to each other, then the forecast patterns paired with them and representing the forecast periods are similar to each other as well. The assumption means that the neighboring input patterns and the forecast patterns paired with them stay in a certain relation, which does not change significantly in time. The forecast accuracy depends on the stability of this relation. The relationship is strengthened when learning sample contains patterns representing the same day of a week as the query

pattern. The strength of the relationship is also affected by an appropriate definition of input and forecast patterns, selection of the representative learning examples and elimination of outliers. The statistical analysis proposed in this work allow us to evaluate the validity of the assumption and strength of the relationship between input and forecast patterns for a given time series. If the results of this analysis are positive, like for the Polish power system data, the sense of construction and using PSBFMs is justified.

Note that the time series associated with electricity demand represents different "hard" time series expressing multiple seasonal variations. Thus the proposed approach of forecasting can be applied not only to STLF but also to other forecasting problems: economical, business, industrial, meteorological etc.

Using patterns we can construct not only similarity-based models but also other forecasting models, e.g. based on the classical statistical methods like linear regression or machine learning methods such as neural networks [19] and decision trees [20].

REFERENCES

[1] B.F. Hobbs, S. Jitprapaikulsarn, S. Konda, V. Chankong, K.A. Loparo, D.J. Maratukulam, "Analysis of the Value for Unit Commitment of Improved Load Forecasting," *IEEE Trans. Power Systems*, vol.14, no.4, pp. 1342-1348, 1999.

[2] R.J. Hyndman, A.B. Koehler, J.K. Ord, R.D. Snyder, *Forecasting with Exponential Smoothing: The State Space Approach*. Springer, 2008.

[3] J.W. Taylor, "Short-Term Electricity Demand Forecasting using Double Seasonal Exponential Smoothing," *Journal of the Operational Research Society* 54, pp. 799-805, 2003.

[4] J.W. Taylor, "Short-Term Load Forecasting with Exponentially Weighted Methods," *IEEE Trans. Power Systems*, vol.27, no.1, pp. 458-464, 2012.

[5] P.G. Gould, A.B. Koehler, F. Vahid-Araghi, R.D. Snyder, J.K. Ord, R.J. Hyndman, "Forecasting Time-Series with Multiple Seasonal Patterns," *Eur. J. Oper. Res.*, vol.191, pp. 207-222, 2008.

[6] J.W. Taylor, R.D. Snyder, "Forecasting Intraday Time Series with Multiple Seasonal Cycles using Parsimonious Seasonal Exponential Smoothing," *Omega*, vol.40, issue 6, 2012.

[7] A.M. De Livera, R.J. Hyndman, R.D. Snyder, "Forecasting Time Series with Complex Seasonal Patterns Using Exponential Smoothing," *Journal of the American Statistical Association*, vol.106, Issue 496, pp. 1513-1527, 2011.

[8] J. Nowicka-Zagrajek, R. Weron, "Modeling Electricity Loads in California: ARMA Models with Hyperbolic Noise," *Signal Processing* 82, pp. 1903–1915, 2002.

[9] C.-M. Lee, C.-N. Ko, "Short-Term Load Forecasting using Lifting Scheme and ARIMA Models," *Expert Systems with Applications* 38, pp. 5902–5911, 2011.

[10] A.C. Harvey, *Forecasting Structural Time Series Models and the Kalman Filter*. Cambridge University Press, 1989.

[11] D.J. Pedregal, P.C. Young, "Modulated Cycles, An Approach to Modeling Periodic Components from Rapidly Sampled Data," *International Journal of Forecasting* 22, pp. 181–194, 2006.

[12] R.B. Cleveland, W.S. Cleveland, J.E. McRae, I. Terpenning, "STL: A Seasonal-Trend Decomposition Procedure Based on Loess." *Journal of Official Statistics*, vol.6, no.1, pp. 3–73, 1990.

[13] G. Dudek, "Pattern Similarity-based Methods for Short-Term Load Forecasting – Part 2: Models," *Applied Soft Computing (in review).*

[14] G. Dudek, *Pattern-Similarity Machine Learning Models for Short-Term Load Forecasting*. Academic Publishing House "Exit", Warsaw 2012 (in Polish).

[15] W. Duch, "Similarity-based Methods: A General Framework for Classification, Approximation and Association," *Control and Cybernetics*, vol. 29, no.4, pp. 937-968, 2000.

[16] W.K. Härdle, *Applied Nonparametric Regression*. Cambridge University Press, 1992.

[17] K. Takezawa, *Introduction to Nonparametric Regression*. Wiley, 2006.

[18] S. Theodoridis, K. Koutroumbas, *Pattern Recognition*. Elsevier Academic Press, 2009.

[19] G. Dudek, "Forecasting Time Series with Multiple Seasonal Cycles using Neural Networks with Local Learning," *Artificial Intelligence and Soft Computing*, Rutkowski L. at al., eds., ICAISC 2013, LNCS, vol. 7894, pp. 52-63, Springer, 2010.

[20] G. Dudek, "Short-Term Load Forecasting using Random Forests," *Advances in Intelligent Systems and Computing*, Filev D. et al., eds., Intelligent Systems'2014, vol. 323, pp. 821-828, 2015.

Grzegorz Dudek received his PhD degree in electrical engineering from the Czestochowa University of Technology, Poland, in 2003 and habilitation degree in computer science from the Łódź University of Technology, Poland, in 2013. Currently, he is an associate professor at the Department of Electrical Engineering, Czestochowa University of Technology. His research interests include pattern recognition, machine learning, artificial intelligence, and their application in classification, regression, forecasting and optimization problems especially in the field of power systems.