Similarity-based Approaches to Short-term Load Forecasting

Grzegorz Dudek Electrical Department, Czestochowa University of Technology, Poland

1 Introduction

Load forecasting plays a key role in the control and scheduling of power systems and is an extremely important tool for energy suppliers, system operators, financial institutions, and other participants in the electric energy generation, transmission, distribution, and markets. The forecasting of hourly load carried out for one hour to seven days ahead is usually referred to as the short-term load forecasting (STLF). Precise short-term 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. Understanding the load behavior as the basic driver of electricity prices has become more important in restructured power markets.

Several factors affect load behavior. These include weather (temperature, wind speed, cloud cover, and humidity), time, demography, economy, electricity prices, geographical conditions, and type and daily habits of consumers.

Many STLF models have been designed. Conventional STLF models use smoothing techniques (e.g. [Christianse, 1971]), regression methods (e.g., [Papalexopoulos & Hesterberg, 1990]), and statistical analysis. Regression methods are usually applied to model the relationship between load consumption and other factors (e.g., weather, day type, and customer class) [Engle et al., 1992]. ARIMA and related models, where the load is modeled by an autoregressive moving average difference equation, are very popular [Gross & Galiana, 1987]. These models are based on the assumption that the data have an internal structure, such as autocorrelation, trend, and seasonal variation.

In recent years, artificial intelligence methods (AI) have been widely applied to STLF [Metaxiotis et al., 2003]. AI methods of forecasting have shown the capability to give better performance in dealing with non-linearity and other difficulties in modeling of the time series. They do not require any complex mathematical formulations or quantitative correlation between inputs and outputs. The AI methods most often used in STLF can be divided as follows: neural networks (multilayer perceptron, radial basis function, Kohonen, counterpropagation, recurrent) (e.g. [Peng et al., 1992; Khotanzad et al., 1997; Dudek, 2000; Osowski et al., 1996], fuzzy systems (e.g., [Miranda & Monteiro, 2000; Kiartzis & Bakirtzis, 2000; and Dudek, 2006b]), and expert systems (e.g., [Rahman & Bhatnager, 1988; Ho et al., 1990]).

Expert systems are heuristic models, which are usually able to consider both quantitative and qualitative factors. A typical approach is to attempt to imitate the reasoning of a human operator. Expert systems acquire knowledge from domain experts and encode it into formal steps of logic (the set of If-Then rules). Neural networks, on the other hand, do not rely on human experience but attempt to learn a functional relationship between system inputs and outputs by themselves. Fuzzy logic models map a set of input variables to a set of output variables. These variables need not be numerical and may be expressed in a natural language. Most commonly, a fuzzy logic model includes mapping of input values to output values using If-Then logic statements.

To overcome some limitations of individual methods, hybrid models such as neural networks combined with fuzzy systems [Kodogiannis & Anagnostakis, 2002; Liang & Cheng, 2003], neural networkfuzzy expert systems [Dash et al., 1995; Tamimi & Egbert, 2000], evolving wavelet-based networks [Li & Fang, 2003], fuzzy logic and ARMAX model [Yang & Huang, 1998] or stochastic optimization methods (genetic algorithm, particle swarm optimization), and neural networks [Erkmen & Ozdogan, 1997; Bashir & El-Hawary, 2007] have been constructed.

New STLF methods are still being created. Some of them are based on machine learning and pattern recognition techniques, for example, regression trees [Dudek, 2004], cluster analysis methods [Dudek, 2006a, Dudek, 2008b], support vector machines (e.g., [Pai & Hong, 2005]), fractal geometry [Dobrzańska, 1991], point function method [Łyp, 2003], canonical distribution of the random vector method (Popławski & Dąsal, 2004), and artificial immune system [Dudek, 2008a].

A survey of load forecasting methods is presented in several studies [Feinberg & Genethliou, 2005; Kodogiannis & Anagnostakis, 2002; Metaxiotis et al., 2003].

This paper presents a class of similarity-based methods (SB) of STLF, which belong to nonparametric regression methods where the regression function is estimated from data using mutual similarities between the data points. The data points in these models represent daily patterns of the load time series. In section 2, the main idea of SB forecasting methods is outlined. Definitions of patterns are described in section 3. Section 4 presents an analysis of the similarities between patterns. The SB methods of STLF, including k-nearest neighbor estimators, fuzzy estimators, and methods using clustering algorithms, are presented in section 5. In section 6, the application examples are described, wherein the SB methods are used to forecast the daily load curve for the next day.

2 The Idea of the Similarity-based Forecasting Methods

The load time series are characterized by annual, weekly, and daily cycles due to changes in industrial activities and climatic conditions. In Figure 1, the load time series of the Polish power system is shown.



Figure 1: The load time series of the Polish power system in the yearly (a) and weekly (b) intervals

The SB methods use analogies between time series sequences with periodicities. A course of a time series 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, which usually contain one period (in the considered STLF problem, the period equals 24 hours). To eliminate weekly and annual variations, the sequence elements are preprocessed to obtain their patterns. The pattern is a vector with components that are functions of real time series elements, that is, hourly loads in this case. The input and output (forecast) patterns are defined as $\mathbf{x} = [x_1 x_2 \dots x_{24}]$ and $\mathbf{y} = [y_1 y_2 \dots y_{24}]$, 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) is constant and equals τ . The SB methods are based on Assumption 1:

If the process pattern \mathbf{x}_a in a period preceding the forecast moment is similar to the pattern \mathbf{x}_b from the history of this process, then the forecast pattern \mathbf{y}_a is similar to the forecast pattern \mathbf{y}_b .

Patterns \mathbf{x}_a , $\mathbf{x}_{b,a}$ and \mathbf{y}_b are determined from the history of the process. Pairs $\mathbf{x}_a - \mathbf{x}_b$ and $\mathbf{y}_a - \mathbf{y}_b$ are defined in the same way and are shifted in time by the same number of series elements (usually this is a multiple of the daily period).

The similarity measures are based on the distance measures (most often Euclidean or Manhattan), correlation measures, or a function similarity measure.

3 Patterns of Time Series Sequences

The way the \mathbf{x} and \mathbf{y} patterns are defined depends on the time series nature (seasonal variations and trends) and the forecast horizon. Functions transforming series elements into patterns should be defined so that patterns can carry most information about the process, and the model quality becomes maximal. Moreover, functions transforming forecast sequences into patterns \mathbf{y} should ensure the possibility of calculating the real forecast of the time series elements.

In the STLF problem, the forecast patterns **y** encode real loads in the following hours of the forecast day $i+\tau$: $\mathbf{L}_{i+\tau} = [L_{i+\tau,1} L_{i+\tau,2} \dots L_{i+\tau,24}]$, and the input patterns **x** map the hourly loads of the day preceding the forecast day $-\mathbf{L}_i = [L_{i,1} L_{i,2} \dots L_{i,24}]$. In general, the input pattern can be defined based on an interval longer than 24 hours, for example, 168 hours. Moreover, the time series elements contained in this interval can be selected to ensure the best quality of the model [Dudek 2006c]. Vectors **y** are encoded using actual process parameters (from the nearest past), which allows the consideration of the current variability of the process and ensures the possibility of decoding.

Some of the pattern definitions $\mathbf{x}_i = [x_{i,1} x_{i,2} \dots x_{i,24}]$ and $\mathbf{y}_i = [y_{i,1} y_{i,2} \dots y_{i,24}]$ are presented below.

$$x_{i,h} = \frac{L_{i,h} - \overline{L}_i}{\sqrt{\sum_{l=1}^{24} (L_{i,l} - \overline{L}_i)^2}} , \qquad (1a)$$

$$y_{i,h} = \frac{L_{i+\tau,h} - L_i}{\sqrt{\sum_{l=1}^{24} (L_{i,l} - \overline{L_i})^2}} , \qquad (1b)$$

$$x_{i,h} = \frac{L_{i,h}}{L'} , \qquad (2a)$$

$$y_{i,h} = \frac{L_{i+\tau,h}}{L''} , \qquad (2b)$$

$$x_{i,h} = L_{i,h} - L'$$
, (3a)

$$y_{i,h} = L_{i+\tau,h}L'' , \qquad (3b)$$

where *i* is the day number; h = 1, 2, ..., 24 is the hour of the day; τ is the forecast horizon; $L_{i,h}$ is the load at hour *h* of day *i*; \overline{L}_i is the mean load of day *i*; $L' \in \{\overline{L}_i, L_{i-\lambda,h}, L_{i,h-1}\}$ where λ usually equals 1 or 7; and $L'' \in \{\overline{L}_i, L_{i+\tau-\lambda,h}\}$ where $\lambda \ge \tau$.

Definition (1a) expresses the normalization of the vectors \mathbf{L}_i . After normalization, they obtain unity length, zero mean, and the same variance. When we used the standard deviation of the vector \mathbf{L}_i components in the denominator of equation (1a), we received vector \mathbf{x}_i with the unity variance and zero mean.

The components of the patterns defined using Equations (2a) and (3a) express, respectively, the indices and differences of loads in the following hours of the *i*-th day with respect to the mean daily load \overline{L}_i , loads of the preceding day $L_{i-1,h}$, loads of the day from a week ago $L_{i-7,h}$, or loads of the preceding hour $L_{i,h-1}$.

Forecast patterns (1b), (2b), and (3b) are analogous to input patterns (1a), (2a), and (3a), respectively. However, they are encoded using the actual loads determined from the process history, which enables the decoding of the forecast vector $\mathbf{L}_{i+\tau}$ after the forecast of pattern \mathbf{y} is determined.

Other definitions of the patterns were presented by Dudek [Dudek, 2006 c; Dudek, 2006d].

4 Analysis of the Pattern Similarities

To confirm the correctness of Assumption 1, we will conduct the analysis of dependence on the similarities between patterns \mathbf{x} and \mathbf{y} . Here, similarity is a synonym of closeness with respect to a specified metric. The analysis is performed on the statistical sample consisting of the pattern pairs:

$$((\mathbf{x}_i, \mathbf{y}_i), (\mathbf{x}_j, \mathbf{y}_j)), \qquad (4)$$

where i = j = 1, 2, ..., n; $i \neq j$; and *n* is the number of the pattern pair.

The sample size is m = n(n-1). We define the random variables D_x and D_y , which express the distance between the **x** patterns and the distance between **y** patterns in the same pair, respectively. The realization vector of the pairs of these random variables, that is, distances between each pattern pair (4), has the following form:

$$[(d(\mathbf{x}_i, \mathbf{x}_j), d(\mathbf{y}_i, \mathbf{y}_j))] = [(d(\mathbf{x}_1, \mathbf{x}_2), d(\mathbf{y}_1, \mathbf{y}_2)) (d(\mathbf{x}_1, \mathbf{x}_3), d(\mathbf{y}_1, \mathbf{y}_3)) \dots (d(\mathbf{x}_n, \mathbf{x}_{n-1}), d(\mathbf{y}_n, \mathbf{y}_{n-1}))],$$
(5)

where $d(\mathbf{x}_i, \mathbf{x}_j)$ is the distance between patterns \mathbf{x}_i and \mathbf{x}_j ; and $d(\mathbf{y}_i, \mathbf{y}_j)$ is the distance between patterns \mathbf{y}_i and \mathbf{y}_j .

To show the stochastic interdependence of the random variables D_x and D_y , the null hypothesis H_0 is formulated: The observed differences in numbers of occurrence of the sample elements in the specified categories of random variables' D_x and D_y values are caused by random nature of the sample. This hypothesis is verified using the chi-square test creating a contingency table, which shows joint empirical distribution of random variables D_x and D_y . The number of categories of D_x is g and that of D_y is h. The boundaries of the categories for D_x are quantiles of order 0, 1/g, 2/g,..., and 1 and that for D_y are quartiles of order 0, 1/h, 2/h, ..., and 1. The strength of association between D_x and D_y is measured using the Cramer contingency coefficient V and the Pearson correlation coefficient ρ :

$$V = \sqrt{\frac{\chi^2}{m[\min(g,h) - 1]}} , \qquad (6)$$

$$\rho = \frac{\sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} \left(d(\mathbf{x}_{i}, \mathbf{x}_{j}) - \overline{D}_{x} \right) \left(d(\mathbf{y}_{i}, \mathbf{y}_{j}) - \overline{D}_{y} \right)}{(m-1)S_{Dx}S_{Dy}} , \qquad (7)$$

where \overline{D}_x , \overline{D}_y are the mean values of D_x and D_y ; and S_{Dx} , S_D are the standard deviations of D_x and D_y , respectively.

Table 1 contains an example of the values of the statistics mentioned above for the time series of hourly loads of the Polish power system from the period 2002-2006, the forecast horizon $\tau = 1$, the Euclidean and Manhattan distances, and pattern definitions (1), (2) where $L' = L'' = \overline{L_i}$ and (2) where $L' = L'' = L_{i-1,h}$. The significance level of $\alpha = 0.05$, and 49 degrees of freedom (8 categories for D_x and D_y) is assumed. The critical value of χ^2 -statistic is 66.34. The calculated values of the χ^2 test lie in all cases in the critical region, which justifies rejection of the null hypothesis in favor of the alternative hypothesis. The values of V and ρ coefficients indicate significant, moderately strong, and positive correlation between D_x and D_y , which emphasizes for patterns (1). Both metrics gave similar results.

Patterns	Distance	X	V	ρ
(1)	Euclidean	$1.04 \cdot 10^{6}$	0.35	0.67
(1)	Manhattan	$9.88 \cdot 10^5$	0.35	0.65
(2) where $L' = L'' = \overline{L_i}$	Euclidean	$7.25 \cdot 10^{5}$	0.30	0.49
	Manhattan	$6.87 \cdot 10^{5}$	0.29	0.50
(2) where $L' = L'' = L_{i-1,h}$	Euclidean	8.32·10 ⁵	0.32	0.24
	Manhattan	$7.97 \cdot 10^{5}$	0.31	0.25

Table 1: Statistical values of the analyzed time series and sample (4)

The interdependence between D_x and D_y becomes stronger if we analyze the distances only for pattern pairs representing the same days of the week (Monday to Sunday). In this case, the sample has the same form as in (4), but *j* is now the index of patterns representing the same day of the week as pattern *i* (this variant is denoted by V2, while the variant considered earlier is denoted by V1). For the time series analyzed in variant V2, we received the higher value of the Cramer's V (0.45) and a similar value of the correlation coefficient (for patterns (1) and the Euclidean distance).

In the next part, we proved the statistically significant dependence between random variables D_x and D_y , where D_x expresses the distance between pattern \mathbf{x}_i and its K nearest neighbors $\mathbf{x}_{i^*,k}$, k = 1, 2, ..., K, and D_y expresses the distance between pattern \mathbf{y}_i and patterns $\mathbf{y}_{i^*,k}$ associated with patterns $\mathbf{x}_{i^*,k}$. The elements of the sample are the pattern pairs:

$$((\mathbf{x}_i, \mathbf{y}_i), (\mathbf{x}_{i^*,k}, \mathbf{y}_{i^*,k})), \qquad (8)$$

where i = 1, 2, ..., n; k = 1, 2, ..., K; $\mathbf{x}_{i^*,k}$, $\mathbf{y}_{i^*,k}$ are the *k*-th nearest neighbor of pattern \mathbf{x}_i and pattern \mathbf{y} associated with it, respectively.

The sample size is *n* K. The realization vector of pairs of these random variables has the following form:

$$[(d(\mathbf{x}_{i}, \mathbf{x}_{i^{*},k}), d(\mathbf{y}_{i}, \mathbf{y}_{i^{*},k}))] = [(d(\mathbf{x}_{1}, \mathbf{x}_{1^{*},1}), d(\mathbf{y}_{1}, \mathbf{y}_{1^{*},1})) (d(\mathbf{x}_{1}, \mathbf{x}_{1^{*},2}), d(\mathbf{y}_{1}, \mathbf{y}_{1^{*},2})) \dots (d(\mathbf{x}_{n}, \mathbf{x}_{n^{*},K}), d(\mathbf{y}_{n}, \mathbf{y}_{n^{*},K}))].$$
(9)

To show the interdependence of random variables D_x and D_{y_x} the null hypothesis analogous to the null hypothesis formulated above for sample (4), is stated. The analysis was performed on the same time series as that previously for two variants V1 and V2 and for K = 5. Table 2 presents the statistical values in the case of the Euclidean distance is used (results for the Manhattan distance are similar).

Patterns	Variant	X ²	V	ρ	\overline{D}_x	\overline{D}_{v}
(1)	V1	1583	0.20	0.34	0.13	0.53
	V2	2425	0.25	0.62	0.16	0.38
(2) where $L' = L'' = \overline{L_i}$	V1	588	0.12	0.21	0.05	0.19
	V2	1163	0.18	0.53	0.06	0.13
(2) where $L' = L'' = L_{i-1,h}$	V1	353	0.10	0.20	0.06	0.18
	V2	904	0.15	0.37	0.07	0.14

Table 2: Statistical	values for	analyzed time	series and	sample (8)
		2		

Clearly, stronger dependence between random variables occurs in variant V2 for patterns (1). Table 2 shows the mean values of D_x and D_y . In variant V2, lower mean distances between patterns \mathbf{y}_i and $\mathbf{y}_{i^*,k}$ are observed, which translate into lower errors in the SB forecasting models.

5 Pattern Similarity-based Forecasting Methods

The SB methods of the STLF can be divided into several classes. One uses a set of reference pattern pairs (\mathbf{x}, \mathbf{y}) . For a given input pattern \mathbf{x}^* , the most similar patterns \mathbf{x} in the reference set are found, and the forecast pattern $\hat{\mathbf{y}}$ is formed from the patterns \mathbf{y} associated with them [Malko et al., 1996; Dudek 2006b]. In another class, only patterns \mathbf{x} are used. The parameters of the function transforming real time series sequences into patterns are forecast as well [Osowski et al., 1996]. The model presented in Dudek [2006a], which uses only patterns \mathbf{y} , avoids the forecasting of these parameters.

The original patterns are used as the reference patterns, or the new patterns are defined by clustering the original patterns. Examples of models using the original patterns are the k-nearest neighbor method [Dudek, 2009b] or a method based on the fuzzy sets [Dudek, 2006b]. Dudek used different methods for pattern clustering, for example, hierarchical clustering, the self organizing Kohonen map and the neural gas method, the counter propagation neural network, the radial basis neural network, the k-means method, and the artificial immune system. Clustering allows the reduction of the number of reference patterns, decreases the data noise influence, and improves generalization.

The SB forecasting methods are based on the nonparametric approach to regression function estimation. Flexibility of nonparametric models is very useful in the preliminary analysis of a dataset and may be helpful in the construction of parametric models, which are usually more convenient to use but not necessarily more precise. The general model of nonparametric regression is in the following form:

$$y = m(x) + \varepsilon , \qquad (10)$$

where y is the response variable; x is the predictor; ε is the error, which is assumed to be normally and independently distributed with a zero mean and constant variance; and m(x) = E(Y | X = x) is a regression curve.

The aim of nonparametric regression is to estimate the function m(.). Most methods implicitly assume that this function is smooth and continuous. The most popular nonparametric regression models are the kernel estimators, k-nearest neighbor estimators, orthogonal series estimators, and spline smoothing [Härdle, 1994].

Selected SB methods of the STLF are described below.

5.1 STLF Model based on k-Nearest Neighbor Estimators

The k-nearest neighbor estimator (k-NN) of m(.) is a weighted average in a varying neighborhood defined by the x-variables, which are among the k-nearest neighbors of x. In this case, the estimator of the regression function has the following form (Härdle 1994):

$$\hat{m}(x) = n^{-1} \sum_{i=1}^{n} w_i(x) y_i \quad , \tag{11}$$

where $w_i(x)$ are the weights defined as follows:

$$w_{i}(x) = \begin{cases} n/k & \text{if } i \in \Omega_{k}(x) \\ 0 & \text{otherwise} \end{cases},$$
(12)

where $\Omega_k(x)$ is the set of indexes of *k* observations nearest to *x*.

In the STLF problem described above, vectors \mathbf{x} are the predictors, and \mathbf{y} are the response variables. In this case, the definition of estimator m(.) is expanded by introducing the possibility of weight differentiation:

$$m(\mathbf{x}^*) = \frac{\sum_{i \in \Omega_k(\mathbf{x}^*)} W_i(\mathbf{x}^*) \mathbf{y}_i}{\sum_{i \in \Omega_k(\mathbf{x}^*)} W_i(\mathbf{x}^*)} , \qquad (13)$$

where \mathbf{x}^* is the input pattern associated with forecast pattern $\hat{\mathbf{y}}$.

The weights depend on the distance between pattern \mathbf{x}^* and the patterns from its nearest neighborhood in the following way:

$$w_i(\mathbf{x}^*) = 1 - (1 - p) \frac{d(\mathbf{x}^*, \mathbf{x}_i)}{d(\mathbf{x}^*, \mathbf{x}^k)} , \qquad (14)$$

$$w_i(\mathbf{x}^*) = (p-1)\sin\left(\arccos\left(\frac{d(\mathbf{x}^*, \mathbf{x}_i)}{d(\mathbf{x}^*, \mathbf{x}^k)} - 1\right)\right) + 1, \qquad (15)$$

$$w_i(\mathbf{x}^*) = (1-p)\sin\left(\arccos\left(\frac{d(\mathbf{x}^*, \mathbf{x}_i)}{d(\mathbf{x}^*, \mathbf{x}^*)}\right)\right) + p \quad , \tag{16}$$

where $i \in \Omega_k(\mathbf{x}^*)$; \mathbf{x}^k is *k*-th nearest neighbor of pattern \mathbf{x}^* in the reference set including pairs (\mathbf{x}, \mathbf{y}) from the history of the process; and $p \in (0, 1)$ is the parameter controlling the weight differentiation level, as if p = 0 the weights are most differentiated, if p = 1 the weights are equal, and equation (13) is reduced to (11).

Figure 2 shows how the weights change depending on their definition, distance $d(\mathbf{x}^*, \mathbf{x}_i)$, and the value of parameter *p*.



Figure 2: Values of the weights of the different definitions ((14) - solid lines, (15) - dashed lines, (16) - dotted lines) and two values of parameter p (p = 0 - black, p = 0.75 - gray)

In another approach the weights depend on the rank $r \in (1, k)$ of the pattern in the nearest neighbor ranking rather than on the distance between patterns:

$$w_i(\mathbf{x}) = k - r + 1 \ . \tag{17}$$

The number of nearest neighbor k plays the role of a smoothing parameter, which controls the smoothness of the estimated curve. Usually, k depends on the number of observations. A large value of k decreases the variance of the model but increases its bias.

The procedure of forecasting using the k-NN estimator runs in the following steps:

- 1. Preprocessing-preparation of the patterns x and y
- 2. For input pattern \mathbf{x}^* representing the load pattern of the day type *s* (Monday to Sunday), the *k* nearest neighbors in the reference set representing the same day type *s* are found.
- 3. The weight values of the nearest neighbors are calculated using (14) (17).
- 4. Estimator (13) is calculated.
- 5. The forecast of the daily load curve $\hat{\mathbf{L}}$ is calculated from Equation (1b), (2b), or (3b).

The reference set composed of pairs (\mathbf{x}, \mathbf{y}) often contains untypical and noised patterns. To reduce the influence of these patterns on the forecast, each pair (\mathbf{x}, \mathbf{y}) is marked with a degree of confidence, expressing the representativeness of this pair. If the pair (\mathbf{x}, \mathbf{y}) contains a pattern that is untypical, the confidence degree has a lower value than those when the pair contains typical patterns. These degrees of confidence are used for the determination of the forecast. Modified Equation (13) has the following form:

$$m(\mathbf{x}^*) = \frac{\sum_{i \in \Omega_k(\mathbf{x}^*)} w_i(\mathbf{x}) v_i(\mathbf{x}, \mathbf{y}) y_i}{\sum_{i \in \Omega_k(\mathbf{x}^*)} w_i(\mathbf{x}) v_i(\mathbf{x}, \mathbf{y})} , \qquad (18)$$

where $v_i(\mathbf{x}, \mathbf{y}) \in (0, 1)$ is the degree of confidence assigned to the reference pattern pair $(\mathbf{x}_i, \mathbf{y}_i)$.

The patterns with a higher confidence degree now have bigger influence on the forecast pattern forming. Two methods are proposed in Dudek (2006c and 2009b) to estimate the confidence degrees: one is based on the correlation coefficients between the distances of patterns \mathbf{x} and \mathbf{y} , and the other is based on the forecast errors determined on the training set.

5.2 STLF Model based on Fuzzy Estimators

Estimator m(.) can be defined in an alternative way using fuzzy sets:

$$m(\mathbf{x}^*) = \frac{\sum_{i=1}^{n} \mu(\mathbf{x}^*, \mathbf{x}_i) \mathbf{y}_i}{\sum_{i=1}^{n} \mu(\mathbf{x}^*, \mathbf{x}_i)} , \qquad (19)$$

where *n* is the number of patterns **x** representing the same day type *s* as pattern \mathbf{x}^* in the reference set, and $\mu(\mathbf{x}^*, \mathbf{x}_i)$ is the membership function of patterns **x** to the neighborhood of pattern \mathbf{x}^* depending on their mutual distance.

The membership function is defined as follows:

$$\mu(\mathbf{x}^*, \mathbf{x}_i) = \exp\left[-\left(\frac{d(\mathbf{x}^*, \mathbf{x}_i)}{\sigma}\right)^{\alpha}\right],$$
(20)

$$\mu(\mathbf{x}^*, \mathbf{x}_i) = \left[1 + \left(\frac{d(\mathbf{x}^*, \mathbf{x}_i)}{\sigma}\right)^{\alpha}\right]^{-1}, \qquad (21)$$

$$\mu(\mathbf{x}^*, \mathbf{x}_i) = \begin{cases} 1 - \left(\frac{d(\mathbf{x}^*, \mathbf{x}_i)}{r}\right)^{\alpha} & \text{for } d(\mathbf{x}^*, \mathbf{x}_i) < r \\ 0 & \text{for } d(\mathbf{x}^*, \mathbf{x}_i) \ge r \end{cases},$$
(22)

where σ and r are the width parameters, and α is the shape parameter.

Patterns \mathbf{y}_i associated with patterns \mathbf{x}_i closer to pattern \mathbf{x}^* have bigger influence on the forming of response $\hat{\mathbf{y}}$ calculated as the mean of patterns \mathbf{y}_i weighted by the membership degrees $\mu(\mathbf{x}^*, \mathbf{x}_i)$.

The membership function (20) is a Gaussian function with the centre at point \mathbf{x}^* . An alternative Cauchy membership function (21) is characterized by fatter tails than the Gaussian function, which results in bigger influence of more distant patterns on the forecast. The parameter *r* in (22) defines the neighborhood radius. In this case, only these patterns \mathbf{y} take part in forecast forming, which are associated with patterns \mathbf{x}_i lying inside the hypersphere of radius *r* and the centre at point \mathbf{x}^* . If $\alpha = 1$, function (22) is linear.

The membership degrees and estimator m(.) can be calculated using the equations derived based on the fuzzy version of the k-means method [Bezdek, 1981]:

$$\mu(\mathbf{x}^*, \mathbf{x}_i) = \left[\sum_{j=1}^{N} \left(\frac{d(\mathbf{x}^*, \mathbf{x}_j)}{d(\mathbf{x}^*, \mathbf{x}_j)}\right)^{\frac{2}{q-1}}\right]^{-1},$$
(23)

$$m(\mathbf{x}^*) = \frac{\sum_{i=1}^{N} \mu^q(\mathbf{x}^*, \mathbf{x}_i) \mathbf{y}_i}{\sum_{i=1}^{N} \mu^q(\mathbf{x}^*, \mathbf{x}_i)} , \qquad (24)$$

where q > 1 is the parameter determining the fuzziness degree, and $d(\mathbf{x}^*, \mathbf{x}_i) > 0$.

Membership function (23) depends not only on the distance between \mathbf{x}_i and \mathbf{x}^* but also on the distances of the remaining points in the reference set from point \mathbf{x}^* and on the reference set size.

Example plots of the membership functions are presented in Figure 3.



Figure 3: Example plots of the membership functions (a) - (20), (b) - (21), (c) - (22), and (d) - (23)

5.3 STLF Models using Clustering of Patterns

The goal of clustering is to distinguish groups of the time series sequence patterns representing common properties. The group characteristics, represented by centroids, memorize information about the features

of the load patterns and can be useful in the forecasting procedures, as shown below. Three forecasting procedures are proposed.

5.3.1 Forecasting Procedure FP1

In the first forecasting procedure, the patterns \mathbf{x}_i and the corresponding forecast patterns \mathbf{y}_i are concatenated and form patterns $\mathbf{v}_i = [x_{i,1} x_{i,2} \dots x_{i,24} y_{i,1} y_{i,2} \dots y_{i,24}]$. During the forecasting of the load curve for the day type *s* (Monday to Sunday), the subset of patterns is grouped, with the y-parts as the forecast patterns of the s-type days and the x-parts as the patterns of the preceding days (e.g., only the v-patterns containing the x-pattern of the Saturday and the y-pattern of the Sunday are grouped for forecasting the Sunday load curve).

After the clustering phase, each cluster *C* is represented by the centroid **m**, mapping the patterns assigned to this cluster. Similar to the v-patterns, the centroids **m** have two parts: $\mathbf{m}_{\mathbf{x}}$ corresponding to patterns **x** and $\mathbf{m}_{\mathbf{y}}$ corresponding to patterns **y**.

During the forecasting phase, the input pattern \mathbf{v}^* only has the first part \mathbf{x}^* . This pattern is assigned to the group represented by the closest (to the x-part) centroid \mathbf{m}^* . The second part of the centroid \mathbf{m}^* is the estimator $m(\mathbf{x}^*)$.

5.3.2 Forecasting Procedure FP2

During the forecasting of the load curve for the s-type day in the second procedure, the patterns **x** and **y** are grouped independently into *K* and *L* clusters, respectively, where **y** are the patterns of s-type days and **x** are the patterns of the preceding days. Two populations of clusters represented by centroids $\mathbf{m}_{\mathbf{x}}$ and $\mathbf{m}_{\mathbf{y}}$ are created. After grouping, the empirical conditional probabilities $P(C_{\mathbf{y},l}|C_{\mathbf{x},k})$ that associate the forecast pattern \mathbf{y}_i to cluster $C_{\mathbf{y},l}$ when the corresponding \mathbf{x}_i pattern is associated to cluster $C_{\mathbf{x},k}$, are determined. These probabilities are calculated for each pair of clusters based on the grouping of the training set. The forecast of the pattern $\hat{\mathbf{y}} = [\hat{y}_1 \ \hat{y}_2 \dots \hat{y}_{24}]$ paired with the input pattern \mathbf{x}^* , which is assigned to group $C_{\mathbf{x},k^*}$, is calculated as the mean of centroids $\mathbf{m}_{\mathbf{y}}$ weighted by the conditional probabilities:

$$m(\mathbf{x}^*) = \frac{\sum_{l=1}^{L} P(C_{\mathbf{y},l} \mid C_{\mathbf{x},k^*}) \mathbf{m}_{\mathbf{y},l}}{\sum_{l=1}^{L} P(C_{\mathbf{y},l} \mid C_{\mathbf{x},k^*})} \quad (25)$$

Centroids $\mathbf{m}_{\mathbf{y}}$ have a stronger influence on the forecast forming of these clusters, for which the occurrence of the cluster $C_{\mathbf{x},k^*}$ in the training set is more frequent.

This approach was inspired by a paper [Lendasse et al., 1998], where the Kohonen net was used as a clustering method.

5.3.2 Forecasting Procedure FP3

In this procedure, only y-patterns are defined. The patterns are grouped into *K* clusters. The cluster's C_k characteristic is the centroid $\mathbf{m}_k = [m_{k,1} \ m_{k,2} \ \dots \ m_{k,24}]$. After the clusterization phase, the clusters are labeled. The label contains information about the days represented by patterns belonging to the cluster: the day types *s* and the day numbers of the year *u*. To forecast the pattern representing day type s^* and number u^* , all labels are searched. For each cluster C_k , the number of entries l_k is memorized, which satisfies the conditions $s = s^*$ and $u \in \Omega$, where Ω is an interval containing the numbers of *g* successive days before and *g* successive days after the day u^* , including u^* .

The estimator m(.) is calculated as the mean of centroids \mathbf{m}_k weighted by the numbers of entries l_k :

$$m(\mathbf{x}^*) = \frac{\sum_{k=1}^{K} l_k \mathbf{m}_k}{\sum_{k=1}^{K} l_k} \quad (26)$$

This approach can be applied without pattern clustering. To forecast a pattern representing day type s^* and number u^* , the patterns that represent day type s^* and belong to the interval Ω are then chosen from the reference set. The forecast is calculated by averaging these patterns.

The neural gas algorithm is described below as an example of pattern clustering.

5.3.3 STLF Model using Neural Gas Algorithm (Dudek 2009a)

The neural gas algorithm was introduced in a study [Martinetz et al., 1993] as an unsupervised learning method based on idea of the Kohonen's self organizing map. The neural gas consists of a set of M units (neurons) $C = (c_1, c_2, ..., c_M)$ with associated weight vectors (called reference or codebook vectors) $\mathbf{w}_i \in \mathbb{R}^n$, indicating their position in the input space. In the training process, the weight vectors are iteratively updated. For a given input point (\mathbf{v} , \mathbf{x} , or \mathbf{y} pattern depending on the forecasting model), the neurons are sorted according to their distance from this point. All neurons are moved towards the input point depending on their position r_i (rank) in the sorted list according to the following:

$$\mathbf{w}_{i}(k+1) \leftarrow \mathbf{w}_{i}(k) + \eta(k)G(i,k)[\mathbf{z} - \mathbf{w}_{i}(k)] , \qquad (27)$$

where $k = 1, 2, ..., k_{\text{max}}$ is the iteration number; **z** is the input point; and $\eta(k)$ is the learning rate calculated as follows:

$$\eta(k) = \eta_{\max} \left(\frac{\eta_{\min}}{\eta_{\max}} \right)^{\frac{k}{k_{\max}}} .$$
(28)

G(k) is the neighborhood function defined as follows:

$$G(i,k) = e^{\frac{r_i}{\lambda(k)}} , \qquad (29)$$

where $r_i = 0, 1, ..., M$ -1 is the rank of the *i*-th unit, and $\lambda(k)$ is the positive number:

$$\lambda(k) = \lambda_{\max} \left(\frac{\lambda_{\min}}{\lambda_{\max}}\right)^{\frac{k}{k_{\max}}} .$$
(30)

 η and λ decrease after every training step to ensure convergence. η_{max} and λ_{max} are their initial values, while η_{min} and λ_{min} are their final values.

The learning algorithm of the neural gas used in the forecasting procedures can be summarized as follows:

- Initialize set *C* that contains *M* units c_i with the weight vectors w_i located in the centre of the training points (v, x or y depending on the forecasting procedure). Initialize the iteration counter k = 0 and the algorithm parameters η_{max}, λ_{max}, η_{min}, λ_{min}, and k_{max}.
- 2. For each training point, repeat steps 3 and 4.
- 3. Order all elements of set *C* according to the distance of their weight vectors \mathbf{w}_i to the input point and give them the ranks r_i .

- 4. Adapt the weight vectors according to (1).
- 5. Increase the iteration number k and calculate the new values of η , λ , and G according to (28), (30), and (29), respectively.
- 6. If $k < k_{\text{max}}$ and the adaptation of the weight vectors is significant (bigger than the threshold), continue from step 2.

For the time dependent parameters, suitable initial and final values have to be chosen. In the experimental part of this work, the values suggested in [Martinetz et al., 1993] were adopted.

6 Application Examples

The SB forecasting models described above were tested on the time series presented in Table 3. Usually, the smaller the power system, the more irregular and harder it is to forecast the load time series. The measure of the load time series regularity can be the forecast error (MAPE, which is traditionally used in STLF models) determined by the naïve method. The forecast rule in this case is as follows: the load curve of the day of forecast is the same as seven days ago. The mean forecast errors, calculated according to this naïve rule, are presented in the last row of Table 4.

Data symbol	Data description
А	Time series of the hourly loads of the Polish power system from the period 2002-
	2006, the mean load of the system ~16 GW
В	Time series of the hourly loads of the Polish power system from the period 1997-
	2000, the mean load of the system ~15,5 GW
С	Time series of the hourly loads of the local power system from the period July 2001-
	January 2003, the mean load of the system ~1,2 GW
D	Time series of the hourly loads of the local power system from the period June
	1998-July 2002, the mean load of the system ~300 MW
Е	Time series of the hourly load demands of the chemical plant from the period 1999-
	2001, the mean load demand of the plant ~80 MW

Table 3: Description of the data used in experiments

Each dataset was divided into two subsets: training and test. The first sequences of the time series (typically two-thirds of the whole time series) were included in the training set, and the latest sequences were included in the test set. Usually, the model parameters are determined on the training sets in the leave-one-out procedure. Sometimes the parameter values are set arbitrarily, for example, in the k-NN model where, according to Silverman (1986), it is assumed $k = N^{0.5}$. The Euclidean distance is used as a similarity measure. The patterns were defined using (1a) and (1b) or (2a) and (2b), where $L' = \overline{L_i}$ (these definitions of the patterns produced the most accurate models in the preliminary tests).

Table 4 presents the results of the forecasting for some SB models. For comparison, the forecasts using the simple multilayer perceptron were calculated. This neural network contained of only one linear neuron and was trained using the Bayesian regularization. For each hour of the day, a separate net (24 inputs and 1 output) was created and trained. This simple net structure is one of the best compared with other structures tested in [Dudek, 2006c] because of its good generalization properties.

Forecasting procedure/		Ti	me series		
Algorithm	Α	В	С	D	Ε
k-NN estimator	1,84	2,14	4,21	3,70	8,30
Fuzzy estimator	1,76	2,14	4,08	3,63	8,24
FP1/k-means	1,99	2,39	4,62	4,16	9,45
FP1/Fuzzy c-means	2,18	2,49	4,49	4,23	9,16
FP1/Hierarchical clustering	2,44	2,52	4,99	4,63	9,43
FP1/Kohonen net	2,66	2,71	5,31	4,82	9,42
FP1/SOM	2,11	2,39	4,41	4,28	8,84
FP1/Neural gas	1,94	2,55	5,12	3,98	9,18
FP1/Artificial immune system	1,88	2,29	4,46	4,00	8,60
FP2/k-means	2,03	2,37	4,36	4,08	8,94
FP2/Fuzzy c-means	2,39	2,47	4,41	4,17	9,07
FP2/Hierarchical clustering	2,40	2,41	4,46	4,21	10,10
FP2/Kohonen net	2,21	2,50	4,31	4,63	8,86
FP2/SOM	2,06	2,50	4,36	4,32	8,90
FP2/Neural gas	2,24	2,49	4,72	4,09	8,96
FP3/without clustering	2,29	2,53	4,25	4,73	10,24
Multilayer perceptron	2,02	2,24	4,89	3,71	8,32
Naïve method	4,25	4,38	6,59	7,45	17,46

Table 4: Results (MAPE) of STLF using the SB methods (best results in italics)

The ranking of the SB methods is shown in Figure 4, where the horizontal axis represents the mean percentage difference between the MAPE of the model and the best model.

The highest accuracy for all time series shows the model based on fuzzy estimators. The analysis of this model has been the subject of recent studies by Dudek [Dudek, 2006b; Dudek, 2006c; Dudek, 2007]. Among the membership functions described in section 5.2, the best results are achieved for the Gaussian and Cauchy functions. Assuming the constant value of the exponent in the Gaussian function ($\alpha = 2$) and the optimization of only one parameter (σ), only a minor deterioration of performance is caused: 2.0% on the training set and 0.6% on the test set on average. The popular distance measure between patterns, the Manhattan and the Euclidean distances, ensures the best results. The membership function shape does not seem to have a significant influence on the model performance provided that the function is monotonically decreasing and has unbounded support.

The model sensitivity to the membership function width is limited. A 20% change of σ causes an insignificant change in the MAPE: 1.0% on the training set and 2.3% on the test set on average.

The model sensitivity to incomplete input information, that is, when load vector \mathbf{L}^* from where pattern \mathbf{x}^* is created, has indefinite values of *l* components, is also limited. Due to the lack of *l* component values, the distance $d(\mathbf{x}_t, \mathbf{x}^*)$ is calculated without these components. It affects the distance distributions and consequently the values of membership degrees and the forecast pattern $\hat{\mathbf{y}}$. Experiments show that errors increase nonlinearly along with the number of missing components. For small values of *l*, only the insignificant increase in errors is observed. The mean relative increase of the MAPE for the input vector incomplete in 50% (*l* = 12) is 9.8% on the training set and 5.5% on the test set. This leads to an important

conclusion: the model does not require the estimation of the missing component values of the load vectors if their number is not big.

The sensitivity to data noise depends on the width of the membership function, and it is bigger for data with a higher degree of regularity and a stronger relationship between patterns, which are modeled by narrow membership functions.

Information about a forecast variable can be contained in an interval longer than 24 hours preceding the forecast moment. To check this, the input patterns are extended by adding components from the week interval (168 hours preceding the forecast moment). Not all input pattern components carry relevant information about the response variable. Moreover, the same information can be included in several components. To select relevant components, two deterministic and two stochastic methods are used [Dudek 2006c]: the forward feature selection and backward feature selection methods [Devijver & Kittler 1982], and the genetic algorithm and the (μ , λ) evolution strategy [Michalewicz 1994], respectively. In the last method, each pattern component has a weight considered in the distance calculation. These weights are optimized to minimize the MAPE. The results of the component selection are ambiguous. Although the errors decreased on the training set, the error on the test set did not always diminish. Similarly, the enhancement of the model by adding the degree of confidence to the reference patterns did not produce the expected improvement of the test set results.



Figure 4: Ranking of the SB methods

6 Conclusions

The SB methods of the STLF are based on the assumption that if the patterns of the time series sequences are similar to each other, then the patterns of sequences following them are similar to each other as well. This means that the patterns of the neighboring sequences stay in a certain relation, which does not change significantly in time. The more stable this relation, the more accurate the forecasts are. If for a given time series the statistical analysis confirms the hypothesis that the dependence between similarities of input patterns and the similarities between forecast patterns associated with them are not caused by a

random character of the sample, the sense of building and using models based on the similarities of patterns of this time series is justified.

The similarity of patterns y, which determines forecast errors, is bigger when the nearest neighbors are searched only among patterns representing the same day of the week as the input pattern. It results from the bigger stability of relationship between input and forecast patterns. This relation can be shaped by proper pattern definitions and strengthened by the elimination of outliers.

The SB forecasting methods are characterized by simplicity. The best model based on fuzzy estimators in the sufficiently accurate version has only one parameter—the width of the membership function. This parameter is easy to estimate, and the model sensitivity to its deviation from the optimal value is limited. Models with a fewer number of parameters have better generalization properties.

The disadvantage of the SB methods is their limited extrapolation capability. However, most models, for example, neural networks, have problems with extrapolation. Another problem is how to input additional information to the model that is not homogeneous with time series elements (loads), for example, wheatear factors.

References

- [Bashir & El-Hawary, 2007] Bashir, Z.A., El-Hawary, M.E. (2007). Short-Term Load Forecasting Using Artificial Neural Network Based on Particle Swarm Optimization Algorithm. In *Canadian Conference on Electrical and Computer Engineering* CCECE 2007, 272–275.
- [Bezdek, 1981] Bezdek, J.C. (1981). Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York.
- [Christianse, 1971]Christianse, W. (1971). Short Term Load Forecasting Using General Exponential Smoothing, IEEE Transactions on PAS, 900–910.
- [Dash et al, 1995] Dash, P.K., Liew, A.C., Rahman, S., Dash, S. (1995). Fuzzy and Neuro-Fuzzy Computing Models for Electric Load Forecasting. *Engineering Applications of Artificial Intelligence*, vol. 8, no. 4, 423–433.
- [Devijver & Kittler, 1982] Devijver, P.A., Kittler, J. (1982) Pattern Recognition: A Statistical Approach, Prentice-Hall, London.
- [Dobrzańska, 1991] Dobrzańska, I. (1991). Hausdorff Dimension as a Tool of Step Prediction. *Badania Operacyjne i Decyzje*, vol. 2, 31–62 (in Polish).
- [Dudek, 2006] Dudek G. (2006). Data Preprocessing in Similarity-Based Methods of Short-Term Load Forecasting. *Przegląd Elektrotechniczny*, no. 9, 15–19 (in Polish).
- [Dudek, 2009a] Dudek, G. (2009). A Comparison of the Neural Gas and Self Organizing Map Methods for Next Day Load Curve Forecasting. *Przegląd Elektrotechniczny*, vol. 85, no. 3, 153–156.
- [Dudek, 2006a] Dudek, G. (2006). Application of Hierarchical Clustering Methods to the Daily Electrical Load Profile Forecasting. *Przegląd Elektrotechniczny*, vol. 82, no. 9, 9–11 (in Polish).
- [Dudek, 1997] Dudek, G. (1997). Application of the Hecht-Nielsen Neural Network to Short Term Load Forecasting. In 8th International Conference on Present-Day Problems of Power Engineering APE'97, vol. 4, 65–72 (in Polish).
- [Dudek, 2008a] Dudek, G. (2008) Artificial Immune System for Short-Term Electric Load Forecasting. In Rutkowski, L., Tadeusiewicz, R., Zadeh, L., Zurada, J. (eds) Lecture Notes in Artificial Intelligence – Proceedings of the 9th International Conference on Artificial Intelligence and Soft Computing ICAISC 2008, vol. 5097, Springer, 1007–1017.
- [Dudek, 2008b] Dudek, G. (2008) Next Day Electric Load Curve Forecasting using k-Means Clustering. In *Proceedings of the* 4th Polish and International PD Forum-Conference on Computer Science PD FCCS'2008 (in print).
- [Dudek, 2004] Dudek, G. (2004) Regression Tree as a Forecasting Tool. In National Conference on Forecasting in Power Engineering PE'04, 99–105, Częstochowa (in Polish).

- [Dudek, 2000] Dudek, G. (2000) Short Term Load Forecasting using Radial Basis Neural Networks. In 5th National Conference on Forecasting in Power Engineering PE'00, 59–68, Częstochowa, Poland (in Polish).
- [Dudek, 2006b] Dudek, G. (2006) Short-Term Electrical Load Forecasting using Fuzzy Clustering Method. Przegląd Elektrotechniczny, vol. 9, 26–28 (in Polish).
- [Dudek, 2006c] Dudek, G. (2006) Short-Term Load Forecasting using Fuzzy Clustering and Genetic Algorithms. *Final report of the Polish State Committee for Scientific Research founded grant no. 3T10B02329*. Dept. Elect. Eng., Częstochowa University of Technology, (unpublished, in Polish).
- [Dudek, 2009b] Dudek, G. (2009) Short-Term Load Forecasting using Nearest Neighbor Estimators. In 14th International Conference on Present-Day Problems of Power Engineering APE '09, vol. 3, 31–39 (in Polish).
- [Dudek, 2007] Dudek, G. (2007) The Analysis of the Short-Term Power System Load Forecasting Model based on the Fuzzy Clustering. *Badania Operacyjne i Decyzje*, no. 2, 15–34 (in Polish).
- [Engle et al., 1992] Engle, R.F., Mustafa, C., Rice, J. (1992). Modeling Peak Electricity Demand. Journal of Forecasting, vol. 11, 241–251.
- [Erkmen & Ozdogan, 1997] Erkmen, I., Ozdogan, A. (1997). Short Term Load Forecasting using Genetically Optimized Neural Network Cascaded with a Modified Kohonen Clustering Process. Proceedings of the 1997 IEEE International Symposium on Intelligent Control, 107–112.
- [Feinberg & Genethliou, 2005] Feinberg, E.A., Genethliou, D. (2005). Load Forecasting. In Chow, J.H., Wu, F.F., Momoh, J. (eds) Applied Mathematics for Restructured Electric Power Systems, 269–285, Springer US.
- [Gross & Galiana, 1987] Gross, G., Galiana, F.D. (1987). Short-Term Load Forecasting. *Proceedings of the IEEE*, vol. 75, no. 12, 1558–1573.
- [Härdle, 1994] Härdle, W. (1994). Applied Nonparametric Regression. Cambridge University Press.
- [Ho et al., 1990] Ho, K.-L., Hsu, Y.-Y., Chen, C.-F., Lee, T.-E., Liang, C.-C., Lai, T.-S., Chen, K.-K. (1990). Short-Term Load Forecasting of Taiwan Power System using a Knowledge Based Expert System. *IEEE Transactions on Power Systems*, vol. 5, 1214–1221.
- [Khotanzad et al., 1997] Khotanzad, A., Afkhami-Rohani, R., Lu, T.L., Davis, M.H., Abaye, A., Maratukulam, D. (1997). ANNSTLF – A Neural Network Based Electric Load Forecasting System. *IEEE Transactions on Neural Networks*, vol. 8, no. 4, 835–846.
- [Kiartzis & Bakirtzis, 2000] Kiartzis, S.J., Bakirtzis, A. G. A Fuzzy Expert System for Peak Load Forecasting. Application to the Greek Power System. In 10th Mediterranean Electrotechnical Conference MELECON 2000, 1097 –1100.
- [Kodogiannis & Anagnostakis, 2002] Kodogiannis, V.S., Anagnostakis, E.M. (2002). Soft Computing Based Techniques for Short-Term Load Forecasting. *Fuzzy Sets and Systems*, vol. 128, 413–426.
- [Lendasse et al., 1998] Lendasse, A., Verleysen, M., de Bodt, E., Cottrell, M., Gregoire, P. (1998). Forecasting Time-Series by Kohonen Classification. Proceedings of the European Symposium on Artificial Neural Networks, 221–226, Bruges, Belgium.
- [Li & Fang, 2003] Li, Y., Fang, T. (2003). Wavelet and Support Vector Machines for Short–Term Electrical Load Forecasting. Proceedings of the International Conference on Wavelet Analysis and its Applications, 399–404.
- [Liang & Cheng, 2002] Liang, R.-H., Cheng, C.-C. (2002). Short-Term Load Forecasting by a Neuro-Fuzzy Based Approach. Electrical Power and Energy Systems, vol. 24, no. 2, 103–111.
- [Łyp, 2003] Łyp, J. Methodology of the Electric Load Analysis and Forecasting for the Local Power Systems. (2003) Ph.D. Dissertation, Dept. Elect. Eng., Częstochowa University of Technology (in Polish).
- [Malko et al., 1996] Malko, J., Mikołajczyk, H., Skorupski, W. (1996). Progress in Employment of Neural Network for Power System Load Forecasting. In 2nd Conference on Neural Networks and Their Applications, vol. 1, 317–323.
- [Martinetz et al., 1993] Martinetz, T.M., Berkovich, S.G., Schulten, K.J. (1993) "Neural-Gas" Network for Vector Quantization and its Application to Time-Series Prediction. *IEEE Transactions on Neural Networks*, vol. 4, no. 4, 558–569.

- [Metaxiotis et al, 2003] Metaxiotis, K., Kagiannas, A., Askounis, D., Psarras, J. (2003). Artificial Intelligence in Short Term Load Forecasting: A State-of-the-Art Survey for the Researcher. *Energy Conversion and Management*, vol. 44, 1525–1534.
- [Michalewicz, 1994] Michalewicz, Z. (1994). Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag, New York.
- [Miranda & Monteiro, 2000] Miranda, V., Monteiro, C. (2000). Fuzzy Inference in Spatial Load Forecasting. *Power Engineering Society Winter Meeting*, IEEE, vol. 2, 1063–1068.
- [Osowski et al., 1996] Osowski, S., Siwek, K., Kądzielawa, A. (1996). Neural Network Approach to Load Forecasting. In 2nd Conference on Neural Networks and Their Applications, 355–360, Częstochowa, Poland.
- [Pai & Hong, 2005] Pai, P.-F., Hong, W.-C. (2005). Support Vector Machines with Simulated Annealing Algorithms in Electricity Load Forecasting. *Energy Conversion and Management*, vol. 46, 2669–2688.
- [Papalexopoulos & Hesterberg, 1990] Papalexopoulos, A.D., Hesterberg, T.C. (1990). A Regression Based Approach to Short Term Load Forecasting. *IEEE Transactions on Power Systems*, vol. 5, 1535–1547.
- [Peng et al., 1992] Peng, M., Hubele, N.F., Karady, G.G. (1992). Advancement in the Application of Neural Networks for Short-Term Load Forecasting. *IEEE Transactions on Power Systems*, vol. 7, 250–257.
- [Popławski & Dąsal, 2004] Popławski, T., Dąsal, K. (2004). Canonical Distribution in Short-Term Load Forecasting. In Szkutnik, J., Kolcun, M. (eds.) *Technical and Economic Aspect of Modern Technology Transfer in Context of Integration with European Union*, 147–153, Mercury-Smekal Pub House, Kosice, Slovak Republic.
- [Rahman & Bhatnager, 1988] Rahman, S., Bhatnager, R.. (1988). An Expert System Based Algorithm for Short-Term Load Forecast. *IEEE Transactions on Power Systems*, vol. 3, 392–399.
- [Silverman, 1986] Silverman, B.W. (1986). Density Estimation for Statistics and Data Analysis. Chapman & Hall.
- [Tammi & Egbert, 2000] Tamimi, M., Egbert, R. (2000). Short Term Electric Load Forecasting via Fuzzy Neural Collaboration. Electric Power System Research, vol. 56, 243–248.
- [Yang & Huang, 1998] Yang, H.T., Huang C.M. (1998). A New Short-Term Load Forecasting Approach using Self-Organizing Fuzzy ARMAX Models. *IEEE Transactions on Power Systems*, vol. 13, 217–225.