Int. J. Appl. Math. Comput. Sci., 2009, Vol. 19, No. 2, 303–315
DOI: 10.2478/v10006-009-0026-2

amcs

ENSEMBLE NEURAL NETWORK APPROACH FOR ACCURATE LOAD FORECASTING IN A POWER SYSTEM

KRZYSZTOF SIWEK [*], STANISŁAW OSOWSKI [*,**], RYSZARD SZUPILUK [***]

[*] Institute of the Theory of Electrical Engineering, Measurements and Information Systems
Warsaw University of Technology, pl. Politechniki 1, 00–661 Warsaw, Poland
e-mail: sto@iem.pw.edu.pl

[**] Institute of Electronic Systems
Military University of Technology, ul. Kaliskiego 21, 00–908 Warsaw, Poland

[***] Department of Business Informatics
Higher School of Economics, Al. Niepodległości 123, 02–554 Warsaw, Poland
e-mail: szupiluk@iem.pw.edu.pl

The paper presents an improved method for 1–24 hours load forecasting in the power system, integrating and combining different neural forecasting results by an ensemble system. We will integrate the results of partial predictions made by three solutions, out of which one relies on a multilayer perceptron and two others on self-organizing networks of the competitive type. As the expert system we will apply different integration methods: simple averaging, SVD based weighted averaging, principal component analysis and blind source separation. The results of numerical experiments, concerning forecasting the hourly load for the next 24 hours of the Polish power system, will be presented and discussed. We will compare the performance of different ensemble methods on the basis of the mean absolute percentage error, mean squared error and maximum percentage error. They show a significant improvement of the proposed ensemble method in comparison to the individual results of prediction. The comparison of our work with the results of other papers for the same data proves the superiority of our approach.

Keywords: neural networks, blind source separation, ensemble of predictors, load forecasting.

1. Introduction

The prediction of the 1–24 hours ahead load demand plays an important role in the economy of power system generation and distribution (Mandal *et al*., 2006). Thanks to the precise forecasting for any hour of the day, we can deliver really needed power and in this way reduce the cost of the energy. Many prediction methods exist and are used in practice. The most common are linear methods based on autoregressive models of time series, such as AutoRegressive with eXogenous input (ARX), AutoRegressive Moving Average with eXogenous input (ARMAX) or AutoRegressive Integrated Moving Average (ARIMA) (Gonzalez-Romera *et al*., 2006; Hippert *et al*., 2001; Ljung, 1999). More advanced approaches apply nonlinear models based mainly on artificial neural networks, such as the MultiLayer Perceptron (MLP), Radial Basis Function (RBF) networks, the Support Vector Machine (SVM) or self-organizing neural networks of the Kohonen type (Cottrell *et al*., 1995; Fidalgo and Pecas Lopez, 2005; Kandil *et al*., 2006; Lendasse *et al*., 2002; Osowski and Siwek, 2002; Yalcinoz and Eminoglu, 2005). Neural networks are nonlinear structures, capable of taking into account more complex relations existing among the analyzed data. Thanks to this feature, they are able to generate more accurate prediction.

These methods differ by the particularity of the network structure and the basic nonlinear functions used in prediction, the principle of operation, the way of learning, and rely their prediction ability on different aspects of the processing of the learning data. For example, the application of a Kohonen network exploits the concept of the clusterization of similar data points (Cottrell *et al*., 1995; Osowski and Siwek, 1999), while the MLP, RBF and the SVM take into account the universal approximation ability of these networks (Osowski, 2006; Yalci-

noz and Eminoglu, 2005). On the other hand, the MLP network performs global approximation, while the RBF network is a typical local approximation tool. The ways of adapting the parameters of all these forms of solution differ also significantly (Haykin, 2002; Osowski, 2006).

The most often used approach is to train many networks and then take the one, which guarantees the best reproduction results on the data not taking part in learning (verification data). A more general approach is to take into account all partial prediction results, combine them into one ensemble system of presumably better quality and treat the combined output as the final forecast (Haykin, 2002; Kuntcheva, 2004). This paper develops and investigates the latter philosophy. Instead of discarding less fortunate prediction results, we analyze them and take the results of such analysis into account during the preparation of the final forecasting. The ensemble of neural predictors is composed of several individual neural networks. The prediction data generated by each predictor of the ensemble are combined together to form one forecasted power pattern for 24 hours ahead. We will investigate here different integration methods: simple and weighted averaging, Principal Component Analysis (PCA) decomposition and Blind Source Separation (BSS) (Cichocki *et al*., 2009; Haykin, 2002; Osowski, 2006). The numerical results of all these integration schemes will be presented and discussed.

2. Integration methods of prediction

The general ensemble system of forecasting applies many individual predictors and combines them into one final forecasting system. Let us assume that there are M individual predictive channels combined into one forecasting system by the integrating part of the network as shown in Fig. 1.

We assume that each of the M predictive neural networks generates a 24 hours load pattern on the basis of the information delivered by the user to the input of each predictor. The output signals of each individual predictor form the vectors $\mathbf{x}_i$ $(i = 1, 2, \ldots, M)$ of the same size (24 components corresponding to 24 hours of the day). These vectors are combined in the integrating unit to form one

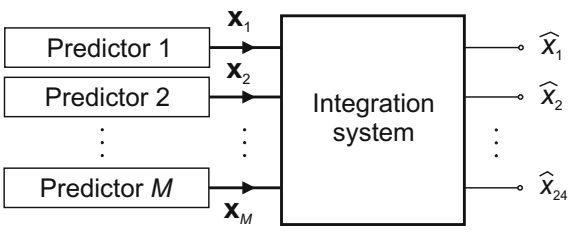


Fig. 1. Ensemble of predictive networks for integrated forecasting.

output vector $\hat{\mathbf{x}} = [\hat{x_1}, \hat{x_2}, \ldots, \hat{x}_{24}]^T$ of the final forecasting.

Various integration methods may be applied in practice. In this paper, we will check and compare methods belonging to three main groups. One is based on the averaging of the results of individual predictors, the second on the application of PCA to the data created by the combined predictors, and the third one on the application of independent component analysis of the time series formed by the individual predictors.

2.1. Integration based on averaging. In this approach the final forecast is defined as the average of the results produced by all different predictors. Two kinds of averaging techniques are used. The simplest one is the ordinary mean of the partial results. In such a case the final prediction vector $\hat{\mathbf{x}}$ for the particular 24 hours load pattern is defined as

$$\hat{\mathbf{x}} = \frac{1}{M} \sum_{i=1}^{M} \mathbf{x}_i. \tag{1}$$

This formula makes use of the stochastic distribution of predictive errors. The process of averaging reduces the final error of forecasting. It works quite well if all predictive networks are of comparable accuracy. If it is not true, the final results may be inferior with respect to the best individual predictor. In such a case better results may be obtained by applying weighted averaging, that is, by taking the summation of terms in (1) with different weights following from the estimated accuracy of each predictor. This accuracy may be measured on the basis of particular predictor performance on the data from the past. The most reliable predictor should be considered with the highest weight, and the least accurate one with the least attention. The forecasted load for the j-th hour of the day can be now defined in the following form:

$$\hat{x}_j = \sum_{i=1}^{M} w_j^{(i)} x_j(i), \tag{2}$$

where the upper index means the i-th neural network. The weights are adjusted individually for each hour and should be adapted in a way to provide the best result of forecasting. The easiest way to determine the values of the weights ($i = 1, 2, \ldots, M$ and $j = 1, 2, \ldots, 24$) is to solve the set of linear equations for each hour of the day corresponding to the learning data. The equations written for $j = 1, 2, \ldots, 24$ and all p training data may be presented in the following matrix form:

$$\mathbf{X}_j \mathbf{w}_j = \mathbf{d}_j, \tag{3}$$

where

$$\mathbf{X}_j = \begin{bmatrix} x_j^{(1)}(1) & x_j^{(2)}(1) & \dots & x_j^{(M)}(1) \\ x_j^{(1)}(2) & x_j^{(2)}(2) & \dots & x_j^{(M)}(2) \\ \dots & \dots & \dots & \dots \\ x_j^{(1)}(p) & x_j^{(2)}(p) & \dots & x_j^{(M)}(p) \end{bmatrix},$$

$$\mathbf{w}_j = \begin{bmatrix} w_j^{(1)} \\ w_j^{(2)} \\ \dots \\ w_j^{(M)} \end{bmatrix}, \quad \mathbf{d}_j = \begin{bmatrix} d_j(1) \\ d_j(2) \\ \dots \\ d_j(p) \end{bmatrix}$$

in which $x_j^{(i)}(k)$ is the load predicted by the i-th neural network for the j-th hour at the k-th learning data and $d_j(k)$ is the destination value (the accurate load) at the j-th hour for the k-th learning data. The solution of the above equation is straightforward by applying singular value decomposition of the data matrix $\mathbf{X}_j$ and pseudoinverse (Golub and Van Loan, 1991),

$$\mathbf{w}_j = \mathbf{X}_j^{+}\mathbf{d}_j. \tag{4}$$

After solving such equations for $j = 1, 2, \dots, 24$, we get all coefficients forming the weighting vectors $\mathbf{w}_i$ used for the final forecasting (Eqn. (2)).

2.2. Integration based on PCA. In this solution the weighted voting of the individual predictors is substituted with linear transformation of the data provided by PCA. PCA represents a classical statistical technique for analyzing the covariance structure of multivariate statistical observations, enhancing the most important elements of information (Diamantras and Kung, 1996). Assuming that an N-dimensional input vector $\mathbf{x}$ is transformed into a K-dimensional output vector $\mathbf{z}$, PCA is defined as follows:

$$\mathbf{z} = \mathbf{W}\mathbf{x}, \tag{5}$$

where $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_K]^T$ is the transformation matrix formed by K eigenvectors of the covariance matrix $\mathbf{R}_x$ associated with K largest eigevalues. The reduced size vector $\mathbf{z}$ is composed of K principal components, beginning from the most important, z_1, and ending on the least importance component, z_K. The reconstruction of the original vector $\mathbf{x}$, denoted here by $\hat{\mathbf{x}}$, on the basis of principal components and the orthogonal transformation matrix $\mathbf{W}$ is described by the relation

$$\hat{\mathbf{x}} = \mathbf{W}^T\mathbf{z}. \tag{6}$$

The reconstructed vector is deprived of the least important information associated with the reduced eigenvalues of the covariance matrix $\mathbf{R}_x$. The cut information usually corresponds to the noise contaminating the measurements.

In our implementation of PCA we form the $\mathbf{x}$ vectors as a combination of the predicted 24 hours loads (vectors $\mathbf{x}_i$) generated by individual neural predictors. The number of such combined vectors is equal to p, i.e., the number of learning data. The size of the vector $\mathbf{x}$ is thus equal to $n_x = 24 \times M$. In the learning phase of PCA, we form the covariance matrix $\mathbf{R}_x$ of p data vectors generated by M predictors. The size of $\mathbf{R}_x$ is equal to $n_x \times n_x$. Then we perform eigenvalue analysis of this matrix. As a result, we get n_x eigenvalues λ_i and the same number of the eigenvectors $\mathbf{w}_i$ associated with them. We arrange the eigenvalues in decreasing order. The K eigenvectors associated with largest eigenvalues of this matrix ($K < n_x$) form the PCA transformation matrix $\mathbf{W}$. At this stage, the system is ready for on-line operation. The actual vectors $\mathbf{x}$ formed by M neural predictors are first transformed into a reduced size vector $\mathbf{z}$ by using (5) and then reconstructed back by applying the relation (6).

After this double transformation, the vector $\hat{\mathbf{x}}$ is deprived of the least important components (the noise existing in the original vector $\mathbf{x}$) and thanks to this it may establish a more accurate forecast of the load pattern. PCA acts here like a lowpass filter eliminating the noise. The result of this transformation contains filtered versions of individual prediction vectors $\hat{\mathbf{x}}$ for $i = 1, 2, \dots, M$. The final forecast for 24 hours can be calculated using a simple averaging procedure of the results, corresponding to each predicted pattern contained in the extended vector $\hat{\mathbf{x}}$.

2.3. Integration based on BSS.

2.3.1. Principle of the method. The approach based on blind source separation (Cichocki *et al.*, 2009; Cichocki and Amari, 2003) integrates different methods of prognosis into one forecasting system by combining the results of BSS of time series. The results of prognosis (vectors $\mathbf{x}_i$) generated by each predictive network for the period used in training create time series that are put in parallel to the BSS system. The number of inputs to BSS is equal to the number M of the prognosis networks applied. The BSS system decomposes the original stream of signals of length q, forming a matrix $\mathbf{X} \subset \mathbb{R}^{M \times q}$ (q is the number of prognosis hours used in learning, $q = 24p$), into independent components using a matrix $W \subset \mathbb{R}^{M \times M}$.

The independent component signals, generated by BSS, form a matrix $\mathbf{Y}$ of M rows and q columns. This is the linear transformation described by $\mathbf{Y} = \mathbf{W}\mathbf{X}$. Each row of the matrix $\mathbf{X}$ represents independent component series. Some of these series represent essential information and some represent noise. Reconstructing the original time series back into a real prognosis on the basis of essential independent components will only provide a prognosis deprived of the noise, that is, of presumably better quality. The problem is that we do not know in advance which component is the noise and which represents the useful

information. It is possible to solve the problem by trying in reconstruction all combinations of independent components and accepting the one which provides the best results of prediction on the learning data. Another approach is to find out which component is of the noisy character using statistical tests (Nikias and Petropulu, 1993), for example, correlation analysis. The signals of noisy channels are then replaced with zeros in the reconstruction phase.

The reconstruction of the original data matrix $\mathbf{X}$ is done by using the inverse operation, called deflation (Cichocki *et al*., 2009; Cichocki and Amari, 2003),

$$\hat{\mathbf{X}} = \mathbf{W}^{-1}\hat{\mathbf{Y}}. \tag{7}$$

In this equation $\hat{\mathbf{X}}$ denotes the reconstructed time series matrix and $\hat{\mathbf{Y}}$ is the independent component matrix built out of the original matrix $\mathbf{Y}$ by zeroing row or rows corresponding to noise. In recovering signals we may try all sensible combinations of independent components, substituting the rejected components (appropriate rows of $\mathbf{Y}$) with zeros. The combination corresponding to the best result of prediction on the learning data is assumed as the final solution. In the reconstruction phase on the testing data, only this combination will be used.

Figure 2 presents a graphical illustration of the proposed method. The input signals $x_i(k)$ for BSS (block of the matrix $\mathbf{W}$) are formed from the stream of data for the particular hour k generated by the predictors. The switches in the figure represent possible elimination of the appropriate independent components at the reconstruction stage of the data.

At the learning stage, the stream of each channel is formed by the components of the vectors $\mathbf{x}_i$, i.e., the output signals of neural predictors for succeeding days combined together. In the retrieval mode, they are the components of 24 hours load patterns predicted by each predictor. Note that the BSS method, very popular in many signal processing problems, has been proposed in the load forecasting for the first time here.

2.3.2. Blind source separation algorithm. The blind source separation system decomposes the streams of input signals of M channels into M independent components. The basic assumption is that the input signals are mixtures of some unknown basic original independent sources which are to be recovered by the separation algorithm. There are many different solutions developed for BSS (Belouchrani *et al*., 1997; Choi *et al*., 2002; Cichocki *et al*., 2009; Cichocki and Amari, 2003). We have tried some of them installed actually in ICALAB (Cichocki *et al*., 2009), and the best results have been obtained during the application of the second order blind identification algorithm (Choi *et al*., 2002). SOBI applies two covariance matrices: $\mathbf{R}_x(0)$, where $\mathbf{R}_x(0) = 1/p\sum_{k=1}^{p}\mathbf{x}(k)\mathbf{x}^T(k)$, and $\mathbf{R}_x(L) = 1/p\sum_{k=1}^{p}\mathbf{x}(k)\mathbf{x}^T(k-L)$ for some selected value of L (for example, $L = 1$ or 2). We perform the eigenvalue decomposition for $\mathbf{R}_x(0)$,

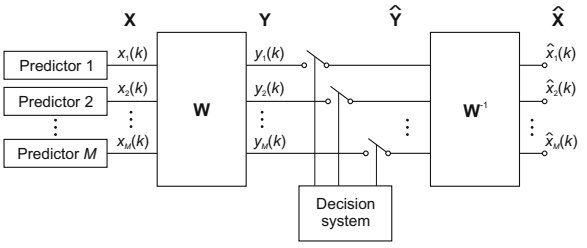


Fig. 2. General scheme of the proposed solution.

$$\mathbf{R}_x(0) = \mathbf{U}_x\mathbf{D}_x\mathbf{U}_x^T. \tag{8}$$

The standard whitening of the vectors $\mathbf{x}_i$ is realized by applying the linear transformation (Cichocki and Amari, 2003),

$$\overline{\mathbf{x}}_i = \mathbf{Q}_x\mathbf{x}_i = \mathbf{D}_x^{-1/2}\mathbf{U}_x^T\mathbf{x}_i, \tag{9}$$

where $\mathbf{Q}_x = \mathbf{D}_x^{-1/2}\mathbf{U}_x^T$. On the basis of this, we define two covariance matrices: one for the vectors $\bar{\mathbf{x}}_i$ and the other for the same vectors with the assumed shift L,

$$\mathbf{R}_{\bar{x}}(0) = \frac{1}{p}\sum_{k=1}^{p}\bar{\mathbf{x}}(k)\bar{\mathbf{x}}^T(k) = \mathbf{Q}_x\mathbf{R}_x(0)\mathbf{Q}_x^T, \tag{10}$$

$$\mathbf{R}_{\bar{x}}(L) = \frac{1}{p}\sum_{k=1}^{p}\bar{\mathbf{x}}(k)\bar{\mathbf{x}}^T(k-L) = \mathbf{Q}_x\mathbf{R}_x(L)\mathbf{Q}_x^T. \tag{11}$$

An orthogonal eigenvalue transformation is then applied to diagonalize the matrix $\mathbf{R}_{\bar{x}}(L)$. It takes the form

$$\mathbf{R}_{\bar{x}}(L) = \mathbf{U}_{\bar{x}}\mathbf{D}_{\bar{x}}\mathbf{U}_{\bar{x}}^T. \tag{12}$$

The demixing matrix $\mathbf{W}$ is then described as (Cichocki and Amari, 2003)

$$\mathbf{W} = \mathbf{U}_{\bar{x}}^T\mathbf{Q}_x. \tag{13}$$

The estimated independent sources for each k-th hour of the day for $k = 1, 2, \ldots, 24$ are described by the relation

$$\mathbf{y}(k) = \mathbf{U}_{\bar{x}}^T\mathbf{Q}_x\mathbf{x}(k), \tag{14}$$

where $\mathbf{x}(k) = [x_1(k), x_2(k), \ldots, x_M(k)]$ is the vector formed by the k-th components of the vectors generated by individual predictors (k means a particular hour). The mixed signals can be reconstructed back using the relation

$$\hat{\mathbf{x}}(k) = \mathbf{W}^{+}\mathbf{y} = \mathbf{Q}_x^{+}\mathbf{U}_{\bar{x}}\mathbf{y}(k), \tag{15}$$

where + means the pseudoinverse (Golub and Van Loan, 1991). Observe that for the number of mixed signals equal to the number of independent components, it is a simple matrix inversion.

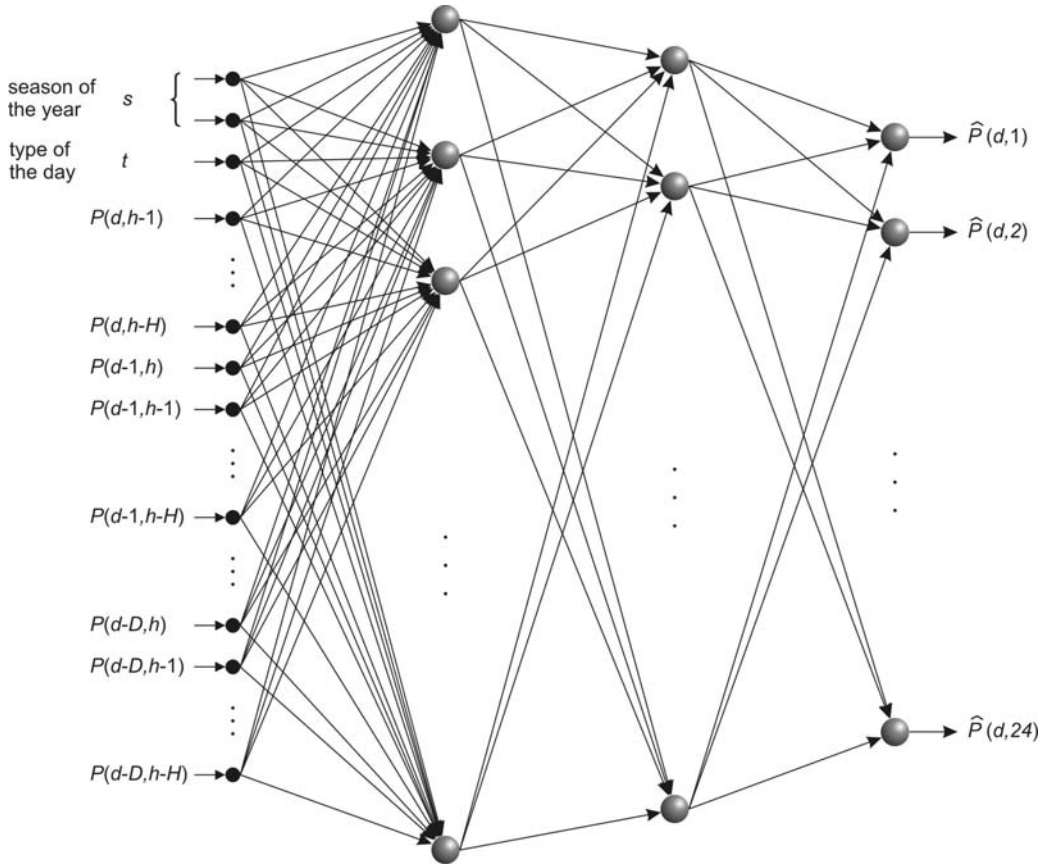


Fig. 3. MLP structure for forecasting the 24-hour load pattern.

3. Individual neural predictors

To obtain accurate results of forecasting, we have to apply individual predictors of superior performance, since the better the results of partial predictions, the better the expected forecasting accuracy of the whole system. There are many different methods used currently for the prediction of the time series of the power demand. To the most well-known belonged in the past linear prediction methods, like ARX or ARMAX (Ljung, 1999). We have tried ARMAX for the data of the Polish power system related to the last four years. However, the results were not encouraging. The average prediction error calculated for the whole year ranged from 5% to 6%, depending on a particular year.

Actually, neural network based predictors are regarded as best. Supervised learning (Afkhami-Rohani and Maratukulam, 1998; Kiartzis *et al*., 1997; Fidalgo and Pecas Lopez, 2005; Osowski and Siwek, 2002) as well as competitive self-organizing (Cottrell *et al*., 1995; Osowski and Siwek, 1999) networks are most widely known. In our work we have used three neural prediction methods. One is based on the MLP and applies the most popular supervised learning. The other two rely on the application of the hybrid approach exploiting self-organizing networks working either in a crisp or a fuzzy mode.

3.1. Multilayer perceptron based method. The first predictor type makes use of the universal approximation ability of the MLP network (Haykin, 2002; Osowski, 2006). To represent the generally unknown function of the next day load pattern, it maps the past loads of the system into the present forecasted load at the d-th day and the the h-th hour. Our general MLP model of the load is assumed here in the following mathematical form (Osowski and Siwek, 2002):

$$\hat{P}(d,h) = f(\mathbf{w}, t, s, P(d,h-1), \ldots, P(d,h-H), P(d-1,h) \ldots, P(d-D,h-H)), \quad (16)$$

where $\mathbf{w}$ represents the vector of synaptic weights of the network, H and D are the number of past hours and days, respectively, influencing the prediction process, t denotes the type of the day (workday or holiday) and s signifies the season of the year (autumn, winter, spring or summer). The value $\hat{P}(d,h)$ represents the predicted loads and the values $P(d-i,h-j)$ written without hat—the known values of the load from the past. All data samples have been normalized dividing the real load by the mean value of the data base of the Polish power system, formed years, taking part in the experiments. The forecasting model does not take into account temperature, although in general it might have a significant impact on the accuracy.

We have omitted temperature simply, because the prediction is concerned with data corresponding to the territory of the whole country. Temperature changes a lot in different regions of Poland, so it would be difficult to adjust the proper value of temperature for a particular day. However, in the case of forecasting energy consumption for a small region, the inclusion of temperature and its gradient in the model would be beneficial and easy to consider in our model.

The expression (16) may be associated with the MLP network of the particular structure presented in Fig. 3. The neural network architecture of this figure contains a certain number of input nodes. One node is used for binary coding of the type of day (working day or holiday) and two—the season of the year (winter, spring, summer and autumn). Some nodes represent the loads of some past days (up to *D*) and previous hours (up to *H*). All these signals put to the input nodes form the input vector $\mathbf{x}$. The output layer contains 24 linear neurons. Their quantity is equal to the number of hours of prediction (24 hours ahead). The signals of output neurons represent the normalized forecasted 24 hours load pattern.

The MLP network consists of many simple neuron-like processing units of a sigmoidal activation function grouped together in layers. The number of hidden layers and neurons of sigmoidal non-linearity are usually subject to adjustment in an experimental way by training different structures and choosing the smallest one, still satisfying the learning accuracy.

The information put to the input of the network is processed locally in each unit by computing the dot product between the corresponding input vector and the weighting vector of the neuron. Before training, the weights are initialized randomly. Training the network to produce a desired output vector $\mathbf{d}_i$ (load pattern of the next 24 hours) when presented with an input vector $\mathbf{x}_i$ involves systematical changing of the weights of all neurons until the network produces the desired output within a given tolerance. The procedure is repeated over the entire training set. Learning is just reduced to the minimization of the Euclidean error measure over the entire learning set. The most effective learning approach applies gradient information and uses second order optimization algorithms, like Levenberg-Marquard or conjugate gradient ones (Osowski, 2006). The gradient vector in a multilayer network is computed using the backpropagation algorithm.

The important point in designing the optimal network structure is adjusting the length of the input vector $\mathbf{x}$, which depends explicitly on the chosen values of D and H. The structure of the network is treated as optimal if it provides the most accurate prediction for the data not taking part in the learning process. There are some methods for optimal selection of input variables (Drezga and Rahman, 1998; Guyon and Elisseeff, 2003). Such tools include covariance analysis, PCA, projection pursuit, the application of linear SVM feature ranking, etc. In this work we have applied correlation analysis studying the degree of correlation of the learning errors of the MLP with different numbers of D and H. On the basis of numerical simulations, we have found that the optimal number of input nodes in our case is 19, which corresponds to $D = 3$ and $H = 4$.

The optimal number of hidden layers and neurons in these layers was found using the trial and error approach by learning many different structure MLP networks and accepting the one which has provided the least value of the error on the validation data, extracted from the learning data set (20% of the learning pairs). On the basis of these numerical experiments we have found the optimal structure containing two hidden layers of 20 and 15 sigmoidal neurons, respectively. In this way the optimal structure of the MLP network used in the prediction is described as 19-20-15-24. Note finally that we have not distinguished holidays or special days like Christmas or Easter. The 24-elements of the load patterns for these specific days were predicted in the same way as for regular days. This was done just to create more difficult conditions of forecasting.

3.2. Neural predictor based on self-organization. The second type of predictor makes use of self-organization of the learning data (Cottrell *et al*., 1995; Osowski and Siwek, 1999). The main task of the self-organizing network is to learn the characteristics of the daily loads of the system. The days of the same type belonging to the same seasons of the year have similar load characteristics and form clusters, grouping similar data. Each cluster is then represented by one neuron acting in a competitive mode. To make the prediction independent of the general trend, changing from year to year, we transform the input data by cutting out the mean value and dividing the result by the standard deviation of the data for this day. In this way the so-called profile $p(d, h)$ for the d-th day and the h-th hour is defined (Cottrell *et al*., 1995; Nikias and Petropulu, 1993),

$$p(d,h) = \frac{P(d,h) - P_m(d)}{\sigma(d)}, \tag{17}$$

where $P(d, h)$ is the real load of the d-th day at the h-th hour, $P_m(d)$ is the mean value of the load of the d-th day and $\sigma(d)$ is the standard deviation of the load of this day. Observe that in an extreme case the profile value can be negative.

The set of 24 profiles for each hour of the day represent the profile vector $\mathbf{p}(d) = [p(d,1), p(d,2), \ldots, p(d,24)]^T$. These vectors form the training data of the self-organizing network of the Kohonen type. In training such a network we have used the neural gas algorithm (Haykin, 2002; Osowski, 2006).

In practice, we have trained the self-organizing network containing 100 neurons. This number was established on the basis of introductory numerical experiments performed using different numbers of neurons and choosing the one providing the best results of prediction. Figure 4 presents the map of the profile vectors for these 100 neurons trained using the data of the Polish power system from the last three years. Each profile vector represents a 24-hour load pattern of the cluster center (central neuron). It depicts the data closest to its weighting vector in the chosen metric space.

If we want to make the prediction for the 24 hours load pattern of the particular day, we simply take the reversed form of Eqn. (17). The prediction of the load for the d-th day and the h-th hour may be expressed in the form

$$\hat{P}(d,h) = \hat{\sigma}(d)\hat{p}(d,h) + \hat{P}_m(d), \tag{18}$$

where the variables with hat mean predicted values. Successful application of this method needs solving three tasks. One is accurate prediction of the profiles $\hat{p}(d,h)$, the second—good estimation of the mean value $\hat{P}_m(d)$ and the third—the estimation of the standard deviation $\hat{\sigma}_m(d)$ for the particular day under consideration.

The latter two problems have been solved by applying the MLP network. Two separate MLP networks: one for the mean and the second for the standard deviation of the same structure have been designed. To obtain accurate mean (standard deviation) prediction we have taken the input vector to the network composed of eight nodes, representing the mean (standard deviation) of the load for the previous days (up to D days) and of the previous weeks (up to W weeks), the binary code of the actual season of the year and the type of day. In practice we have assumed $W = 1$ and $D = 2$. The destination associated with the output neuron represents the predicted value of the mean (standard deviation) for the d-th day, respectively. The mean and standard deviation values used in the experiments have been normalized by proper linear scaling of the real data to the range $[0,\ 1]$. The general MLP structure for mean load prediction of the d-th day is presented in Fig. 5. The symbol $\hat{P}_m(d,w) = \hat{P}(d)$ means the mean value of the load for the d-th day of the w-th week under consideration. An identical structure has been used for the prediction of the standard deviation for the d-th day of the w-th week.

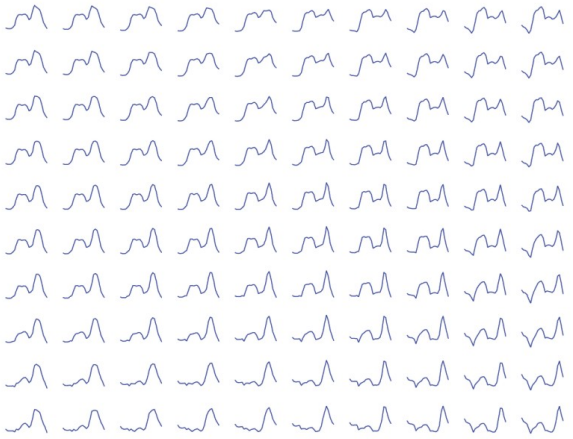


Fig. 4. Map of profile vectors of 100 self-organizing neurons.

The number of hidden neurons has been chosen on the ground of the good generalization ability of the network. It was adjusted after a series of numerical experiments using the validation data. As a result of such experiments the final structure of the MLP network used in the prediction of $\hat{P}_m(d)$ and $\hat{\sigma}(d)$ was 8-12-1.

The profile prediction problem has been solved by us in two different ways. In the Crisp Self-Organization (CSO) approach (Cottrell *et al.*, 1995; Osowski and Siwek, 1999) we estimate the profiles by averaging the winner vectors for this particular day (for example, all Mondays of July) on the past learning data,

$$\hat{\mathbf{p}}(d) = \frac{\sum_{i=1}^{m} k_{di}\mathbf{w}_i}{\sum_{i=1}^{m} k_{di}}. \tag{19}$$

In this expression, k_{di} is the number of appearances of the i-th neuron among the winners in the past for this particular the d-th day and $\mathbf{w}_i$ is the weighting vector of the i-th self-organizing neuron (learned load pattern of the i-th cluster). Only winners have been taken into account in this method.

In the second, slightly different, approach, we have fuzzified the process of the determination of the profile vector. At the prediction stage we take into account not only the winner but also the activity of some losers, closest to the winner. The learning phase is performed in the same way as it was done in the first case. However, as a result of learning, we memorize not only the signal y of the winner, but also of some limited number (say q) of neurons closest to the winner, keeping their relative activities. On the basis of these distributed activities of neurons at the presentation of the input vector to the self-organizing network we define the membership degree of the i-th neuron in the form

$$\mu_i = \frac{y_i}{\sum_{i=1}^{q} y_i}. \tag{20}$$

The highest membership value corresponds to the winner, but q losers take also some nonzero values. The stage of the profile prediction for the d-th day (Osowski and Siwek, 1999) takes into account not only the winners but also their q neighbors and their relative activities

$$\hat{\mathbf{p}}(d) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{q} \mu_i^{(j)}\mathbf{w}_i^{(j)}}{\sum_{i=1}^{m}\sum_{j=1}^{q} \mu_i^{(j)}}. \tag{21}$$

The parameter $\mu_i^{(j)}$ denotes the membership degree of the j-th neuron taking part in prediction of the profile vector

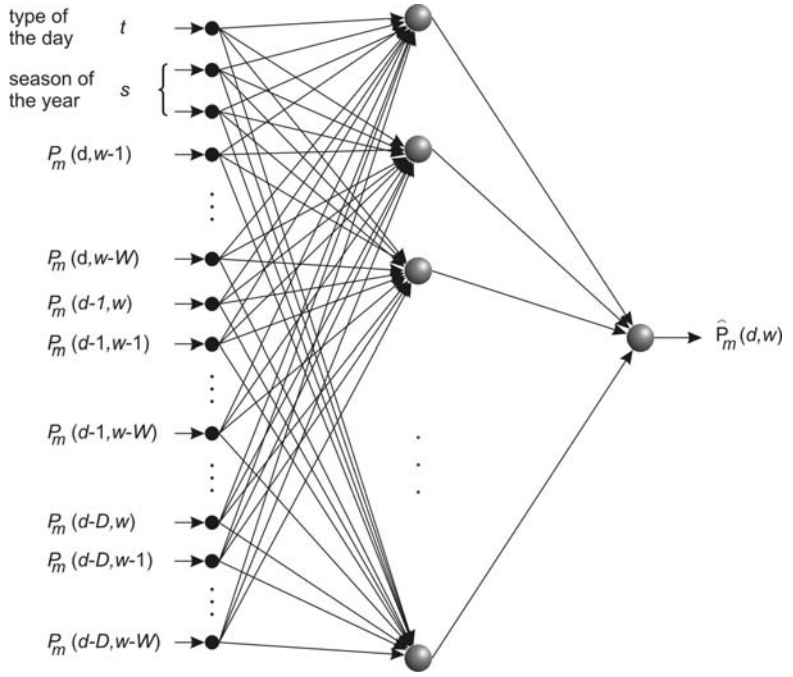


Fig. 5. MLP structure for the mean load of a particular d-th day. An identical structure is used for standard deviation prediction.

for the d-th day. The index i means the notation of particular day profiles of the past taking part in prediction. The variable m is the number of similar days (for example, Mondays of July) from the learning data base of the past. The relations (20) and (21) are of the fuzzy nature, thus the method is called Fuzzy Self-Organization (FSO).

4. Results of numerical experiments

The numerical experiments have been performed for the data of the Polish power system collected over four years. The whole data base has been split into two parts: the learning set containing the data of three years and the testing set composed of data of one year, not taking part in learning. We have used three different neural predictors combined into one ensemble network performing the final forecasting. One is based on the application of the MLP structure and two others on self-organization (CSO and FSO). They have been combined together using different methods of integration, discussed in Section 2. The results have been compared on the basis of the committed errors. There are three most important types of errors from the practical point of view. Let us denote by $P(h)$ and $\hat{P}(h)$ the real and predicted load at the h-th hour, respectively, and by n—the total number of hours of prediction. We have adopted the following definitions of errors:

- the Mean Absolute Percentage Error (MAPE),

$$\text{MAPE} = \frac{1}{n}\sum_{h=1}^{n}\frac{\mid P(h) - \hat{P}(h) \mid}{P(h)} \cdot 100\%; \quad (22)$$

- the Mean Squared Error (MSE),

$$\text{MSE} = \frac{1}{n}\sum_{h=1}^{n}\left[P(h) - \hat{P}(h)\right]^2; \quad (23)$$

- the Normalized Mean Squared Error (NMSE),

$$\text{NMSE} = \frac{\text{MSE}}{(mean(P))^2}, \quad (24)$$

  where $mean(P)$ represents the mean value of the load in the time period of prediction;

- the MAXimum Percentage Error (MAXPE),

$$\text{MAXPE} = \max\left\{\frac{\mid P(h) - \hat{P}(h) \mid}{P(h)} \cdot 100\%\right\}. \quad (25)$$

The errors have been calculated separately for the learning and testing data. Here we will present only the testing errors, related to the data not taking part in learning, since this information is most important from the practical point of view.

Table 1 presents the obtained results of the load forecasting for the last (testing) year in the form of the mean absolute percentage error, maximum error (both in %) and the mean square error measured as the mean of the squared errors (in MW^2). These results represent individual predictors. The superiority of the MLP is evident. Note that both self-organizing approaches are based on the same principle of operation and, as a result, represent similar accuracy.

The first experiments of integration have been performed by applying the averaging principle. The obtained results in the form of three error measures are presented in Table 2. There is an evident improvement of the prediction quality in terms ofthe MAPE and MSE. Both methods of integration deliver similar results, although weighted average is a bit better.

The next experiments are concerned with PCA based integration. Different numbers of principal components have been considered. Table 3 presents the results for three different numbers: 5, 11 and 20 of principal components. The best results correspond to 11 components, and this number was regarded as the optimal one. The results related to the PCA method are similar to the averaging approach in terms of the MAPE, MSE and MAXPE. No significant decrease in errors has been observed in comparison to averaging.

The last form of integration (BSS) needs special signal processing. The learning data streams of the 1–24 hours ahead load forecast, corresponding to MLP, CSO and FSO approaches, have been submitted to the input of the BSS system in the form of three parallel time series. The SOBI algorithm of separation has been applied. As a result, we have got the separation matrix $\mathbf{W}$ and three independent component streams packed in the matrix $\mathbf{Y}$, whose graphical forms for one year period of time are presented in Fig. 6. It looks that only Channel 1 depicts the components of essential information and the rest—some kind of noise or outliers with a very small portion of useful information.

To confirm this statement, we have made some additional experiments and calculated the autocorrelation functions of these three streams of data. The results of these experiments in the form of correlation values for positive (right) and negative (left) delays are presented in Fig. 7. The middle point of the figure represents the delay equal to zero. The correlation function of the noise is very characteristic, since there are only few dominating stripes corresponding to delays close to zero (for purely white noise there is only one high magnitude stripe at zero delay). The distribution of the correlation function of other delays forms the noisy sequence of small value. It is now evident that the signal of Channel 2 may be regarded as white noise. The signal of Channel 3, visible in Fig. 6, is the mixture of white noise and some outliers with (maybe) a very small portion of some useful information. On the other hand, the correlation function of the time series of Channel 1 is very typical for the deterministic signals forming the useful information used in prediction.

Using the obtained decomposition, we may reconstruct three streams of original data (forecasted values) by applying the relation (15) and omitting some basic components (the rows of $\mathbf{Y}$) during the deflation process. Since in a general case (without performing correlation analysis) we do not know which independent stream represents essential information, we may try all possible combinations of them. Along with three independent components there are also three possible solutions corresponding to the application in the reconstruction stage of the combinations of two streams and three solutions after the reconstruction of the data on the basis of one stream only. The combination of streams resulting in the best prediction accuracy of

Table 1. Testing errors of the load forecasting for the Polish power system using individual predictors.

Method	MAPE [%]	MSE [MW^2]	NMSE	MAXPE [%]
MLP	1.98	1.65e5	0.64e-3	16.92
CSO	2.35	2.45e5	0.96e-3	18.10
FSO	2.34	2.43e5	0.95e-3	18.08

Table 2. Results of integration using averaging.

Method	MAPE [%]	MSE [MW^2]	NMSE	MAXPE [%]
Ordinary mean	1.86	1.48e5	0.58e-3	16.98
Weighted average	1.84	1.47e5	0.57e-3	16.97

Table 3. Results of integration using PCA with different numbers of principal components.

Method	MAXPE [%]	MSE [MW^2]	NMSE	MAXPE [%]
PCA (5 comp.)	2.15	1.92e5	0.75e-3	17.08
PCA (11 comp.)	1.81	1.46e5	0.57e-3	16.23
PCA (20 comp.)	1.89	1.52e5	0.59e-3	16.88

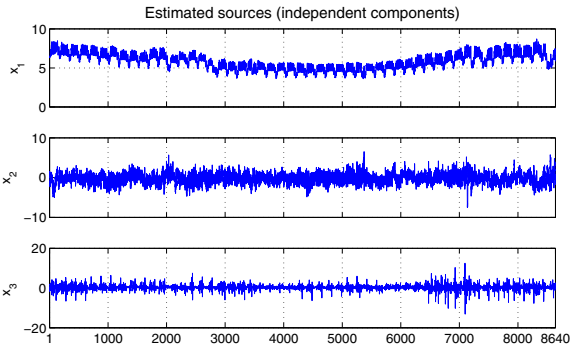


Fig. 6. Independent components of load pattern prognosis for one year data.

the reconstructed time series of the power demand may be assumed as the optimal solution, and this one should be tested on new data (in our case the testing data correspond to the last year, not taking part in learning). For illustrating these phenomena, the results of testing the BSS based forecasting system for all six combinations of independent components at the reconstruction stage are collected in Table 4.

They are presented in the form of the MAPE, MSE, NMSE and MAXPE. As can be seen, the best case (distinguished in bold) corresponds to the case of omitting the component no. 2, which can be treated as evident noise. Note that the relevant information is contained mainly in the first channel. Reconstructing the prognosis on the basis of this channel generates the results only slightly worse than in the best combination 1–3 (the MAPE difference of 0.01% within the tolerance limit). On the other hand, combining two noisy signals of Channel 2 and 3 or taking only individual noisy channels 2 or 3, we reconstruct the forecast of the 24 hours ahead load pattern, which represents the noise only (MA PE above 99%).

From the practical point of view, in our further prognosis we can rely only on the component no. 1, regarding the others as noise. This conclusion is not very surprising if we observe that only three forecasting methods have been applied in the numerical experiments. The decomposition of three mixed signals has resulted in the essential information (Channel 1), the evident noise (Channel 2) and some combination of the noise and outliers visible in the channel no. 3.

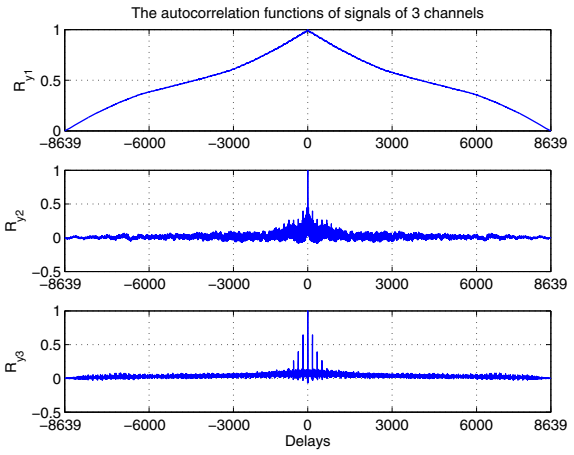


Fig. 7. Distribution of values of the autocorrelation function of separated streams of data (a) Channel 1, (b) Channel 2, (c) Channel 3.

Table 4. Errors of forecasting the power demand for the testing data using the BSS based integration system.

Combination	MAXPE [%]	MSE [MW^2]	NMSE	MAXPE [%]
1-2	1.99	1.67e5	0.65e-3	17.18
1-3	**1.73**	**1.25e5**	**0.49e-3**	**16.21**
2-3	99.87	2.64e8	1.03	110.93
1	1.74	1.26e5	0.49e-3	16.42
2	99.73	2.63e8	1.02	110.05
3	99.97	2.65e8	1.04	100.87

In practice, especially with a high number of the prognosis methods applied, the combination approach presented above may be too tedious and time consuming. In such a case, the best approach is to apply the autocorrelation analysis of all separated signals. The results of this analysis indicate which signals are deterministic and which are stochastic (noise). The noisy signals should be simply zeroed and only the deterministic signals should be combined together in the deflation procedure.

Figure 8 presents the results of experiments in the form of MAPEs for individual predictors and different forms of their combinations (averaging, PCA and BSS). It is evident that the integration improves significantly the accuracy of forecasting. The least efficient is the simple averaging (mean) of the results of all predictors. More powerful is weighted averaging, and the best results correspond to the application of the BSS approach.

Table 5 depicts relative improvements in the best final results (BSS based integration) over individual forecasting methods for the testing data of one year and all three categories of errors. There is a significant improvement in forecasting results in all categories. The highest improvement has been observed for the MSE. If we compare the results with the best predictor (MLP), we note 13.02% for the MAPE, 23.87% of the MSE and 4.24% of the MAXPE. The improvements over other prediction methods are even more impressive (almost 50% improvement in terms of the MSE).

Figure 9 depicts the relative improvement in forecasting results for the MAPE (Fig. 9(a)) and the MAXPE (Fig. 9(b)) in comparison with the best individual predictor (MLP). Also, both self-organizing methods are included in this comparison. It is evident that the integration

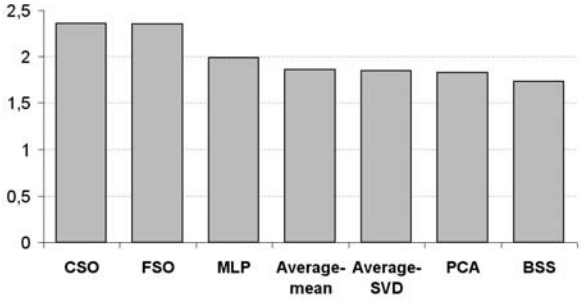


Fig. 8. Summary of MAPE results of power demand forecasting for the testing data not taking part in learning.

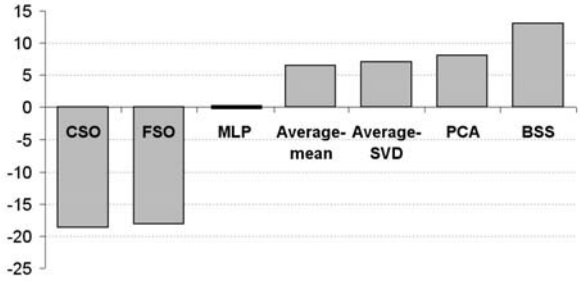
of many predictors, even not equally good, brings an improvement of the final statistical results. In the most extreme case (BSS), the relative improvement with respect to the best individual predictor (MLP) is over 13%. Even the maximum percentage error has been slightly reduced, although this time the improvement is around 4% only.

The next experiment has been performed by omitting one of the self-organizing results. We have taken into account the results of the MLP and FSO since the results of CSO are very close to these of FSO and both methods rely on a similar principle. The observed results are only slightly worse than in the best case (1.75% of the MAPE and 16.36% of the MAXPE). This confirms the fact that the highest potential improvement of results may be expected when independent individual predictors are integrated into one forecasting system.

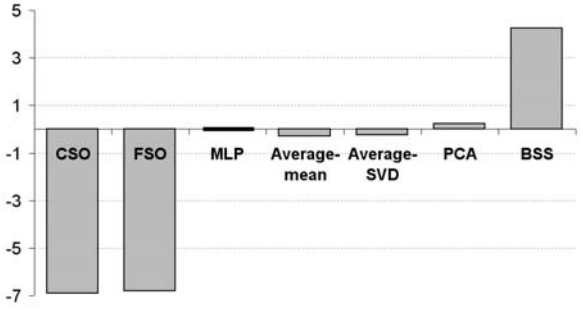
It is interesting to compare the patterns of the load on particular days of the year. Figure 10 depicts two representatives of typical days of the year. Figure 10(a) corresponds to the working days of the chosen week of the year, and Fig. 10(b) — to the weekend days of the same week. Three time series are depicted: the real one, the results corresponding to the best individual predictor, and those related to the best results of the ensemble network (BSS). It is evident that both prognoses follow the real load pattern of the system; however, the integration results trace better the general trend of the load change in comparison to the best individual prognosis made by the MLP.

Table 5. Relative improvement in the BSS method over individual predictors.

Predictor	MAXPE [%]	MSE [MW^2]	MAXPE [%]
MLP	13.02	23.87	4.24
CSO	26.36	48.87	8.38
FSO	25.38	47.27	8.20

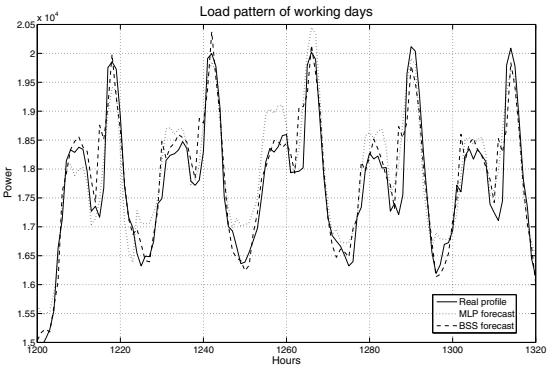


(a)

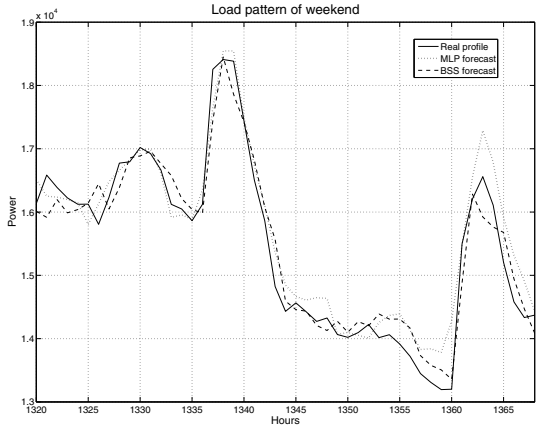


(b)

Fig. 9. Comparison (in %) of different prediction methods with respect to the best individual predictor: (a) MAPE, (b) MAXPE.



(a)



(b)

Fig. 10. Typical load patterns for the working days (a) and the weekend days (b).

5. Conclusions

The paper has presented a neural network ensemble approach to forecasting the 24-hour load pattern of a power system. In this method many different predictors are trained and their results compared to each other. In the classical approach, less fortunate predictors are discarded and the results of the best one are treated as the final results. In the presented approach, we analyze all of them and take the results of such analysis into account during the preparation of the final forecasting by integrating them into the final outcome.

The ensemble of neural predictors is composed of three individual neural networks, although this number may be easily extended to more individual predictors. The prediction data generated by each component of the ensemble are combined together to form one forecasted power pattern for 24 hours ahead. We have tried different methods of integrating the results of individual predictors: simple and weighted averaging, PCA filtering and BSS processing. The best results have been obtained with the application of the BSS method by decomposing the data into streams of statistically independent components and reconstructing the time series omitting the noise.

The experimental results have shown that the performance of individual predictors was improved significantly by the integration of their results. The improvement is observed even during the application of different quality predictors. For the data corresponding to the Polish power system and the application of three different predictors, we have got a 13% relative improvement of the MAPE and more than 23% of the MSE over the best individual predictors (MLP network).

It is interesting to compare the accuracy of our results with the other approaches presented in different papers. The same data of the Polish power system have been predicted in (Lendasse *et al*., 2002; Sorjamaa *et al*., 2007). These papers gave the results only in the form of the normalized mean squared error defined as the real MSE value divided by the square of the mean value. The best resulting NMSE of (Lendasse *et al*., 2002) was equal to NMSE=1.6e–3. In the case of (Sorjamaa *et al*., 2007), the best result of the NMSE was 1.8e–3. Our best result corresponding to the same data was equal to 0.49e–3.

Acknowledgment

This work is supported by the Polish Ministry of Science and Higher Education under a grant for the years 2008–2010.

References

Afkhami-Rohani, K. R. and Maratukulam, D. (1998). ANNSTLF—Artificial neural network short-term load forecaster—Generation three, *IEEE Transactions on Power Systems* **13**(4): 1413–1422.

Belouchrani, A., Abed-Meraim, K., Cardoso, J. and Moulines, E. (1997). A BSS technique using SOS, *IEEE Transactions on Signal Processing* **45**(2): 434–444.

Choi, S., Cichocki, A. and Belouchrani, A. (2002). Second order nonstationary source separation, *Journal of VLSI Signal Processing* **32**(1–2): 93–104.

Cichocki, A. and Amari, S. I. (2003). *Adaptive Blind Signal and Image Processing*, Wiley, New York, NY.

Cichocki, A., Amari, S., Siwek, K. and Tanaka, T. (2009). *ICALAB Toolboxes,* RIKEN, Tokyo, Available at: `http://www.bsp.brain.riken.jp/ICALAB`.

Cottrell, M., Girard, B., Girard, Y., Muller, C. and Rousset, P. (1995). Daily electrical power curve: Classification and forecasting using a Kohonen map, *Proceedings of the International Workshop on Artificial Neutral Networks, IWANN, Malaga, Spain,* pp. 1107–1113.

Diamantras, K. and Kung, S. Y. (1996). *Principal Component Neural Networks*, Wiley, New York, NY.

Drezga, I. and Rahman, S. (1998). Input variable selection for ANN-based short-term load forecasting, *IEEE Transactions on Power Systems* **13**(4): 1238–1244.

Fidalgo, J. N. and Pecas Lopez, J. (2005). Load forecasting performance enhancement when facing anomalous events, *IEEE Transations on Power Systems* **20**(2): 408–415.

Golub, G. and Van Loan, C. (1991). *Matrix Computations*, Academic Press, New York, NY.

Gonzalez-Romera, E., Jaramillo-Moran, M. A. and Carmona-Fernandez, D. (2006). Monthly electric energy demand forecasting based on trend extraction, *IEEE Transations on Power Systems* **21**(4): 1946–1953.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection, *Journal of Machine Learning Research* **3**(3): 1158–1182.

Haykin, S. (2002). *Neural Networks. A Comprehensive Foundation*, Macmillan, New York, NY.

Hippert, H. S., Pedreira, C. E. and Souza, R. C. (2001). Neural networks for short-term load forecasting: A review and evaluation, *IEEE Transations on Power Systems* **16**(1): 44–55.

Kandil, N., Wamkeue, R., Saad, M. and Georges, S. (2006). An efficient approach for short term load forecasting using artificial neural networks, *Electrical Power and Energy Systems* **28**(4): 525–530.

Kiartzis, S. J., Zoumas, C. E., Theocharis, J., Bakirtzis, A. G. and Petridis, V. (1997). Short-term load forecasting in an autonomous power system using artificial neural networks, *IEEE Transations on Power Systems* **12**(4): 1591–1596.

Kuntcheva, L. (2004). *Combining Pattern Classifiers—Methods and Algorithms*, Wiley, New York, NY.

Lendasse, A., Cottrell, M., Wertz, V. and Verleysen, M. (2002). Prediction of electric load using kohonen maps—Application to the Polish electricity consumption, *Proceedings of the American Control Conference, Anchorage, AK, USA,* Vol. 28, pp. 3684–3688.

Ljung, L. (1999). *System Identification—Theory for the User*, PTR Prentice Hall, Upper Saddle River, NJ.

Mandal, P., Senjyu, T., Urasaki, N. and Funabashi, T. (2006). A neural network based several hours ahead electric load forecasting using similar days approach, *Electrical Power and Energy Systems* **28**(3): 367–373.

Nikias, L. and Petropulu, A. P. (1993). *Higher-Order Spectral Analysis—A Nonlinear Signal Processing Framework*, Prentice Hall, Upper Saddle River, NJ.

Osowski, S. (2006). *Neural Networks for Information Processing*, Warsaw University of Technology Press, Warsaw, (in Polish).

Osowski, S. and Siwek, K. (1999). The self-organizing neural network approach to load forecasting in power system, *Proceedings of the International Joint Conference on Neural Networks, Washington, DC, USA,* Vol. 28, pp. 1345–1348.

Osowski, S. and Siwek, K. (2002). Regularization of neural networks for load forecasting in power system, *IEE Proceedings GTD* **149**(3): 340–345.

Sorjamaa, A., Hao, J., Reyhani, N., Li, Y. and Lendasse, A. (2007). Methodology for long-term prediction of time series, *Neurocomputing* **70**(16–18): 2861–2869.

Yalcinoz, T. and Eminoglu, U. (2005). Short term and medium term power distribution load forecasting by neural networks, *Energy Conversion and Management* **46**(8): 1393–1405.

Krzysztof Siwek was born in Poland in 1971. He received the M.Sc. and Ph.D. degrees in electrical engineering from the Warsaw University of Technology, Poland, in 1995 and 2001, respectively. His scientific interest is in neural networks, signal processing and genetic algorithms, and their application to different problems of forecasting. He is an author or co-author of more than 60 scientific papers and two books. He is also a co-author of the ICALAB packages for 1-D and 2-D BSS.

Stanisław Osowski was born in Poland in 1948. He received the M. Sc., Ph.D. and D.Sc. degrees from the Warsaw University of Technology, Poland, in 1972, 1975 and 1981, respectively, all in electrical engineering. Currently he is a professor of electrical engineering at the Institute of the Theory of Electrical Engineering and Electrical Measurements of the Warsaw University of Technology. His research and teaching interests include the areas of neural networks, optimization techniques, and computer-aided circuit analysis and design. He is an author or co-author of more than 200 scientific papers and ten books.

Ryszard Szupiluk was born in Poland in 1971. He received the M.Sc. and Ph.D. degrees in electrical engineering from the Warsaw University of Technology, Poland, in 1995, and 2000, respectively. His scientific interest is in blind source separation, neural networks and signal processing, and their application to different problems of forecasting. He is an author or co-author of more than 40 scientific papers.

Received: 11 September 2008
Revised: 17 January 2009