

Artificial Immune System for Forecasting Time Series with Multiple Seasonal Cycles

Grzegorz Dudek

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

Abstract. Many time series exhibit seasonal variations related to the daily, weekly or annual activity. In this paper a new immune inspired univariate method for forecasting time series with multiple seasonal periods is proposed. This method is based on the patterns of time series seasonal sequences: input ones representing sequences preceding the forecast and forecast ones representing the forecasted sequences. The immune system includes two populations of immune memory cells – antibodies, which recognize both types of patterns represented by antigens. The empirical probabilities that the forecast pattern is detected by the k th antibody from the second population while the corresponding input pattern is detected by the j th antibody from the first population, are computed and applied to the forecast construction. The empirical study of the model including sensitivity analysis to changes in parameter values and the robustness to noisy and missing data is performed. The suitability of the proposed approach is illustrated through applications to electrical load forecasting and compared with ARIMA and exponential smoothing approaches.

Keywords: artificial immune system, seasonal time series forecasting, similarity-based methods.

1 Introduction

In general, a time series can be thought of as consisting of four different components: trend, seasonal variations, cyclical variations, and irregular component. The specific functional relationship between these components can assume different forms. Usually they combine in an additive or a multiplicative fashion. Seasonality is defined to be the tendency of time series data to exhibit behavior that repeats itself every n periods. The difference between a cyclical and a seasonal component is that the latter occurs at regular (seasonal) intervals, while cyclical factors have usually a longer duration that varies from cycle to cycle. The presence of the cyclical and multiple seasonal cycles hampers the construction of forecasting models. In this article we concentrate on the seasonal cycles. Seasonal patterns of time series can be examined via correlograms or periodograms based on a Fourier decomposition.

Many economical, business and industrial time series exhibit seasonal behavior. Examples of data with recurrent patterns are: retail sales, industrial production, traffic,

weather phenomena, electricity load, calls to call center, gas and water consumption. The recurrent patterns in these cases can be observed within daily, weekly and/or annual periods. Seasonality is often connected with the rhythm of life of the population and its relationship to the variability of the seasons, professional activity, traditions and habits.

A variety of methods have been proposed for seasonal time series forecasting. These include [1]: seasonal ARIMA, exponential smoothing, artificial neural networks, dynamic harmonic regression, vector autoregression, random effect models, and many others. The first three approaches are the most commonly employed to modeling seasonal patterns.

According to Box et al. [2] we can extend the base ARIMA model with just one seasonal pattern for the case of multiple seasonalities. Such an extension we can find in [3]. The inconvenience in the time series modeling using multiple seasonal ARIMA is a combinatorial problem of selecting appropriate model orders.

The basic Holt-Winters exponential smoothing was adapted by Taylor so that it can accommodate two seasonalities [3]. Empirical comparison showed that the resulting forecasts for the new double seasonal Holt-Winters method outperformed those from standard Holt-Winters and also those from a double seasonal ARIMA model. An advantage of the exponential smoothing models is, besides their relative simplicity, that they can be nonlinear. On the other hand it can be viewed as being of high dimension, as it involves initialization and updating of a large number of terms (level, periods of the intraday and intraweek cycles). More parsimonious formulation is proposed in [1]. Recently five new exponentially weighted methods for forecasting time series that consist of both intraweek and intraday seasonal cycles were proposed in [4].

Gould et al. described a state space model developed for the series using the innovation approach which enables to develop explicit models for both additive and multiplicative seasonality [5]. The innovation state space approach provides a theoretical foundation for exponential smoothing methods. This procedure improves on the current approaches by providing a common sense structure to the models, flexibility in modeling seasonal patterns, a potential reduction in the number of parameters to be estimated, and model based prediction intervals.

Artificial neural networks (ANNs), being nonlinear and data-driven in nature, may be well suited to the seasonal time series modeling. One of the major advantage of ANNs, that makes they are so often used in practice, is their great capacity to extract unknown and general information from a given data set even in high-dimensional task. The automated ANN learning releases a designer from the cumbersome procedures of a priori model selection. Although there is another problem: the selection of network architecture as well as the learning algorithm. The most popular ANN type used in forecasting task is multilayer perceptron, which has a property of universal approximation. ANNs are able to directly model seasonality, without the prior seasonal adjustment. An example we can find in [6] where authors conduct a comparative study between ANN and ARIMA models. However Nelson et al. [7] conclude

that ANNs trained on deseasonalized data perform significantly better than those with raw data. Zhang and Qi [8] notice that not only deseasonality is important but also detrending. They study the effectiveness of data preprocessing on ANN modeling and forecasting performance. In a large-scale empirical study [9] Zhang and Kline fit a linear trend for detrending and then subtract the estimated trend component from the raw data. For deseasonalizing they employ the method of seasonal index based on centered moving averages.

Decomposition of the time series to cope with seasonalities and trend, is a procedure used not only in ANNs, but also in other models, e.g. ARIMA and exponential smoothing. The components showing less complexity than the original time series can be predicted independently and more accurate. A frequently used approach is to decompose the time series on seasonal, trend and stochastic components (e.g. using STL filtering procedure based on LOESS smoother [10]). Other methods of decompositions apply the Fourier transform [11] or the wavelet transform [12]. The simple way to remove seasonality is to define the separate time series for each observation in a cycle, i.e. in the case of cycle of length n , n time series is defined including observations in the same positions in a cycle.

In this paper we propose an approach based on the patterns of the time series seasonal sequences. Using patterns we do not need to decompose a time series. A trend and many seasonal cycles as well as the nonstationarity and heteroscedasticity is not a problem here when using proper pattern definitions. The proposed approach belongs to the class of similarity-based methods [13] and is dedicated to forecasting time series with multiple seasonal periods. The forecast here is constructed using analogies between sequences of the time series with periodicities. An artificial immune system (AIS) is used for detection of similar patterns of sequences. The clusters of patterns are represented by antibodies (AB). Two population of ABs are created which recognize two populations of patterns (antigens) – input ones and forecast ones. The empirical probabilities that the pattern of forecasted sequence is detected by the k th AB from the second population while the corresponding pattern of input sequence is detected by the j th AB from the first population are computed and applied to the forecast construction. This idea is taken from [14], where the Kohonen net was used as a clustering method.

The merits of AIS lie in its pattern recognition and memorization capabilities. The application areas for AIS can be summarized as [15]: learning (clustering, classification, recognition, robotic and control applications), anomaly detection (fault detection, computer and network security applications), and optimization (continuous and combinatorial). Antigen recognition, self-organizing memory, immune response shaping, learning from examples, and generalization capability are valuable properties of immune systems which can be brought to potential forecasting models. The first AIS model dedicated to the time series forecasting was proposed by Dudek [16]. The novelty of the AIS proposed in this paper in the comparison with [16] is that the immune memory is composed of two AB populations and the cross-reactivity thresholds are adapted to learning data during the immune memory creation process.

2 Similarity-Based Forecasting Methods

The similarity-based (SB) methods use analogies between sequences of the time series with seasonal cycles. 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 of length n , which usually contain one seasonal cycle. Fig. 1 shows a periodical time series, where we can observe annual, weekly and daily variations. This series represents hourly electrical loads of the Polish power system. Our task is to forecast the time series elements in the daily period, so the sequences include 24 successive elements of the daily periods.

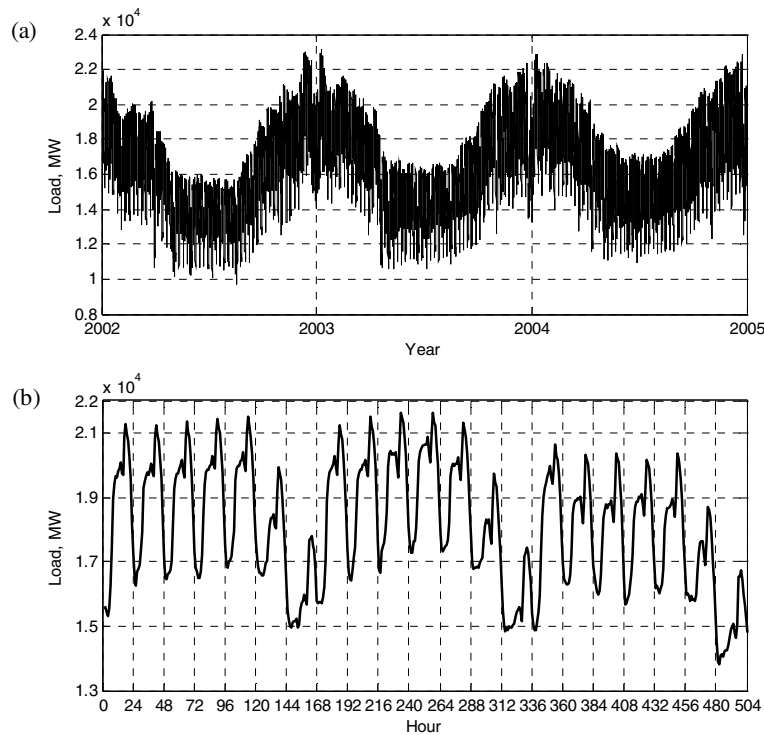


Fig. 1. The load time series of the Polish power system in three-year (a) and three-week (b) intervals

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

methods are based on the following assumption: 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 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.

The way of how the \mathbf{x} and \mathbf{y} patterns are defined depends on the time series nature (seasonal variations, trend), the forecast period and the forecast horizon. Functions transforming series elements into patterns should be defined so that patterns carry most information about the process. Moreover, functions transforming forecast sequences into patterns \mathbf{y} should ensure the opposite transformation: from the forecasted pattern \mathbf{y} to the forecasted time series sequence.

The forecast pattern $\mathbf{y}_i = [y_{i,1} \ y_{i,2} \ \dots \ y_{i,n}]$ encodes the following actual time series elements z in the forecast period $i + \tau$. $\mathbf{z}_{i+\tau} = [z_{i+\tau,1} \ z_{i+\tau,2} \ \dots \ z_{i+\tau,n}]$, and the input pattern $\mathbf{x}_i = [x_{i,1} \ x_{i,2} \ \dots \ x_{i,n}]$ maps the time series elements in the period i preceding the forecast period: $\mathbf{z}_i = [z_{i,1} \ z_{i,2} \ \dots \ z_{i,n}]$. In general, the input pattern can be defined on the basis of a sequence longer than one period, and the time series elements contained in this sequence can be selected in order to ensure the best quality of the model. Vectors \mathbf{y} are encoded using actual process parameters Ψ_i (from the nearest past), which allows to take into consideration current variability of the process and ensures possibility of decoding.

For series with daily, weekly and annual seasons we define some functions mapping the original feature space Z into the pattern spaces X and Y , i.e. $f_x : Z \rightarrow X$ and $f_y : Z \rightarrow Y$:

$$f_x(z_{i,t}, \Psi_i) = \frac{z_{i,t} - \bar{z}_i}{\sqrt{\sum_{l=1}^n (z_{i,l} - \bar{z}_i)^2}}, \quad f_y(z_{i+\tau,t}, \Psi_i) = \frac{z_{i+\tau,t} - \bar{z}_i}{\sqrt{\sum_{l=1}^n (z_{i+\tau,l} - \bar{z}_i)^2}}, \quad (1)$$

$$f_x(z_{i,t}, \Psi_i) = \frac{z_{i,t}}{z'}, \quad f_y(z_{i+\tau,t}, \Psi_i) = \frac{z_{i+\tau,t}}{z''}, \quad (2)$$

$$f_x(z_{i,t}, \Psi_i) = z_{i,t} - z', \quad f_y(z_{i+\tau,t}, \Psi_i) = z_{i+\tau,t} - z'', \quad (3)$$

where: $i = 1, 2, \dots, N$ – the period number, $t = 1, 2, \dots, n$ – the time series element number in the period i , τ – the forecast horizon, $z_{i,t}$ – the t th time series element in the period i , \bar{z}_i – the mean value of elements in period i , $z' \in \{\bar{z}_i, z_{i-1,t}, z_{i-7,t}, z_{i,t-1}\}$, $z'' \in \{\bar{z}_i, z_{i,t}, z_{i+\tau,t}\}$, Ψ_i – the set of coding parameters such as \bar{z}_i , z' and z'' .

The function f_x defined using (1) expresses normalization of the vectors \mathbf{z}_i . After normalization these vectors have the unity length, zero mean and the same variance. When we use the standard deviation of the vector \mathbf{z}_i components in the denominator of equation (1), we receive vector \mathbf{x}_i with the unity variance and zero mean. Note that the nonstationary and heteroscedastic time series is represented by patterns having the same mean and variance.

The components of the x -patterns defined using equations (2) and (3) express, respectively, indices and differences of $z_{i,t}$ and z' or z'' .

Forecast patterns are defined using analogous functions to input pattern functions f_x , but they are encoded using the time series elements or characteristics determined from the process history, what enables decoding of the forecasted vector $\mathbf{z}_{i+\tau}$ after the forecast of pattern \mathbf{y} is determined. To calculate the time series element values on the basis of their patterns we use the inverse functions: $f_x^{-1}(x_{i,t}, \Psi_i)$ or $f_y^{-1}(y_{i,t}, \Psi_i)$. For example the inverse functions for (1) are:

$$f_x^{-1}(x_{i,t}, \Psi_i) = x_{i,t} \sqrt{\sum_{l=1}^n (z_{i,t} - \bar{z}_i)^2} + \bar{z}_i, \quad f_y^{-1}(y_{i,t}, \Psi_i) = y_{i,t} \sqrt{\sum_{l=1}^n (z_{i+\tau,l} - \bar{z}_i)^2} + \bar{z}_i. \quad (4)$$

As a similarity measure between two patterns (real-valued vectors) we can use [17]: the inner product (when the vectors are normalized) or closely related to it cosine similarity measure, Pearson's correlation coefficient or Tanimoto measure. It is useful to define the similarity measures on the base of the distance measures, e.g using linear mapping: $s(\mathbf{x}_a, \mathbf{x}_b) = c - d(\mathbf{x}_a, \mathbf{x}_b)$ or nonlinear mapping: $s(\mathbf{x}_a, \mathbf{x}_b) = 1/(1 + d(\mathbf{x}_a, \mathbf{x}_b))$, where: $s(.,.)$ is a similarity, $d(.,.)$ is a distance and c is a constant greater than the highest value of the distance. The popular distance measures are: Euclidean, Manhattan or Canberra distances. If the components of the vectors are expressed in different units or change in different ranges, in order to offset their impact on the distance, their weighing is recommended.

If for a given time series the statistical analysis confirms the hypothesis that the dependence between similarities of input patterns and similarities between forecast patterns paired with them are not caused by random character of the sample, it justifies the sense of building and using models based on the similarities of patterns of this time series. The statistical analysis of pattern similarities is described in [13].

The forecasting procedure in the case of SB methods can be summarized as follows:

1. Elimination of the trend and seasonal variations of periods longer than n using pattern functions f_x and f_y .
2. Forecasting the pattern \mathbf{y} using similarities between patterns.
3. Reconstruction the time series elements from the forecasted pattern \mathbf{y} using the inverse function f_y^{-1} .

3 Immune Inspired Forecasting Model

The proposed AIS contains immune memory consisting of two populations of ABs. The population of x -antibodies (AB x) detects antigens representing patterns $\mathbf{x} = [x_1, x_2, \dots, x_n]^T - \text{AGx}$, while the population of y -antibodies (AB y) detects antigens representing patterns $\mathbf{y} = [y_1, y_2, \dots, y_n]^T - \text{AGy}$. The vectors \mathbf{x} and \mathbf{y} form the epitopes of AGs and paratopes of ABs. AB x has the cross-reactivity threshold r defining the AB recognition region. This recognition region is represented by the n -dimensional

hypersphere of radius r with center at the point \mathbf{x} . Similarly AB y has the recognition region of radius s with center at the point \mathbf{y} . The cross-reactivity thresholds are adjusted individually during training. The recognition regions contain AGs with similar epitopes.

AG can be bound to many different ABs of the same type (x or y). The strength of binding (affinity) is dependent on the distance between an epitope and a paratope. AB represents a cluster of similar AGs in the pattern space X or Y . The clusters are overlapped and their sizes depend on the similarity between AGs belonging to them, measured in the both pattern spaces X and Y . The k th AB x can be written as a pair $\{\mathbf{p}_k, r_k\}$, where $\mathbf{p}_k = \mathbf{x}_k$, and the k th AB y as $\{\mathbf{q}_k, s_k\}$, where $\mathbf{q}_k = \mathbf{y}_k$.

After the two population of immune memory have been created, the empirical conditional probabilities $P(AB_{y_k} | AB_{x_j})$, $j, k = 1, 2, \dots, N$, that the i th AG y stimulates (is recognized by) the k th AB y , when the corresponding i th AG x stimulates the j th AB x , are determined. These probabilities are calculated for each pair of AB x and AB y on the basis of recognition of the training population of AGs.

In the forecasting phase the new AG x , representing pattern \mathbf{x}^* , is presented to the trained immune memory. The forecasted pattern \mathbf{y} paired with \mathbf{x}^* is calculated as the mean of AB y paratopes weighted by the conditional probabilities and affinities.

The detailed algorithm of the immune system to forecasting seasonal time series is described below.

Training (immune memory creation)

1. Loading of the training populations of antigens.
2. Generation of the antibody populations.
3. Calculation of the cross-reactivity thresholds of x-antibodies.
4. Calculation of the cross-reactivity thresholds of y-antibodies.
5. Calculation of the empirical conditional probabilities $P(AB_{y_k} | AB_{x_j})$.

Test

6. Forecast determination using y-antibodies, probabilities $P(AB_{y_k} | AB_{x_j})$ and affinities.

Fig. 2. Pseudocode of the AIS for the seasonal time series forecasting

Step 1. Loading of the training populations of antigens. An AG x represents a single \mathbf{x} pattern, and AG y represents a single \mathbf{y} pattern. Both populations of AG x and AG y are divided into training and test parts in the same way. Immune memory is trained using the training populations, and after learning the model is tested using the test populations.

Step 2. Generation of the antibody populations. The AB populations are created by copying the training populations of AGs (ABs and AGs have the same structure). Thus the paratopes take the form: $\mathbf{p}_k = \mathbf{x}_k$, $\mathbf{q}_k = \mathbf{y}_k$, $k = 1, 2, \dots, N$. The number of AGs and ABs of both types is the same as the number of learning patterns.

Step 3. Calculation of the cross-reactivity thresholds of x-antibodies. The recognition region of the k th ABx should be as large as possible and cover only the AGx that satisfy two conditions:

- (i) their epitops \mathbf{x} are similar to the paratope \mathbf{p}_k , and
- (ii) the AGy paired with them have epitopes \mathbf{y} similar to the k th ABy paratope – \mathbf{q}_k .

The measure of similarity of the i th AGx to the k th ABx is an affinity:

$$a(\mathbf{p}_k, \mathbf{x}_i) = \begin{cases} 0, & \text{if } d(\mathbf{p}_k, \mathbf{x}_i) > r_k \text{ or } r_k = 0 \\ 1 - \frac{d(\mathbf{p}_k, \mathbf{x}_i)}{r_k}, & \text{otherwise} \end{cases}, \quad (5)$$

where $d(\mathbf{p}_k, \mathbf{x}_i)$ is the distance between vectors \mathbf{p}_k and \mathbf{x}_i , $a(\mathbf{p}_k, \mathbf{x}_i) \in [0, 1]$. Affinity $a(\mathbf{p}_k, \mathbf{x}_i)$ informs about the degree of membership of the i th AGx to the cluster represented by the k th ABx.

The similarity of the i th AGy to the k th ABy mentioned in (ii) is measured using the forecast error of the time series elements encoded in the paratope of the k th ABy. These elements are forecasted using the epitope of the i th AGy. The forecast error is:

$$\delta_{k,i} = \frac{100}{n} \sum_{t=1}^n \frac{|z_{k+\tau,t} - f_y^{-1}(y_{i,t}, \Psi_k)|}{z_{k+\tau,t}}, \quad (6)$$

where: $z_{k+\tau,t}$ – the t th time series element of the period $k+\tau$ which is encoded in the paratope of the k th ABy: $q_{k,t} = f_y(z_{k+\tau,t}, \Psi_k)$, $f_y^{-1}(y_{i,t}, \Psi_k)$ – the inverse function of the pattern \mathbf{y} returning the forecast of time series element $z_{k+\tau,t}$ using the epitope of the i th AGy.

If the condition $\delta_{k,i} \leq \delta_y$ is satisfied, where δ_y is the error threshold value, it is assumed that the i th AGy is similar to the k th ABy, and i th AGx, paired with this AGy, is classified to class 1. When the above condition is not met the i th AGx is classified to class 2. Thus class 1 indicates the high similarity between ABy and AGy. The classification procedure is performed for each ABx.

The cross-reactivity threshold of k th ABx is defined as follows:

$$r_k = d(\mathbf{p}_k, \mathbf{x}_A) + c[d(\mathbf{p}_k, \mathbf{x}_B) - d(\mathbf{p}_k, \mathbf{x}_A)], \quad (7)$$

where B denotes the nearest AGx of class 2 to the k th ABx, and A denotes the furthest AGx of class 1 satisfying the condition $d(\mathbf{p}_k, \mathbf{x}_A) < d(\mathbf{p}_k, \mathbf{x}_B)$. The parameter $c \in [0, 1]$ allows to adjust the cross-reactivity threshold value from $r_{k\min} = d(\mathbf{p}_k, \mathbf{x}_A)$ to $r_{k\max} = d(\mathbf{p}_k, \mathbf{x}_B)$.

Step 4. Calculation of the cross-reactivity thresholds of y-antibodies. The cross-reactivity threshold of k th ABy is calculated similarly to the above:

$$s_k = d(\mathbf{q}_k, \mathbf{y}_A) + b[d(\mathbf{q}_k, \mathbf{y}_B) - d(\mathbf{q}_k, \mathbf{y}_A)], \quad (8)$$

where B denotes the nearest AGy of class 2 to the k th ABY, and A denotes the furthest AGy of class 1 satisfying the condition $d(\mathbf{q}_k, \mathbf{y}_A) < d(\mathbf{q}_k, \mathbf{y}_B)$. The parameter $b \in [0, 1]$ plays the same role as the parameter c .

The i th AGy is classified to class 1, if for the i th AGx paired with it, there is $\varepsilon_{k,i} \leq \varepsilon_x$, where ε_x is the threshold value and $\varepsilon_{k,i}$ is the forecast error of the time series elements encoded in the paratope of the k th ABx. These elements are forecasted using the epitope of the i th AGx. The forecast error is:

$$\varepsilon_{k,i} = \frac{100}{n} \sum_{t=1}^n \frac{|z_{k,t} - f_x^{-1}(x_{i,t}, \Psi_k)|}{z_{k,t}}, \quad (9)$$

where: $z_{k,t}$ – the t th time series element of the period k which is encoded in the paratope of the k th ABx: $p_{k,t} = f_x(z_{k,t}, \Psi_k)$, $f_x^{-1}(x_{i,t}, \Psi_k)$ – the inverse function of pattern \mathbf{x} returning the forecast of time series element $z_{k,t}$ using the epitope of the i th AGx.

The i th AGy is recognized by k th ABY if affinity $a(\mathbf{q}_k, \mathbf{y}_i) > 0$, where:

$$a(\mathbf{q}_k, \mathbf{y}_i) = \begin{cases} 0, & \text{if } d(\mathbf{q}_k, \mathbf{y}_i) > s_k \text{ or } s_k = 0 \\ 1 - \frac{d(\mathbf{q}_k, \mathbf{y}_i)}{s_k}, & \text{otherwise} \end{cases}, \quad (10)$$

$a(\mathbf{q}_k, \mathbf{y}_i) \in [0, 1]$ expresses the degree of membership of pattern \mathbf{y}_i to the cluster represented by the k th ABY.

Procedure for determining the threshold s_k is thus analogous to the procedure for determining the threshold r_k . The recognition region of k th ABY is as large as possible and covers AGy that satisfy two conditions:

- (i) their epitops \mathbf{y} are similar to the paratope \mathbf{q}_k , and
- (ii) the AGx paired with them have epitopes \mathbf{x} similar to the k th ABx paratope – \mathbf{p}_k .

This way of forming clusters in pattern space X (Y) makes that their sizes are dependent on the dispersion of y -patterns (x -patterns) paired with patterns belonging to these clusters. Another pattern \mathbf{x}_i (\mathbf{y}_i) is appended to the cluster ABx_k (ABy_k) (this is achieved by increasing the cross-reactivity threshold of AB representing this cluster), if the pattern paired with \mathbf{x}_i (\mathbf{y}_i) is sufficiently similar to the paratope of the k th ABy_k (ABx_k). The pattern is considered sufficiently similar to the paratope, if it allows to forecast the time series encoded in the paratope with an error no greater than the threshold value. This ensures that the forecast error for the pattern \mathbf{x} (\mathbf{y}) has a value not greater than ε_x (δ_y). Lower error thresholds imply smaller clusters, lower bias and greater variance of the model. Mean absolute percentage error (MAPE) here is used as an error measure ((6) and (9)) but other error measures can be applied.

Step 5. Calculation of the empirical conditional probabilities $P(ABy_k|ABx_j)$. After the clustering of both spaces is ready, the successive pairs of antigens (AGx_i , AGy_j), $i = 1, 2, \dots, N$, are presented to the trained immune memory. The stimulated ABx and

ABy are counted and the empirical frequencies of ABy_k given ABx_j , estimating conditional probabilities $P(ABy_k | ABx_j)$, are determined.

Step 6. Forecast procedure. In the forecast procedure new AGx, representing the pattern \mathbf{x}^* , is presented to the immune memory. Let Ω be a set of ABx stimulated by this AGx. The forecasted pattern \mathbf{y} corresponding to \mathbf{x}^* is estimated as follows:

$$\hat{\mathbf{y}} = \sum_{k=1}^N w_k \mathbf{q}_k, \quad (11)$$

where

$$w_k = \frac{\sum_{j \in \Omega} P(ABy_k | ABx_j) a(\mathbf{p}_j, \mathbf{x}^*)}{\sum_{l=1}^N \sum_{j \in \Omega} P(ABy_l | ABx_j) a(\mathbf{p}_j, \mathbf{x}^*)} \quad (12)$$

and $\sum_{k=1}^N w_k = 1$.

The forecast is calculated as the weighted mean of paratopes \mathbf{q}_k . The weights w express the sum of products of the affinities of stimulated memory cells ABx to the AGx and probabilities $P(ABy_k | ABx_j)$.

The clusters represented by ABs have spherical shapes, they overlap and their sizes are limited by cross-reactivity thresholds. The number of clusters is here equal to the number of learning patterns, and the means of clusters in the pattern spaces X and Y (paratopes ABx and ABy) are fixed – they lie on the learning patterns.

The cross-reactivity thresholds, determining the cluster sizes, are tuned to the training data in the immune memory learning process. In results the clusters in the space X correspond to compact clusters in the space Y , and vice versa. It leads to more accurate mapping $X \rightarrow Y$. The model has four parameters: the error thresholds (δ_y and ε_x) and the parameters tuning the cross-reactivity thresholds (b and c). Increasing the value of these parameters imply an increase in size of clusters, an increase in the model bias and reduction in its variance.

The training routine is deterministic, which means the fast learning process. The immune memory learning needs only one pass of the training data. The runtime complexity of the training routine is $O(N^2n)$. The most costly operation is the distance calculation between each ABs and AGs. The runtime complexity of the forecasting procedure is also $O(N^2n)$.

4 Empirical Study

The described above AIS was applied to the next day electrical load curve forecasting ($\tau = 1$). Short-term load forecasting plays a key role in control and scheduling of power systems and is extremely important for energy suppliers, system operators, financial institutions, and other participants in electric energy generation,

transmission, distribution, and markets. Precise load forecasts are necessary for electric companies to make important decisions connected with electric power production and transmission planning, such as unit commitment, generation dispatch, hydro scheduling, hydro-thermal coordination, spinning reserve allocation and interchange evaluation.

The series studied in this section represents the hourly electrical load of the Polish power system from the period 2002-2004. This series is shown in Fig. 1. The time series were divided into training and test parts. The test set contained 30 pairs of patterns from January 2004 (from January 2 to 31) and 31 pairs of patterns from July 2004. The training set contained patterns from the period from January 1, 2002 to the day preceding the day of forecast.

For each day from the test part the separate immune memory was created using the training subset containing AGy representing days of the same type (Monday, ..., Sunday) as the day of forecast and paired with them AGx representing the preceding days (e.g. for forecasting the Sunday load curve, model learns from AGx representing the Saturday patterns and AGy representing the Sunday patterns). This routine of model learning provides fine-tuning its parameters to the changes observed in the current behavior of the time series.

The distance between ABs and AGs was calculated using Euclidean metric. The patterns were defined using (1). MAPE, which is traditionally used in short-term load forecasting, was a forecast error measure.

The model parameters were determined using the grid search method on the training subsets in the local version of the leave-one-out procedure. In this procedure not all patterns are successively removed from the training set to estimate the generalization error but only the k -nearest neighbors of the test x -pattern (k was arbitrarily set to 5). As a result, the model is optimized locally in the neighborhood of the test pattern. It leads to a reduction in learning time.

In the grid search procedure the parameters were changed as follows:

- (i) $\delta_y = 1.00, 1.25, \dots, 3.00$, $\varepsilon_x = 1.00, 1.25, \dots$, at the constant values of $b = c = 1$, and
- (ii) $b = c = 0, 0.2, \dots, 1.0$, at the optimal values of δ_y and ε_x determined in (i).

It was observed that at lower values of δ_y and c many x -patterns are unrecognized. If $\delta_y \geq 2.25$ and $c = 1$ approximately 99% of the x -patterns are detected by ABx. Increasing δ_y above 2.25 results in increasing the error. Minimum error (MAPE) was observed for $\delta_y = 2.25$, $\varepsilon_x = 1.75$ and $b = c = 1$.

The forecast test MAPE for January was 1.37 and for July was 0.92. Fig. 3 illustrates the empirical conditional probabilities $P(AB_{y_k}|AB_{x_j})$ estimated on the training set for July 1, 2004. You can observe a specific pattern that indicates which ABs in both populations are activated simultaneously (e.g. the activation of x -antibodies #19–31 corresponds the stronger activation of y -antibodies #19–32, 69–80 and 118–122). These are ABs representing daily cycles lying in the same period of the year. Higher probabilities imply a stronger relationship $X \rightarrow Y$ and greater confidence to the forecast. The weights of activated ABs for this forecasting task are shown in Fig. 4. Here

we also can see that the input pattern stimulates ABs representing patterns from the same period of the year as the input pattern. The reconstructed forecast pattern on the background of patterns represented by activated ABs is shown in Fig. 5.

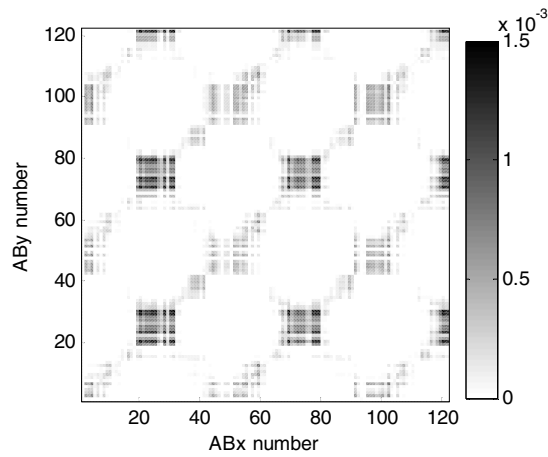


Fig. 3. The empirical conditional probabilities $P(AB_{y_k} | AB_{x_j})$ estimated on the training set for July 1, 2004

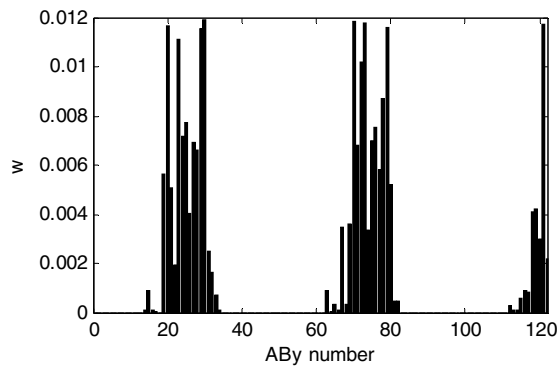


Fig. 4. The weights of activated ABs estimated on the training set for July 1, 2004

4.1 Model Sensitivity to Changes in Parameter Values

The aim of this analysis is to evaluate the influence of the parameter values on the forecasts generated by the model. Fig. 6 shows the test sample forecast errors depending on the parameter values which were changed individually in the ranges: $\epsilon_x \in [0.5\epsilon_x^*, 1.5\epsilon_x^*]$, $\delta_y \in [0.5\delta_y^*, 1.5\delta_y^*]$ and $b = [0.5b^*, \min(1.5b^*, 1)]$, where asterisk denotes the optimal value of the parameter estimated on the training set. It was assumed $c = b$.

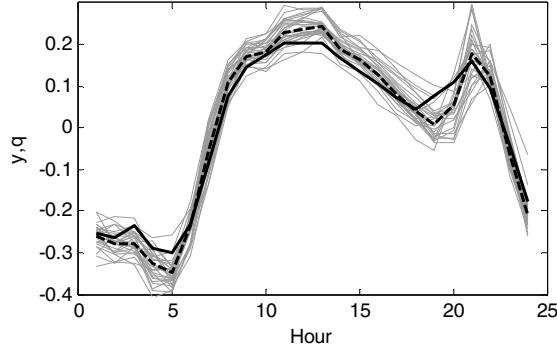


Fig. 5. The activated ABs (gray lines), the reconstructed forecast pattern (dashed line) and the actual forecast pattern (black continuous line) – load forecasting for July 1, 2004

To avoid the situations when for the smaller parameter values many test AGx are unrecognized, it was assumed that an unrecognized AGx is included to the group represented by the nearest ABx, although the recognition region of this ABx does not include the AGx. The same assumption is made for analyses described in Sections 4.2, 4.3 and 5.

Fig. 7 shows the relative percentage difference between the forecasts generated by the optimal model ($\hat{z}_{i+\tau,t}(p^*)$) and the model with non-optimal parameter value ($\hat{z}_{i+\tau,t}(p)$):

$$RPD(p) = \frac{100}{Nn} \sum_{i=1}^N \sum_{t=1}^n \frac{|\hat{z}_{i+\tau,t}(p) - \hat{z}_{i+\tau,t}(p^*)|}{\hat{z}_{i+\tau,t}(p^*)}. \quad (13)$$

where p denotes the parameter.

The change of the parameter value can cause a step change in the model response which makes the lines in Fig. 6 and 7 are not smooth. From Fig. 6 we can see that overestimation of ε_x and δ_y results in more rapidly increase in the forecast error than underestimation. The values of ε_x and δ_y for the minimum training and test errors are not the same. Smaller values are observed for the test set. RPD achieves 0.3–0.4% for the border values of ε_x and δ_y and only 0.074% for b and c .

The model sensitivity to changes in parameters is defined by the sensitivity index:

$$SI_p = \frac{MAPE_{tst\ max} - MAPE_{tst\ min}}{MAPE_{tst\ min}} \cdot 100. \quad (14)$$

Index (14) informs what is the relative percentage difference between the maximum and minimum errors $MAPE_{tst}$ when the parameter varies in a given range. Its values were: 10.10% for ε_x , 7.42% for δ_y and 2.27% for b and c . Thus the sensitivity to the error thresholds are much greater than for b and c .

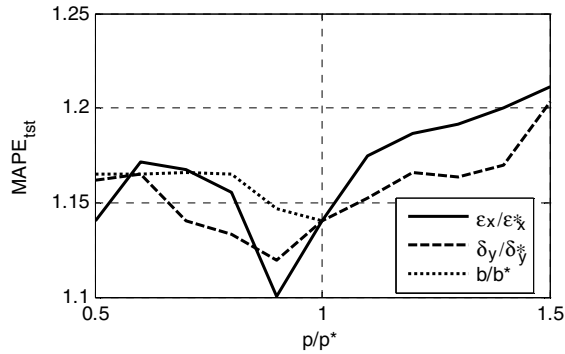


Fig. 6. The forecast error depending on the parameter values

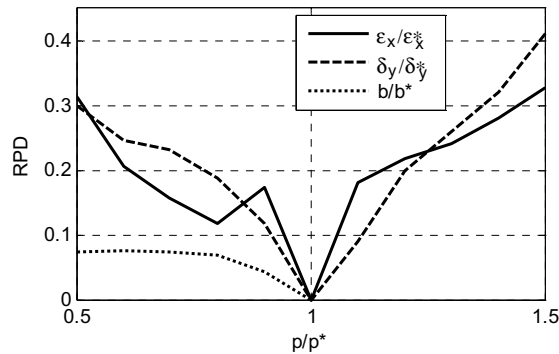


Fig. 7. Relative percentage difference (13) depending on the parameter values

4.2 Model Robustness to Noisy Data

The aim of this analysis is to determine the model robustness to noisy data resulting from the measurement or estimation errors of the time series terms. The noisy patterns are located in the space differently relative to each other than the original patterns. This affects the distances between them, and thus, the probabilities $P(AB_{y_k}|AB_{x_j})$, affinities and the forecasted pattern.

It is assumed that the actual time series elements are disturbed by random errors:

$$z_i' = z_i \xi_i, \tag{15}$$

where $\xi_i \sim N(1, \sigma)$.

The standard deviation σ was changing in the range from 0 to 0.1. It corresponds to a share of noise in the data ($100 |z' - z| / z$) from 0 to 8%. For each σ value 30 training sessions were performed and then mean forecast error was recorded (Fig. 8).

The sensitivity index to noisy data is defined as follows:

$$SI_n(\sigma) = \frac{MAPE_{tst}(\sigma) - MAPE_{tst}(\sigma=0)}{\sigma} \cdot 100. \quad (16)$$

This index expresses the ratio of the change in forecast error due to the noisy data to the intensity of the noise. The mean value of SI_n for all σ , which corresponds to the slope of the straight line approximating the characteristics presented in Fig. 8, was equal to 65.96%.

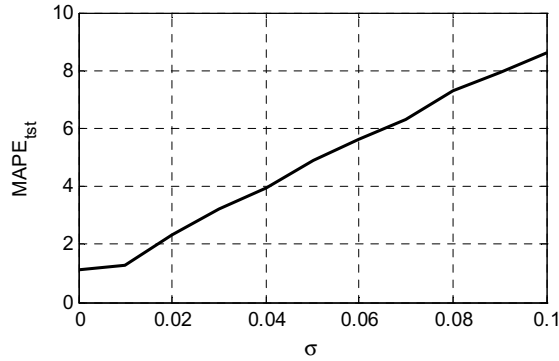


Fig. 8. The forecast error depending on the σ value

4.3 Model Robustness to Missing Data

In this study the robustness to missing input information is analyzed. We assume that some components of the input pattern \mathbf{x}^* are missing. It corresponds to the missing terms of the time series (missing components of vector \mathbf{z}^*). In many models (e.g. ARIMA, exponential smoothing, neural networks) this is a serious problem, and the missing data reconstruction is needed. One strategy for dealing with the missing component value is to assign it the value that is most common among training examples. Another strategy is to assign it the values of the corresponding components of the most similar patterns. The proposed AIS copes well with missing components of the input pattern. In such a case the epitopes of AGx and paratopes of ABx are composed of non-missing components. The immune memory creation and test procedures remain unchanged.

When patterns \mathbf{x} and \mathbf{y} are defined using the mean value of elements \bar{z}_i , the values of non-missing components are different from their original values. This is caused by different value of \bar{z}_i which is determined now without the missing components. As in the case of noisy data, patterns with missing components locate differently in the space than the original ones, which cause the change of the forecast pattern.

To examine the robustness of the model to missing data we remove m components of the vectors \mathbf{z}^* and then we redefine patterns \mathbf{x} and \mathbf{y} . The immune memory is constructed on the training set and the test is performed using \mathbf{x}^* pattern having the same components as \mathbf{z}^* . The m components are chosen by random independently for each

\mathbf{z}^* . The forecast errors depending on the relative number of missing components are shown in Fig. 9. When the number of missing components is low the deterioration in the model accuracy is not observed. Errors begin to grow rapidly when m exceeds 16.

As a measure of the model sensitivity to the missing components the index SI_m is proposed:

$$SI_m(m) = \frac{MAPE_{tst}(m) - MAPE_{tst}(m=0)}{m/n} \cdot 100. \quad (17)$$

The SI_m value for $m = 6$ was -3.82% , for $m = 12$ was 11.82% and for $m = 18$ was 41.81% .

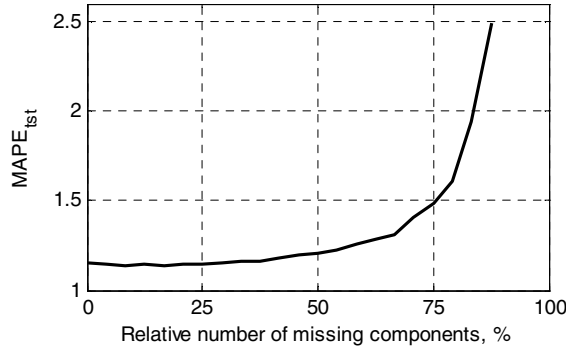


Fig. 9. The forecast error depending on the relative number of missing components

5 Empirical Comparison with Other Models

In this section we compare the proposed AIS with other popular models of the seasonal time series forecasting: ARIMA, exponential smoothing (ES) and double seasonal Holt-Winters (DSHW) method. These models were tested in the next day electrical load curve forecasting problem on three time series of electrical load:

- TS1: time series of the hourly loads of the Polish power system from the period 2002-2004 (this time series was used in analyses described in Section 4). The test sample includes data from 2004 with the exception of 13 untypical days (e.g. holidays).
- TS2: time series of the hourly loads of the local power system from the period July 2001-December 2002. The test sample includes data from the period July-December 2002 except for 8 untypical days.
- TS3: time series of the hourly loads of the local power system from the period 1999-2001. The test sample includes data from 2001 except for 13 untypical days.

Time series TS1 is presented in Fig. 1, TS2 and TS3 in Fig. 10. Time series TS2 and TS3 are more irregular and harder to forecasting than TS1. The measure of the load time series regularity could be the forecast error determined by the naïve method.

The forecast rule in this case is as follows: the forecasted daily cycle 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 1.

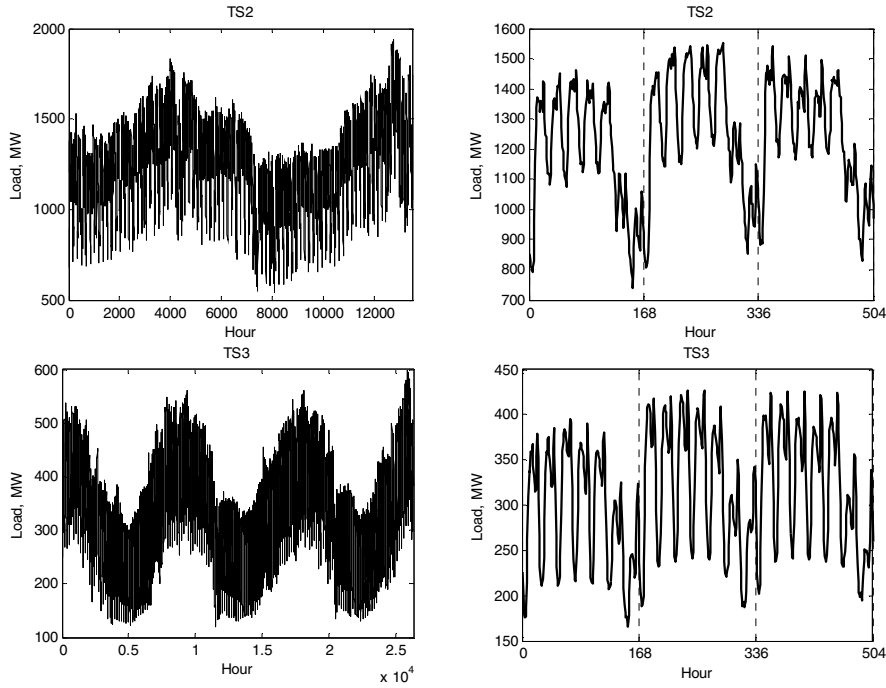


Fig. 10. The TS2 and TS3 time series (three-week details on the right)

In ARIMA the time series were decomposed into 24 series, i.e. for each hour of a day a separate series is created. In this way the daily seasonality is removed. For the independent modeling of these series $ARIMA(p, d, q) \times (P, D, Q)_m$ model is used:

$$\Phi(B^m)\phi(B)(1-B^m)^D(1-B)^d z_t = c + \Theta(B^m)\theta(B)\xi_t, \quad (18)$$

where $\{z_t\}$ is the time series, $\{\xi_t\}$ is a white noise process with mean zero and variance σ^2 , B is the backshift operator, $\Phi(\cdot)$, $\phi(\cdot)$, $\Theta(\cdot)$, and $\theta(\cdot)$ are polynomials of order P , p , Q and q respectively, m is the seasonal period (for our data $m = 7$), d and D are orders of nonseasonal and seasonal differencing, respectively, and c is a constant.

To find the best ARIMA model for each time series we use a step-wise procedure for traversing the model space which is implemented in the **forecast** package for the **R** system for statistical computing [18]. This automatic procedure returns the model with the lowest Akaike's Information Criterion (AIC) value.

ARIMA model parameters, as well as the parameters of the ES and DSHW models described below, were estimated using 12-week time series fragments immediately preceding the forecasted daily period. Untypical days in these fragments were replaced with the days from the previous weeks.

The ES state space models [19] are classified into 30 types depending on how the seasonal, trend and error components are taken into account. These components can be expressed additively or multiplicatively, and the trend can be damped or not. For example, the ES model with a damped additive trend, multiplicative seasonality and multiplicative errors is of the form:

$$\begin{aligned}
 \text{Level:} \quad & l_t = (l_{t-1} + \phi b_{t-1})(1 + \alpha \xi_t) \\
 \text{Growth:} \quad & b_t = \phi b_{t-1} + \beta(l_{t-1} + \phi b_{t-1})\xi_t \\
 \text{Seasonal:} \quad & s_t = s_{t-m}(1 + \gamma \xi_t) \\
 \text{Forecast:} \quad & \mu_t = (l_{t-1} + \phi b_{t-1})s_{t-m}
 \end{aligned} \tag{19}$$

where l_t represents the level of the series at time t , b_t denotes the growth (or slope) at time t , s_t is the seasonal component of the series at time t , μ_t is the expected value of the forecast at time t , $\alpha, \beta, \gamma \in (0, 1)$ are the smoothing parameters, and $\phi \in (0, 1)$ denotes a damping parameter.

The trend component is a combination of a level term l and a growth term b .

In model (19) there is only one seasonal component. For this reason, as in the case of the ARIMA model, time series is decomposed into 24 series, each of which represents the load at the same hour of a day. These series were modeled independently using an automated procedure implemented in the **forecast** package for the **R** system [18]. In this procedure the initial states of the level, growth and seasonal components are estimated as well as the smoothing and damping parameters. AIC was used for selecting the best model for a given time series.

The DSHW model was proposed by Taylor [3]. This is an exponential smoothing formulation that can accommodate more than one seasonal pattern. This approach does not require the decomposition of the problem. The trend component is treated additively and the seasonal components are treated multiplicatively:

$$\begin{aligned}
 \text{Level:} \quad & l_t = \alpha y_t / (d_{t-m_1} w_{t-m_2}) + (1 - \alpha)(l_{t-1} + b_{t-1}) \\
 \text{Growth:} \quad & b_t = \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1} \\
 \text{Seasonality 1:} \quad & d_t = \delta y_t / (l_t w_{t-m_2}) + (1 - \delta)d_{t-m_1} \\
 \text{Seasonality 2:} \quad & w_t = \omega y_t / (l_t d_{t-m_1}) + (1 - \omega)w_{t-m_2} \\
 \text{Forecast:} \quad & \hat{y}_{t+h} = (l_t + hb_t)d_{t-m_1+h} w_{t-m_2+h} + \lambda^h (y_t - (l_{t-1} + b_{t-1})d_{t-m_1} w_{t-m_2})
 \end{aligned} \tag{20}$$

where d_t and w_t are the seasonal components (daily and weekly in our examples) of the series at time t , \hat{y}_{t+h} is the h step-ahead forecast made from forecast origin t , $\alpha, \beta, \delta, \omega \in (0, 1)$ are the smoothing parameters, m_1 and m_2 are the seasonal periods (for our data $m_1 = 24$ and $m_2 = 168$).

The term involving the parameter λ in (20) is a simple adjustment for first-order autocorrelation. All the parameters: $\alpha, \beta, \delta, \omega$ and λ are estimated in a single procedure by minimizing the sum of squared one step-ahead in-sample errors. The initial smoothed values for the level, trend and seasonal components are estimated by averaging the early observations. These calculations were performed using `dshw` function from the **forecast** package for the **R** system.

The model selection and training procedures for AIS were the same as described in Section 4. The model was optimized for each test sample, as well as other models.

In Table 1 results of forecasts are presented: MAPE for the test samples and the interquartile range (*IQR*) of MAPE. The actual and forecasted fragment of TS1 are shown in Fig. 11.

Table 1. Results of forecasting

Model	TS1		TS2		TS3	
	$MAPE_{tst}$	<i>IQR</i>	$MAPE_{tst}$	<i>IQR</i>	$MAPE_{tst}$	<i>IQR</i>
AIS	1.60	1.56	3.35	2.69	3.23	3.10
ARIMA	1.82	1.71	3.41	3.25	3.93	3.68
ES	1.66	1.57	3.16	3.10	3.51	3.23
DSHW	2.23	2.17	3.62	3.45	4.70	4.04
Naïve	3.43	3.42	4.96	3.71	6.62	5.87

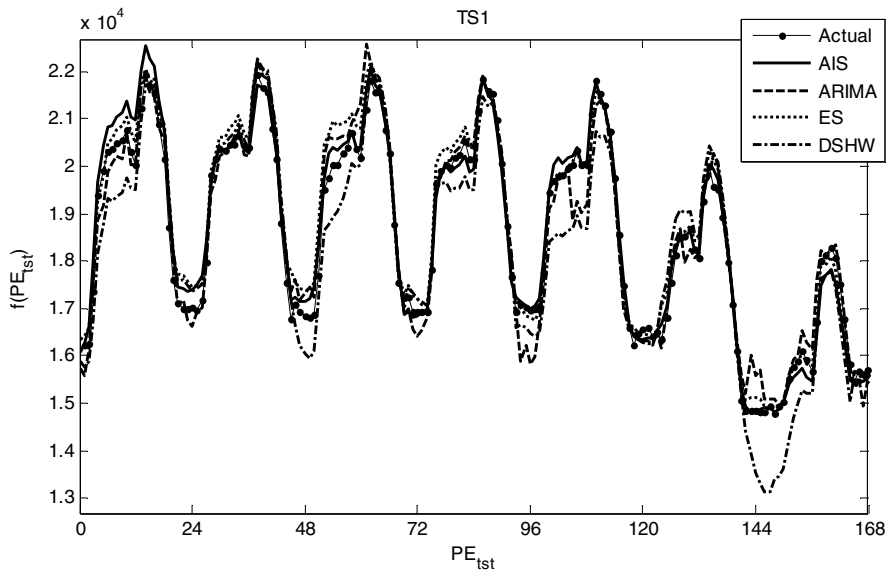


Fig. 11. The actual and forecasted fragment of TS1

In order to indicate the best model we check if the difference between the accuracies of each pair of models is statistically significant using the Wilcoxon rank sum test for equality of medians. The 5% significance level is applied in this study. For only one case: AIS, ES and TS1 the test failed to reject the null hypothesis that errors have the same medians. In all other cases the test indicates the statistically significant difference between errors. Thus the best method for TS1 are AIS and ES, for TS2 is ES, and for TS3 is AIS. The worst method for all cases was DSHW. This model produced several completely incorrect forecasts for TS2 (not included in mean error presented in Table 1). From Table 1 we can see that the naïve method was substantially

outperformed by all other methods. It is worth noting as well that the ES model performed better than ARIMA in all cases. It was observed that the ARIMA and ES models produced worst forecasts than AIS and DSHW for the first hours of a day. The proposed AIS was the best model for TS1 and TS3 but not for TS2. This is probably because of the insufficient number of learning points for TS2 (TS2 time series is twice shorter than TS1 and TS3). In this case there are fewer ABs in the immune memory and the local modeling is less accurate.

In Fig. 12 the density functions of the percentage errors (PE) are shown. For ARIMA and ES the shift of the PE densities on the right is observed. It means that these models produced underestimated forecasts.

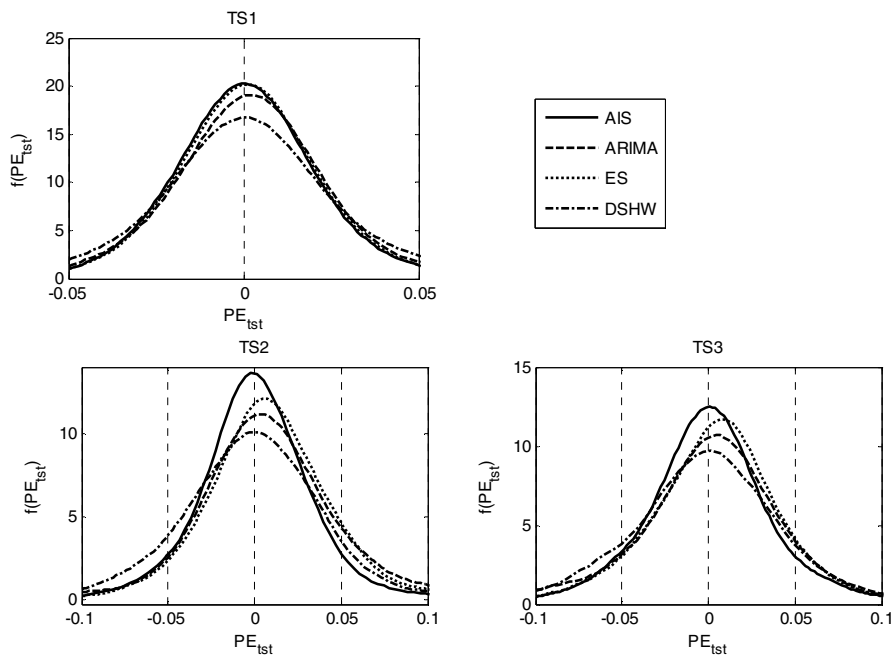


Fig. 12. The empirical PDF of PE_{tst}

6 Conclusions

In this article we deal with forecasting of the seasonal time series which can be non-stationary and heteroscedastic using patterns of the time series seasonal periods. The proposed forecasting method belongs to the class of similarity-based models. These models are based on the assumption that, if 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. It means that patterns of neighboring sequences are staying in a certain relation, which does not change significantly in time. The more stable this relation is, the more accurate forecasts are. This relation can be shaped by proper pattern definitions and strengthened by elimination of outliers.

The idea of using AIS as a forecasting model is a very promising one. The immune system has some mechanisms useful in the forecasting tasks, such as an ability to recognize and to respond to different patterns, an ability to learn, memorize, encode and decode information.

Unlike other clustering methods used in forecasting models [13], [14], the proposed AIS forms clusters taking into account the forecast error. The cluster sizes are tuned to the data in such a way to minimize the forecast error. Due to the deterministic nature of the model the results are stable and the learning process is rapid. The AIS model also offers robustness to missing data.

The disadvantage of the proposed immune system is limited ability to extrapolation. Regions without the antigens are not represented in the immune memory. However, a lot of models, e.g. neural networks, have problems with extrapolation.

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

References

1. Taylor, J.W., Snyder, R.D.: Forecasting Intraday Time Series with Multiple Seasonal Cycles Using Parsimonious Seasonal Exponential Smoothing. Department of Econometrics and Business Statistics Working Paper 9/09, Monash University (2009)
2. Box, G.E.P., Jenkins, G.M., Reinsel, G.C.: Time Series Analysis: Forecasting and Control, 3rd edn. Englewood Cliffs, Prentice Hall, New Jersey (1994)
3. Taylor, J.W.: Short-Term Electricity Demand Forecasting Using Double Seasonal Exponential Smoothing. *Journal of the Operational Research Society* 54, 799–805 (2003)
4. Taylor, J.W.: Exponentially Weighted Methods for Forecasting Intraday Time Series with Multiple Seasonal Cycles. *International Journal of Forecasting* 26(4), 627–646 (2010)
5. Gould, P.G., Koehler, A.B., Ord, J.K., Snyder, R.D., Hyndman, R.J., Vahid-Araghi, F.: Forecasting Time-Series with Multiple Seasonal Patterns. *European Journal of Operational Research* 191, 207–222 (2008)
6. Sharda, R., Patil, R.B.: Connectionist Approach to Time Series Prediction: An Empirical Test. *Journal of Intelligent Manufacturing* 3, 317–323 (1992)
7. Nelson, M., Hill, T., Remus, T., O'Connor, M.: Time Series Forecasting Using NNs: Should the Data Be Deseasonalized First? *Journal of Forecasting* 18, 359–367 (1999)
8. Zhang, G.P., Qi, M.: Neural Network Forecasting for Seasonal and Trend Time Series. *European Journal of Operational Research* 160, 501–514 (2005)
9. Zhang, G.P., Kline, D.M.: Quarterly Time-Series Forecasting with Neural Networks. *IEEE Transactions on Neural Networks* 18(6), 1800–1814 (2007)
10. Cleveland, R.B., Cleveland, W.S., McRae, J.E., Terpenning, I.: STL: A Seasonal-Trend Decomposition Procedure Based on Loess. *Journal of Official Statistics* 6, 3–73 (1990)
11. Atiya, A., El-Shoura, S., Shaheen, S., El-Sherif, M.: A Comparison Between Neural Networks Forecasting Techniques—Case Study: River Flow Forecasting. *IEEE Transactions on Neural Networks* 10(2), 402–409 (1999)
12. Soares, L.J., Medeiros, M.C.: Modeling and Forecasting Short-Term Electricity Load: A Comparison of Methods with an Application to Brazilian Data. *International Journal of Forecasting* 24, 630–644 (2008)

13. Dudek, G.: Similarity-based Approaches to Short-Term Load Forecasting. In: Zhu, J.J., Fung, G.P.C. (eds.) *Forecasting Models: Methods and Applications*, pp. 161–178. iConcept Press (2010), http://www.iconceptpress.com/site/download_publishedPaper.php?paper_id=1009170201
14. Lendasse, A., Verleysen, M., de Bodt, E., Cottrell, M., Gregoire, P.: Forecasting Time-Series by Kohonen Classification. In: *Proc. the European Symposium on Artificial Neural Networks*, pp. 221–226. Bruges, Belgium (1998)
15. Hart, E., Timmis, J.: Application Areas of AIS: The Past, the Present and the Future. *Applied Soft Computing* 8(1), 191–201 (2008)
16. Dudek, G.: Artificial Immune System for Short-term Electric Load Forecasting. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) *ICAISC 2008. LNCS (LNAI)*, vol. 5097, pp. 1007–1017. Springer, Heidelberg (2008)
17. Theodoridis, S., Koutroumbas, K.: *Pattern Recognition*, 4th edn. Elsevier Academic Press (2009)
18. Hyndman, R.J., Khandakar, Y.: Automatic Time Series Forecasting: The Forecast Package for R. *Journal of Statistical Software* 27(3), 1–22 (2008)
19. Hyndman, R.J., Koehler, A.B., Ord, J.K., Snyder, R.D.: *Forecasting with Exponential Smoothing: The State Space Approach*. Springer (2008)