Artificial Immune Clustering Algorithm to Forecasting Seasonal Time Series

Grzegorz Dudek

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

Abstract. This paper concentrates on the forecasting time series with multiple seasonal periods using new immune inspired method. Proposed model includes two populations of immune memory cells – antibodies, which recognize patterns of the time series sequences represented by antigens. The empirical probabilities, that the pattern of forecasted sequence is detected by the *j*th antibody from the first population while the corresponding pattern of input sequence is detected by the *i*th antibody from the second population, are computed and applied to the forecast construction. The suitability of the proposed approach is illustrated through an application to electrical load forecasting.

Keywords: artificial immune system, cluster analysis, 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 m 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. 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. A variety of methods have been proposed for forecasting seasonal time series. These include: exponential smoothing, seasonal ARMA, artificial neural networks, dynamic harmonic regression, vector autoregression, random effect models, and many others.

The proposed approach belongs to the class of similarity-based methods [1] 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 to detection of similar patterns of

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part I, LNCS 6922, pp. 468–477, 2011.

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 forecasted ones. The empirical probabilities, that the pattern of forecasted sequence is detected by the *j*th AB from the first population while the corresponding pattern of input sequence is detected by the *i*th AB from the second population are computed and applied to the forecast construction. Another AIS to forecasting of seasonal time series was presented in [2].

The merits of AIS lie in its pattern recognition and memorization capabilities. The application areas for AIS can be summarized as [3]: 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. A good introduction to AIS are [4] and [5].

2 Similarity-Based Forecasting Methods

The similarity-based (SB) methods use analogies between sequences of the time series with periodicities. A course of a time series can be deduced from the behavior of this time series in similar conditions in the past or from the behavior of other time series with similar changes in time. In the first stage of this approach, the time series is divided into sequences of length n, which usually contain one period. Fig. 1 shows the periodical time series, where we can observe yearly, 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 daily periods.

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 real time series elements. The input and output (forecast) patterns are defined: $\mathbf{x} = [x_1\ x_2\ \ldots\ x_n]^T$ and $\mathbf{y} = [y_1\ y_2\ \ldots\ y_n]^T$, 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 similarity measures are based on the distance or correlation measures.

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 possibility of calculation of real forecast of time series elements.

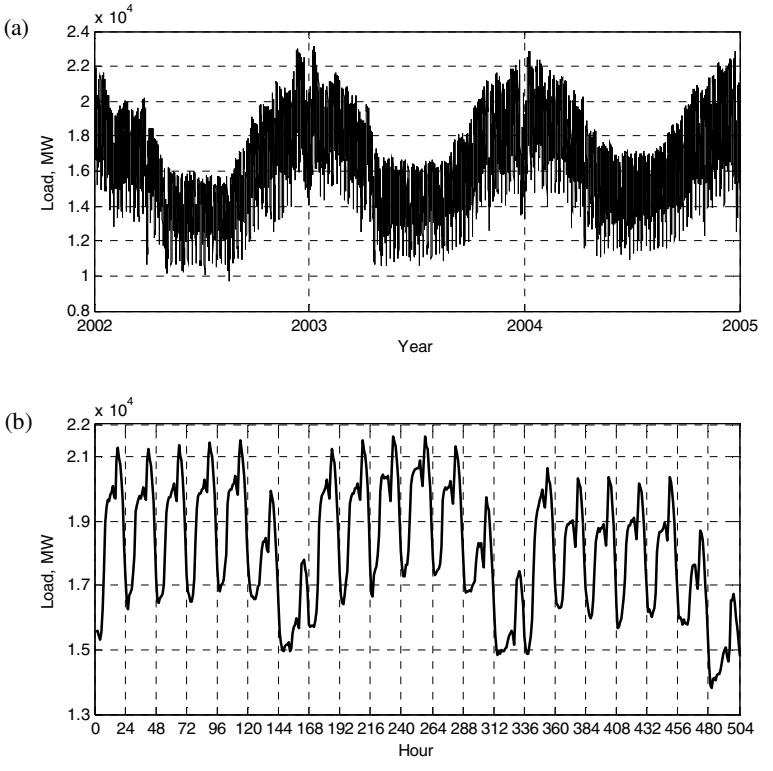

Fig. 1. The load time series of Polish power system in three year (a) and three week (b) intervals

The forecast pattern $\mathbf{y}_i = [y_{i,1}\ y_{i,2}\ \ldots\ y_{i,n}]$ encodes the following real time series elements z in the forecast period $i+\tau$: $\mathbf{z}_{i+\tau} = [z_{i+\tau,1}\ z_{i+\tau,2}\ \ldots\ z_{i+\tau,n}]$, and the input pattern $\mathbf{x}_i = [x_{i,1}\ x_{i,2}\ \ldots\ x_{i,n}]$ maps the time series elements in the period i preceding the forecast period: $\mathbf{z}_i = [z_{i,1}\ z_{i,2}\ \ldots\ z_{i,n}]$. 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.

Some functions mapping the original feature space Z into the pattern spaces X and Y – $f_x : Z \to X$ and $f_y : Z \to Y$ – are presented below.

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

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

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

where: $i = 1, 2, \ldots, M$ – the period number, $t = 1, 2, \ldots, n$ – the time series element number in the period i, τ – the forecast horizon, $z_{i,t}$ – the tth time series element in the period i, $\bar{z}_i$ – the mean value of elements in period i, $z' \in \{\bar{z}_i, z_{i\text{-}1,t}, z_{i\text{-}7,t}, z_{i,t\text{-}1}\}$, $z'' \in \{\bar{z}_i, z_{i+\tau\text{-}1,t}, z_{i+\tau\text{-}7,t}\}$, 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 they 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.

The components of the x-patterns defined using equations (2) and (3) express, respectively, indices and differences of time series elements in the following whiles of the ith period.

Forecast patterns are defined using analogous functions to input pattern functions f_x, but they are encoded using the time series elements 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 $f_y^{-1}(y_{i,t}, \Psi_i)$.

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 [1].

The forecasting procedure in the case of SB methods can be summarized as follows: (i) elimination of the trend and seasonal variations of periods longer than n using pattern functions f_x and f_y, (ii) forecasting the pattern $\mathbf{y}$ using similarities between x-patterns, and (iii) 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 (ABx) detects antigens representing patterns $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$: AGx, while the population of y-antibodies (ABy) detects antigens representing patterns $\mathbf{y} = [y_1, y_2, \ldots, y_n]^T$: AGy. The vectors $\mathbf{x}$ and $\mathbf{y}$ form the epitopes of AGs and paratopes of ABs. ABx 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 ABy 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 kth ABx can be written as a pair $\{\mathbf{p}_k, r_k\}$, where $\mathbf{p}_k = \mathbf{x}_k$, and the kth ABy 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 $P(ABy_k \mid ABx_j)$, $j, k = 1, 2, \ldots, N$, that the ith AGy stimulates (is recognized by) the kth ABy, when the corresponding ith AGx stimulates the jth ABx, are determined. These probabilities are calculated for each pair of ABs on the basis of recognition of the training population of AGs.

In the forecasting phase the new AGx, 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 PCy paratopes weighted by the conditional probabilities and affinities.

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

Step 1. Loading of the training population of antigens. An AGx represents a single $\mathbf{x}$ pattern, and AGy represents a single $\mathbf{y}$ pattern. Both populations AGx and AGy 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 population. 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: $\mathbf{p}_k = \mathbf{x}_k$, $\mathbf{q}_k = \mathbf{y}_k$, $k = 1, 2, \ldots, N$. 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 kth 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 kth ABy paratope – $\mathbf{q}_k$.

The measure of similarity of the ith AGx to the kth ABx is the affinity:

$$a(\mathbf{p}_k, \mathbf{x}_i) = \begin{cases} 0, & \text{if } d(\mathbf{p}_k, \mathbf{x}_i) > r_k \text{ or } r_k = 0 \\ 1 - \dfrac{d(\mathbf{p}_k, \mathbf{x}_i)}{r_k}, & \text{otherwise} \end{cases}, \tag{4}$$

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

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

$$\delta_{k,i} = \frac{100}{n}\sum_{t=1}^{n}\frac{|\, z_{k+\tau,t} - f_y^{-1}(y_{i,t}, \Psi_k)\,|}{z_{k+\tau,t}}, \tag{5}$$

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

If the condition $\delta_{k,i} \le \delta_y$ is satisfied, where δ_y is the error threshold value, it is assumed that the ith AGy is similar to the kth ABy, and ith AGx, paired with this AGy, is classified to class 1. When the above condition is not met the ith 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 the kth ABx is defined as follows:

$$r_k = d(\mathbf{p}_k, \mathbf{x}_A) + c[d(\mathbf{p}_k, \mathbf{x}_B) - d(\mathbf{p}_k, \mathbf{x}_A)], \tag{6}$$

where B denotes the nearest AGx of class 2 to the kth 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 the kth ABy is calculated similarly to the above:

$$s_k = d(\mathbf{q}_k, \mathbf{y}_A) + b[d(\mathbf{q}_k, \mathbf{y}_B) - d(\mathbf{q}_k, \mathbf{y}_A)], \tag{7}$$

where B denotes the nearest AGy of class 2 to the kth 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 ith AGy is classified to class 1, if for the ith AGx paired with it, there is $\varepsilon_{k,i} \le \varepsilon_x$, 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 kth ABx. These elements are forecasted using the epitope of the ith AGx:

$$\varepsilon_{k,i} = \frac{100}{n}\sum_{t=1}^{n}\frac{|\, z_{k,t} - f_x^{-1}(x_{i,t}, \Psi_k)\,|}{z_{k,t}}, \tag{8}$$

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

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

$$a(\mathbf{q}_k, \mathbf{y}_i) = \begin{cases} 0, & \text{if } d(\mathbf{q}_k, \mathbf{y}_i) > s_k \text{ or } s_k = 0 \\ 1 - \dfrac{d(\mathbf{q}_k, \mathbf{y}_i)}{s_k}, & \text{otherwise} \end{cases} . \tag{9}$$

$a(\mathbf{q}_k, \mathbf{y}_i) \in [0, 1]$ 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 **y** are similar to the paratope $\mathbf{q}_k$, and (ii) the AGx paired with them have epitopes **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 paratope with an error no greater than the threshold value. This ensures that the forecast error for the pattern **x** (**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.

MAPE here is used as an error measure ((5) and (8)) but other error measures can be used.

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_i), $i = 1, 2, \ldots, N$, 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 a new AGx, representing the pattern **x***, is presented to the immune memory. Let Ω be a set of ABx stimulated by this AGx. The forecasted pattern **y** corresponding to **x*** is estimated as follows:

$$\hat{\mathbf{y}} = \sum_{k=1}^{N} w_k \mathbf{q}_k , \tag{10}$$

where

$$w_k = \frac{\sum_{j \in \Omega} P(ABy_k \mid ABx_j) a(\mathbf{p}_j, \mathbf{x}^*)}{\sum_{l=1}^{N} \sum_{j \in \Omega} P(ABy_l \mid ABx_j) a(\mathbf{p}_j, \mathbf{x}^*)} . \tag{11}$$

The forecast is calculated as the weighted mean of paratopes $\mathbf{q}_k$. Weights w express the products of the affinity of stimulated memory cells ABx to the AGx and probabilities $P(ABy_k \mid ABx_j)$. $\sum_{k=1}^{N} w_k = 1$.

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 – error thresholds (δ_y and ε_x) and parameters tuning the cross-reactivity thresholds (b and c). Increasing the values of these parameters imply an increase in size of clusters, an increase of the model bias and reduction of its variance.

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

4 Application Example

The described above AIS was applied to the next day electrical load curve forecasting. 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.

The series studied in this paper 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 was divided into training and test parts. The test set contained 31 pairs of patterns from July 2004. The training set contained patterns from the period from 1 January 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 adaptive 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).

The model parameters were determined using the grid search method on the training subsets in the local version of leave-one-out procedure. In this procedure not all patterns are successively removed from the training set 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) δ_y = 1.00, 1.25, …, 3.00, ε_x = 1.00, 1.25, …, δ_y, at the constant values of $b = c = 1$, and (ii) $b = c$ = 0, 0.2, …, 1.0, at the optimal values of δ_y and ε_x determined in point (i).

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

The forecast results are shown in Fig. 2 and 3. *MAPE* for the test part of time series was 0.92 and its standard deviation – 0.72.

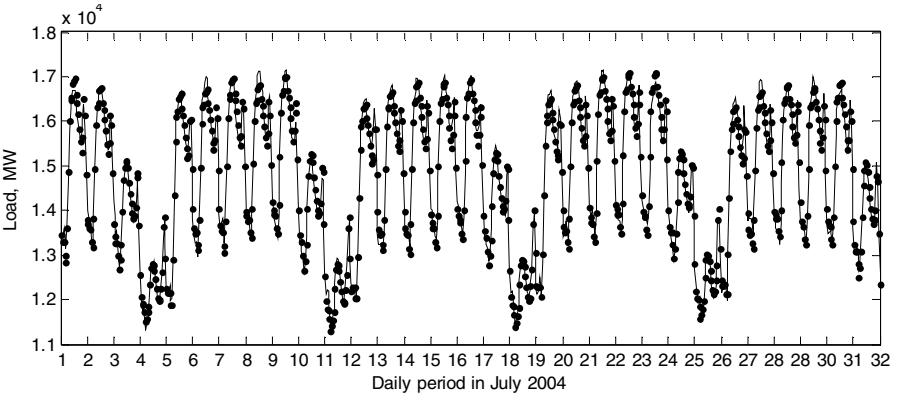

Fig. 2. Test part of the time series – July 2004 (*solid lines*) and its forecast (*dots*)

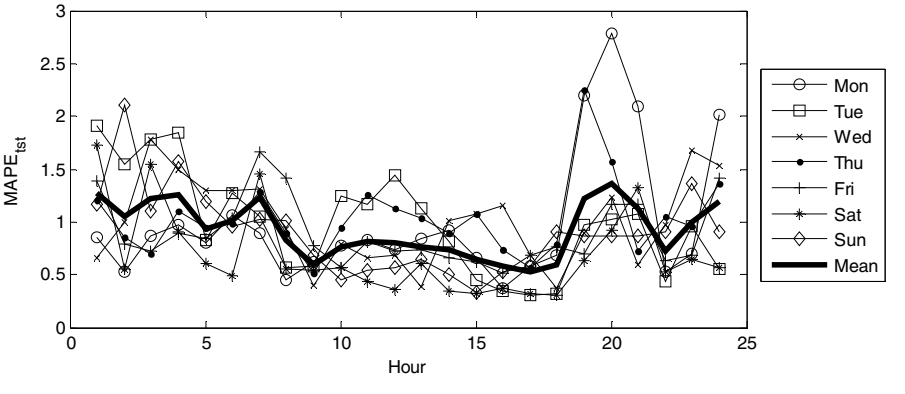

Fig. 3. $MAPE_{tst}$ for each day type and hour of the daily period

5 Conclusion

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 [1], [6], 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 results are stable and the learning process is rapid.

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. Dudek, G.: Similarity-based Approaches to Short-Term Load Forecasting. In: Forecasting Models: Methods and Applications, pp. 161–178. iConcept Press (2010), http://www.iconceptpress.com/site/download_publishedPaper.php?paper_id=100917020141
2. 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)
3. Hart, E., Timmis, J.: Application Areas of AIS: The Past, the Present and the Future. Applied Soft Computing 8(1), 191–201 (2008)
4. Perelson, A.S., Weisbuch, G.: Immunology for Physicists. Rev. Modern Phys. 69, 1219–1267 (1997)
5. De Castro, L.N., Timmis, J.: Artificial Immune Systems as a Novel Soft Computing Paradigm. Soft Computing 7(8), 526–544 (2003)
6. 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, Bruges, Belgium, pp. 221–226 (1998)