IDENTIFICATION OF TYPE DAILY DIAGRAMS OF ELECTRIC CONSUMPTION BASED ON CLUSTER ANALYSIS OF MULTI-DIMENSIONAL DATA BY NEURAL NETWORK

Bohumír Garlík[*], *Miloš Křivan*[†]

Abstract: This article establishes a mathematical description of a self-organizing neural network used for cluster analysis with a subsequent sampling of its effectiveness as an example of identification of the type daily diagrams of electric energy-consumption of complex intelligent buildings within an electric micro grid, namely for a typical work day and a day off on the basis of its annual history. The mentioned type daily diagram can be used to predict power consumption. This method is given in the context of the commonly used procedure for cluster analysis. The experiment was processed in the computer program Artint © 2010.

Key words: *Artificial Neural Network (ANN), Self-Organizing Map (SOM), Kohonen map, counter-propagation neural network, cluster analysis, Type Daily Diagrams (TDD), load prediction, intelligent building, micro grid*

Received: March 3, 2013
Revised and accepted: June 4, 2013

1. Introduction

With the development of computing technology and the growth of its computational power, there has been an increasing focus on artificial intelligence methods since World War 2. These methods include terms such as artificial neural networks, fuzzy

[*]Bohumír Garlík
Department of Microenvironmental and Building Services Engineering, Czech Technical University in Prague, Thákurova 7, 160 00 Prague 6, E-mail: bohumir.garlik@fsv.cvut.cz; Department of Management and Artificial Intelligence, Research Institute for Intelligent Buildings, Brno, Drobného 16, 602 00 Brno, Czech Republic, , E-mail: bohumir.garlik@vuib.cz
[†]Miloš Křivan
Department of Management and Artificial Intelligence, Research Institute for Intelligence Buildings, Brno, Drobného 16, 602 00 Brno, Czech Republic, E-mail: pwrnetexp@gmail.com

sets or evolutionary algorithms. However, their massive utilization in practical applications across all human activities only occurred in the eighties of the previous century, due to the development of personal computers.

Artificial neural networks are used to process and evaluate incomplete, indeterminate or inconsistent information, especially for tasks involving recognition, diagnostics, classification of objects with respect to provided categories or prediction of the time development of the given variable, compression and coding information, noise filtering, extrapolation or interpolation of the trends of a given variable and last but not least the cluster analysis of multidimensional data, as described in this article. More precisely, by recognition we mean the recognition of visual or acoustic information, such as written text or spoken words, and by diagnostics we mean diagnostics of the residual service life of technological equipment or human organisms including bio-medicinal signals such as ECG or EEG.

2. Competitive Model of Neural Network

We define an artificial neural network as the oriented graph with edges and vertices rated dynamically, i.e. as the ordered quintuplet $[V, E, \varepsilon, w, y]$:

- V set of vertices (neurons)
- E set of edges (synapses)
- ε mapping edges with incidence vertices ($\varepsilon : E \to V \times V$)
- w dynamic valuation of edges ($w : \varepsilon(E) \times \mathbf{T} \to \mathbb{R}$)
- y dynamic valuation of vertices ($y : V \times \mathbf{t} \to \mathbb{R}$).

The vector $\vec{w}(T) = [w_{ij}(T)|[i,j] \in V \times V]$ is called the *network configuration* in time T, $(\forall [i,j] \notin \varepsilon(E) \Rightarrow w_{ij}(T) = 0)$ and the vector $\vec{y}(t) = [y_i(t)|i \in V]$ is called the *network state* in time t. The configuration respectively state of the network as a vector function of time T or t is referred to as *adaptive dynamics* respectively *active dynamics* of the neural network. Active or adaptive dynamics of a neural network in continuous time can be defined as a vector solution of the following systems of differential equations [1]:

$$\frac{d}{dt}x_j(t) + x_j(t) = \sum_i f_i(x_i(t - \Delta t))w_{ij} - \vartheta_j \tag{1}$$

respectively

$$\frac{d}{dT}w_{ij}(T) + \beta g_j(x_j(T))w_{ij}(T) = \alpha f_i(x_i(T))g_j(x_j(T)) \tag{2}$$

$i, j \in V, \alpha, \beta \in \langle 0, 1 \rangle$, and then analogously to biological neural network we have:

- x_i potential of the i-th neuron
- f_i activation function of the i-th neuron ($f_i(x_i) = y_i$)
- g_j adaptation function of the j-th neuron
- ϑ_j threshold of the j-th neuron
- w_{ij} synaptic weight links of the i-th neuron to the j-th neuron

α	measure of plasticity of synapses
β	measure of elasticity of synapses
Δt	signal delay time.

If we replace in (1) and (2) the derivations by analogous expressions for discrete time:

$$\frac{d}{dt}x_j(t) \equiv \frac{x_j(t+1)-x_j(t)}{t+1-t} \qquad \frac{d}{dT}w_{ij}(T) \equiv \frac{w_{ij}(T+1)-w_{ij}(T)}{T+1-T}$$

and if we set $\Delta t = 0$, then we obtain the following systems of difference equations and the vectors of its solutions define the active and adaptive dynamics of a neural network in discrete time:

$$x_j(t+1) = \sum_i f_i(x_i(t))w_{ij} - \vartheta_j \quad y_j(t+1) = f_j\left(\sum_i y_i(t)w_{ij} - \vartheta_j\right) \tag{3}$$

respectively

$$w_{ij}(T+1) = (1-\beta g_j(x_j(T)))\, w_{ij}(T) + \alpha f_i(x_i(T))g_j(x_j(T)) \tag{4}$$

$i, j \in V$.

We approximate the dependence of the state on the potential of the neuron by sigmoid function: $f(x) = \frac{1}{(1+e^{-px})}$ where the parameter $p > 0$ expresses the slope of the sigmoid. For a slope approaching zero or infinity we get the activation function in the shape of linearity respectively non-linearity:

$$\lim_{p\to 0} f(x) = \frac{1}{2} \qquad \lim_{p\to\infty} f(x) = 0 \qquad x < 0 \qquad \lim_{p\to\infty} f(x) = 1 \qquad x > 0$$

and we can finally define the following *network function*: $\vec{F}(\vec{x}(t)) = \vec{y}(t+\Delta t)$, where Δt is the response time of the network.

Let us divide the population of the neurons in V to two disjoint populations V_1 and V_2 ($V_1 \cup V_2 = V, V_1 \cap V_2 = \emptyset, |V_1| = n, |V_2| = m$), and let us connect them by edges so that there is an edge from each neuron in V_1 to each neuron in V_2 ($\varepsilon(E_1) = V_1 \times V_2$), i.e. the network is oriented from V_1 to V_2 and V_1 respectively V_2 is then understood as the input respectively output population. Let us, furthermore, connect neurons in V_2 by edges so that there is an edge from each neuron in V_2 to every other neuron in V_2 ($\varepsilon(E_2) = V_2 \times V_2 - \{[j,j] | j \in V_2\}$).

Let us choose the activation function of neurons of population V_1 as an identity, i.e. modified linearity and the activation function of neurons of population V_2 as non-linearity. Then, during the active dynamics for constantly applied stimulus attached to population V_1 we can express the active dynamics (3) for $y_k(0) = 0$ as follows:

$$y_j(t+1) = f_j\left(\sum_k y_k(t)w_{kj} - \vartheta_j\right) \qquad -\vartheta_j = \sum_i x_i(0)w_{ij} \tag{5}$$

$i \in V_1, j, k \in V_2$ and let us call the parameter $-\vartheta_j$ the gain potential of the j-th neuron.

Let us choose the following initial conditions for the network configuration $w_{kj}(0) = -2$, $w_{ij}(0) = r_{ij}$, and let us add templates from the training set specified in the form $\{\vec{a}(T)|T \in \Delta T\}$ for the population V_1, where ΔT is the network adaptation period. If we only let the mutual links between neurons in V_1 and V_2 adapt, and if we select the adaptation function for the neurons in V_2 to match the activation functions, then, assuming elasticity is equal to plasticity ($\alpha = \beta$), we can express the adaptive dynamics (4) as follows:

$$\begin{aligned} w_{kj}(T) &= w_{kj}(T-1) \\ w_{ij}(T) &= w_{ij}(T-1) + \alpha y_j(T)(a_i(T) - w_{ij}(T-1)) \end{aligned} \tag{6}$$

$i \in V_1, j, k \in V_2, T \in \Delta T = \{1, -, N\}$, where N respectively r_{ij} is the number of patterns of training set respectively the value specified of the random number generator.

In each step of the adaptive dynamics (6) it is required to designate the states of neurons in V_2, i.e. the steps of the adaptive dynamics are conditioned by the active dynamics, which, from the perspective of adaptive dynamics, runs infinitely fast. Thus, the state of V_2 is determined synchronously with the state of V_1.

Let us assign to each neuron in V_2 a weight vector $\vec{w}_j = [w_{ij}|i \in V_1]$. Then the neurons in V_2 together with the edges E_2 and the active dynamics (5) form a Hopfield optimization network [4] with the following energy function:

$$E(\vec{y}) = \sum_j \sum_k y_k y_j + \sum_j y_j \vartheta_j \qquad -\vartheta_j = \sum_i a_i(T) w_{ij} = \vec{a}(T) \cdot \vec{w}_j \tag{7}$$

$i \in V_1, j \in V_2, k \in V_2 - \{j\}$.

If the vectors of the training set respectively the weight vectors are normal, then the received potential of each neuron will comply with $-\vartheta = \cos\varphi$ and the distance between the specified vectors can be defined as the angle $\varphi \in \langle 0, \pi \rangle$ between them. The energy function specified above will then reach its minimum if and only if only one neuron in V_2 is excited, specifically the neuron with the maximum potential gain (7) – the so-called *gain* neuron.

The process of energy minimization of the state of V_2 realized by the active dynamics (5), when the excited neuron with the maximum potential gain inhibits (by negative links) other neurons, is called *lateral inhibition.* Lateral inhibition, which designates a corresponding state of the population of V_2 based on the presented training template, replaces the missing template association in the training set – in other words, it replaces the statement of a teacher, and we thus speak of teacher-less learning.

Lateral inhibition in each adaptation step will ensure the adaptation of only the weight vector corresponding to the k-th gain neuron, i.e. of the weight vector as per the above-specified non-Euclidean metric of the closest presented training set template, to which it will advance on the surface of an n-dimensional ball of unit radius by an adaptation step proportional to the plasticity of the synapse (Fig. 1):

$$\vec{w}_k(T) = \vec{w}_k(T-1) + \alpha(\vec{a}(T) - \vec{w}_k(T-1)) \tag{8}$$

and the gain neuron thus won the *competition* for the presented template of the training set. The normality of the adapted weight vector will be ensured by its subsequent normalization.

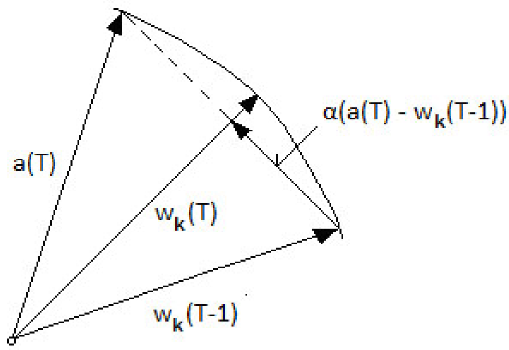

Fig. 1 *Adaptation step.*

The objective function (9) will reach its minimum if and only if the weight vector is on the position with a minimal sum of distances from all vectors of the training set which excite the appropriate neuron, i.e. in the center of the cluster of the specified vectors:

$$G(\vec{w}_j) = \frac{1}{2}\sum_T y_j(T)\sum_i (a_i(T) - w_{ij})^2 \quad -\frac{\partial G(\vec{w}_j)}{\partial w_{ij}} = \sum_T y_j(T)(a_i(T) - w_{ij}) \tag{9}$$

$i \in V_1, j \in V_2, T \in \Delta T$.

Adaptive dynamics (6) is a gradient descent on a lower-bounded objective function (9), and so, assuming that the vectors of the training set form clusters in the n-dimensional space whose size corresponds to the cardinality of V_2, the (initially randomly located) weight vectors will converge towards the centers of these clusters during adaptive dynamics.

Let us define the following categories of normal vectors:

$$C_k = \{\vec{x} \in \Omega | \varphi(\vec{x}, \vec{w}_k) < \varphi(\vec{x}, \vec{w}_j)\} \quad \Omega = \{\vec{x} \in \mathbb{R}^n \| \vec{x}| = 1\} \tag{10}$$

$k \in V_2, j \in V_2 - \{k\}$ and φ is a non-Euclidean metric, i.e. the angle between the vectors.

The function of the network will thus assign, during lateral inhibition, a vector of the canonical basis of an m-dimensional space with a one on the k-th position to an arbitrary normal network input, if and only if the network input lies in the k-th category (10). The function of the network of the competitive model can thus be understood as a classification with respect to the categories specified above.

If we set $|V_2| = m^2$, then we can interpret the neurons in V_2 as elements of a square $m \times m$ grid. Let us define the square neighborhood of the r-th order of the k-th element of the grid as the set containing all grid elements which lie at a distance of less than or equal to order r, i.e. $\sigma(k,r) = \{j \in V_2 | \rho(k,j) \leq r\}$, where ρ is the metric defined on the grid as the neighborhood of elements of the appropriate order, and let us adjust the adaptive dynamics (8) for the k-th gain neuron:

$$\vec{w}_j(T) = \vec{w}_j(T-1) + \alpha_j(T)\left(\vec{a}(T) - \vec{w}_j(T-1)\right) \tag{11}$$

$j \in \sigma(k,r)$ and the plasticity drops globally with the time of the adaptive dynamics and locally with the order of the distance of the appropriate neuron from the gain neuron in the population grid of V_2.

The adjustment of the adaptive dynamics specified above generalize lateral inhibition by the extension of the excitation of the gain neuron to its neighborhood, which links the above-specified metric φ with the above-specified metric ρ. If the vectors of the training set are randomly distributed in the n-dimensional space in accordance with some distribution function, then after the adaptation of the network the weight vectors will be randomly distributed in the same area in accordance with the same distribution function.

If we present a training set on an adapted network in active mode, then the map of the frequency of excitations of neurons in V_2, the so-called *Kohonen map* [2] will provide a mapping of the clusters of vectors of the training set in an n-dimensional space. Such a generalized competitive model, under assumption of a sufficiently large cardinality of V_2, performs the cluster analysis of the training set, i.e. determines the number of clusters and their distribution in the n-dimensional space.

Let us adjust the topology of the already adapted competitive model by adding a population set V_3, connected by edges to the population V_2 so that there is an edge from each neuron in V_2 to each neuron in V_3 ($\varepsilon(E_3) = V_2 \times V_3$). Let the new output population V_3 have the same cardinality as the input population V_1, and thus the population V_2 becomes a hidden population.

Let us set the weights of edges E_3 as follows: $w_{jq(i)} = w_{ij}, i \in V_1, j \in V_2, q(i) \in V_3$, where $q(i)$ is the image of the i-th neuron of population V_1 in population V_3. The output population V_3 together with the weighted edges E_3 thus forms an image of the output population V_1 together with the weighted edges E_1 mirrored over the hidden population V_2, a phenomenon which we call *counter propagation* [3] of the synaptic weights of edges E_1 to edges E_3 in the direction of the orientation of edges. Let us select the activation functions of neurons in V_3 identically to the activation functions of neurons in V_1. Then, during active dynamics after the stabilization of the state of the population of V_2, the excitation of the k-th gain neuron will lead to the following values of potentials of neurons in V_3:

$$x_{q(i)} = \sum_j y_j w_{jq(i)} = w_{kq(i)} = w_{ik}$$

$i \in V_1, j \in V_2$, then stimulus $\vec{x} \in C_k$ implies the following network function: $\vec{F}(\vec{x}) = \vec{w}_k$.

The function of the network in the competitive model with forward propagation of weights will thus assign a *prototype* (the closest weight vector) to each normal network input. Prototypes lie in the centers of the appropriate clusters and thus represent these clusters – they are their typical representatives.

The competitive model with forward propagation of weights and the Kohonen map may be used to reduce the cardinality of multidimensional data, which may be replaced by a set of prototypes of their elements with cardinality of m^2.

3. Experiment

The goal of this experiment is the identification of type daily diagrams of hourly consumption of electric energy by a complex of intelligent buildings within an electric micro-grid on a workday in the middle of the workweek, i.e. Wednesday, and on a non-work day, specifically using a non-work day before a non-work day and also a non-work day before a work day, i.e. on Saturday and Sunday, based on the recorded annual history of hour consumption of electric energy by the complex.

To allow the measuring of the efficiency of the utilized cluster analysis method, the annual history of hourly consumption of electric energy has been artificially modeled so that a typical daily diagram may be compared to a certain standard. The default standards of daily diagrams of hourly consumptions were the characteristic hourly developments of the consumption of the above-listed three days, where each hourly consumption of each of these was randomly modified by a random number generator with a normal probability distribution, as many times as was necessary to fill the annual history of hourly consumption, i.e. two hundred sixty times for Wednesday and fifty two times for Saturday and Sunday each, i.e. three hundred sixty four daily diagrams in total. A demonstration of the source code, in Fortran, of the used computer program is shown below:

```
.
.
.
DO J=1,NH                    !NH - NUMBER OF HOURS (24)
X=HL(J)                      !HL - HOURLY LOAD
DO I=1,ND                    !ND - NUMBER OF DAYS (260/52)
DD(I,J)=RANDOM(SEED,X)       !DD - YEAR HISTORY OF DAILY DIAGRAMS
ENDDO
ENDDO
.
.
.
```

```
FUNCTION RANDOM(IX,X)
AM=X                    !AM - MEAN VALUE
SD=0.01*AM              !SD - STANDARD DEVIATION
A=0.0
DO I=1,12
IY=IX*65539
IF(IY)1,2,2
1 IY=IY+2147483647+1
2 Y=IY
Y=Y*0.4656613E-9
IX=IY
A=A+Y
ENDDO
RANDOM=(A-6.0)*SD+AM
RETURN
END
```

The attached images each contain four examples of randomly modeled daily diagrams of Saturday (Fig. 2), Sunday (Fig. 3) and Wednesday (Fig. 4).

This model of the annual history of hourly consumption of energy, i.e. a rectangular matrix with 364 rows and 24 columns, which represents our multidimensional data, then forms the training set during adaptive dynamics presented to the competitive model of the artificial neural network on a day-by-day basis, i.e. the input always being one day containing 24 attributes. The topology of the network used is then formed by two layers of neurons: the input layer with 24 neurons and the output layer with 225 neurons, which is organized in a square 15 by 15 grid. The plasticity dropped exponentially during network adaptation, from the default value of 1 to its final value of 0.005, and the order of the neighborhood of the gain neuron during network adaptation dropped exponentially from its default value of 7 to the final value of 0, i.e. the neighborhood of the gain neuron of the default order covered the whole output layer of the network and by the end of the network adaptation it degenerated to only contain the gain neuron.

After network adaptation, the active dynamics process was initiated by representing the training samples. This has led to the appearance of a Kohonen Map (Fig. 5), which depicts three well-separated clusters, two of which are smaller and likely correspond to Saturday and Sunday, and one of which is larger and likely corresponds to Wednesday. The map may be interpreted as a map of the landscape with a marked elevation (above sea-level) for each spot, where each island represents a cluster of daily diagrams and the highest point, marked by a white circle on the figure, represents the prototype representing the given cluster, i.e. the type daily diagram of the appropriate period. The map should then be understood as the surface of a globe, i.e. the top respectively left rim is identified with the bottom respectively top rim. Neurons with a zero number of excitations correspond to weight vectors which were never the closest to any of the presented templates in the training set, i.e. during adaptive dynamics these did not come sufficiently close to the clusters of training templates. Forward propagation may be used to extract the necessary weight vectors from the configuration of the learned network during active dynamics. These results in the sought daily diagrams are provided

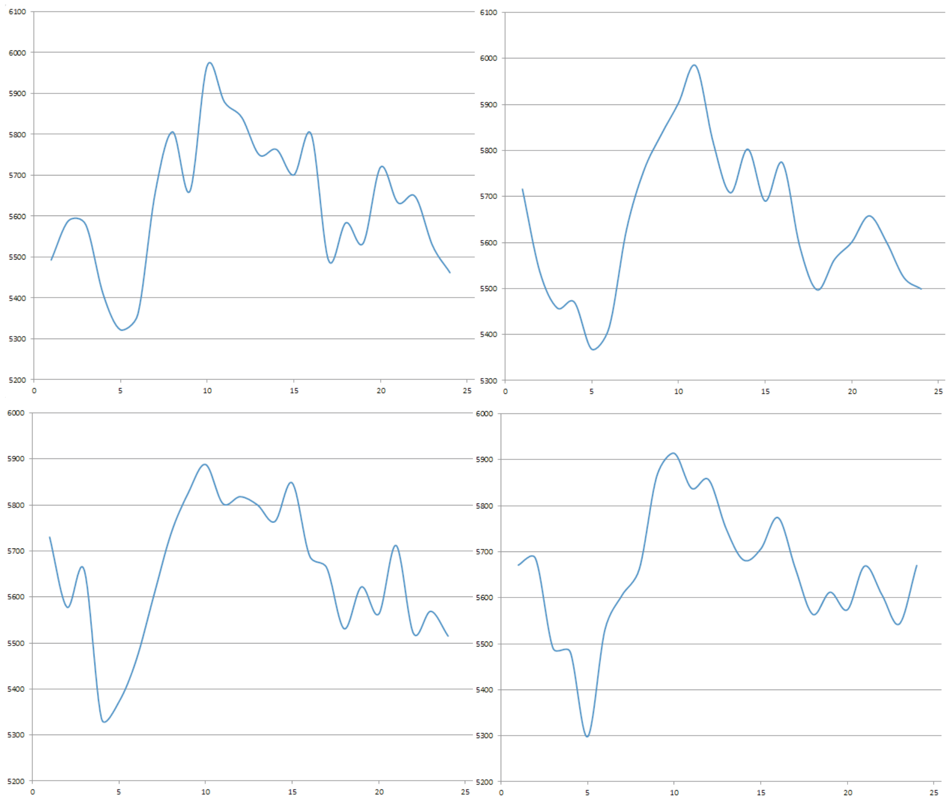

Fig. 2 *Examples of randomly modeled diagrams of Saturday.*

in Figs. 6-8 together with a comparison to the appropriate standard.

Although individual daily diagrams of the annual history (Figs. 2-4) are mutually relatively different, and for instance the fourth example of a Saturday daily diagram (Fig. 2) is more similar to a characteristic Sunday daily diagram, the resulting type daily diagrams are very similar to the appropriate standards (Figs. 6-8). This documents the high efficiency of the utilized cluster analysis method. Tab. I contains a numerical comparison of the type daily diagrams with the appropriate standards, and their average respectively maximum deviation is 0.2% respectively 0.5%.

4. Conclusion

The mathematical description provided above and the experiment which was carried out implies that the cluster analysis method used in the article clearly fulfills the five generally formulated attributes of an ideal method:

- The method does not require any a priori information from the user

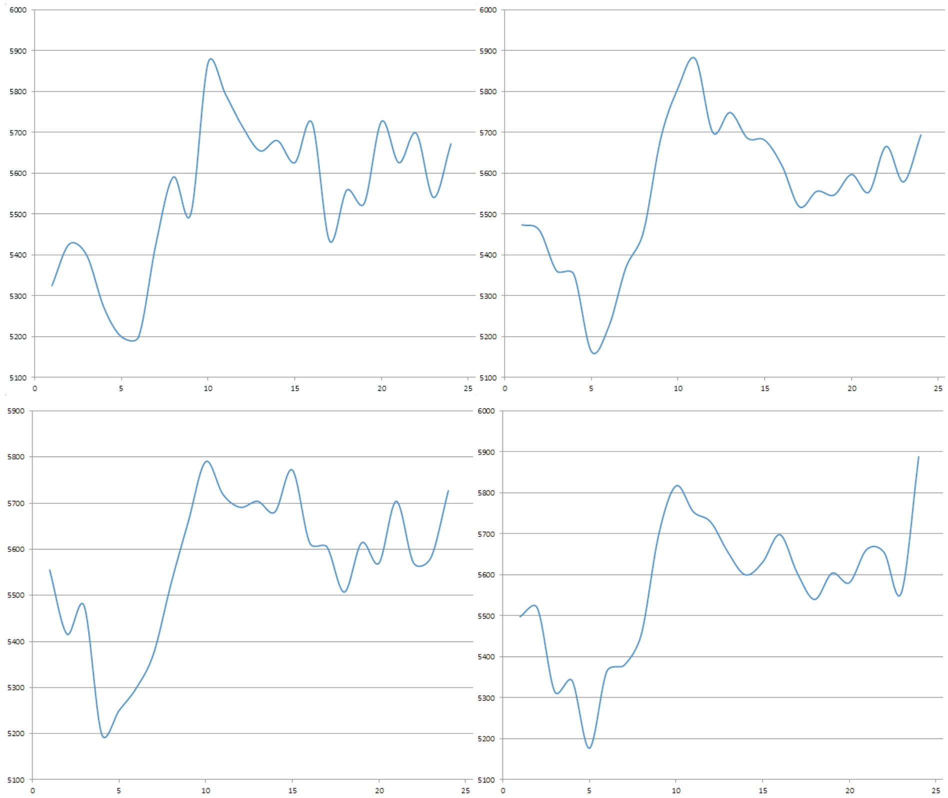

Fig. 3 *Examples of randomly modeled diagrams of Sunday.*

- The method identifies clusters of arbitrary shape and density of contained objects
- The method is not sensitive to the order of the presented observations of objects
- The method is robust towards remote observations of objects
- The method is capable of analyzing a set of a large number of observations with a large number of variables during a single presentation

and, as may be seen from the Kohonen map, it is also hierarchical, since each cluster in the map also contains sub-clusters.

The most frequently used methods of cluster analysis include the non-hierarchical *k-means* method, which is included in basically all statistical programs. Its algorithm is based on an a priori selection of the number of clusters, including random generation of their centroids and subsequent inclusion of individual objects into clusters based on their distance (Euclidean metric) from the centroids, which are continuously updated based on the average values of coordinates of all objects in the given cluster. This method does not comply with even one of the properties

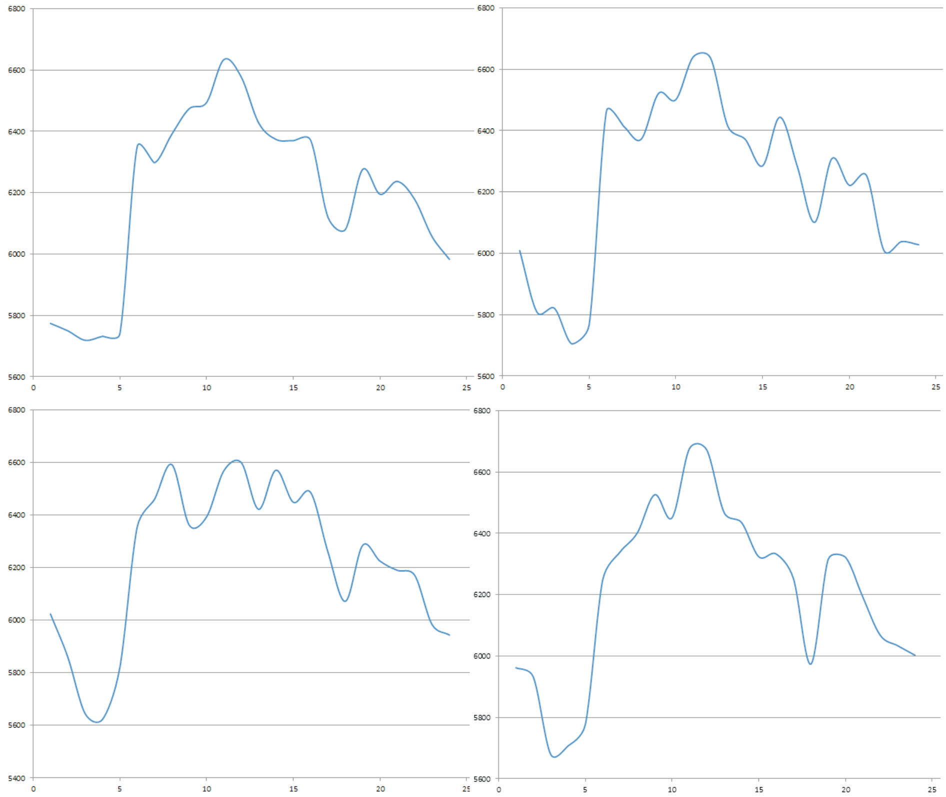

Fig. 4 *Examples of randomly modeled diagrams of Wednesday.*

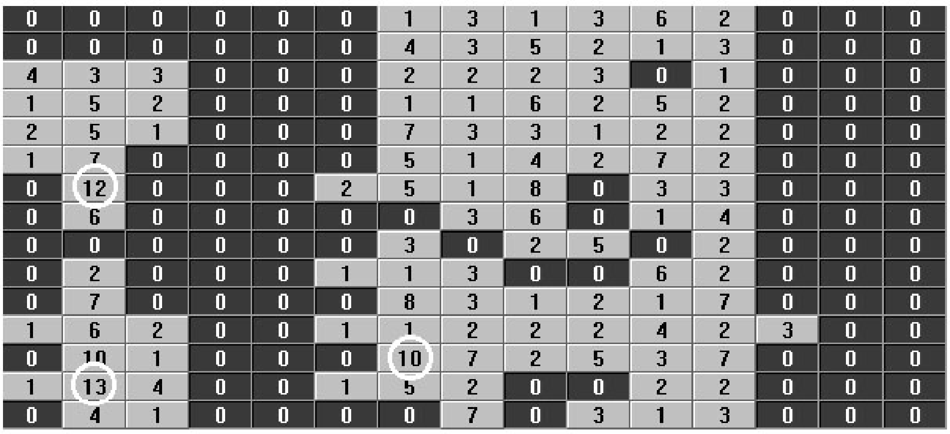

Fig. 5 *Kohonen map.*

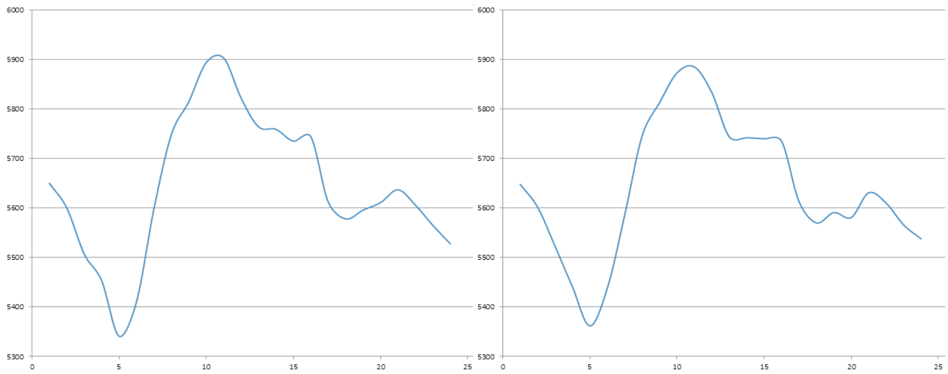

Fig. 6 *Type daily diagram and standard of Saturday.*

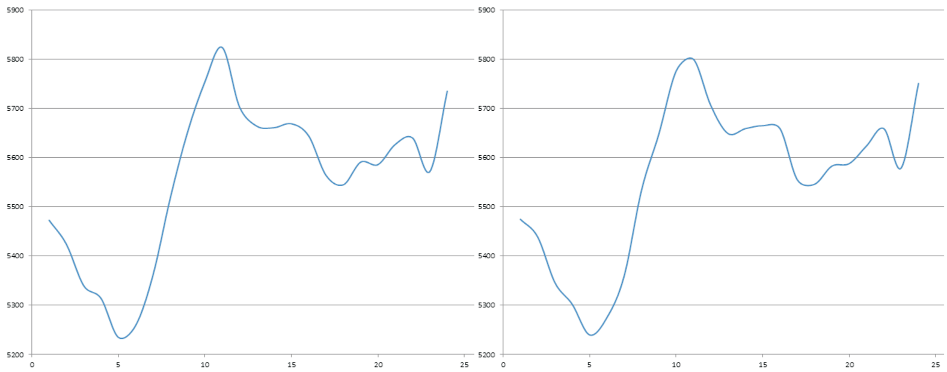

Fig. 7 *Type daily diagram and standard of Sunday.*

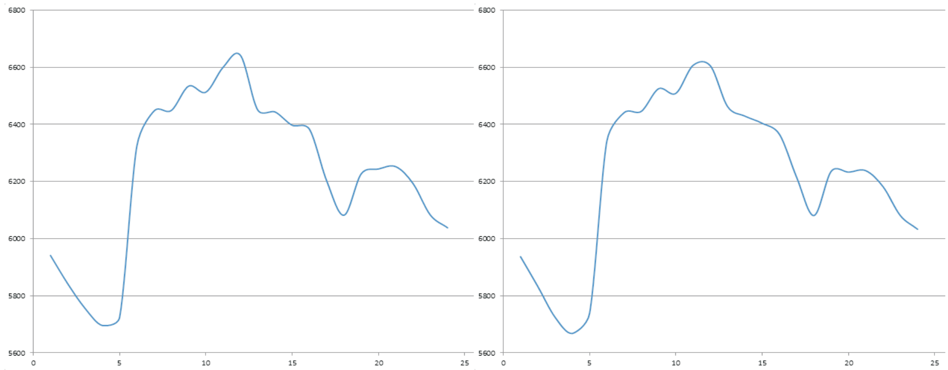

Fig. 8 *Type daily diagram and standard of Wednesday.*

	Saturday			Sunday			Wednesday	
TDD	standard	difference	TDD	standard	difference	TDD	standard	difference
[kW]	[kW]	[%]	[kW]	[kW]	[%]	[kW]	[kW]	[%]
5650	5648	0,04	5473	5475	0,04	5941	5937	0,07
5600	5602	0,04	5424	5439	0,28	5842	5833	0,15
5507	5523	0,29	5340	5346	0,11	5756	5725	0,54
5453	5441	0,22	5314	5302	0,23	5696	5668	0,49
5341	5362	0,39	5235	5240	0,10	5724	5738	0,24
5411	5438	0,50	5259	5275	0,30	6323	6341	0,28
5599	5586	0,23	5361	5360	0,02	6447	6441	0,09
5749	5746	0,05	5518	5534	0,29	6449	6446	0,05
5815	5813	0,03	5652	5648	0,07	6534	6525	0,14
5894	5873	0,36	5754	5776	0,38	6513	6509	0,06
5903	5885	0,31	5824	5800	0,41	6600	6607	0,11
5822	5833	0,19	5703	5706	0,05	6642	6605	0,56
5764	5744	0,35	5664	5649	0,27	6451	6462	0,17
5759	5742	0,30	5661	5659	0,04	6444	6429	0,23
5735	5740	0,09	5669	5665	0,07	6397	6404	0,11
5744	5735	0,16	5644	5659	0,27	6383	6365	0,28
5612	5613	0,02	5564	5555	0,16	6202	6214	0,19
5578	5570	0,14	5545	5546	0,02	6082	6081	0,02
5596	5591	0,09	5591	5583	0,14	6227	6236	0,14
5611	5581	0,54	5586	5588	0,04	6244	6233	0,18
5637	5631	0,11	5627	5624	0,05	6252	6238	0,22
5606	5610	0,07	5640	5659	0,34	6193	6182	0,18
5565	5566	0,02	5572	5579	0,13	6083	6081	0,03
5528	5538	0,18	5735	5751	0,28	6038	6033	0,08
	MEAN=	0,20		MEAN=	0,17		MEAN=	0,19
	MAX=	0,54		MAX=	0,41		MAX=	0,56

Tab. I *Type daily diagram and standard.*

of an ideal method, and is only suitable for the analysis of sufficiently distant homogenous clusters of approximately spherical shape, which reduces its utility value.

The cluster analysis of neural networks is thus, in comparison e.g. with the k-means method, completely universal in accordance with the nature of the location of objects in the examined file (it may also identify a completely uniform distribution of objects in an n-dimensional space), however the interpretation of its results partially depends on a subjective reading of the Kohonen map.

References

[1] Grossberg S.: The Adaptive Brain, North-Holland, 1987.

[2] Kohonen T.: The Self-Organizing Map, IEEE, **78**, 9, 1990, pp. 1464-1480.

[3] R. Hecht-Nielsen R.: Counter-propagation networks, Appl. Opt. 26 (23), 1987, pp. 4979-4984.

[4] Hopfield J. J.: Artificial Neural Networks, IEEE Circuits & Devices, **4**, 3-10, 1988.