Method Description

General Information

Type of Entry (*Academic, Practitioner, Researcher, Student*)	**Student (PhD student), Academic**
First Name	**Paweł**
Last Name	**Pełka**
Country	**Poland**
Type of Affiliation (*University, Company-Organization, Individual*)	**University**
Affiliation	**Department of Electrical Engineering, Czestochowa University of Technology,**

Team Members (*if applicable*):

1st Member	
First Name	
Last Name	
Country	
Affiliation	
2nd Member	
First Name	
Last Name	
Country	
Affiliation	

Information about the method utilized

Name of Method	**Ponpo**
Type of Method (*Statistical, Machine Learning, Combination, Other*)	**Machine Learning**
Short Description (up to 200 words)	The model could be categorized as a k-NS based model. The forecasting model is based on the pre-processed time series, where input and output variables are defined as patterns representing unified fragments of the time series. Relationships between inputs and outputs, which are simplified due to patterns, are modelled using weighting function defining membership of learning points to the neighborhood of the query point.

Extended Description:

To reproduce 100 time series forecasts please use execution_REPRO2.m or execution_REPRO2.exe file. You have to get a training data files to execution_REPRO2.exe file folder (Hourly-train1.csv, …,Yearly-train1.csv). The program gets 100 random numbers between (1-100 000) and creates forecast to time series. The program execution_REPRO2 saves reproduced results to repro.csv file.

Patterns of Time Series Fragments

Fragments of time series are represented by patterns, which are vectors of n or m components. Each component is a preprocessed time series point. Input and output patterns are defined: $\mathbf{x} = [x_1\ x_2\ \ldots\ x_n]^T$ and $\mathbf{y} = [y_1\ y_2\ \ldots\ y_m]^T$, respectively. The patterns are paired ($\mathbf{x}_i$, $\mathbf{y}_i$), where $\mathbf{y}_i$ is a pattern of the time series fragment succeeding the fragment represented by pattern $\mathbf{x}_i$. A distance in time between these fragments is equal to the forecast horizon τ.

Pattern similarity-based forecasting methods are based on the following assumption [10]: If the input patterns $\mathbf{x}_a$ and $\mathbf{x}_b$ are similar to each other, then patterns $\mathbf{y}_a$ and $\mathbf{y}_b$ paired with them are similar to each other as well. The above assumption allows us to forecast the pattern $\mathbf{y}_a$ on the basis of known patterns $\mathbf{x}_a$, $\mathbf{x}_b$ and $\mathbf{y}_b$, which are determined from the history. In N-WE we calculate similarity between query pattern $\mathbf{x}_a$ and patterns $\mathbf{x}$ from the training set. Then to get the forecasted pattern $\mathbf{y}_a$ we aggregate training patterns $\mathbf{y}$ using weights dependent on the similarity between x-patterns. The method of preprocessing the time series fragments to get their patterns are dependent on the time series specificity (seasonal variations, trend), forecast period and horizon.

Let us consider the task of prediction of m consecutive points of the monthly electricity demand time series. Let us denote the forecasted fragment by $Y_i = \{E_{i+1}\ E_{i+2}\ \ldots\ E_{i+m}\}$, and the preceding fragment of n points by $X_i = \{E_{i-n+1}\ E_{i-n+2}\ \ldots\ E_i\}$. An input pattern $\mathbf{x}_i = [x_{i,1}\ x_{i,2}\ \ldots\ x_{i,n}]^T$ represents the fragment X_i. The components of this vector are preprocessed points of the sequence X_i. Some examples of preprocessing are:

$$x_{i,t} = \frac{E_{i-n+t}}{\overline{E}_i} \tag{1}$$

$$x_{i,t} = \frac{E_{i-n+t} - \overline{E}_i}{D_i} \tag{2}$$

where $t = 1, 2, ..., n$, $\overline{E}_i$ is the mean value of the points in sequence X_i, and $D_i = \sqrt{\sum_{j=1}^{n}(E_{i-n+j} - \overline{E}_i)^2}$ is a measure of their dispersion.

Pattern components defined using (1) are the points of the sequence X_i divided by the mean value of this sequence. The pattern defined using (2) is a normalized vector $[E_{i-n+1}\ E_{i-n+2}\ \ldots\ E_i]^T$. It has the unity length and the mean value equal to zero. Moreover, all x-patterns have the same variance.

The output pattern $\mathbf{y}_i = [y_{i,1}\ y_{i,2}\ \ldots\ y_{i,m}]^T$ represents the forecasted sequence Y_i. The components of y-pattern can be defined similarly to the components of x-patterns:

$$y_{i,t} = \frac{E_{i+t}}{\overline{E}_i} \tag{3}$$

$$y_{i,t} = \frac{E_{i+t} - \overline{E}_i}{D_i} \tag{4}$$

In formulas (3)-(4) $\overline{E}_i$ and D_i are determined from the sequence X_i, and not, as one might expect Y_i. These values at the moment of forecasting are known and enable us to calculate the forecast of demand based on the forecast of y-pattern returned by the forecasting model.

We use for this the transformed equations (5)-(8). For example, in the case of (8) the forecast of demand is calculated as follows:

$$\widehat{E}_{i+t} = \widehat{y}_{i,t} D_i + \overline{E}_i \tag{5}$$

Patterns $\mathbf{x}_i$ (representing the sequence preceding the forecasted one) and $\mathbf{y}_i$ (representing the forecasted sequence) are paired ($\mathbf{x}_i$, $\mathbf{y}_i$). The set of these pairs determined from history is used for learning the forecasting model.

kNS with weights

The nearest neighbour estimate $m(\mathbf{x})$ is defined as the weighted average of the y-patterns in a varying neighbourhood of the query x-pattern. Typically, this neighbourhood is defined through the x-patterns which are among the *k* nearest neighbours of the query pattern. The value of *k* determines the number of training patterns from which the regression function is constructed and controls the degree of smoothing. The *k*-NN estimator gives the regression function, which is discontinuous. In the points where the set of the nearest neighbours changes, the jumps on the function graph are observed. To avoid this inconvenience, a membership of the training points to the neighbourhood of the query point was introduced. In this approach, each training point belongs to the query point neighbourhood with a degree depending on the distance between these points.

The regression function $m(\mathbf{x})$ has the nonparametric form:

$$m(\mathbf{x}) = \sum_{j=1}^{N} w(\mathbf{x}, \mathbf{x}_j)\mathbf{y}_j \tag{6}$$

where the weighting function $w(\mathbf{x},\mathbf{x}_j)$ is dependent on the similarity or distance between patterns $\mathbf{x}$ and $\mathbf{x}_j$. Usually it decreases monotonically with the distance. When using *k-ns* approach, the weighting function has a form of the membership function, e.g a function:

$$w_i(\boldsymbol{x}) = \left(\frac{1 - \frac{d(x,x_i)}{d(x,x^k)}}{1 + 2\frac{d(x,x_i)}{d(x,x^k)}} - 1 \right) + 1, \tag{7}$$

where $i \in \Theta_k(x)$, $\boldsymbol{x}^k$ is *k*-th nearest neighbor pattern $\boldsymbol{x}$ in traning data, $d(\boldsymbol{x}, \boldsymbol{x}_i)$ is an Euclidean distance between patterns i $\boldsymbol{x}$, a $\boldsymbol{x}_i$

Estimator (10) is a linear combination of vectors $\mathbf{y}_j$ weighted by the membership degree (11) which nonlinearly map the distance $d(\mathbf{x},\mathbf{x}_j)$. The greater the distance, the lower the weight.

The training set contains pairs of patterns ($\mathbf{x}_i$, $\mathbf{y}_i$), which are historical for the forecasted sequence, i.e. these ones for which $i = n$, n+1, ..., i^*–m, where i^* is an index of the last month before the forecasted sequence. The forecasting task is to generate the forecasts for months i^*+1, i^*+2, ..., i^*+m.

The forecasting procedure consists of four steps:

1. Preprocessing of load time series into x- and y-patterns.
2. Calculating the weights for the training x-patterns using membership function (5).
3. Calculating the forecasted y-pattern from (10).
4. Decoding the forecasted y-pattern using transformed equations (3)-(4) to get the monthly electricity demand for consecutive months: i*+1, i*+2, ..., i*+mAs we can see from (6) and (7)

References:

1. Dudek, G., Pełka, P.: Medium-term electric energy demand forecasting using Nadaraya-Watson estimator. In: Rusek S., Gono R. (eds) Proceedings of 18th International Scientific Conference on Electric Power Engineering (EPE), pp. 300-305, IEEE, New York (2017).
2. Pełka, P., Dudek, G.: Prediction of monthly electric energy consumption using pattern based fuzzy nearest neighbour regression, 2nd International Conference of Computational Methods in Engineering Science (CMES), (2017).