Novel Consumer Classification Scheme for Smart Grids

Kálmán Tornai*†, Lóránt Kovács*,
András Oláh*†, Rajmund Drenyovszki*, István Pintér*, Dávid Tisza*†, János Levendovszky‡,
*Kecskemet College, Department of Informatics
†Pázmány Péter Catholic University, Faculty of Information Technology and Bionics
‡Department of Networked Systems and Services, Budapest University of Technology and Economics

***Abstract*—Classifying different type of consumers (households, office buildings and industrial plants) is an important task in Smart Grids. In this paper, we propose a novel classification scheme based on nonlinear prediction for consumption time-series obtained from a smart meter. The candidate predictors were tested under different assumptions regarding the statistical behavior of the underlying consumption time-series. As a result a feedforward neural network based predictor has been shown to be the most promising solution. In order to demonstrate the power of the proposed method simulations have been carried out. The consumption data came from a bottom up model, where Markov model of individual appliances and real measurements of photo-voltaic generators have been applied. The numerical results prove that our method is capable of distinguishing an office-building with installed photo voltaic mini power plant from an office-building which is lack of such power plant.**

***Index Terms*—Linear prediction, Nonlinear prediction, Feedforward Neural Network, Radial Basis Function Network, Smart Grid, Classification**

I. Introduction

Smart grid is a new vision of future evolution of electricity systems. In contrast to the present grids, smart grid can efficiently incorporate the renewable energy resources and can adaptively manage the balance (supply and demand) of the system. The new features root in the two-way communication system and an intelligent measurement system as an integrated part of the energy system. The new objectives and possibilities of the system imply new applications which are based on new signal processing methods. In the presence of smart metering a huge amount of data will be acquired. The information can be caught by sophisticated algorithms.

The aim of the relevant research is to define the fields of interest at the methods by which the relevant information can be reached. Such tasks are identifying consumer categories, identification of outliers in certain groups, identification of misuse of services. In this paper we deal with only one of these relevant tasks, which is automated detection of consumer classes which is the basic tool to recognize category changes, consumption behavior changes, or irregularities of the grid. The classification of different categories supposes that statistical change points in time-series has been detected resulting in stationary segments. In this paper, we cannot focus on the segmentation problem, we only refer to the relevant literature such as [1].

Our goal is to establish a new classification scheme that is capable to distinguish between two categories of office buildings with or without local photovoltaic generator. The classification will be carried out by consumption time-series, obtained by a smart meter. In this paper, a new classification scheme will be investigated, which is based on the different prediction error levels, obtained in different class of consumers. Three different type of predictors will be tested such as Linear Predictor (LP), Radial Basis Function Neural Network Predictor (RBFP) and Feedforward Neural Network Predictor (FFNNP), from which the FFNNP will be proven to be the most promising tool.

The different predictors will be analyzed in different circumstances: as an initial level the consumption data is supposed to be an autoregressive process. As a second level a more realistic scenario will be used: the consumption will be modeled as the sum of independent Markovian two-state (On/Off) models that represent the appliance in the offices. The parameters of the Markovian model were fitted on real power consumption data. The paper will be organized as follows: In Section III, the basic concept for the classification scheme and the applied model of the system will be introduced. In Section IV, the applied predictors will be described in detail. In Section V, the performance of the different predictors will be analyzed, finally in Section VI, conclusions will be drawn.

II. Related work

Based on several surveys on classification of time series we summarize the existing methods. The algorithms, which are designed for clustering time series have two approaches as follows: i) redesign and modify existing algorithms in order to handle the sequential data, where the aim is to exploit the sequential nature of the data; ii) the data is transformed to fit the model of existing algorithms. [2] The algorithms can be categorized as i) distance based classification; ii) feature based classification; and iii) model based classification. In the following sections the categories will be briefly discussed.

A. Distance based classification

The decision of distance based classification algorithms depends on the distances between data elements. In this case new measurements have to be saved to calculate the distance between two samples. The metric (measure of distance) has

large influence of performance of the distance based classification algorithms. [3] In general the euclidean distance is used, however special problems require more complicated metrics such as dynamic time warping distance. [4], [5] When the sequence contains symbols instead of numerical values, sequence alignment is needed such as Needleman-Wunsch.

B. Feature based classification

Algorithms which are feature based, such as ANN and Decision Trees transform the sequential data into a feature-set. The selection of the appropriate feature set is the key of the methods and mainly influences the performance. [6] The wavelet decomposition of pattern matching is the most often used methods to do the transformation.

One of the pattern extraction based method is the Minimal Distinguishing Subsequence (MDS) method [7]. This method is mostly suitable for biological sequences such as DNA or protein sequences. The discriminative approach, which decides that a new sequence belongs to certain class or not uses the (Euclidean) distance between two sub-sequence.

The transformation of the time-series into frequency domain can reduce the dimensionality. The methods: DFT (Discrete Fourier Transform), DWT (Discrete Wavelet Transform), SVD (Singular Value Decomposition) are well known in signal processing [8].

Using kernel methods such as SVM (Support Vector Machines) to feature extraction may increase the number of features to be handled by the method. These methods also widely used on biological data such as mentioned before. [9]

C. Model based classification

Model based classification methods construct a model using training set of data of a cluster. After training the new data is classified upon the best fitting model. The methods can be divided into statistical and neural network based algorithms [2].

Notable statistical models are Gaussian, Poisson, Markov and Hidden Markov Model. They are constructed as to model the probability distribution of the data [10]. According to [11] the methods can be divided into predictive and descriptive models. The first one tries to estimate the unavailable data from the existing ones, the second one tries to mine the patterns and relationships in the data.

Hidden Markov Models can be used successfully in case of biological data. They can deal with variable length input, however requires the assumption on the probabilities of the states, and the probabilities of certain state depends on only the previous state [10]. HMM also requires prior knowledge to choose the specific input features [12]. The artificial neural networks (ANN), especially recurrent neural networks (RNN) are close to the statistical models [13], however they not require a priori knowledge about the data. Furthermore the input noise has less influence to the performance of the methods [14].

D. Our classification scheme

In our approach the data sequences contain only numerical values and the method is semi-distance based as the method make its decision based on the difference of predicted and real measurement values. Furthermore by applying artificial neural networks our method is also model-based. However this model is constructed without any explicit information about the behavior of the processes which generate the values of the time-series. The artificial neural network is treated as a black box and the training algorithm which extract the information needed for the model, only requires the input and desired output. During the training phase there are no assumptions about the process, which generates the time-series, only the observable values are available. That approach makes our solution different from existing methods.

III. System Description

In this section the basic system model, the data acquisition framework and the data model will be introduced.

A. Data acquisition framework

Figure 1 depicts the scheme of data acquisition in the model. There are several office-buildings equipped with smart meters, which are supposed to measure and transmit the power consumption time-series. The statistics of the power consumption may differ in the case of different buildings, which depends on the size of the building, number and type of appliances, the habit, location, number of workers, etc.

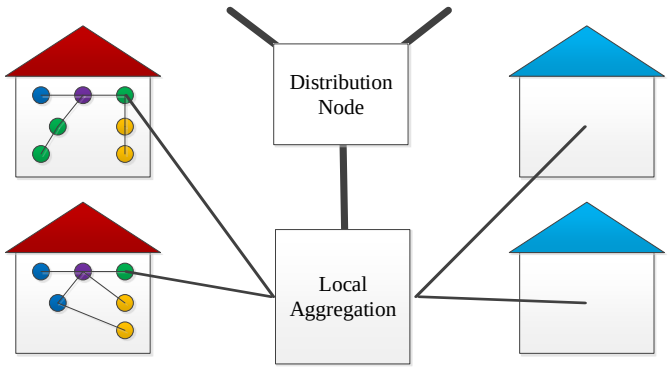

Fig. 1. The model of the Smart Grid Network. Several electrical appliances, two different classes of buildings and two distribution nodes are depicted.

The measurements will be sent to a local power distribution node, where the data will be analyzed. The data have to be cleansed (outlier values have to be removed) and a continuous classification will be maintained thus the change of the behavior of offices-buildings can be detected instantaneously.

B. Scheme of the proposed method

The proposed method exploits the different statistical properties of the power consumption time-series acquired from different classes of consumers. If a predictor has been trained with time-series of class i, then the following assumptions

are made: i) time-series of class i can be predicted with low error rate; ii) time-series which do not belong to class i can be predicted with significantly superior error. As a result a prediction based solution not only capable of classification but can detect measurement errors or can estimate the expected consumption. That universality is the main advantage of our scheme over the existing solutions. For the sake of simplicity only two classes have been considered in the formal model. Each class members were generated using the same statistics (process parameters). Thus the resulting algorithmic flow of the scheme is depicted by Figure 2. One predictor is trained for every class, using representative sequences. After the training phase for class i, the upcoming values of a new sequence will be compared with their predicted values (using the ith predictor) resulting in a prediction error sequence. The mean of prediction error will be used as decision variable to decide whether the upcoming sequence belongs to the class i.

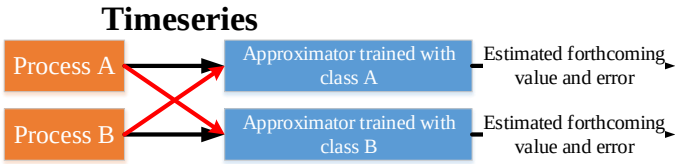

Fig. 2. Algorithmic scheme of the proposed scheme.

Formally, the two classes of processes are denoted by $\mathcal{C}^{(i)}$ and $\mathcal{C}^{(j)}$ respectively, the predictors are denoted by $\Gamma^{(i)}(.)$ and $\Gamma^{(j)}(.)$ respectively. The training set of a predictor is assembled by using samples from the corresponding time-series:

$$\tau_{(i)} = \left\{ \left(x_1^{(i)} \dots x_{1+m}^{(i)}; x_{2+m}^{(i)} \right), \dots \right\},$$

$$\tau_{(j)} = \left\{ \left(x_1^{(j)} \dots x_{1+m}^{(j)}; x_{2+m}^{(j)} \right), \dots \right\}, \tag{1}$$

where

$$x^{(i)} \in \mathcal{C}^{(i)} \wedge x^{(j)} \in \mathcal{C}^{(j)} \tag{2}$$

The information inherent in the correlation between the elements of the time-series can be used to predict the forthcoming values. Therefore the classification problem boils down to comparing the real x_k value with $\hat{x}_k$, a value predicted by using L preceding values. This predicted value is calculated by applying the learned free parameters ($\mathbf{W}$) of the predictor

$$\hat{x}_k = \Gamma\left(\mathbf{W}, x_{k-1,\dots,k-L}\right). \tag{3}$$

If the difference between real values and its prediction exceeds a given threshold (Δ), then the value does not conform to the properties of the process learned by the predictor. Formally, the result of the decision for the ith time-series:

$$\hat{y}_k^{(i)} = \operatorname{sgn}\left(-\left| x_k^{(i)} - \hat{x}_k^{(i)} \right| + \Delta^{(i)} \right) \in \{0, 1\}. \tag{4}$$

For each class a different threshold value may exist, which is denoted by $\Delta^{(i)}$ for the ith class.

C. Alternative algorithmic flow for linear prediction

In this section an alternative algorithmic flow will be discussed. Using linear predictor, the coefficients of the predictor can be used directly as decision variables, since the space of the linear filter coefficients span an Euclidean space. The basic idea is illustrated by Figure 3.

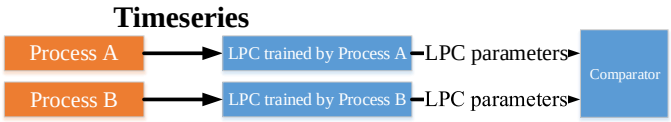

Fig. 3. Alternate algorithmic flow indicating linear prediction

This scheme has the advantage of reducing the computational time of the method, however it cannot be used in the case of nonlinear predictors due to the complexity and variability of their structures.

IV. Implementation of the Scheme

To implement the predictor, several techniques can be used [15], such as i) linear prediction; ii) logarithmic prediction; iii) SVM based prediction; iv) feedforward neural network based prediction [16]–[18]. However we limited our scope to those that are able to be executed and evaluated in real time. In this section we briefly outline the prediction methods which satisfy these criteria and are involved in the performance analysis.

A. Linear prediction based detection

For linear prediction (LP) [19], a Finite Impulse Filter (FIR) can be used as follows:

$$\hat{x}_k^{(i)} = \sum_{l=1}^{L} w_l^{(i)} \cdot x_{k-l}^{(i)}, \tag{5}$$

where the weights $w_m^{(i)}$ are optimized in order to minimize the mean squared error of the prediction. The optimization is based on a training sequence $x_k^{(i)}$ as follows:

$$\mathbf{w}_{\text{opt}}^{(i)} : \min_{\mathbf{w}^{(i)}} \sum_{k=1}^{M} \left| x_k^{(i)} - \hat{x}_k^{(i)} \right|^2 \tag{6}$$

The optimization of the LP filter coefficients applies the autocorrelation function [20] yielding the Yule-Walker equations [21], which can be solved by the Levinson-Durbin algorithm [22].

B. Nonlinear prediction by RBF neural networks

The Radial Basis Function (RBF) Neural Network can be used as universal function approximator [23], and the training of the network can be carried out rapidly by solving a set of linear equations. RBF consists of the following layers:

1) Layer with radial basis function $\varphi(.)$.
2) Output layer containing one linear perceptron to summarize the outputs of the RBF.

The output of the structure is computed as follows:

$$\mathrm{Net}(\mathbf{x},\mathbf{w}) = \hat{x}_k^{(i)} = \sum_{l=1}^{N} w_l^{(i)} \varphi(\mathbf{x}, \mathbf{c}_l^{(i)}) + w_0^{(i)}, \tag{7}$$

where φ function is a sphere-symmetric function. In our implementation the most widespread Gaussian function has been applied:

$$\varphi_G = \exp\left(-\frac{r^2}{2\delta^2}\right) = \exp\left(-\frac{\|\mathbf{x}_l - \mathbf{c}_l\|^2}{{\delta_l}^2}\right) \tag{8}$$

The RBF using as a predictor (RBFP) was trained by the following method: [24]

- First the output of the network is determined for each element of the training set.
- The element of the training set is found, which has the largest prediction error. Then a new neuron is added with weights equal the values previous element of training set.
- The weights of the linear layer will be recalculated.

C. Nonlinear prediction by FFNN

The Feedforward Neural Network Predictors (FFNNP) are capable of approximation any function in L^2 with arbitrary precision [25], and can be trained by rapidly converging algorithms. That advantage allows us to apply the prediction based classification, which is capable to handle power consumption time-series well. After preliminary simulations the FFNNP is implemented with three hidden layers in our scheme. In the first two hidden layers nonlinear sigmoid activation function while in the last hidden layer linear activation function were used. The output of the network can be expressed as

$$\mathrm{Net}(\mathbf{x},\mathbf{w}) = \varphi_L\left(\sum_{l=1}^{N_L} \mathbf{w}_l^{(L)} \cdot \varphi_S\left(\sum_{j=0}^{N_{L-1}} \mathbf{w}_{lj}^{(L-1)} \varphi_S\left(\sum_{m=1}^{N_1} \mathbf{w}_{lm}^{(1)} \cdot \mathbf{x}_m\right)\right)\right), \tag{9}$$

where

$$\varphi_S(u) = \frac{2}{1+e^{-2\alpha u}} - 1 \tag{10}$$

is the sigmoid activation function. To adapt the weights of the artificial neurons the Levenberg-Marquardt backpropagation learning function have been applied [26], which is often cited as the fastest backpropagation algorithm.

V. Performance analysis

In this section the results of the performance analysis of our scheme will be introduced. The following metrics were applied in order to characterize the performance of different predictors and schemes:

- root mean square (RMS) of the prediction error:

$$\mathrm{RMS}\left(x, \hat{x}\right) = \sqrt{\frac{1}{n}\sum_n \left(x_n - \hat{x}_n\right)^2} \tag{11}$$

- RMS of difference of LP coefficients

$$\mathrm{RMS}\left(w^i, w'^i\right) = \sqrt{\frac{1}{n}\sum_n \left(w_n^i - w_n'^i\right)^2} \tag{12}$$

- minimal detection error rate, where for obtaining $\mathbf{W}_j$ the τ_j training set has been used.

$$\varepsilon = \min_{\Delta}\left(\frac{E_1 + E_2}{N * N}\right) \tag{13}$$

where

$$E_1 = \sum_i^N \sum_{j \in \mathcal{C}_i} \mathrm{sgn}\left(\mathrm{RMS}\left(\Gamma\left(\mathbf{W}_j, x^i\right) - x_i\right) > \Delta\right) \tag{14}$$

$$E_1 = \sum_i^N \sum_{j \notin \mathcal{C}_i} \mathrm{sgn}\left(\mathrm{RMS}\left(\Gamma\left(\mathbf{W}_j, x^i\right) - x_i\right) \leq \Delta\right) \tag{15}$$

As a first step, the different predictors has been compared by applying different models of the aggregated consumption series. After it, real power consumption data were used for testing.

In the case of artificially generated time-series we applied two levels of modeling i) AR processes and ii) independent two-state (On/Off) Markovian models for the individual appliances; the sum of the appliance-level data was used as the aggregate consumption time-series.

The simulations are carried out in Matlab environment. The time-series are generated both i) autoregressive (AR) processes with different parameters and ii) Markovian model of appliances.

A. Markovian model for consumption time-series

Unfortunately, it is not common to have access to annotated consumption data in public databases, which is why close-to-real consumption time-series has to be generated artificially. In our contribution the consumption of individual appliances is assumed to be a two-state Markovian process which can model the time dependencies of the time-series. The sum of the independently modeled appliance consumption time-series has been used as the model of aggregate consumption of a consumer entity. The Markovian parameters were fitted on real measurements coming from the REDD database [27], containing several weeks of appliance-level power consumption data for 6 different homes.

Formally, a single appliance is modeled with a discrete random variable

$$X^{\mathrm{Markovian}} \in \{0, h\}, \tag{16}$$

where 0 means that the consumer is switched off, and h is the energy level of the powered device.

The power consumption of an office building were modeled by the sum of several Markovian processes representing computers, refrigerators and lighting devices. Figure 4 depicts the consumption of a single appliance, and Figure 5 shows the time-series generated by summing 100 independently generated Markovian On/Off processes.

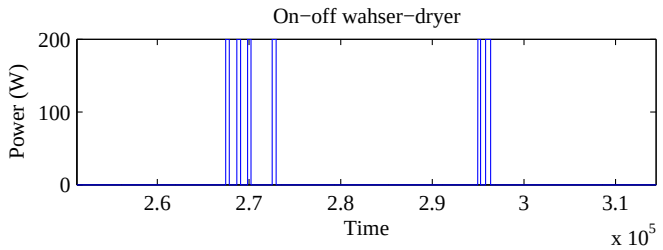

Fig. 4. Markovian On/Off model of a single appliance

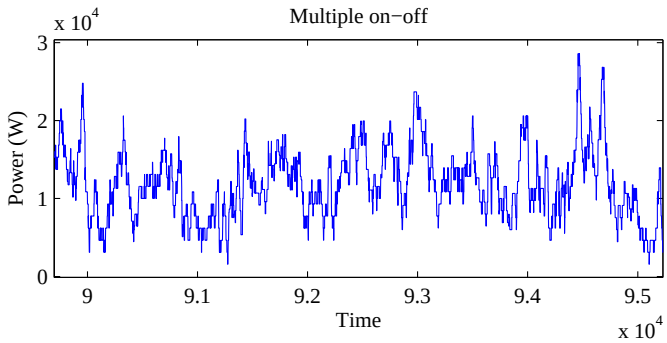

Fig. 5. Aggregation of multiple time-series generated by independent Markovian models

B. Simulation environment and results

For each tests N number of time-series were generated for both classes. The prediction errors have been investigated in order to determine whether significant difference exists between the two classes or not. Therefore a cross-test was performed: every time-series was used as a training sequence for a predictor. After the training phase the time-series were used to evaluate the trained predictors. The prediction has been done for all time-series. Because of the RMS of the error can be computed for all possible pair of training and test set, then a $N \times N$ matrix with these RMS values can be obtained:

$$\mathbf{R}_{i,j} = \mathrm{RMS}\left(\Gamma\left(\mathbf{W}_j, x^i\right) - x_i\right), \tag{17}$$

where $\mathbf{W}_j$ is obtained by using

$$\tau_j = \left\{\left(x_1^j \ldots x_{1+m}^j; x_{2+m}^j\right), \ldots\right\} \tag{18}$$

The visualization of matrix $\mathbf{R}$ characterize the detection capability of the methods. If we get four quadrants, where the RMS of prediction error is low when the training set and test set belong to the same Class (upper left, and lower right regions) and the RMS of prediction error is high otherwise (upper right and lower left regions) then the method is capable of distinguishing the classes. As the edge is sharper between the regions (i.e. the error dramatically changes at the border of regions) the classification is more reliable.

1) Tests on AR processes: The linear predictor performs low error as expected in the case of AR process. The AR parameters of the two classes were $[0.3, -0.3, 0.1, 0.2, -0.1]$ for $\mathcal{C}_1$ and $[0.9, -0.8, 0.1, -0.1, 0.2]$ for $\mathcal{C}_2$. Figure 6 shows the resulting autocorrelation functions of two classes.

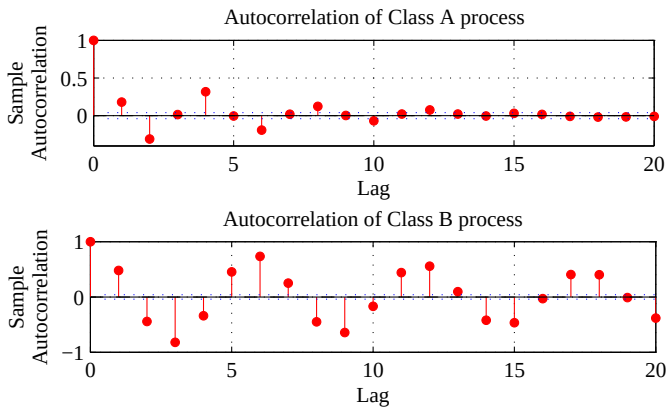

Fig. 6. Autocorrelation plots of sample time-series from each power consumption class

The LP is capable of prediction with the RMS of errors (both in linear and dB values) depicted in Figure 7.

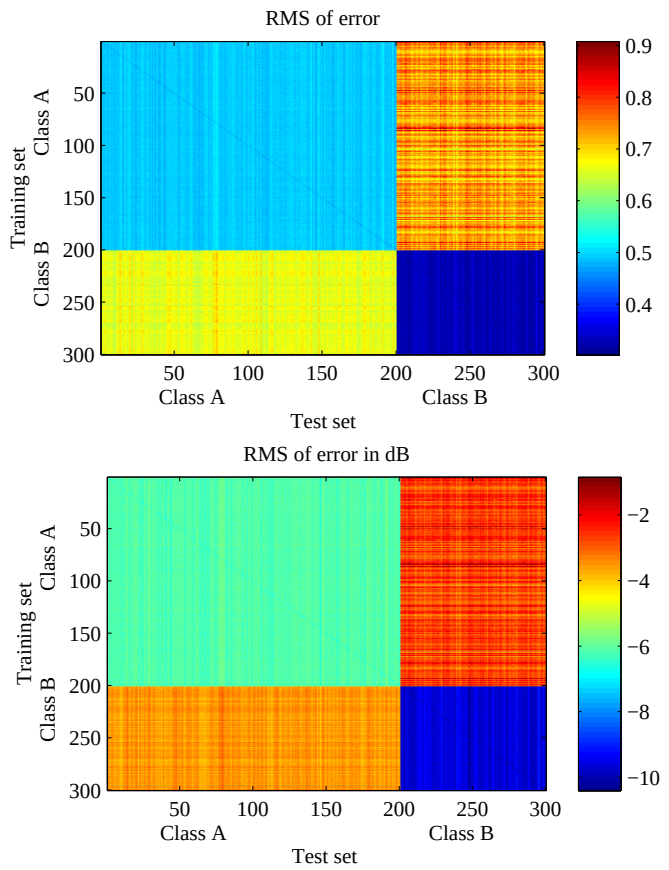

Fig. 7. Demonstration of class separation capabilities of LPC predictor on time-series generated by AR processes

Both RMS values and dB values show that it is possible to have an appropriate Δ threshold value in order to clearly distinguish the classes. Using $\Delta = 0.57$ the detection error rate is $\varepsilon \leq 10^{-5}$. Instead of using the previous metric and algorithmic flow we have investigated the filter parameters of the trained LP. The unsurprising results are depicted in Figure 8. The magnitude of error values is significantly lower, but that

does not impair the classification capability of the scheme. The results also demonstrate that the prediction capability is redundant in case of AR processes. The same tests were evaluated using the predictor coefficients. As one can expect, the same low detection error rate can be seen in Figure 8.

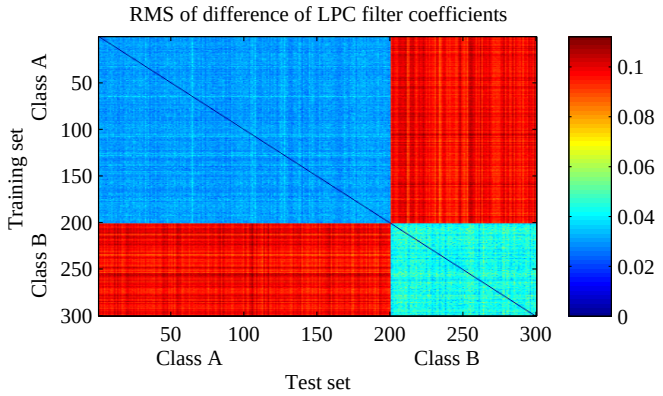

Fig. 8. The difference in RMS of the LPC filter coefficients

2) Tests on time-series generated by Markovian model: In this section it will be demonstrated that the LP is not able to solve the problem when the time-series are generated by a more realistic Markovian model. The results are shown on Figure 9 and Figure 10. Both results ($\Delta = 6248, \varepsilon = 0.195$ and $\Delta = 0.1383, \varepsilon = 0.265$) indicates that the deployment of nonlinear predictor such as RBFP and FFNNP is necessary.

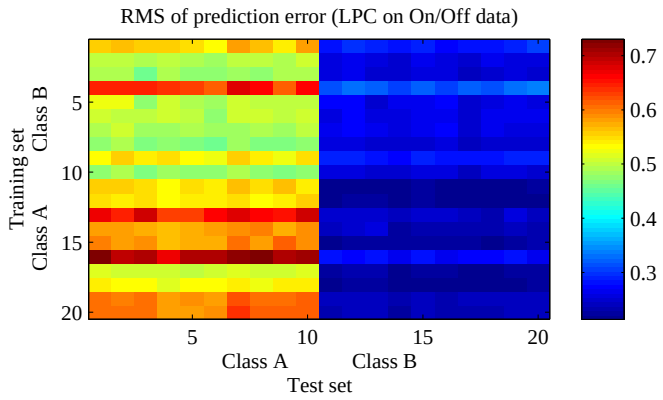

Fig. 9. RMS of LP errors in the case of Markovian model

Since linear predictor cannot predict the sum of Markovian processes, nonlinear RBF and FFNN predictors came to the fore. The results of RBFP are shown on Figure 11. Unfortunately, RBF cannot satisfactory deal with the noncontinuous segments of the time-series (see Figure 5), the performance is poor: the minimum of the detection error (when $\Delta = 1.76$) is $\varepsilon = 0.691$.

The results for prediction of sum of Markovian processes by FFNN are shown in Figure 12. At a threshold of $\Delta = 0.231$ the classification error rate is under $\varepsilon = 0.037$), which seems to be promising for a many applications. (However further refinement of parameters such as number of hidden layers and neurons is required.)

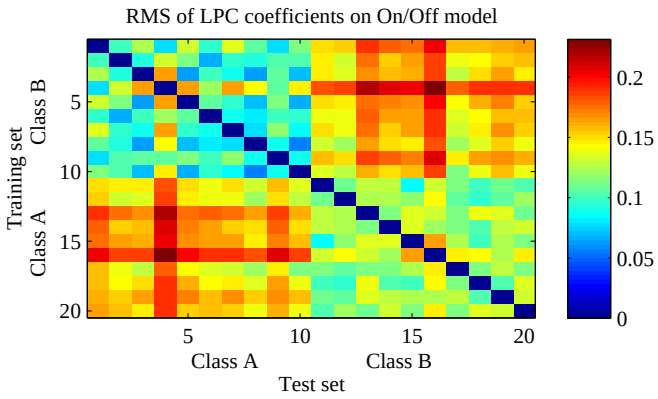

Fig. 10. RMS of coefficient differences of LP in the case of Markovian model

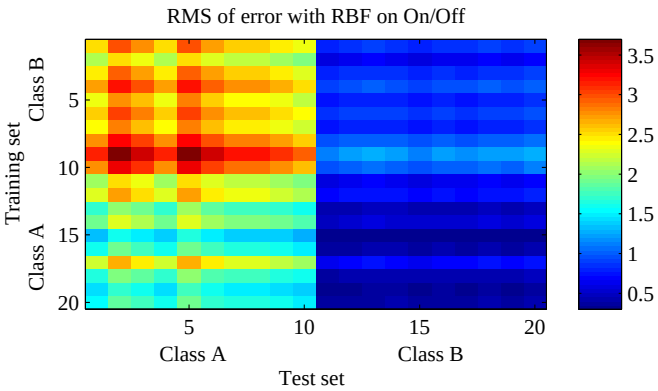

Fig. 11. RMS of error using RBFP in the case of Markovian model

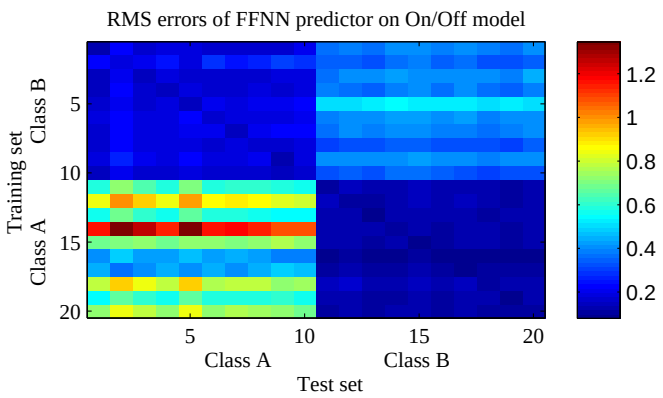

Fig. 12. RMS of error using FFNNP in the case of Markovian model

C. Real data tests and results

It also has been investigated whether the scheme is capable of solving the problem in real applications or not. In order to do that time-series by summing Markovian two-state (On/Off) models, and real data of photovoltaic power generator were used. From half of the generated time-series the photovoltaic data have been subtracted. In such way two classes were produced: i) an office building with solar power generator installed on the roof; and ii) an office building without such energy source. Furthermore, the possibility of low decision

error rate caused by different expected value and variance was excluded by aligning the expected values and variances to the same interval as it is shown on Figure 13. (The number of appliances was the same for all offices: 50 computers, 100 lightning devices, 10 air conditioner.)

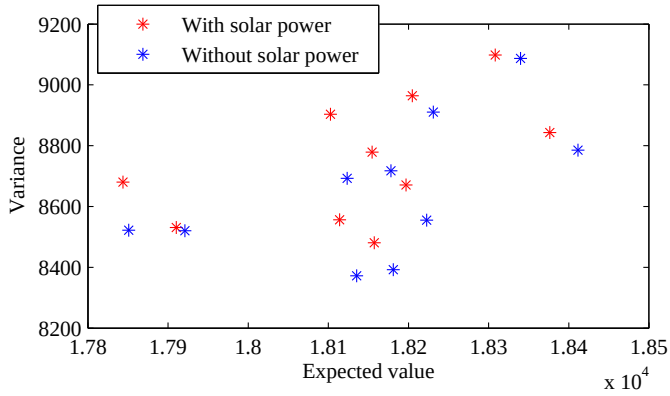

Fig. 13. Diagram of expected value and variance of Markovian data with and without solar power

The obtained results are introduced by Figure 14. The decision error rate is $\varepsilon = 0.046$ (int he case of decision threshold $\Delta = 0.196$), which underlines that our scheme using FFNN nonlinear predictor is capable of solving the task of classifying different class of electricity consumers.

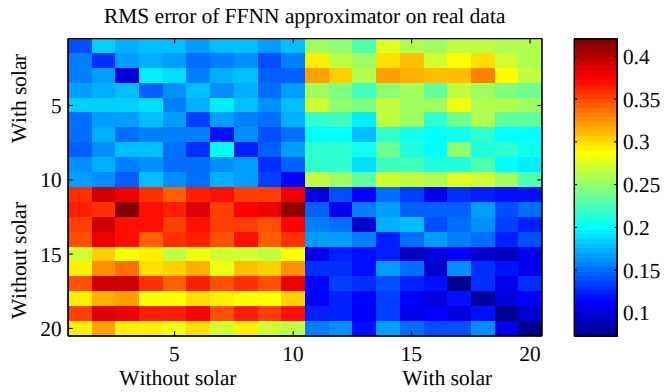

Fig. 14. RMS error using FFNNP in the case of real data

VI. Conclusion

In this paper, linear and nonlinear predictors have been applied in order to classify different classes of electricity consumers that have the same first and second order statistics but have different distributions and time dependencies. Both RMS of prediction error and difference in coefficient vector has been used as decision variable. Our scheme has been tested by different consumption models, from which the most realistic one (sum of Markovian two-state processes and photovoltaic generator measurements) could be successfully classified only by Feedforward Neural Network predictor. As a final result it has been shown that FFNN based classification scheme is capable of low error rate detection in real applications.

In the future we are going to extend the validation tests by performing simulations with real power consumption data as well. (The real measurements will be provided by a Hungarian energy supplier company.)

Acknowledgment

This publication/research has been supported by the European Union and the Hungarian Republic through the project TÁMOP-4.2.2.C-11/1/KONV-2012-0004 - National Research Center for Development and Market Introduction of Advanced Information and Communication Technologies. This source of support is gratefully acknowledged.

References

[1] I. Pinter, L. Kovacs, A. Olah, R. Drenyovszki, D. Tisza, and K. Tornai, "Application of jensen-shannon divergence in smart grids," in *Proceedings of 5th Scientific and Expert Conference TEAM 2013*, 2013, pp. 287–290.

[2] T. W. Liao, "Clustering of time series data - a survey," *Pattern Recognition*, vol. 38, no. 11, pp. 1857–1874, 2005.

[3] Z. Xing, J. Pei, and E. Keogh, "A brief survey on sequence classification," *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 1, pp. 40—48, 2010.

[4] S. Chan, B. Kao, C. Yip, and M. Tang, "A brief survey on sequence classification," in *Database Systems for Advanced Applications, 2003.(DASFAA 2003) Proceedings Eighth International Conference on*. IEEE, 2003, pp. 119–124.

[5] T. Li, S. Ma, and M. Ogihara, "Wavelet methods in data mining," in *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds. Springer, 2005, pp. 603–626.

[6] D. Eads, K. Glocer, S. Perkins, and J. Theiler, "Grammar-guided feature extraction for time series classification," in *In Proceedings of the 9th Annual Conference on Neural Information Processing Systems*, 2005.

[7] X. Ji, J. Bailey, and G. Dong, "Mining minimal distinguishing subsequence patterns with gap constraints," in *Data Mining, Fifth IEEE International Conference on*, 2005, p. 8.

[8] K. Yang and C. Shahabi, "A pca-based similarity measure for multivariate time series," in *ACM International Workshop On Multimedia Databases: Proceedings of the 2 nd ACM international workshop on Multimedia databases*, vol. 13, 2004, pp. 65–74.

[9] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, "Text classification using string kernels," *The Journal of Machine Learning Research*, vol. 2, pp. 419–444, 2002.

[10] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[11] S. Laxman and P. Sastry, "A survey of temporal data mining," *Sadhana*, vol. 31, no. 2, pp. 173–198, 2006.

[12] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labeling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006.

[13] D. Ruck, S. Rogers, K. Kabrisky, M. Oxley, and B. Suter, "The multilayer perceptron as an approximation to an optimal bayes estimator." *IEEE Transactions on Neural Networks*, vol. 1, no. 4, pp. 296–298, 1990.

[14] C. Giles, S. Lawrence, and A. Tsoi, "Noisy time series prediction using recurrent neural networks and grammatical inference," *Machine Learning*, vol. 44, no. 1, pp. 161–183, 2001.

[15] M. Last, A. Kandel, and H. Bunke, *Data mining in time series databases*. World Scientific, 2004.

[16] R. G. Brown, *Smoothing, Forecasting and Prediction of Discrete Time Series*. Dover Publication Inc., 2004.

[17] K. R. Müller, A. J. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik, *Artificial Neural Networks – LNCS: Predicting Time Series with Support Vector Machines*. Springer, 1997.

[18] I. H. Witten and E. Frank, *Data Mining, Practical Machine Learning Tools and Techniques*, 2nd ed. Elsevier, 2005.

[19] J. Makhoul, "Linear prediction: A tutorial review," *Proceedings of the IEEE*, vol. 63, no. 4, pp. 561–580, 1975.

[20] D. O'Shaughnessy, "Linear predictive coding," *Potentials, IEEE*, vol. 7, no. 1, pp. 29–32, 1988.

[21] H. Monson, *Statistical Digital Signal Processing and Modeling*. John Wiley & Sons, 1996.

[22] L. Ljung, *System Identification: Theory for the User*. Prentice-Hall, 1987.

[23] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Systems*, vol. 2, pp. 321—355, 1988.

[24] O. Buchtala, A. Hofmann, and B. Sick, "Fast and efficient training of rbf networks," in *Artificial Neural Networks and Neural Information Processing — ICANN/ICONIP 2003*. Springer Berlin Heidelberg, 2003, vol. 2714, pp. 43–51.

[25] S. Haykin, *Neural Networks, A Comprehensive Foundation*, 2nd ed. Pearson, Prentince Hall, 2005.

[26] S. J. Nocedal, Jorge; Wright, *Numerical Optimization, 2nd Edition*. Springer, 2006.

[27] J. Zico Kolter and Matthew J. Johnson, "REDD: a public data set for energy disaggregation research," in *SustKDD workshop*, San Diego, California, 2011. [Online]. Available: http://redd.csail.mit.edu/kolter-kddsust11.pdf

Kálmán Tornai has received the M.Sc. in Computer Engineering from the Pázmány Péter Catholic University, Faculty of Information Technology in 2008. He is a PhD. candidate, and assistant lecturer at the same institute. His research is oriented to data mining, efficient data processing and efficient algorithms, especially in wireless, mobile and smart systems.

Lóránt Kovács has received the Ph.D. degree in Electrical Engineering from the Budapest University of Technology and Economics in 2008. His research is oriented to adaptive signal processing related to wireless communications and smart grid.

András Oláh received his Ph.D. from Technical University of Budapest in 2007. He is currently working as an associate professor at Pázmány Péter Catholic University, Faculty of Information Technology. His research interests include linear and nonlinear signal processing in communication technologies, routing and MAC protocols for ad hoc networks and optimization methods and algorithms applied to wireless communications.

Rajmund Drenyovszki has received the B.Sc. degree in Computer Science Engineering from Kecskemet College in 2003, and the M.Sc. degree from Budapest University of Technology and Economics in 2010. In 2012 he started the Ph.D. studies at the University of Pannonia, Hungary within the Doctoral School of Information Science and Technology. His research is oriented toward the utilization of cyberphisical systems in smart grids to achieve higher efficiency.

István Pintér received his M.Sc. degree in electrical engineering and Ph.D. degree in informatics in 1983, and 1997, respectively from Technical University of Budapest, Hungary. Currently he is a professor at Kecskemét College, Hungary. His research areas include non-linear signal processing and machine learning. He is a member of IEEE and IEEE Computer Society.

Dávid Tisza received his M.Sc. from the Pázmány Péter Catholic University, Faculty of Information Technology in 2007. He is currently a Ph.D. student at the doctoral school of the same institute. His research interests include linear and nonlinear signal processing in communication technologies, artificial intelligence and neural networks, routing and MAC protocols for ad hoc networks and optimization methods and algorithms applied to wireless communications.

János Levendovszky received his Master and Ph.D. from the Budapest University of Technology And Economics. Currently is a professor at the same university. His present field of interests ranges from adaptive algorithms to discrete optimization. He researches and gives lectures on the topics on neural network theory, algebraic coding theory and digital signal processing.