Load Forecasting With The Aid of Neuro-Fuzzy Modelling

A. R. Koushki[1], M. Nosrati Maralloo[1], B.Hashemitabar[2], C. Lucas[3]
[1]Department of Computer Engineering, Islamic Azad University, research and science campus, Tehran, Iran.Email: a_r_koushki@yahoo.com ,nosraty_mehdi. @yahoo.com
[2] Department of Biomedical Engineering, Amirkabir University of Technology Tehran, Iran. Email: hashemit.b@gmail.com
[3] Control and Intelligent Processing Center of Excellence, Electrical and Computer Eng. Department, University of Tehran Tehran, Iran.Email: lucas@ipm.ir

ABSTRACT
One of the important requirements for operational planning of electrical utilities is the prediction of hourly load up to several days, known as Short Term Load Forecasting (STLF). Considering the effect of its accuracy on system security and also economical aspects, there is an on-going attention toward putting new approaches to the task. Recently, Neuro-Fuzzy modelling has played a successful role in various applications over nonlinear time series prediction. This paper presents a neuro-fuzzy model for the application of short-term load forecasting. This model is identified through Locally Liner Model Tree (LoLiMoT) learning algorithm. The model is compared to a multilayer perceptron, Generalized Regression Networks (GRNN) and Kohonen Classification and Intervention Analysis. The models are trained and assessed on load data extracted from EUNITE network competition.

Keywords - Short term load forecasting, Neuro-Fuzzy modelling, LoLiMoT, GRNN, intervention analysis.

1. INTRODUCTION

STLF has an essential role in the operation and planning of electric power systems. It is one of the requirements for activities such as economic dispatch, unit commitment, fuel allocation and maintenance scheduling. There is a great attention to new approaches for the enhancement of forecasting accuracy because of economical and industrial aspects[9,10]. Various modeling approaches are proposed in the literature. ARIMA Models [1,2] are one of the traditional approaches for forecasting issues. ARIMA models and the other classic approaches such as Kalman Filters [3] suffer from nonlinear behavior of dynamical systems. Nonlinear parametric models have attracted a great attention to load forecasting. Artificial Neural networks (ANNs) are applied successfully to STLF. Artificial neural networks are being applied to forecasting problems since their distributed structure of weights and neurons permits to approach complex relationships between variables without specifying them explicitly in advanced[5]. Multiple neural approaches are found in the literature such as [4,5], et. al.

This paper presents a neuro-fuzzy model for short-term load forecasting. This model is identified through Locally Liner Model Tree (LoLiMoT) learning algorithm. The model is compared to a multilayer perceptron, Generalized regression neural networks (GRNN) and an intervention analysis. These models are utilized for time series prediction benchmark in EUNITE competition that is the electricity load forecast competition organised by the European Network on Intelligent Technologies for Smart Adaptive Systems (EUNITE). The electricity load forecast is a challenging problem introduced by the Eastern Slovakian Electricity Corporation, which can bring a very significant financial profit using more accurate prediction technology. The problem is to forecast maximum daily electricity load based on previous data available for electricity load and average daily temperature. The average daily temperature and every half an hour load for the time period January 1997 until December 1998 are given. List of public holidays for the same period of time are also provided. The actual task is to supply the prediction of maximum daily values of electrical loads for January 1999.

The rest of the paper is outlined as follows. Section II describes the Multilayer Perceptron Network. In Section III, intervention analysis is discussed. GRNN and Neuro-fuzzy modeling with LoLiMoT learning algorithm is considered in Section IV and V. Finally, Section VI, presents the simulation results followed by Section VII that concludes the paper.

2.MULTILAYER RPERCEPTRON NETWORK

Using a feedforward artificial neural network to enable forecasting of electrical loads, the choice was between the two most common types; multi-layer perceptrons (MLPs) and radial basis function (RBF) networks[6]. The fundamental difference is the way in which hidden units combine values coming from preceding layers in the network, MLP's use inner products, while RBF's use Euclidean distance. A multi-Layer perceptrons network was created, these are one of the most important types of neural network models[11] . It consists of a set of sensory nodes, the input layer, one or more hidden layers of computing nodes, and an output layer of computing nodes (Fig.1). They are commonly termed "Multilayer Feed Forward Networks. The input signal propagates through the network in a forward direction, hence the term feed-forward. They are trained using an algorithm known as the error back-propagation algorithm. This process is made up of two passes through the network layers, one forward and one backward. In the forward pass, the input is applied to the sensory nodes, and its effect propagates through the network layers, the response then appears at the output nodes. This output is compared against a desired value, to produce an error signal which then propagates backward through the network. In the forward pass, the synaptic weights of the network are fixed, but during the backward pass they are adjusted so that the network output moves closer towards the desired response (Fig.1).

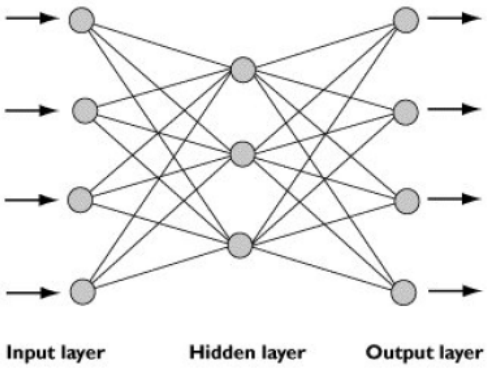

Fig. 1 The structure of Feedforward neural network.

3. CLUSTERING APPROACH

3.1 Kohonen Classification

The Kohonen algorithm [7] is a powerful self-organization process, which has the special property of effectively creating spatially organized "internal representations" of various features of input patterns, and their abstractions. The Kohonen algorithm carries out a distribution of an input space V_I in another space of smaller dimension V_M, preserving the topological relationships among the input vectors. That is, similar input vectors are distributed to close points in the output space(Figure 2). The output space V_M is represented by a two-dimensional array of neurons. The topology conservation is carried out by means of a non-supervised competitive learning, in which each input vector x is compared with the weight vectors w_i of each neuron in the network. The neuron whose weight vector is the nearest to the vector x that is called Best Matching Unit (BMU) is selected, modifying its weights and those of its neighbors according to equation

$$W(t + 1) = W(t) + \alpha(t)(x(t) - W(t)) \qquad (1)$$

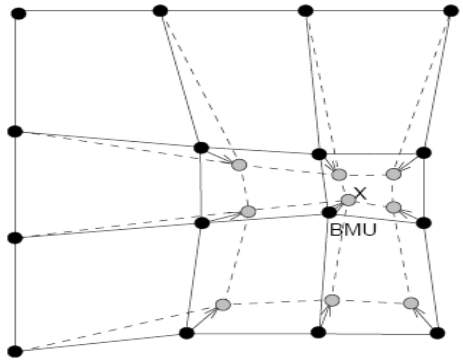

Fig. 2. Updating the best matching unit (BMU) and its neighbours towards the input sample marked with x. The solid and dashed lines correspond to situation before and after updating, respectively.

3.2 intervention analysis

The intervention analysis is based on characteristics about the electrical demand. The main characteristics of electrical demand are the following.

Weekly periodicity with seasonal patterns: The load curves repeat every seven days and are dissimilar for each season.

Nonstationary behaviour: The load changes with the national economy conditions. It grows if economy does, and decreases with economical recession. In the time series analysis methodology, to make stationary series, a differencing operator is applied.

Meteorological influence: It is well known that meteorological variables are the ones that have more influence on the electrical demand in annual average. The influence of the meteorological variables is more important in hours of light and the afternoon than at the first hours of the day and the sleep hours. The peak load occurs when the meteorological variables take bigger influence. That is why, the temperatures (only one available) play a fundamental role in a correct prediction.

The optimal prediction must be adjusted according to these characteristics. Because the number of load pattern is very little, the adjustment cannot be performed by a neuronal paradigm. Thus we use an intervention analysis based on statistical methods.

4. GENERALIZED REGRESSION NEURAL NETWORKS (GRNN)

In 1990, Donald F. Specht proposed a method to formulate the weighted-neighbor method in the form of a neural network. He called this a Probabilistic Neural Network. Here is a diagram of a PNN/GRNN network:

General Regression Neural Networks have similar architectures to MLP, but there is a fundamental difference: Probabilistic networks perform classification where the target variable is categorical, whereas general regression neural networks perform regression where the target variable is continuous. All GRNN networks have four layers:

Input layer – There is one neuron in the input layer for each predictor variable. In the case of categorical variables, *N*-1 neurons are used where *N* is the number of categories. The input neurons (or processing before the input layer) standardizes the range of the values by subtracting the median and dividing by the interquartile range. The input neurons then feed the values to each of the neurons in the hidden layer.

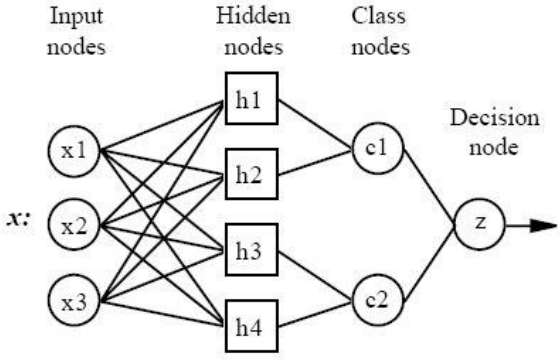

Fig. 3. Structure of GRNN model

Hidden layer – This layer has one neuron for each case in the training data set. The neuron stores the values of the predictor variables for the case along with the target value. When presented with the ***x*** vector of input values from the input layer,

a hidden neuron computes the Euclidean distance of the test case from the neuron's center point and then applies the RBF kernel function using the sigma value(s). The resulting value is passed to the neurons in the pattern layer.

Pattern layer / Summation layer – there are only two neurons in the pattern layer. One neuron is the denominator summation unit and the other is the numerator summation unit. The denominator summation unit adds up the weight values coming from each of the hidden neurons. The numerator summation unit adds up the weight values multiplied by the actual target value for each hidden neuron.

Decision layer – the decision layer divides the value accumulated in the numerator summation unit by the value in the denominator summation unit and uses the result as the predicted target value.

GRNN networks have advantages and disadvantages compared to multilayer perceptron networks:

- It is usually much faster to train a GRNN network than a MLP network.
- GRNN networks often are more accurate than MLP networks.
- GRNN networks are relatively insensitive to outliers (wild points).

5. LOCALLY LINEAR NEURO-FUZZY WITH MODEL TREE LEARNING

The fundamental approach with Locally Linear NeuroFuzzy (LLNF) model is dividing the input space into small linear subspaces with fuzzy validity functions. Any produced linear part with its validity function can be described as a fuzzy neuron. Thus the total model is a neurofuzzy network with one hidden layer, and a linear neuron in the output layer which simply calculates the weighted sum of the outputs of locally linear neurons:

$$\hat{y}_i = \omega_{i_0} + \omega_{i_1} u_1 + \omega_{i_2} u_2 + \ldots + \omega_{i_p} u_p \tag{2}$$

$$\hat{y} = \sum_{i=1}^{M} \hat{y}_i \phi_i \left(\underline{u}\right) \tag{3}$$

This structure is depicted in Fig. 4, where $\underline{u} = \begin{bmatrix} u_1 & u_2 & \cdots & u_p \end{bmatrix}^T$ is the model input, M is the number of LLM neurons, and ω_{ij} denotes the LLM parameters of the ith neuron. The validity functions are chosen as normalized Gaussians; normalization is necessary for a proper interpretation of validity functions:

$$\phi_i \left(\underline{u}\right) = \frac{\mu_i \left(\underline{u}\right)}{\sum_{j=1}^{M} \mu_j \left(\underline{u}\right)} \tag{3}$$

$$\mu_i \left(\underline{u}\right) = \exp\left(-\frac{1}{2} \left(\frac{\left(u_1 - c_{i1}\right)^2}{\sigma_{i1}^2} + \ldots + \frac{\left(u_p - c_{ip}\right)^2}{\sigma_{ip}^2} \right) \right)$$

$$= \exp\left(-\frac{1}{2} \frac{\left(u_1 - c_{i1}\right)^2}{\sigma_{i1}^2} \right) \times \ldots \times \exp\left(-\frac{1}{2} \frac{\left(u_p - c_{ip}\right)^2}{\sigma_{ip}^2} \right) \tag{4}$$

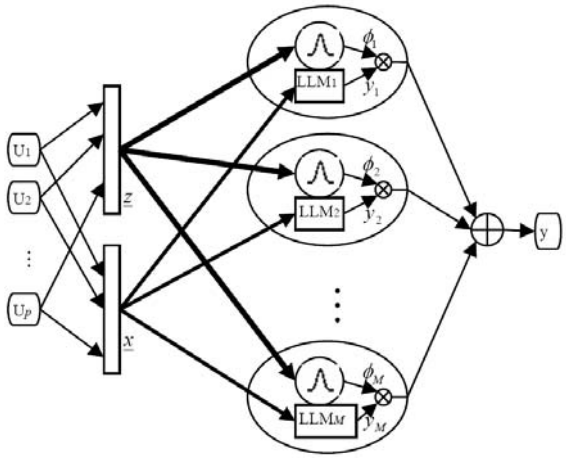

Fig. 4. Structure of locally linear neuro-fuzzy model

The $M \times p$ parameters of the nonlinear hidden layer are the parameters of Gaussian validity functions: center (C_{ij}) and standard deviation (σ_{ij}). Optimization or learning methods are used to adjust the two sets of parameters, the rule consequent parameters of the locally linear models (ω_{ij} s) and the rule premise parameters of validity functions (C_{ij} s and σ_{ij} s). Global optimization of linear consequent parameters is simply obtained by least squares technique. The global parameter vector contains $M \times \left(p + 1\right)$ elements:

$$\underline{\omega} = \begin{bmatrix} \omega_{10} & \omega_{11} & \cdots & \omega_{1p} & \omega_{20} & \omega_{21} & & \\ & & & \cdots & \omega_{M0} & \cdots & \omega_{Mp} \end{bmatrix}^T \tag{5}$$

and the associated regression matrix $\underline{X}$ for N measured data samples is

$$\underline{X} = \begin{bmatrix} \underline{X}_1 & \underline{X}_2 & \cdots & \underline{X}_M \end{bmatrix} \tag{6}$$

$$\underline{X}_i = \begin{bmatrix} \phi_i \left(\underline{u}(1)\right) & \ldots & u_p(1)\phi_i \left(\underline{u}(1)\right) \\ \phi_i \left(\underline{u}(2)\right) & \ldots & u_p(2)\phi_i \left(\underline{u}(2)\right) \\ \vdots & & \vdots \\ \phi_i \left(\underline{u}(N)\right) & \ldots & u_p(N)\phi_i \left(\underline{u}(N)\right) \end{bmatrix} \tag{7}$$

Therefore

$$\underline{\hat{y}} = \underline{X}.\underline{\hat{\omega}} \quad ; \quad \underline{\hat{\omega}} = \left(\underline{X}^T \underline{X}\right)^{-1} \underline{X}^T \underline{y} \tag{8}$$

An incremental tree based learning algorithm is appropriate for tuning the rule premise parameters, i.e. determining the validation hypercube for each locally linear model[8]. In each iteration the worst performing locally linear neuron is determined to be divided. All the possible divisions in the p dimensional input space are checked and the best is performed. The splitting ratio can be simply adjusted as $\frac{1}{2}$, which means that the locally linear neuron is divided into two equal halves on the selected input dimension. Based on such a division the centers (c_{ij}) and standard deviations (σ_{ij}) of the new neurons are computed and the fuzzy validity functions for the new structure are updated according to the equations 10 and 11. The center of validity functions are the centers of the new hyper-cubes, and the standard deviations are usually set as 0.7. The algorithm is as follows [1]:

1. *The initial model:* start with a single locally linear neuron, which is a globally optimal linear least squares estimation over the whole input space with $\Phi_1(\underline{u}) = 1$, and $M = 1$.
2. *Find the worst neuron*: Calculate a local loss function e.g. MSE for each of the $i = 1,\ldots,M$ locally linear neurons, and find the worst performing neuron.
3. *Check all divisions*: The worst neuron is considered for further refinement. The validation hypercube of this neuron is divided into two halves with an axis orthogonal split. Divisions in all dimensions are tried, and for each of the *p* divisions the following steps are carried out:
   a. Construction of the multi-dimensional validity functions for both generated hyper cubes.
   b. Local estimation of the rule consequent parameters for both newly generated neurons.
   c. Calculation of the total loss function or error index for the current overall model.
4. *Validate the best division*: The best of the p alternatives in step 3 is selected. If it results in reduction of loss functions or error indices on training and validation data sets, the related validity functions and neurons are updated, the number of neurons is incremented $M = M + 1$.
5. Test the termination condition: If the termination condition is met, then stop, else go to step 2.

Five iterations of the procedure for an application with two dimensional input space is depicted in Fig.5. This automatic learning algorithm provides the best linear or nonlinear model with maximum generalization, and performs well in prediction applications.

The error index used in the experiments of this study is Normalized Mean Square Error (NMSE), which is defined as

$$NMSE = \left(\frac{\sum_{i=1}^{n} (y - \hat{y})^2}{\sum_{i=1}^{n} (y - \bar{y})^2} \right) \qquad (9)$$

Where y, $\hat{y}$, and $\bar{y}$ are observed data, predicted data and average of observed data respectively.

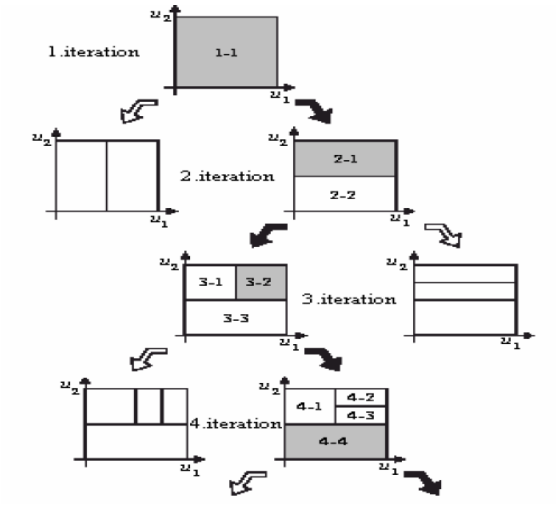

Fig. 5. Operation of the LOLIMOT algorithm in the first five iterations for a two dimensional input space.

6. RESULTS

6.1.Data Description

The historical data have been supplied by world-wide competition within the EUNITE network. The problem to be solved is the forecasting of maximum daily electrical load (peak load) based on electrical load values and temperatures data. The historical data are half an hour loads and average daily temperatures of the time period 1997-1998, including the holidays for the same period of time. Besides, average daily temperatures data of the years 1995 and 1996 are available, too. The actual task is to supply the prediction of maximum daily values of electrical loads for January 1999 (31 data values altogether).

6.2.Load Values

Fig.5 shows the maximum daily load, in MW, for the years 1997 and 1998. One can see that the maximum load for the first days of January 1997 is bigger than January 1998, while that, in general, the maximum load for the rest of months is contrary. This issue can be seen more clearly in the Fig.5. This is due to the different temperatures (table 1).

The LOLIMOT algorithm is implemented as a MATLAB m-file and is used to predict the load of January 1999. The number of iterations is also optimized by an intelligent program: the model will be checked by the test data in each iteration and the training will be stopped when the mean square error (MSE) of test data starts to increase. In this way, the over-fitness is avoided and the most accurate prediction is prepared. All of these models are compared in their optimum performance.

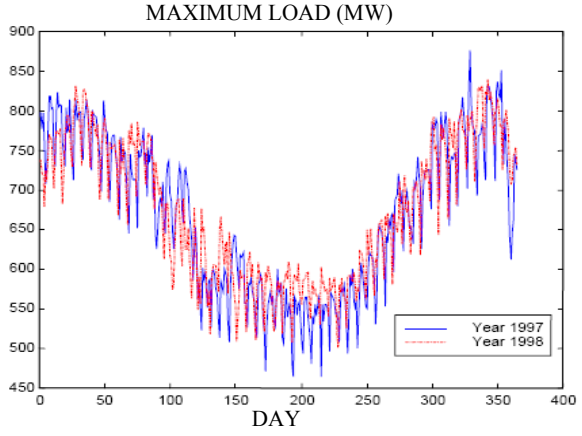

Fig.6. Maximum daily loads from time period 1997-1998.

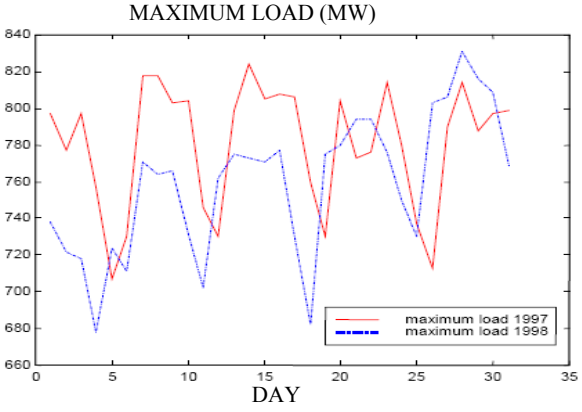

Fig. 7. Maximum load for years 1997 and 1998.

TABLE 1. Average daily load and its percentages of increase during several time periods. (*) minus sign indicates decrease.

	Avereage daiy 1997 (MW)	Avereage daiy 1998 (MW)	Percentage increase(%)
Annual Load	28484	28766	1%
Jannuary Load	34491	32899	-4.61%
December Load	32919	34424	4.57%

The other networks have been implemented to be compared with LoLiMoT; the MLP network with Back propagation learning method, a GRNN model and a kohonen classification and intervention analysis approach. Figure 8 depicts the results achieved by the above models on peak-load forecasting. As can be seen the forecast load curves produced by the LoLiMoT model follow the actual ones and the results from LoLiMoT are very promising. Table 2 contains numerous error criterions for these three methods.

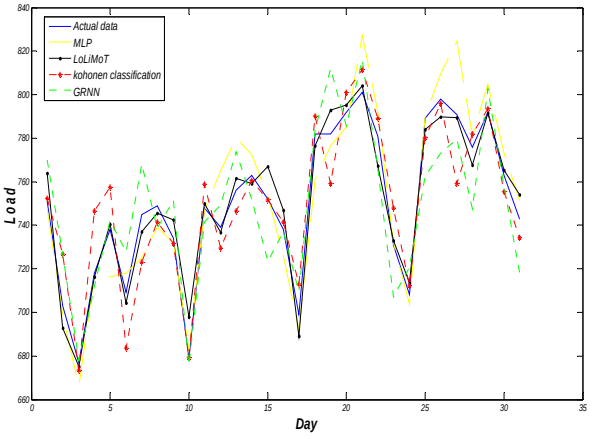

Fig 8. The result achieved by MLP, GRNN, Kohonen classification and LoLiMoT models on daily peak load forecasting.

Table 2 Forecasting errors

	Mean	Max	MAPE	MSE
MLP	0.5246	15.11	6.58	14.45
GRNN	0.26	13.26	4.54	11.83
Classification	0.10	10.54	3.45	10.67
LoLiMoT	0.77	3.14	1.98	6.44

7. CONCLUSION

This paper describes the methodology, implementation, and results of a load forecast procedure. Using historical data of half an hour loads, We use a neuro-fuzzy model that is identified through Locally Liner Model Tree (LoLiMoT) learning algorithm and compare it's result with multilayer perceptron, GRNN model and Kohonen Classification and Intervention Analysis.

The experiments show that the performance of the LoLiMoT model on short-term load forecasts is much better than that of the multilayer perceptron and slightly better than that of the GRNN and kohonen classification. The superior performance displayed by the model seems to be justified by its very flexible and interpretable structure.

REFERENCES

[1] Nelles O (2001) Nonlinear system identification. Springer, Berlin Heidelberg New York.

[2] Khan M.R. (2001), Short term load forecasting for large distribution systems using artificial neural networks and fuzzy logic", PhD Thesis, UVEE, FEI, VUT Brno, Czech Republic.

[3] T. Koskela, M. V, M. Linari, and M. Marsili, "A versatile microwave plethysmograph for the monitoring of physiological parameters (Periodical style)," *IEEE Trans. Biomed. Eng.*, vol. 49, no. 10, pp. 1204–1210, Oct. 2002.

[4] V. Medina, R. Valdes, J. Azpiroz, and E. Sacristan, "Title of paper if known," unpublished.

[5] A. Gholipour,A. Abbaspour, B.N.Araabi and C.Lucas "Enhancements in the prediction of solar activity by locally linear model tree" proceeding of MIC2003:22nd Int.Conf. onmodeling,IdentificationandControl,Innsbruck,Austria,2003,pp.158-161.

[6] Khan M.R., Zak L. and Ondrusek C. (2001), Fuzzy logic based short-term electric load forecasting", 4th InternationalScientific Conference "Elektro-2001" Faculty of Electrical Engineering, University of Zilina, Slovak Republic, ISBN: 80-7100-836-2, pp. 19-24.

[7] Cottrell, M., Girard, B., Rousset, P.: Forecasting of Curves Using a Kohonen Classification. Journal of Forecasting, (1998), Vol. 17, 429-439

[8] A. Gholipour, C. Lucas, B. N. Araabi, M. Mirmomeni, and M. Shafiee, "Extracting the main patterns of natural time series for longterm neuro fuzzy prediction," *Neural Computing and Applications*, doi: 10.1007/s00521-006-0062-x, Aug. 2006.

[9] A. K Sinha, "Short Term Load Forecasting Using Artificial Neural2 Networks", in Proc. Of IEEE Int. Conf on Industrial Technology, Goa,India, 2000, vol. 1, pp. 548-553.

[10] I. Drezga, and S. Rahman, "Short Term Load Forecasting with Local Hours [7] .ANN Predictors", IEEE Trans. Power Systems, vol. 14, pp. 844-850.

[11] P. Werbos, Beyond regression: new tools for prediction andanalysis in the behavioural sciences, Ph.D. Thesis, Harvard University, 1974.