Mutual Information Based Input Selection in Neuro-Fuzzy Modeling for Long Term Load Forecasting

M. Nosrati Maralloo[1] , A. R. Koushki[1] , C. Lucas[2] , M. M. Pedram[3]
[1]Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran. Email: nosratymehdi@gmail.com ; a_r_koushki@yahoo.com
[2] Control and Intelligent Processing Center of Excellence, Electrical and Computer Engineering Department, University of Tehran, Tehran, Iran.Email: lucas@ipm.ir
[3] Computer Engineering Department,Faculty of Engineering, University of Tarbiat Moallem, Karaj/Tehran, Iran. Email: pedram@tmu.ac.ir

ABSTRACT
Long-term load forecasting demand is necessary for the correct operation of electric utilities. There is an on-going attention toward putting new approaches to the task. Recently, Neuro-fuzzy modeling has played a successful role in various applications over nonlinear time series prediction. In modeling, irrelevant inputs cause the deterioration of performance. Therefore, to have an accurate model, some strategies are needed to choose a set of most relevant inputs. Mutual Information (MI) is very effective in evaluating the relevance of each input from the aspect of information theory. This paper presents a neuro-fuzzy model with locally linear model tree (LoLiMoT) learning algorithm for the long term load forecasting of North-American electric utility. Proper inputs which consider historical data are selected by MI.

Keywords
long term load forecasting, Neuro-Fuzzy modeling, LoLiMoT, Mutual Information.

1. INTRODUCTION
Long term load forecasting demand is necessary for the correct operation of electric utilities. Long-term forecasting is usually used for planning, such as determining the future sites of generators; accurate forecasting of electricity prices would enable power marketers and companies to make sound business decisions in a volatile environment [1]. Long-term load forecasting plays an important role in power systems for system planning, construction scheduling of new generating capacity and electricity purchasing of generating units [2]. Many techniques exist for the approximation of the underlying process of a time series: linear methods such as ARX, ARMA, etc. [12,13], and nonlinear ones such as artificial neural networks [7,12]. The common difficulty to all the methods is the determination of sufficient and necessary information for an accurate prediction. There is a great attention to new approaches for the enhancement of forecasting accuracy because of economical and industrial aspects. Various modeling approaches are proposed in the literature. ARIMA models [17,18] are one of the traditional approaches for forecasting issues. ARIMA models and the other classic approaches such as Kalman Filters suffer from nonlinear behavior of dynamical systems [15]. Nonlinear parametric models have attracted a great attention to load forecasting [15]. Artificial Neural networks (NNs) [3-7] have succeeded in several power system problems, such as planning, control, analysis, protection, design, load forecasting, security analysis, and fault diagnosis[22]. 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 advance. Multiple neural approaches are found in the literature such as, et. al. Artificial neural networks and neuro-fuzzy models utilize a learning mechanism.
In learning process, generalization performance on a finite sized training set is closely related to the number of free parameters in the network. The performance of the learning is greatly affected by the selection of model inputs. Input variables that provide little information about the network output generate unneeded weights [8]. The objective of Input selection is to identify a subset of original variables from a given input data set while removing irrelevant and/or redundant variables.
In this paper, locally linear neuro-fuzzy modeling with input selection methodology based on MI is used for long term prediction. This method gives more reasonable inputs and improves generalization performance.
The rest of the paper is outlined as follows. Section 2 describes MI and its estimation. In section 3, an algorithm for input selection using MI is discussed. Neuro-fuzzy modeling with LoLiMoT learning algorithm is considered in section 4. Finally, section 5, presents the simulation results followed by section 6 that concludes the paper.

2. MUTUAL INFORMATION AND ITS ESTIMATION
2.1 Definition of Mutual Information
In probability theory, especially in the information theory the MI can be used for evaluating the dependencies between random variables. In fact, the MI between two random variables, such as X and Y, can be considered as a measure of the amount of knowledge on Y provided by X (or conversely on the amount of knowledge on X provided by Y). If X and Yare to be independent, therefore X contains no information about Y and vice versa; thus the MI between them is zero.
The definition of MI begins from the Shannon Entropy [9] in the information theory [23]. The MI of two random variables X and Y is defined as:

$$I(X; Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = H(X)+ H(Y) - H(X; Y) \quad (1)$$

Where H(X) and H(Y) are the entropies of X and Y, and H(X|Y), H(Y|X) are the conditional entropies, and H(X;Y) is the joint entropy of X and Y that is defined by:

$$H(X) = -\int P_X(x) \log P_X(x)\, dx \quad (2)$$

$$H(Y) = -\int P_Y(y) \log P_Y(y)\, dy \quad (3)$$

$$H(X;Y) = -\int\int P_{X,Y}(x,y) \log P_{X,Y}(x,y)\, dx\, dy \quad (4)$$

Where $P_{X,Y}(x, y)$ and $P_X(x)$ and $p_Y(y)$ are the joint probability density function and marginal density functions of X and Y, respectively. The marginal density functions are given by:

$$P_X(x) = -\int P_{X,Y}(x,y)\, dy \quad (5)$$

$$P_Y(y) = -\int P_{X,Y}(x,y)\, dx \quad (6)$$

MI is the Kullback-Leibler distance between the joint Distribution $P_{X,Y}(x, y)$ and the product distribution $p_X(x).p_Y(y)$. By substituting equations (2) to (4) into (1), the MI equation will be:

$$I(X;Y) = \int\int P_{X,Y}(x,y) \log \frac{p_{X,Y}(x,y)}{p_X(x)\, p_Y(y)}\, dx\, dy \quad (7)$$

In discrete forms the integrations are replaced by summation over all possible values that appear in data. Therefore, it is only required to estimate $P_{X,Y}(x,y)$ in order to estimate the MI between X and Y, by (5) to (7). Histogram- and kernel-based methods are widespread to estimate probability density functions [19]. However, their use is usually restricted to one or two-dimensional probability density functions.

2.2. Estimating MI

A recent estimator based on entropy is used, that is estimated from k-nearest neighbors' statistics [16]. It estimates the MI between two random variables of any dimensional space. The basic idea is to estimate entropy from the average distance to the k-nearest neighbors (over all of data).

In practice, one has a set of N input-output pairs, $z_i=(x_i,y_i)$, i=1,...,N, which are assumed to be realizations of a random variable Z=(X, Y) with density Px y (x, y) Either X and Y have values in R or in R^P, and the algorithm will use the natural norm (Euclidean norm) in those spaces. Input-output pairs are compared through the maximum norm:

$$\|z-z'\|_\infty = \max\{\|x-x'\|, \|y-y'\|\} \quad (8)$$

It can be considered that k is a fixed positive integer, then $z_{k(i)}=(x_{k(i)},y_{k(i)})$ is the k-th nearest neighbor of z_i (with maximum norm). It can be denoted that:

$$\varepsilon_i/2 = \|z-z_{k(i)}\|_\infty \quad (9)$$

$$\varepsilon_i^x/2 = \|x_i - x_{k(i)}\|, \qquad \varepsilon_i^y/2 = \|y_i - y_{k(i)}\| \quad (10)$$

$\varepsilon_i/2$ is the distance from z_i to its k-th neighbor and $\varepsilon_i^x/2$ and $\varepsilon_i^y/2$ are the distances between the same points projected into X and Y subspaces. Obviously, $\varepsilon_i = \max\{\varepsilon_i^x, \varepsilon_i^y\}$. n_i^x, n_i^y are the numbers of sample points with $\|x_i-x_j\| \le \varepsilon_i^x/2$ and $\|y_i-y_j\| \le \varepsilon_i^y/2$. The estimation for MI is then:

$$\hat{I}(X;Y) = \psi(k) - \frac{1}{K} - \frac{1}{N}\sum_{i=1}^{N}[\psi(n_i^x)+\psi(n_i^y)] + \psi(N) \quad (11)$$

Where Ψ is the digamma function:

$$\psi(x) = \frac{\Gamma'(x)}{\Gamma(x)} = \frac{d}{dx}\ln\Gamma(x), \psi(1) = -0.5772156 \quad (12)$$

Where:

$$\Gamma(x) = \int_0^\infty u^{x-1}e^{-u}\, du \quad (13)$$

With a small value for k, this estimator has a large variance and a small bias, whereas a large value of k leads to a small variance and a large bias [10]. In this paper, k= 6 is used.

3. INPUT VARIABLES SELECTION ALGORITHM

This section is devoted to describe the input variables selection algorithm. This algorithm has been used beforehand for feature selection in classification and pattern recognition problems [11], [20], and [21] which is proposed by Battiti in 1994. The objective of this algorithm is to maximize relevance between inputs and output and minimizes the redundancy of selected inputs. This algorithm computes $I(T;l)$ and $I(l;l')$, where l and l' are individual inputs and T is output. The goal of these two terms is to select relevant input with the output which has least dependency with other selected inputs [21]. The algorithm is as follows:

1) *Initialization*: Set L to 'initial set of n inputs'; S to 'empty set'; and T to 'output'.
2) *Computation of the mutual information with the output*:For each input $l \in L$ compute $I(T;l)$.
3) *Choice of the first input*: Find the input l that maximizes $I(T;l)$; Set $L \leftarrow L-\{l\}$, $S \leftarrow \{l\}$.
4) *Greedy selection*: Repeat until desired number of input variables is selected:
   a) *Computation of the mutual information between variables*: For all couples of variables (l, S) with $l \in L$, $s \in S$; compute $I(l;s)$, if it is not already available.
   b) *Selection of the next input*: Chose the input $l \in L$ as the one that maximizes $I(T;l) - \frac{\beta}{|S|}\sum_{s\in S} I(l,s)$; set $L \leftarrow L-\{l\}$, $S \leftarrow S \cup \{l\}$.
5) *Output the set S containing the selected inputs.*

To consider redundancy between input variables, Battiti imports β as a parameter to adjust the relative importance of mutual information between the candidate input and the already selected inputs with respect to the mutual information with the output. If $\beta = 0$ the algorithm only attempts to maximize mutual information with output, so the redundancy between input variables is never considered. If β increases, the total mutual information between already selected inputs influences the selection procedure much and the redundancy is then reduced [11], [21].

4. LOCALLY LINEAR NEURO-FUZZY WITH MODEL TREE LEARNING

4.1. Neuro-fuzzy modeling

The main idea for utilizing the locally linear neurofuzzy(LLNF) model for function approximation is dividing the input space into small linear subspaces with fuzzy validity functions $\varphi_i(u)$. These functions describe the validity of each linear model in its region. The validity

function applied here is the normalized Gaussian function, which is defined as

$$\mu(x) = \exp(-\frac{(x-c)^2}{2\sigma^2}) \quad (14)$$

where c is the center and s is the standard deviation of the Gaussian. The Gaussian function is the membership function (degree of membership of a specific object to the fuzzy sets) used in this study. Each local linear subspace with its validity function is called 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 models (LLMs) as:

$$\hat{y}_i = \omega_{i_0} + \omega_{i_1} u_1 + \omega_{i_2} u_2 + \ldots + \omega_{i_p} u_p \quad (15)$$

$$\hat{y} = \sum_{i=1}^{M} \hat{y}_i \phi_i(\underline{u}) \quad (16)$$

where $\underline{u} = [u_1 \ u_2 \ \ldots \ u_p]^T$ is the model input, M is the number of LLM neurons, and wij 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:

$$\phi_i(\underline{u}) = \frac{\mu_i(\underline{u})}{\sum_{j=1}^{M} \mu_j(\underline{u})} \quad (17)$$

$$\mu_i(\underline{u}) = \exp\left(-\frac{1}{2}\left(\frac{(u_1 - c_{i1})^2}{\sigma_{i1}^2} + \ldots + \frac{(u_p - c_{ip})^2}{\sigma_{ip}^2}\right)\right) \quad (18)$$

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

Each Gaussian validity function has two sets of parameters, centers (c_{ij}) and standard deviations (σ_{ij}) which are the 2M.p parameters of the nonlinear hidden layer. Optimization or learning methods are used to adjust the two sets of parameters, the rule-consequent parameters of the locally linear models (w_{ij}) and the rule premise parameters of validity functions (c_{ij} and σ_{ij}). A least squares optimization method is used to adjust the parameters of local linear models (w_{ij}), and a learning algorithm (described below) is used to adjust the parameters of validity functions (c_{ij} and σ_{ij})[14]. Global optimization of linear parameters is simply obtained by the least squares technique. The complete parameter vector contains M(p + 1) elements:

$$\underline{\omega} = \begin{bmatrix} \omega_{10} & \omega_{11} & \cdots & \omega_{1p} & \omega_{20} & \omega_{21} & \cdots & \omega_{Mp} \end{bmatrix}^T \quad (19)$$

and the associated regression matrix X for N measured data samples is

$$\underline{X} = [\underline{X}_1 \ \ \underline{X}_2 \ \ \ldots \ \ \underline{X}_M] \quad (20)$$

$$\underline{X}_i = \begin{bmatrix} \phi_i(\underline{u}(1)) & \ldots & u_p(1)\phi_i(\underline{u}(1)) \\ \phi_i(\underline{u}(2)) & \ldots & u_p(2)\phi_i(\underline{u}(2)) \\ \vdots & & \vdots \\ \phi_i(\underline{u}(N)) & \ldots & u_p(N)\phi_i(\underline{u}(N)) \end{bmatrix} \quad (21)$$

Thus

$$\underline{\hat{y}} = \underline{X}.\underline{\hat{\omega}} \ ; \ \underline{\hat{\omega}} = (\underline{\hat{X}}.\underline{X} + \alpha \underline{I})^{-1} \underline{X}^T \underline{y} \ ; \ \alpha << 1 \quad (22)$$

where α is the regularization parameter for avoiding any near singularity of matrix X^TX and in this study is empirically set to 0.002. The structure of LLNF is shown in Fig. 1. The remarkable properties of locally linear neuro fuzzy model, its transparency and intuitive construction, lead to the use of least squares technique for rule antecedent parameters and incremental learning procedures for rule consequent parameters. In this paper, Locally Linear Model Tree (LoLiMoT) algorithm as an incremental tree-based algorithm is used to tune the rule premise parameters, i.e. determining the validation hypercube for each locally linear model [14], [15]. 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 fuzzy validity functions for the new structure are updated; their centers are the centers of the new hyper cubes, and the standard deviations are usually set as 0.7.
For more detail refer to [15].

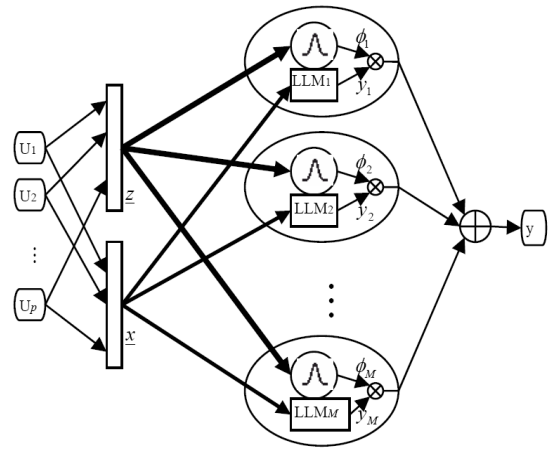

Fig. 1. Structure of locally linear neuro-fuzzy model

4.2. Learning Algorithm

Locally Linear Model Tree (LOLIMOT) is a progressive tree construction algorithm that partitions the input space by axis bisection in all directions of input space. It implements a heuristic search for the rule premise parameters and avoids a time-consuming nonlinear optimization. The LOLIMOT algorithm is described in five steps according to [14]:

1. Start with an initial model: Start with a single LLM, which is a global linear model over the whole input space with $\varphi_1(u) = 1$, and set M = 1. If there is a priori input space partitioning, it can be used as the initial structure.
2. Find the worst LLM: Calculate a local loss function, for example, mean square error (MSE), for each of the i = 1, .. ., M LLMs and find the worst performing LLM.
3. Check all divisions: The worst LLM is considered for further refinement. The hyper rectangle (more than a three-dimensional rectangle or cube) of this LLM is split 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. First, construct the multidimensional membership functions for both generated hyper rectangles and construct all validity functions: In part a, only the membership function of LLM that is split would change and the membership function of other neurons do not change, but all of the validity functions change that must be updated for all LLMs by equation (17). Second, estimate the rule-consequent parameters for newly generated LLMs and third, calculate the loss function for the current overall model.
4. Find the best division: The best of the p alternatives checked in step 3 is selected, and the related validity functions and LLMs are constructed. The number of LLM neurons is incremented M = M + 1.
5. Test the termination condition: If the termination condition

is met, then stop; otherwise, go to step 2. The termination condition is reaching to a predefined error between output (y) and LLNF output with M neuron ($\hat{y}$), that is, when the condition $\| y - \hat{y} \| << \varepsilon$ is satisfied. In practice we used a predefined number of neurons to LOLIMOT, plotted the error as a function of this number, and kept increasing the number of neurons until satisfactory performance was obtained. A suitable number of LLMs would be fit to training data on the basis of a validation set. The best number of LLMs is that in which the root mean square error (RMSE) for the validation set starts to increase. Details can be found in work by Nelles[14].

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 selected. The splitting ratio can be simply set to 0.5, which means that the locally linear neuron is divided into two halves. The fuzzy validity functions for the new construction are updated; their centers are the centers of the new hyper cubes (more than a three-dimensional cube), and the standard deviations are usually set to 0.7 times the width of the hypercube in that dimension.

Fig. 2 illustrates the operation of the LOLIMOT algorithm in the first four iterations for a two-dimensional input space. In iteration 1, a global linear model is fit to data. Then for refinement, input space is split into halves, and a local linear model is fit in each hyper rectangle. In iteration 2, first, the best possible splitting method is selected (e.g., in Fig.2 , iteration 2 splitting along the u_2 axis is assumed to be better), then in the selected model, the worst LLM should be used for further refinement (shaded rectangle or 2-1, for instance), and the algorithm continues with a default number of LLMs.

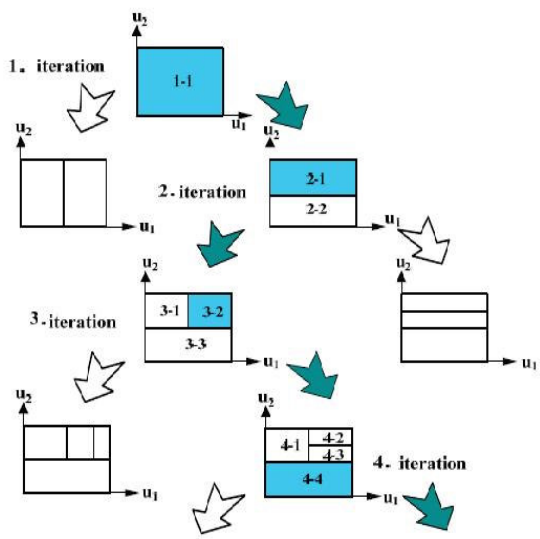

Fig. 2. Operation of the LOLIMOT algorithm in the first five iterations for a two dimensional input space.

5. RESULTS

5.1. Data set

In order to compare the proposed method with the one in [22], the same data set is used, i.e. North-American electric utility load data.

The electric peak-load values range in the intervals [1528,4635] MW. The load data is preprocessed using ordinary normalization (minimum and maximum values in the [−0.5,0.5] range). There is no particular treatment for holidays.

The dataset is shown in Fig. 3.

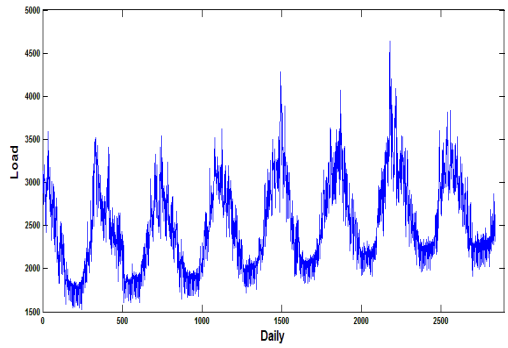

Fig.3. Dataset of The North-American electric

5.2. Experiments

The proposed methods are applied to the above data. The aim of the input selection in the case long term load forecasting is to find which lagged values of load time series are suitable. By applying the mutual information technique, 3 of the inputs that have the most significant impact on the output are identified and used. The results show a notable improvement over those attained without the input selection stage. Fig. 4 to Fig. 7 shows the predictive power of the LoLiMoT algorithm, and the effect of input selection via the mutual information technique.

Four experiments reported in [22] are carried out by the proposed method in this paper. In the first experiment, the training set contains 104 weekly peak-loads from 1985 to 1986. The testing set contains 52 weekly peak-loads from 1987. The objective to foresee the 104 weekly peak-loads from 1988 to 1989. The second experiment is quite similar to the first. In it, the training set includes 208 weekly peak-loads from 1985 to 1986, and from 1988 to 1989. The testing set remains the same, and forecasting spans the time horizon from 1990 to 1991. The first and second experiments are done based on 52 step ahead forecast.

In the third experiment, the training set contains 24 monthly peak-loads from 1985 to 1986. The testing set contains 12 monthly peak-loads from 1987. Objective to foresee the 24 monthly peak-loads from 1988 to 1989. The fourth experiment is quite similar to the third as well. The training set includes 48 monthly peak-loads from 1985 to 1986, and from 1988 to 1989. The testing set remains the same, and forecasting spans the time horizon from 1990 to 1991. These two experiments are carried out based on 12 step ahead forecast.

Table 1 shows forecasting errors, above experiments in weekly and monthly peak load forecasting.

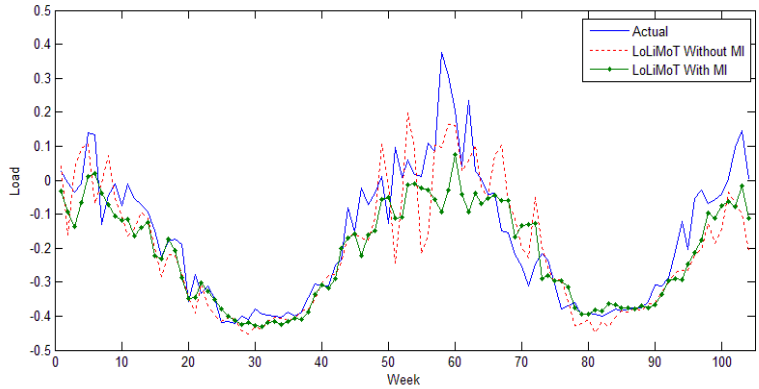

Fig. 4. First experiment.

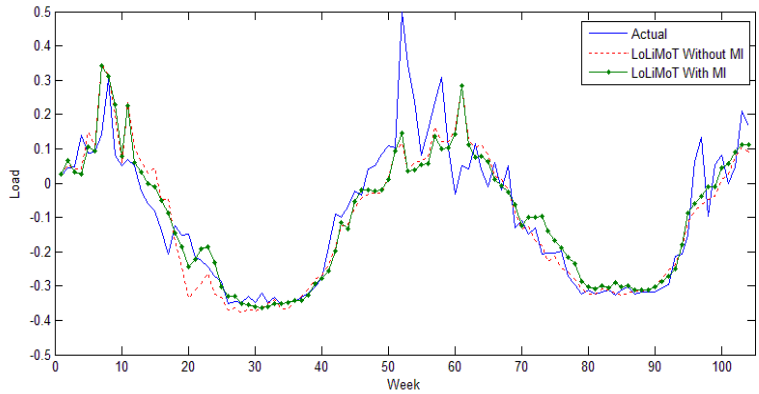

Fig. 5. Second experiment.

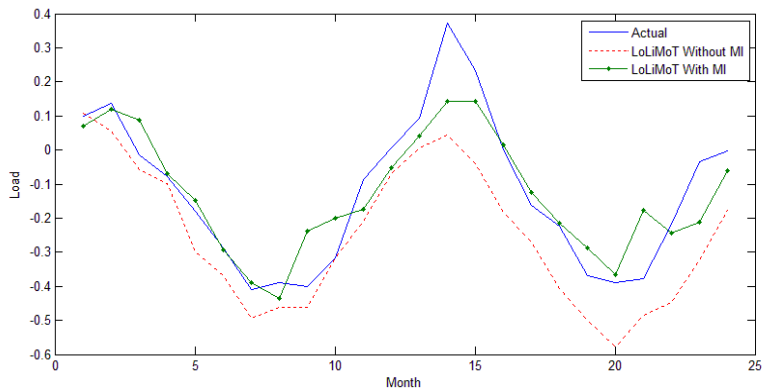

Fig. 6. Third experiment.

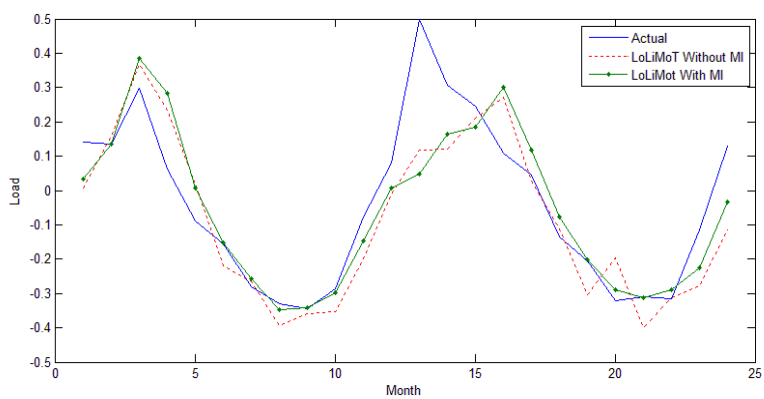

Fig. 7. Fourth experiment.

Table 1
Forecasting errors – mean, maximum, and minimum absolute error, mean square error (MSE) – in the four experiments.

Model	Errors	Experiments			
		1	2	3	4
LoLiMoT with MI	Mean	0.063	0.056	0.070	0.084
	Max	0.462	0.352	0.221	0.452
	Min	0.000	0.000	0.000	0.000
	MSE	0.010	0.007	0.009	0.016
LoLiMoT without MI	Mean	0.071	0.059	0.12	0.102
	Max	0.343	0.383	0.320	0.382
	Min	0.000	0.000	0.000	0.000
	MSE	0.012	0.008	0.023	0.017

6. CONCLUSION

One of the most successful applications of the neuro-fuzzy model to real-world problems is in the area of electric load forecasting. In this paper, a neuro-fuzzy model with locally linear model tree (LoLiMoT) learning algorithm was implemented for long term load forecasting. Inputs were selected using MI for the model considering historical data of the North-American electric utility. In this approach 3 of the more relevant inputs were kept in order to increase the speed and computational power of the LoLiMoT algorithm. Experimental results show that this method has satisfactory results.The experiments also show that the performance of the LoLiMoT with MI based input selection on long term load forecasts is better than that of the LoLiMoT without MI input selection.

REFERENCES

[1] Mohammad Shahidehpour, Hatim Yamin, Zuyi Li, "Market Operations in Electric Power Systems: Forecasting, Scheduling, and Risk Managementg" , pp. 57-113 ,2002.

[2] N.X.Jia, R.Yokoyama, Y.C.Zhou, "A Novel Approach to Long Term Load Forecasting Where Functional Relations and Impact Relations Coexist" ,IEEE, 2008.

[3] M. Cottrell, B.Y. Girard, M. Mangeas, C. Muller, Neural modeling for time series: a statistical stepwise method for weight elimination, IEEE Trans. Neural Networks 6 (6) pp. 1355–1364, 1995

[4] M. Cottrell, B. Girard, P. Rousset, Forecasting of curves using a Kohonen classification, J. Forecasting 17,pp. 429–439, 1998.

[5] A. Lendasse, E. de Bodt, V. Wertz, M. Verleysen, Nonlinear Financial time series forecasting - application to the Bel20 stock market index, European J. Econom. Social Systems 14 (1) , pp. 81–91, 2000.

[6] J. Sjoberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.Y. Glorennec, H. Hjalmarsson, A. Juditsky, Non linear black box modeling in system identification: an unified overview, Automatica 33, pp. 1691–1724, 1997.

[7] A.S. Weigend, N.A. Gershenfeld, Times Series Prediction: Forecasting the future and Understanding the Past, Addison-Wesley, Reading, MA, 1994.

[8] A. S. Weigend, B. A. Huberman, D. E. Rumelhart, "Predicting the future: a connectionist approach", Int. Journal of Neural Systems, vol. 1, pp. 193-209, 1990.

[9] Shannon, C. E., (1948) A Mathematical Theory of Communication. The Bell Systems and Technology. 27 pp.379–423.

[10] F. Rossi, A. Lendasse, D. Francois, V. Wertz, and M. Verleysen, "Mutual information for the selection of relevant variables in spectrometric nonlinear modeling", Chemometrics and Intelligent Laboratory Systems, vol. 80, pp. 215-226, 2006.

[11] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Trans. On Neural Networks*, vol. 5, pp. 537–550, 1994.

[12] Antti Sorjamaa, Jin Hao, Nima Reyhani, Yongnan Ji, Amaury Lendasse," Methodology for long-term prediction of time series", Elsevier, 2007.

[13] L. Ljung, System Identification Theory for User, Prentice-Hall, Englewood Cliffs, NJ, 1987

[14] O. Nelles, *Nonlinear system identification,* Springer Verlag, Berlin,2001.

[15] A. H. Vahabie, M. M. Rezaei Yousefi,B. N. Araabi and C. Lucas,"Mutual Information Based Input Selection in Neuro-Fuzzy Modeling for Short Term Load Forecasting of Iran National Power System", IEEE International Conference on Control and Automation Guangzhou, CHINA - May 30 to June 1, 2007

[16] Kraskov, A., Stögbauer, H., Grassberger, P., Estimating mutual information, *Phys. Rev*., E. 69, 066138, 2004.

[17] G.E.P. Box, G. Jenkins, Time Series Analysis: Forecasting and Control, Cambridge University Press, Cambridge, 1976.

[18] L. Ljung, System Identification—Theory for User, Prentice-Hall, Englewood CliPs, NJ, 1987.

[19] Scott, D. W., (1992) Multivariable Density Estimation:Theory, Practice, and Visualization. New York: Wiley.

[20] A. Al-Ani, M. Deriche, "An optimal feature selection technique using the concept of mutual information," *Int. Symposium on Signal Processing and its Applications (ISSPA)*, Kuala Lumpur, Malaysia, Aug., pp. 477-480, 2001.

[21] N. Kwak, C. H. Choi, "Input feature selection for classification problems," *IEEE Trans. on Neural Networks*, v.

[22] Ot´avio A.S. Carpinteiro, Rafael C. Leme , Antonio C. Zambroni de Souza ,Carlos A.M. Pinheiro , Edmilson M. Moreira ,"Long-term load forecasting via a hierarchical neural model with time integrators", Elsevier B.V., 2006

[23] Cover, T. and Thomas, J., *Elements of Information Theory* , John Wiley, 1990.