Short term Load Forecasting of Iran's Power Systems with Neurofuzzy Application and SSA Analysis

Hossein Chegini
Computer Eng. Dept, Faculty of Eng., Islamic Azad University of Tehran-Science and Research Branch, Tehran, Iran.
hossein.computer@gmail.com

Abdolreza Shirvani
Computer Eng. Dept, Faculty of Eng., Islamic Azad University of Tehran-Science and Research Branch, Tehran, Iran.
abdols@gmail.com

Mir Mohsen Pedram
Computer Eng. Dept., Faculty of Eng., Tarbiat Moallem University of Tehran, Karaj/Tehran, Iran.
pedram@tmu.ac.ir

G. Reza Yousefi
Electrical Eng. Dept, Islamic Azad University-Saveh Branch, Saveh, Iran.
reza.yousefi@gmail.com

Caro Lucas
Electrical and Computer Engineering Department, Control and Intelligent Processing Center of Excellence, University of Tehran, Tehran, Iran.
lucas@ipm.ir

Abstract

Financial marketing, space weather, load forecasting, earthquake and etc are important and complicated systems that their prediction helps to avoid and prevent their pernicious behavior. For instance, load forecasting that we used in this paper can lead to know sensitive and important periods of load time series sooner and control them by operation of powerhouse. The short term prediction (equally one day in this paper) can aim this target because of its accuracy. The technique used in this paper is a branch of Takagi-Sugeno neurofuzzy application which builds a regression tree from a candidate observed data. The name of this application is LoLiMot (Locally linear model tree) and is used to predict 1 day of Iran Power System load. The data is obtained from 17 generator devices dispersed in some cities like Tehran, Mashhad and other big cities with one hour sampling. A new method named SSA (Singular Spectral Analysis) first separate strong signals from noises and feed these signals called PC (principle components) to LoLiMot network. The results obtained from SSA+LoLiMot is so better than the LoLiMot alone. The case study on data described above proves this claim.

Keywords

Forecasting, Neurofuzzy, Regression tree, Singular spectral analysis, Time series

1. INTRODUCTION

The use of neurofuzzy modeling has been used among many complicated problems like prediction .Because of using both statistical aspect and neural networks it is capable of implementing many nonlinear systems like space weather, financial data , load data and others which have high chaotic degree .

In this paper an implementation of a Locally Linear Model Tree (LoLiMot) network is used to improve our result comparing with other networks like MLP (multi layer perceptron) and RBF (radial basis function). Then we used the SSA (singular spectral analysis) which tries to decompose signals from noises and do the prediction on principle components.

In our case study we used our method on a known data to obtain a comparison and verify our application's accuracy. At the end we used our method on load data and showed our results.

In section 2 we discuss LoLiMot algorithm. In section 3 we show the SSA analysis and the principle components the LoLiMot use to predict. In section 4 a case study is shown to get a comparison between our algorithm and an algorithm applied on a known dataset. The conclusion is presented in section 5 which indicate our results were satisfactory.

2. NEUROFUZZY MODEL

The fundamental approach with the locally linear neuro fuzzy (LLNF) model is dividing the input space into small linear subspaces with fuzzy validity functions as in [1]. 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:

$$\hat{y} = \omega_{i0} + u_1\omega_{i1} + u_2\omega_{i2} + \cdots + u_p\omega_{ip} \quad (1)$$

$$\hat{y} = \sum_{i=1}^{M} \underline{\hat{y}}_i \ \varphi_i(\underline{u}) \quad (2)$$

where $\underline{u} = [u_1\, u_2\ \cdots\ u_p]$ is the model input, M is the number of locally linear neurons, and ω_{ij} denotes the linear estimation parameters of the ith neuron. The validity functions are chosen as normalized Gaussians:

$$\varphi_i(\underline{u}) = \frac{\mu_i\,(\underline{u})}{\sum_{j=1}^{M}\mu_j\,(\underline{u})} \quad (3)$$

$$\mu_i(\underline{u}) = \exp(-\frac{1}{2}\left(\frac{(u_i - c_{i1})^2}{\sigma_{i1}^2} + \cdots + \frac{(u_p - c_{i1})^2}{\sigma_{ip}^2}\right)) \quad (4)$$

The *M.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 using least-squares technique discussed in [1].

An incremental tree-based learning algorithm, e.g. locally linear model tree (LoLiMoT), is appropriate for tuning the rule premise parameters, i.e. determining the validation hypercube for each locally linear model. 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 which means that the locally linear neuron is divided into two equal halves. The fuzzy validity functions for the new structure are updated as is suggested in [3] and [4].

Their centers are the centers of the new hypercubes, and the standard deviations are usually set as 0.7. The algorithm is as follows:

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 $\varphi_1(u) = 1$ 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 hypercubes.
(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 $M = M + 1$; and the algorithm continues from Step 2, otherwise the learning algorithm is terminated.

The steps above are illustrated in Fig. 1. A two dimensional input space (u1 and u2) is assumed. Four steps are done in this input space and the divisions till 4^{th} step are shown. As we see in step two the space 2-1, in step 3 the space 3-2, and in step 4 the space 4-4 are candidate for division. As we go further steps in LoLiMot algorithm the spaces become smaller. In Fig. 1 there is a two dimensional input space and therefore two cases should be compared.

3. SINGULAR SPECTRAL ANALYSIS

SSA is defined as a new tool to extract information from short and noisy chaotic time series [10], [11] and [5]. It relies on the Karhunen–Loeve decomposition of an estimate of covariance matrix based on M lagged copies of the time series. Thus, as the first step, the embedding procedure is applied to construct a sequence $\{\tilde{X}(t)\}$ of M-dimensional vectors from time series $\{\tilde{X}(t): t = 1, \cdots, N\}$:

$$\tilde{X}(t) = \left(X(t), X(t+1), \cdots, X(t+M-1)\right) \quad , \quad t = 1, \cdots, \acute{N}, \quad \acute{N} = N - M + 1 . \tag{5}$$

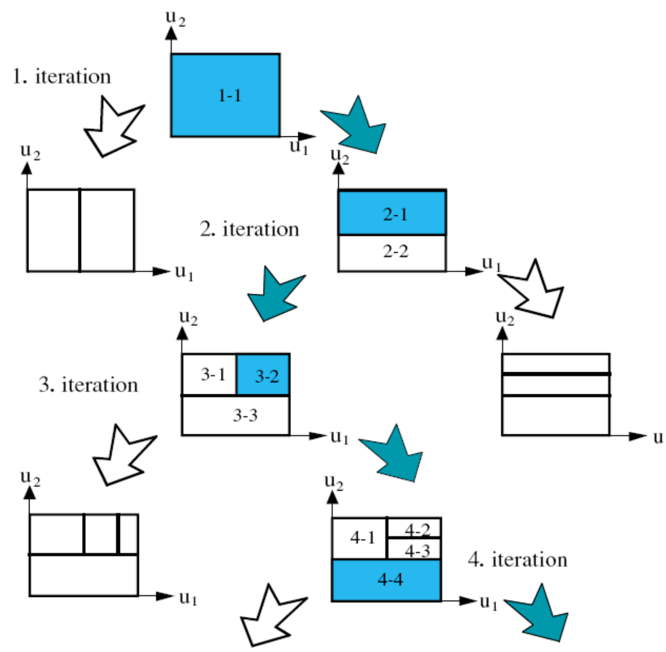

Fig.1.Representation of LoLiMot algorithm on a two-dimensional input space.

The $\acute{N} \times M$ trajectory matrix (D) of the time series has the M-dimensional vectors as its columns, and is obviously a Hankel matrix (the elements on the diagonals i+j = const are equal). In the second step,the $M \times M$ covariance matrix CX is calculated and its eigenelements $\{(\lambda_k, \rho_k) : k = 1, \ldots, M\}$ are determined by singular value decomposition (SVD).

Each eigenvalue , λ_k, estimates the partial variance in the direction of , ρ_k, and the sum of all eigenvalues equals the total variance of the original time series. In the third step, the time series is projected onto each eigenvector and yields the corresponding principal components:

$$A_k(k) = \sum_{j=1}^{M} X(t+j-1)\rho_k(j) \tag{6}$$

Each of the principal components, being a nonlinear or linear trend or a periodic or quasi-periodic pattern,has narrow band frequency spectra and well-defined characteristics to be estimated. As the fourth step, the time series is reconstructed by combining the associated principle components:

$$R_K(t) = \frac{1}{M_t} \sum_{k \epsilon K} \sum_{j \epsilon L_t}^{U_t} A_k(t-j+1)\rho_k(j) \tag{7}$$

The normalization factor (M_t) and the lower (L_t) and upper bounds (U_t) of the reconstruction procedure differ for the center and edges of the time series, and are defined by the following formula:

(M_t, L_t, U_t)

$$= \begin{cases} \left(\frac{1}{t}, 1, t\right), & 1 \le t \le M-1 \\ \left(\frac{1}{M}, 1, M\right), & M \le t \le \acute{N} \\ \left(\frac{1}{N-t+1}, t-N+M, M\right) & N+1 \acute{\le} t \le N \end{cases} \quad (8)$$

TABLE 1

NUMERICAL RESULTS BETWEEN MLP,RBF AND LOLIMOT

	Testing error
MLP	3.2155
RBF	2.4518
LoLiMot	5.486e-002

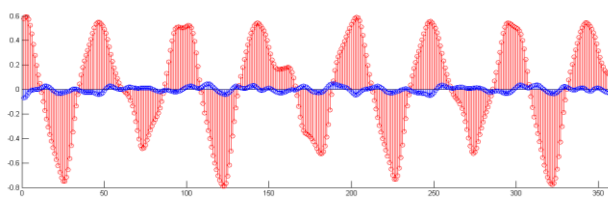

Fig2. Comparison of errors between LoLimot and LoLiMot+SSA algorithm for the same data set. The red stems show the LoLiMot error and the blue stems show the LoLiMot+SSA algorithm .

4. CASE STUDY

In this section we present our results in Mackey glauss time series for 30 step ahead prediction. In Table 1 comparison is among LoLiMot, MLP and RBF. Because LoLiMot has better results, it is the selective network to predict the load data.

After applying the SSA and choosing 12 strongest PCs (principle components) for using in LoLiMot network, the error decreased and the numerical results in Table 2 shows this. The output of LoLiMot+SSA algorithm is shown in Fig. 3 and Fig.4. In these figures blue curves shows the time series and the green curves shows the network's ouotput.

Fig. 2 depicts the error comparison of both algorithms for 400 data points. Red stems in Fig. 2 show the error for LoLiMot itself and the blue ones show error of LoLimot and SSA together. It is clear that the second method decreases the error significantly.

TABLE 2

THE ERROR FROM BOTH ALGORITHMS AFTER EFFICIENT EPOCHES

	LoLiMot error	LoLiMot+SSA error
Testing set	5.486e-002	5.76e-003

In next step we used both methods to predict one day ahead the load data. For predicting this time series with LoLiMot+SSA, we first implemented the SSA algorithm and obtained required information (PCs). Fig. 5 depicts the first 60 PCs as eigenvalues. The first12 principle components are selected and others are avoided for enhancing the signal to noise ratio. .

In this method as the number of components increase the data get more noise attribute and ignoring it can avoid overfitting problem. In this case 200 data points are used as training set, 100 data points as validation set and 100 data points as testing set and the horizon of prediction is 24 step ahead. Fig. 5 shows the principle components for this data set that are used for prediction. Fig. 6 also depicts three avoided PCs. It is clear that these signals are noisy. Table 3 also shows numerical results of LoLiMot+SSA algorithm on Load data for both training and testing set.

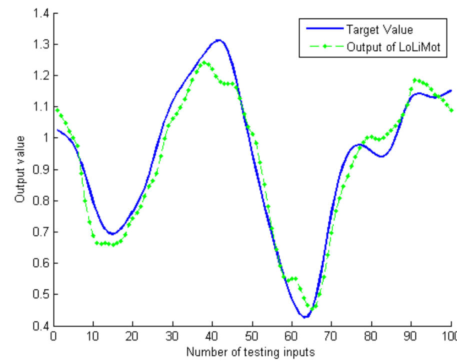

Fig.3. The Output of LoLiMot for Mackey-glauss time series in 30 step ahead prediction .

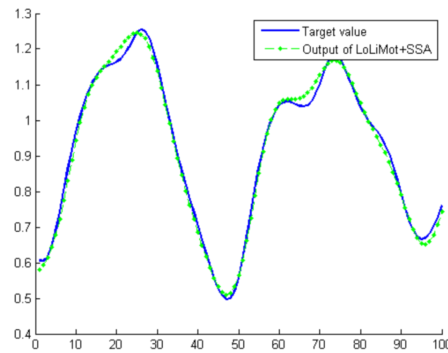

Fig.4. The Output of LoLiMot+SSA for Mackey-glauss time series in 30 step ahead prediction .

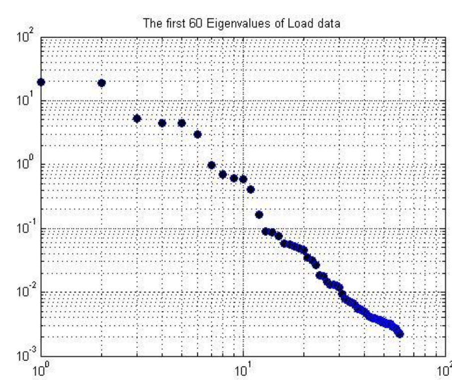

Fig.5. The first 60 PCs obtained from Load data with SSA analysis.

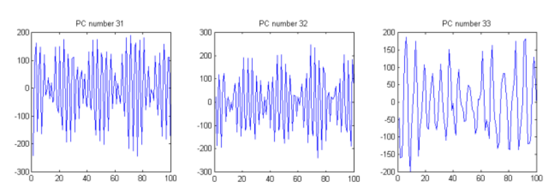

Fig.6 The principle components with number 31,32,33. As it shows these principles get more noise attributes and they are ignored in training (Components after 12 are ignored in this paper)

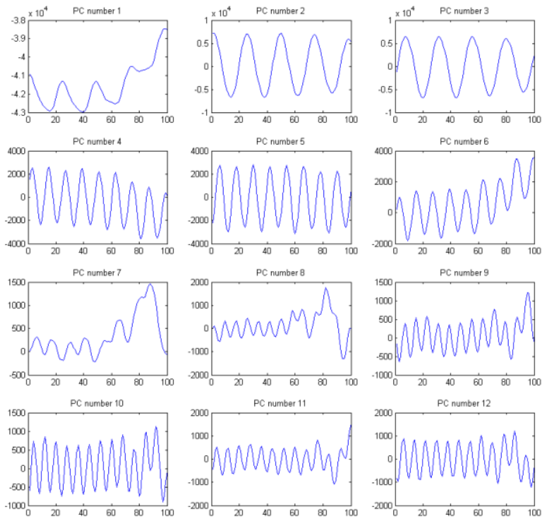

Fig. 7 The first 12 principle components of load data obtained with SSA analysis. These components are selected for training the LoLiMot network .

TABLE 3
NUMERICAL RESULTS FOR LOAD DATA

		LoLiMot+SSA
Training set		1.9835e-002
Testing set		2.625e-002

Fig. 8 shows the network output of LoLiMot+SSA for load data on 200 training data points .Fig. 9 shows the same network's output on 100 testing data points. Again in these figures the green curves shows the network's output and the blue curves the time series.

5. CONCLUSION

With short term prediction, a reliable accuracy can be obtained and in critical situations the network can be used as an alerting system in a powerhouse too. In this paper we tried to use a powerful tool for predicting the load data. First by SSA we separated most powerful principles from noises in a time series. Then with a locally linear model tree algorithm the dynamics of each principle components captured and in the last step the main prediction for the original time series reconstructed. In this paper the performance of LoLiMot with some earlier methods like MLP and RBF was compared to show its high efficiency and then a second comparison was made between loLiMot and LoLiMot+SSA to show how well the second method can perform the LoLiMot algorithm. The macky-glauss time series was selected as a basis for comparison between LoLiMot and LoLIMOt+SSA and after observing the improvements the second method was applied to load data.We hope to outperform the application in long term load forecasting and reaching more accurate results.

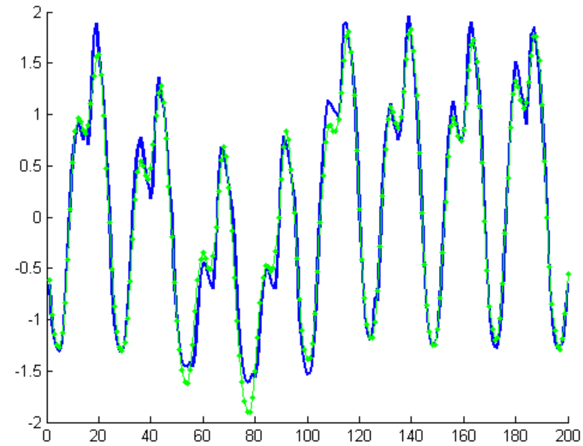

Fig.8. The Output of LoLiMot+SSA for training data set in 24 step ahead prediction .

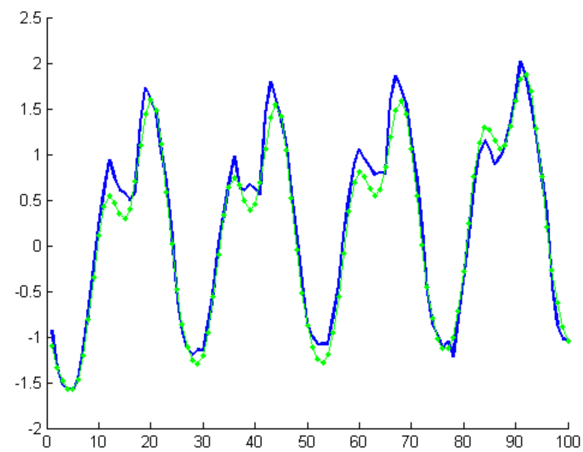

Fig.9. The Output of LoLiMot+SSA for testing data set in 24 step ahead prediction.

REFERENCES

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

[2] Haykin S (1994) Neural networks: a comprehensive foundation. Macmillan, New York

[3] Javad Sharifie, Caro Lucas, and Babak N. Araabi(2006)Locally linear neurofuzzy modeling and prediction of geomagnetic disturbances based on solar wind conditions

[4] Masoud Mirmomeni,Caro Lucas, Babak Nadjar Araabi, and Masoud Shafiee (2007) Forecasting sunspot numbers with the aid of fuzzy descriptor models

[5] Javad Sharifi, Babak N. Araabi, and Caro Lucas(2006) Multi-step prediction of Dst index using singular spectrum analysis and locally linear neurofuzzy modeling

[6] ALI GHOLIPOUR, BABAK N. ARAABI and CARO LUCAS(2006) Predicting Chaotic Time Series Using Neural and Neurofuzzy Models: A Comparative Study

[7] A. Pedram, M. R. Jamali, T. Pedram, S. M. Fakhraie, and C. Lucas(2006) Local Linear Model Tree (LOLIMOT) Reconfigurable Parallel Hardware

[8] Lennart Ljung (2000) ,system Identification theory for the user

[9] Vautard, R., Ghil, M., 1989. Singular spectrum analysis in nonlinear dynamics with applications to paleoclimatic time series. Physica D 35, 395–424.

[10] Vautard, R., Yiou, P., Ghil, M., 1992. Singular spectrum analysis: a toolkit for short noisy chaotic signals. Physica D 58, 95–126.