Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2015) 23: 201 – 211

doi:10.3906/elk-1301-134

Short-term load forecasting using mixed lazy learning method

Seyed-Masoud BARAKATI[1,*], **Ali Akbar GHARAVEISI**[2], **Seyed Mohammad Reza RAFIEI**[3]
[1]Department of Electronics Engineering, Sistan and Baluchestan University, Zahedan, Iran
[2]Department of Computer Engineering, Shahid Bahonar University of Kerman, Kerman, Iran
[3]Department of Electrical Engineering, Islamic Azad University, Garmsar Branch, Iran

Received: 20.01.2013 • **Accepted:** 30.03.2013 • **Published Online:** 12.01.2015 • **Printed:** 09.02.2015

Abstract: A novel short-term load forecasting method based on the lazy learning (LL) algorithm is proposed. The LL algorithm's input data are electrical load information, daily electricity consumption patterns, and temperatures in a specified region. In order to verify the ability of the proposed method, a load forecasting problem, using the Pennsylvania-New Jersey-Maryland Interconnection electrical load data, is carried out. Three LL models are proposed: constant, linear, and mixed models. First, the performances of the 3 developed models are compared using the root mean square error technique. The best technique is then selected to compete with the state-of-the-art neural network (NN) load forecasting models. A comparison is made between the performances of the proposed mixed-model LL as the superior LL model and the radial basis function and multilayer perceptron NN models. The results reveal significant improvements in the precision and efficiency of the proposed forecasting model when compared with the NN techniques.

Key words: Lazy learning, radial basis function, multilayer perceptron, neural networks, mixed model lazy learning, electric power load forecasting

1. Introduction

Accurate load forecasting models are essential for the planning and development of power systems. The purchase, sale, production, and distribution of electrical energy depend on the accurate forecasting of the demand patterns [1]. Electrical load forecasting techniques are designed for short-term (i.e. from 1 h to 1 week), mid-term (i.e. from 1 week to 1 year), and long-term (i.e. for more than 1 year) time frames.

Short-term load forecasting is used to determine the capacity and the level of electrical energy provisioning to meet the expected demand. Automatic generation control and cost-effective load distribution depend on the accuracy and efficiency of short-term load forecasting. Researchers and practitioners have proposed a wide range of techniques to forecast short-term electrical loads and demands. The techniques are mostly based on statistical and time-series analysis [1], learning algorithms, or expert systems [2]. Neural networks (NNs) [3–7], fuzzy expert systems [8–10], wavelet-based networks [11–13], or a combination of these methods [14–17] are examples of expert systems that have been investigated for short-term prediction in the literature.

The lazy learning (LL) method is a local learning technique where a prediction is extracted by locally interpolating the neighboring examples of a received query [18,19]. The query is considered to be related according to a distance measure. When a prediction for a specific query point is required, a set of local models, each with a different polynomial degree (i.e. constant, linear, or squared), is defined. The polynomials include

*Correspondence: smbaraka@ece.usb.ac.ir

a different number of neighbors. The generalization ability of each model is then assessed through a local cross-validation process. A prediction is then obtained either by combining or selecting the different models based on the statistics of their cross-validation errors. This method has been widely used in many applications, such as multilabel learning, systems modeling and control [20,21], [22], air quality measurements and predictions [23], time-series prediction [24–27], and data labeling and filtering [26].

In this paper, a novel LL method using constant, linear, and mixed (linear and constant) models is applied to predict the electric load of the Pennsylvania-New Jersey-Maryland (PJM) Interconnection for the 2005–2007 period. The PJM Interconnection is a regional US transmission organization that coordinates the movement of wholesale electricity in all or parts of Delaware, Illinois, Indiana, Kentucky, Maryland, Michigan, New Jersey, North Carolina, Ohio, Pennsylvania, Tennessee, Virginia, West Virginia, and the District of Columbia [28]. The data available for the year 2001 through to the first half of 2004 are used to train the model. Short-term load forecasting is performed for the following 2 years (i.e. 2005 and 2006). Finally, a comparison is made between the simulation results of the 3 proposed LL methods with that of the existing radial basis function (RBF) and multilayer perceptron (MLP) NN techniques. All of the simulations are performed in the MATLAB environment.

In the following section, the LL algorithm is introduced. In Section 3, important factors in electric load forecasting are discussed. Application of the LL method to the short-term load forecasting and simulation results are discussed in Section 4. Section 5 is dedicated to a comparison of the proposed method with the RBF and MLP NN techniques. Conclusions of this research are discussed in Section 6.

2. The LL algorithm

LL suspends all of the computations until an explicit request for a prediction is received. Once a request is made, LL interpolates the local samples. The interpolation is conducted including the relevant samples and in accordance with a specified distance measure. Each prediction therefore requires a local modeling process consisting of a parametric and a structural identification [18,29]. The parametric identification involves the parameters' optimization process of the local approximation, and the structural identification involves the selection of [27,30]:

- a family of local approximations,
- a metric to evaluate and select the relevant samples, and
- a bandwidth by indicating the size of the region.

The structural identification determines whether the data are correctly modeled from the members of the chosen family of approximations. In other words, when a prediction is required for a specific query point, a set of local models is identified, each with a different polynomial degree and different number of neighbors. The suitability of each model is then assessed through a local cross-validation process. The local model can be built around the points where the approximation is requested. Finally, a prediction is obtained either by combining or selecting the different models based on the cross-validation error statistics of the models.

The least square error is normally used for identification of the local models. The key advantage of the LL models is in their simplicity and low computational requirements.

2.1. Local weighted regression

Consider 2 variables and $y \in R$, where the mapping $f : R^m \to R$, is known only through a set of n samples, obtained as follows:

$$y_i = f(x_i) + \varepsilon_i, \tag{1}$$

where $\forall i$, ε_i is a random variable, such that $E[\varepsilon_i] = 0$ and $E[\varepsilon_i \varepsilon_j] = 0$, $\forall j \neq i$ (E denotes the expectation operator), $E[\varepsilon_i^r] = \mu(x_i)$, $\forall r \geq 2$, and $\mu_r(.)$ is the unknown rth moment of the distribution of ε_i that is defined as a function of x_i. The problem of local regression can be stated as the problem of estimating the value that the regression function, $f(x) = E[y\,|x]$, assumes for a specific query point x_q, using the information pertaining only to a neighborhood of x.

For example, for a polynomial of degree 1, given a query point x_q, the parameters β_1 of a local first-degree polynomial approximating $f(.)$ in the neighborhood of the query point can be obtained by solving the following local polynomial regression:

$$\sum_{i=1}^{n} \{(y_i - x_{1,i}^{'}\beta_1)^2 K(\frac{D(x_i, x_q)}{h})\}, \tag{2}$$

where given a metric on the space R^m, $D(x_i, x_q)$ is the distance from the query point to the ith sample, $K(.)$ a weight function, and $h$the bandwidth. The vectors $x_{1,i}$can be obtained by preappending a constant value of 1 to each vector as a constant in the regression. The matrix from the above solution for the stated weighted least square problem is expressed by Eq. (3):

$$\widehat{\beta}_1 = (X_1^{'} W^{'} W X_1)^{-1} X_1^{'} W^{'} W y = (Z^T Z)^{-1} Z^T v = V Z^T v, \tag{3}$$

where X is a matrix in which the ith row is $x(i)^T$; y is a vector in which the ith element is $y(i)$; W is a diagonal matrix in which the ith diagonal element is $W_{ii} = \sqrt{K(D(x_i, x_q)/h)}$, $Z = WX$, $v = Wy$; and the matrix $X^T W^T W X = Z^T Z$ is assumed to be nonsingular, so that $V = (Z^T Z)^{-1}$ can be defined.

Once obtained, the local first-degree polynomial approximation, a prediction of $y_q = f(x_q)$, is finally given by:

$$\widehat{y}_{1,q} = x_{1,q}^{'} \widehat{\beta}_1. \tag{4}$$

This process is summarized in Figure 1, where the RLSE denotes the relative least square error.

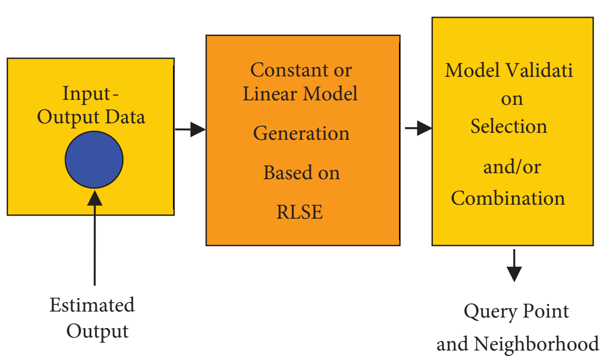

Figure 1. The LL process.

2.2. Local mode generation

The principle aim of the model is to generate a set of suitable candidate model structures for our problem. Traditionally, there have been a number of popular ways to search for the best model from a large collection of model structures. Maron and Moore [31] distinguished between 2 main categories of the model generation procedure: 1) brute force methods, which require a heavy computational effort to perform an exhaustive search in the space of model structures, and 2) search methods, which generate a number of possible candidates within a space defined with respect to some structural parameter, e.g., the number of neurons in a feedforward NN or the number of basis functions in a block-feedback NN.

The problem of local structural identification can be considered as a problem of bandwidth selection. Subsequently, the problem of bandwidth selection can be described as a search problem in the space of $B(k)$, so that the number of neighbors lies between k_m and k_M, where the latter is the minimum and the former is the maximum neighborhood area.

By specifying the range of quantity k over the interval $[k_m, k_M]$, the local model generation process returns a set of local models where the parameters of the model are fit in the set of neighbors within the bandwidth $B(k)$.

Both the constant and linear model generation are applied to the situations where parametric identification is conducted using Eq. (5):

$$\widehat{y}_q = \frac{\sum_{i=1}^{N} K(x_i, q, B) y_i}{\sum_{i=1}^{N} K(x_i, q, B)}, \tag{5}$$

where N is the number of samples in the training set. Figures 2a and 2b show the constant and linear fitting models, respectively. In the constant model, constant parameters are used for modeling, while in the linear model, a linear fitting is employed.

2.3. Recursive least squares for model generation

The adoption of recursive least square (RLS) algorithms for model identification and adaptive control systems was explained in [32]. The LL procedure does not include a temporal sequence. In order to provide a spatial sequence in linear models, RLS is deployed when observations are made at specific time intervals rather than in batches. The query neighbors are sorted according to the distance $d(x_i, q)$.

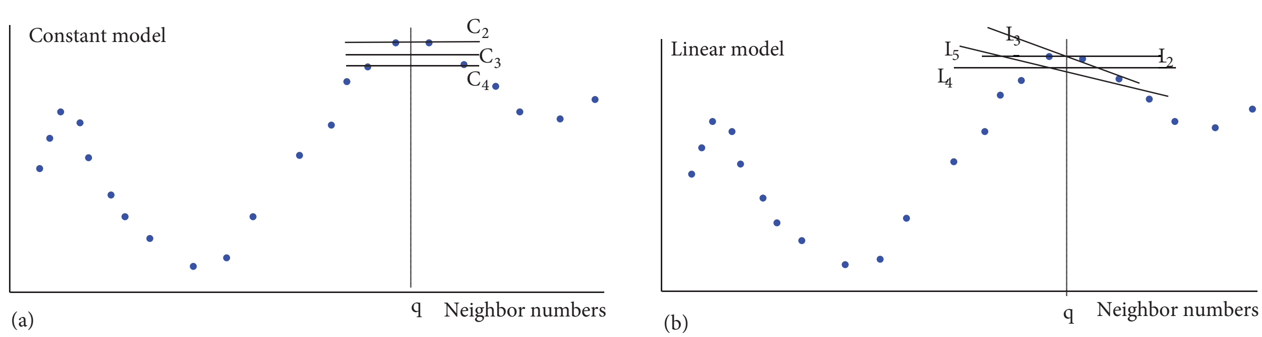

Figure 2. The constant (a) and linear (b) model's generations for different numbers of neighbors.

By using the sorted neighbors, a standard RLS can be used to obtain the parameters of the model that are fitted on the *k+1* nearest neighbors by updating the parameters of the model with k samples. Note that the RLS is not used here to update a model from time t to time *t+1*. With this concept, one can describe this method as RLSs in space, as opposed to the more traditional RLSs in time. The main assumption to be made for using a recursive approach in the local model generation is the adoption of a uniform weight kernel function, expressed as:

$$K(x_i, q, B) = \begin{cases} 1 & d(x_i, q) \leq B \\ 0 & \text{others} \end{cases} . \tag{6}$$

The main advantage for adopting the weighting kernel function is using a direct method to obtain the parameters for the model with the *k+1* as the nearest neighbors by simply updating the parameters of the estimated model using the k nearest neighbors. For the constant LL (CLL) model, the RLS can be defined as:

$$\widehat{y}_q^c(k+1) = \frac{k\,\widehat{y}_q^c(k) + y(k+1)}{k+1}. \tag{7}$$

Considering the linear LL (LLL) model, let $\widehat{\beta}(k)$ be the least square vector of parameters $\widehat{\beta}$ identified with k neighbors in Eq. (3). The RLS algorithm allows for an efficient way to identify the vector $\widehat{\beta}(k+1)$, using the *k+1* nearest neighbors. This is achieved on the basis that the vector $\widehat{\beta}(k)$ is estimated by the k nearest neighbors.

By implementing one step of the standard RLS algorithm, one can obtain the following:

$$\begin{cases} V(k+1) = V(k) - \frac{V(k)x(k+1)x^T(k+1)V(k)}{1+x^T(k+1)V(k)x(k+1)}, \\ \gamma(k+1) = V(k+1)x(k+1), \\ e(k+1) = y(k+1) - x^T(k+1)\widehat{\beta}(k), \\ \widehat{\beta}(k+1) = \widehat{\beta}(k) + \gamma(k+1)e(k+1), \\ \widehat{y}_q^l(k+1) = q^T\widehat{\beta}(k+1), \end{cases} \tag{8}$$

where $V(k) = (Z^T Z)^{-1}$, $x(k+1)$ is the *(k+1)th* nearest neighbors of the query point, and $\widehat{y}_q^l(k)$ represents the predicted query point returned by a linear model estimated on the basis of the k nearest neighbors.

3. Important factors in electric load forecasting

The electric load of a system may be influenced by a number of factors that can be categorized into 4 categories [32,33]:

1. Long-term socioeconomic factors (i.e. growth-contraction, price fluctuation, and demographics),
2. Time factors (i.e. demand patterns due to seasonal, peak, or off-peak weekly consumptions),
3. Weather and climate factors (i.e. weather, humidity, or environmental and climate changes), and
4. Other random events (mean time between failures, down times, and sudden and unplanned changes in consumption patterns).

Obtaining all of the data is important in achieving an accurate load forecasting.

4. Application of the LL method to short-term load forecasting

As previously mentioned, electrical short-term load forecasting plays a decisive role in the planning, cost-saving, and improvement of the security operation condition of a power system. LL, as introduced in Section 2, has been shown to be viable for nonlinear time series prediction and, in particular, for electrical short-term load forecasting. The strengths of the LL approach in load forecasting are its predictive accuracy, its fast design, the easy model update procedure, and the readability of the model structure.

In this study, the CLL, LLL, and mixed-model LL (MLL) methods, the last of which is a combination of the CLL and LLL methods, for short-term electrical load forecasting are discussed. In the proposed method, at least one important factor in each category of load forecasting mentioned in Section 3 is considered. The obtained data are processed and analyzed statistically. The adopted factors are price fluctuation; seasonal, week-day, and week-end demand patterns; and the temperature profile of the working environment. The correlation factors between these factors and the electrical load are then integrated into a weight matrix in the LL forecasting models.

In the following subsection, first the CLL and LLL methods are described, and then the MLL method for short-term load is employed for electrical load forecasting.

4.1. imulation and discussions of the proposed CLL, LLL, and MLL methods

To evaluate the proposed method, the hourly electrical load data extracted from the PJM system are used. Each obtained data point can be considered as a point in the multidimensional R^m space. The electric power load data at each hour is denoted as y (output), as shown in Figure 3.

The available data for a 4-year period are used as the training set. The information about the following 2 years is used for model validation purposes. The data and their associated weights obtained by the correlation analysis algorithm described previously are fed into the CLL, LLL, and MLL models. Figures 4–7 show sample results of the proposed methods and provide a comparison of the forecast results of the 3 models.

In all of the figures, the red dotted line shows the actual load, the gray line shows the prediction results obtained by the CLL, the blue line plots the LLL results, and the black solid line shows the load forecasting result for the MLL model, where the demand values are divided by 10,000 MW. Table 1 demonstrates the root mean square error (RMSE) for the resultant load forecast values of the 3 models. Table 1 reveals that the MLL model provides a more accurate forecast of the electric power load compared with the CLL and the LLL models. However, the accuracy of the MLL forecast model depends on the best levels-of-mix of the constant and linear methods. The proposed MLL model is selected in this paper as the superior LL technique to compete with the state-of-the-art RBF and MLP NN techniques for electrical power load forecasting.

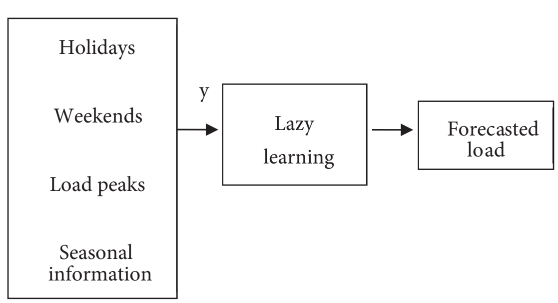

Figure 3. Load forecasting scheme via LL.

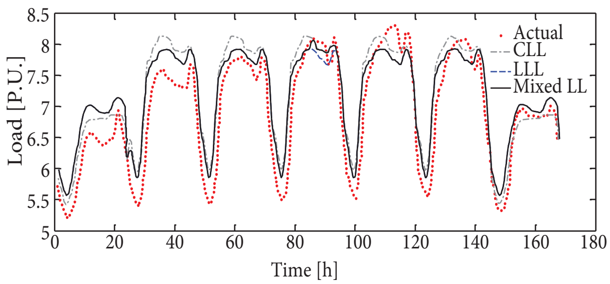

Figure 4. Electrical load forecast for the first week of autumn 2005.

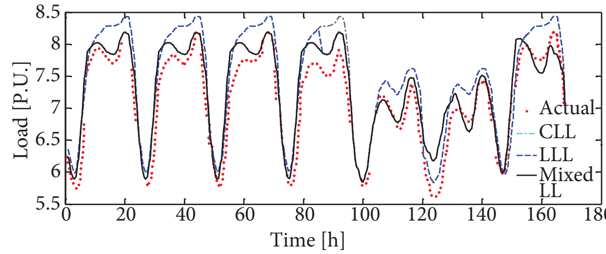

Figure 5. Electrical load forecast for the first week of summer 2005.

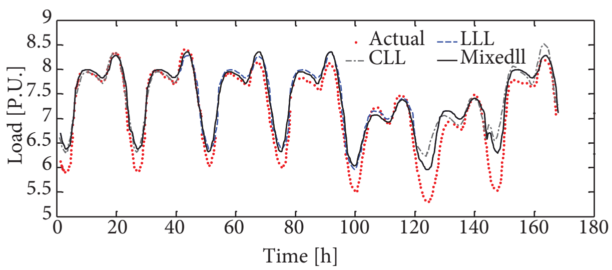

Figure 6. Electrical load forecast for the first week of spring 2006.

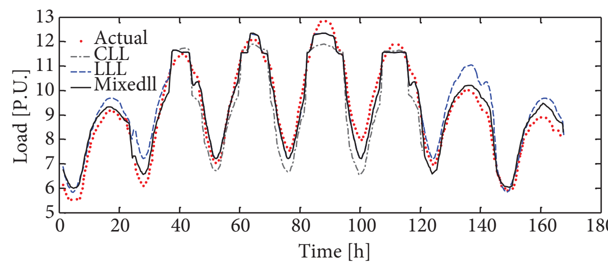

Figure 7. Electrical load forecast for the first week of winter 2007.

Table 1. RMSE for different load forecasts.

Method/week	Autumn 2005	Summer 2005	Spring 2006	Winter 2007
CLL	3.752%	3.638%	3.507%	6.407%
LLL	3.420%	3.036%	3.205%	5.673%
MLL	3.402%	2.983%	2.341%	4.015%

5. Comparison of the MLL and NN-based approaches

In this section, the simulation results from the proposed MLL are compared with those of the RBF and MLP NN models. NNs have been widely applied for the purpose of electrical load forecasting [5–7,34–36]. RBF NNs adopt RBFs as activation functions [37,38]. RBF NNs have been adopted for function approximation [39], time-series prediction [36], and control systems.

In addition to the actual values (observed) of the electrical power load, Figure 8 displays the electrical power load forecast by both the RBF NN technique and the proposed MLL for the first day of June 2007. The RMSEs against the actual values of the load are 7.397% for RBF and 2.182% for MLL at the same computational rate (see Table 2). The results of the simulation show a 5.215% improvement in the accuracy of the electrical power load forecasting by the proposed MLL. Comparing the MLP NN [40,41] with the MLL, the RMSE is also reduced by 5.217%, showing a good improvement in the accuracy of the forecasting technique (see Figure 9 and Table 2). Moreover, the computation time for the proposed MLL technique is 5 times faster than the existing MLP NN technique.

The results of the simulation run, RMSE values, and simulation run times for the 5 tested techniques are reported in Table 2. The simulation time is measured based on a 3.3-GHz Pentium 5 processor.

Figure 10 shows the differences between the RMSE of the forecast values against the actual electric power load of the 5 discussed techniques for a typical week in April 2007. The results are proof of the superiority of the proposed MLL method against the other 4 methods, i.e. LLL, CLL, RBF NN, and MP NN.

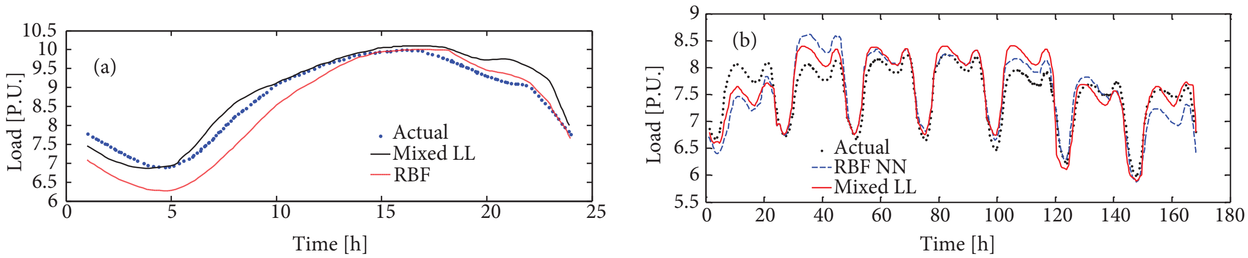

Figure 8. Comparison of the proposed MLL method and the RBF NN for a) the first day of June and b) the third week of April 2007.

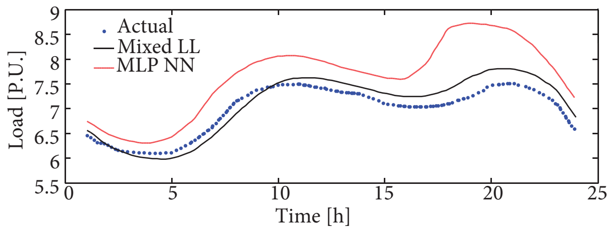

Figure 9. Comparison of the proposed method and the MLP NN (first day of January 2007).

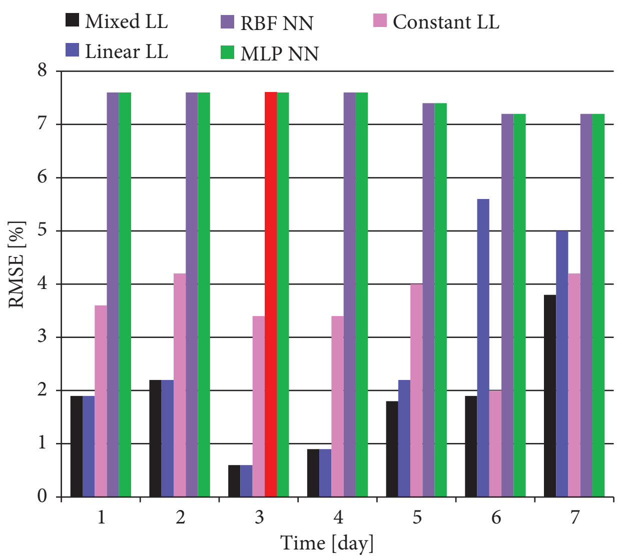

Figure 10. Comparison of the RMSE values of the proposed methods, RBF, and MLP (third week of April 2007).

6. Conclusion

A novel method based on the LL algorithm is proposed in this paper to forecast short-term electrical power loads. Each hourly load is assigned to a point where the state space is determined by the influencing factors of the daily demand on electricity. A weight matrix is derived based on the correlations between these factors and the load demand (inputs). The inputs are then fed into the lazy predictor using the derived weighting matrix. The proposed LL models are implemented and compared with the prevalent RBF NN and MLP NN load forecasting models.

Table 2. Detailed values of different load forecasting methods on the first day of January 2007.

	Forecasted load (MW)					Actual load
Hour	CLL	LLL	MLL	RBF	MLP	(MW)
1	70,658	67,027	65,699	67,560	67,560	64,037
2	69,192	65,112	62,659	64,871	64,871	62,040
3	68,225	63,849	60,583	63,392	63,392	61,196
4	67,831	63,198	59,906	63,004	63,004	60,973
5	68,402	63,930	59,774	64,115	64,115	61,098
6	69,955	65,447	61,373	67,613	67,613	63,343
7	72,331	67,728	64,538	73,300	73,300	66,548
8	74,971	70,275	68,780	76,990	76,990	71,411
9	77,909	72,670	72,580	79,510	79,510	73,733
10	79,054	74,503	75,358	80,549	80,549	74,758
11	79,540	75,754	76,149	80,651	80,651	74,918
12	79,181	76,053	76,042	79,641	79,641	74,213
13	78,396	75,657	75,288	78,321	78,321	73,329
14	77,375	74,991	74,167	77,251	77,251	72,701
15	76,575	74,440	73,032	76,185	76,185	71,160
16	76,235	74,288	72,381	75,714	75,714	70,385
17	76,261	74,641	72,481	79,277	79,277	70,362
18	77,514	75,246	73,794	86,316	86,316	71,082
19	79,873	75,832	76,296	87,161	87,161	72,316
20	81,597	76,568	77,864	86,756	86,756	74,692
21	81,849	77,108	78,139	85,794	85,794	75,091
22	80,968	76,821	76,925	82,623	82,623	73,845
23	80,023	76,055	74,077	77,735	77,735	71,383
24	71,547	69,864	68,175	71,667	71,667	65,126
RMSE	6.194%	2.858%	2.182%	7.397%	7.399%	—
Simulation time (s)	12.5	12.5	12.5	13	120	—

The simulation results show a significant improvement in both reduced RMSE and computational speed. An overall 5% improvement in the accuracy of load forecasting using the MLL technique compared with the RBF and MLP NN techniques has been achieved. A further improvement is observed in the computation rate of the proposed MLL technique. It is 5 times faster than the MLP NN technique.

References

[1] Box GEP, Jenkins GM. Time Series Analysis-Forecasting and Control. San Francisco, CA, USA: Holden-Day, 1970.

[2] Kandil MS, El-Debeiky SM, Hasanien NE. Long-term load forecasting for fast developing utility using a knowledge-based expert system. IEEE T Power Syst 2002; 17: 491–496.

[3] Xiao Z, Zhong SYB, Sun C. BP neural networks with rough set for short term load forecasting. Expert Syst Appl 2009; 36: 273–279.

[4] Xia C, Wang J, Menery KM. Short, medium and long term load forecasting model and virtual load forecaster based on redial bases function neural network. Int J Elec Power 2010; 32: 743–750.

[5] Ho KL, Hsu YY, Yang CC. Short-term load forecasting using a multilayer neural network with an adaptive learning algorithm. IEEE T Power Syst 1992; 7: 141–149.

[6] Chen ST, Yu DC, Moghaddamjo AR. Weather sensitive short-term load forecasting using non-fully connected artificial neural network. IEEE T Power Syst 1992; 7: 1098–1105.

[7] Hippert HS, Pedreira CE, Souza RC. Neural networks for short-term load forecasting: a review and evaluation. IEEE T Power Syst 2001; 16: 44–55.

[8] Al-Kamdari AM, Soliman SA, Al-Hawary ME. Fuzzy short-term electrical load forecasting. Int J Elec Power 2004; 26: 111–122.

[9] Pai PF. Hybrid ellipsoidal fuzzy systems in forecasting regional electricity loads. Energ Convers Manage 2006; 47: 2283–2289.

[10] Chuangxin BY, Cao GY. Short-term load forecasting using a new fuzzy modeling strategy. In: 5th World Congress on Intelligent Control and Automation Conference; June 2004; Hangzhou, China. pp. 5045–5049.

[11] Gao RG, Soukalas LHT. Neural-wavelet methodology for load forecasting. J Intell Robot Syst 2001; 10: 149–157.

[12] Bashir Z, El-Hawary ME. Short-term load forecasting by using wavelet neural networks. In: Canadian Electrical and Computer Engineering Conference; March 2000; Halifax, Canada. pp. 163–166.

[13] Tao D, Xiuli W, Xifan W. A combined model of wavelet and neural network for short term load forecasting. In: International Conference on Power System Technology; October 2002. pp. 2331–2335.

[14] Liang RH, Cheng CC. Short-term load forecasting by a neuro-fuzzy approach. Int J Elec Power 2002; 24: 103–111.

[15] Xiao MW, Min BX, Shun ML. Short-term load forecasting with artificial neural network and fuzzy logic. In: International Conference on Power System Technology; October 2002. pp. 1101–1104.

[16] Ansarimehr P, Barghinia S, Habibi H, Vafadar N. Short-term load forecasting for Iran National Power System using artificial neural network and fuzzy expert system. In: International Conference on Power System Technology; October 2002. pp. 1082–1085.

[17] Barzamini R, Menhaj MB, Khosravi A, Kamalvand SH. Short-term load forecasting for Iran National Power System and its regions using multilayer perceptron and fuzzy inference systems. In: IEEE International Joint Conference on Neural Networks; August 2005; Montreal, Canada. pp. 2619–2624.

[18] Birattari M, Bontempi G, Bersini H. Lazy learning: a local method for supervised learning. In: Jain LC, Kacprzyk J, editors. New Learning Paradigms in Soft Computing. Heidelberg, Germany: Physica-Verlag GmbH, 1999. pp. 97–136.

[19] Aha DW. Lazy Learning. Dordrecht, the Netherlands: Kluwer Academic Publishers, 1997.

[20] Bertolissi E. Data-driven techniques for direct adaptive control: the lazy and the fuzzy approaches. Fuzzy Sets Syst 2001; 3: 3–14.

[21] Bontempi GL. The local paradigm for modeling and control: from neuro-fuzzy to lazy learning. Fuzzy Set Syst 2001; 4: 59–71.

[22] Bontempi GL. Lazy learning for control design. In: European Symposium on Artificial Neural Networks; April 1998; Bruges, Belgium. pp. 73–78.

[23] Corani G. Air quality prediction in Milan: feed-forward neural networks, pruned neural networks and lazy learning. Ecol Model 2005; 185: 513–529.

[24] Sorjamaa A. Pruned Lazy Learning Models for Time Series Prediction. Helsinki, Finland: Helsinki University of Technology, 2007.

[25] Aha DW, editor. Special issue on lazy learning. Artif Intell Rev 1997; 11: 1–6.

[26] Riverola FF, Iglesias EL, Díaz F, Méndez JR, Corchado JM. Spam hunting: an instance-based reasoning system for spam labelling and filtering. Decis Support Syst 2006; 43: 722–736.

[27] Bontempi G, Birattari M, Bersini H. Local learning for iterated time-series prediction. In: International Conference on Machine Learning; 1999; San Francisco, CA, USA. pp. 32–38.

[28] Capacity Adequacy Planning Department. PJM LOAD/ENERGY forecasting model. White paper. Valley Forge, PA, USA: PJM Interconnection, 2007.

[29] Birattari M, Bontempi GL. Lazy Learning. Brussels, Belgium: Free University of Brussels, 1999.

[30] Birattari M, Bontempi G. Lazy learning vs. speedy: a fast algorithm for recursive identification and recursive validation of local constant models. Technical report TR/IRIDIA/99-6. Brussels, Belgium: IRIDIA-ULB, 1999.

[31] Maron O, Moore A. The racing algorithm: model selection for lazy learners. Artif Intell Rev 2006; 11: 193–225.

[32] Goodwin GC, Sin KS. Adaptive Filtering Prediction and Control. Upper Saddle River, NJ, USA: Prentice Hall, 1984.

[33] Ho KL, Hsu YY, Chen CF, Lee TE, Liang CC, Lai TS, Chen KK. Short term load forecasting of Taiwan power system using a knowledge-based expert system. IEEE T Power Syst 1990; 5: 1214–1221.

[34] Goia A, May C, Fusai G. Functional clustering and linear regression for peak load forecasting. Int J Forecasting 2010; 26: 700–711.

[35] Kariniotakis GN. Load forecasting using dynamic high-order neural networks. In: Proceedings of the IEEE 2nd International Forum on the Applications of Neural Networks to Power Systems; April 1993; Yokohoma, Japan. pp. 801–805.

[36] Hippert HS, Tailor JW. An evaluation of Bayesian techniques for controlling model complexity and selecting inputs in a neural network for short-term load forecasting. Neural Networks 2010; 23: 386–395.

[37] Buhmann MD, Ablowitz MJ. Radial Basis Functions Theory and Implementations. Cambridge, UK: Cambridge University, 2003.

[38] Chen S, Cowan CFN, Grant PM. Orthogonal least squares learning algorithm for radial basis function networks. IEEE T Neural Networ 1991; 2: 224–230.

[39] Poggio T, Girosi F. Networks for approximation and learning. Proc IEEE 1990; 78: 1484–1487.

[40] Jones RD, Lee YC, Barnes CW, Flake GW. Function approximation and time series prediction with neural networks. In: International Joint Conference on Neural Networks; June 1990; San Diego, CA, USA. pp. 649–665.

[41] Haykin S. Neural Networks: A Comprehensive Foundation. Upper Saddle River, NJ, USA: Prentice Hall, 1998.