Energy Conversion and Management 51 (2010) 2707–2714

Contents lists available at ScienceDirect

Energy Conversion and Management

journal homepage: www.elsevier.com/locate/enconman

Electricity price forecasting using Enhanced Probability Neural Network

Whei-Min Lin [a], Hong-Jey Gow [a], Ming-Tang Tsai [b,*]

[a] Department of Electrical Engineering, National Sun Yat-Sen University, Kaohsiung, Taiwan, ROC
[b] Department of Electrical Engineering, Cheng-Shiu University, Kaohsiung, Taiwan, ROC

ARTICLE INFO

Article history:
Received 20 September 2009
Accepted 1 June 2010
Available online 29 June 2010

Keywords:
Orthogonal Experimental Design (OED)
Locational Marginal Price
Probability Neural Network
Electricity price forecasting

ABSTRACT

This paper proposes a price forecasting system for electric market participants to reduce the risk of price volatility. Combining the Probability Neural Network (PNN) and Orthogonal Experimental Design (OED), an Enhanced Probability Neural Network (EPNN) is proposed in the solving process. In this paper, the Locational Marginal Price (LMP), system load and temperature of PJM system were collected and the data clusters were embedded in the Excel Database according to the year, season, workday, and weekend. With the OED to smooth parameters in the EPNN, the forecasting error can be improved during the training process to promote the accuracy and reliability where even the "spikes" can be tracked closely. Simulation results show the effectiveness of the proposed EPNN to provide quality information in a price volatile environment.

1. Introduction

Deregulating the power market creates competition and a trading mechanism for market players. It moves from the cost-based operation to a bid-based operation [1,2]. "Electricity" has become a commodity that the price could be volatile in the energy market where the sudden "spikes" could even appear. To forecast price accurately is an important task for producers, consumers, and retailers. According to price forecast, participants can develop their bidding strategies to maximize the profit with lower risks. Price forecasting also helps investors to better planning the Grids.

In the US, Pennsylvania–New Jersey–Maryland (PJM) power market [3] is commonly known as one of the successful market models. PJM market that operates the competitive market is coordinated by an Independent System Operator (ISO) to determine the Locational Marginal Price (LMP) according to the status of system nodes. The LMP at each node will then reflect not only the price of voluntary bids but also the overhead of delivering energy to locations. Generally, LMP includes three components: an energy cost component, a transmission congestion component, and the marginal loss component. LMP in a pool is different for different locations while the energy cost can be identical for all the nodes. A feasible and practical method for LMP forecasting will funnel a better risk management for all market participants.

Many factors including load, historical prices and temperature could impact the LMP. The loads and prices in the wholesale market are mutually intertwined activities. Loads are heavily affected by the weather parameter, so the prices are strongly volatile with the changing weather. Another factor is the time of use at various levels of day, week, month, season, and year. Price could rise hundred of times the normal value to reflect the volatility. In PJM, LMP is introduced at nodes. Congestion occurs when a transmission flow exceeds its limits. Line flow information becomes an important factor in price forecasting. It is complicated to perform LMP forecasting, especially finding the best strategy in a world of uncertainties. Reported techniques to forecast day-ahead prices include time series models [4], weighted nearest neighbors techniques [5], auto regressive integrated moving average models (ARIMA) [6], Mixed ARIMA models [7,8], and Markov models [9]. These approaches can be very accurate with sufficient information and computation time, however, there is no approach with satisfactory performance in dealing with the spikes. Recently, Artificial Neural Network (ANN) has been applied to forecast prices in various markets [10–16]. ANN is a simple, powerful, and flexible tool for forecasting, providing better solutions to model complex nonlinear relationships than the traditional linear models. ANNs have weaknesses in the determination of network architecture and network parameters. Running in a dynamic environment, especially for online applications, traditional ANN network can become the bottleneck in adaptive applications [17]. Among ANNs, the Probability Neural Network (PNN) can function as a classifier, and it has the advantages of a fast learning process, requiring only a single-pass network training stage without any iteration for adjusting weights, and it can adapt itself to architectural changes. However, PNN can only classify the data for categorical input data type. Since, there is no mechanism for adjusting parameters in the course of recalling. To deal with the continuity problem, a large

* Corresponding author.
E-mail address: tsaymt@post.csu.edu.tw (M.-T. Tsai).

0196-8904/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.enconman.2010.06.006

amount of information needs to be obtained during the learning stage. In order to solve this problem, this paper uses Orthogonal Experimental Design (OED) to improve the traditional PNN.

OED is an effective tool for robust design and an engineering methodology for optimizing process conditions which are minimally sensitive to various causes of variations. The characteristics of OED are: (1) results obtained through few experiments; (2) good recurrence of result in the same experimental environment; (3) simple construction of mathematical model with the application of orthogonal array; (4) simple analytical procedure. Combining PNN [18] and OED [19], an Enhanced PNN (EPNN) is proposed in this paper. OED is used to adjust the smoothing parameters in EPNN learning stage to improve the training ability, and a good performance with a close spike tracking capability can be seen. This paper developed day-ahead forecasts for electricity price using EPNN, based on similar-day PJM market model. Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), and Root Mean Square Error (RMSE) obtained from the forecasting results demonstrate that EPNN can efficiently forecast the price for any day of a week.

2. Orthogonal Experimental Design

EPNN contains the learning stage and the recalling stage. Although the structure of traditional PNN is simple, it may cause errors in non-classification problems because it has no mechanism for adjusting the concatenated key parameters. To improve this process, this paper uses OED to find the optimal smoothing parameter σ_k to increase the accuracy of prediction.

Orthogonal experiment uses a small amount of experimental data to construct a mathematical model, which is derived from orthogonal experiment to calculate experiment values. OED contains two main parts: (1) orthogonal array and (2) factor analysis. In engineering experiments, the cause that influences the result is the factor of orthogonal array; while various states of the factor are called the "level number". Factor analysis is carried out after the orthogonal array set-up, and the "factor effects" of each factor can be obtained from the analytical result. The impact of each factor on the experiment is deduced from those effects.

An orthogonal array with F factors and Q levels can be described as $L_M(Q^F)$, where "L" denotes a Latin square, M is the chosen number of combinations of levels. The notion of using orthogonal arrays has been associated with Latin Square from the outset. We let $L_M(Q^F) = [e_{ij}]_{MxF}$ where the jth factor in the ith combination has level value e_{ij} and $e_{ij} \in \{1, 2, \ldots, Q\}$. An example of an orthogonal array can be seen as

$$L_4(2^3) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 2 & 1 & 2 \\ 2 & 2 & 1 \end{bmatrix} \quad (1)$$

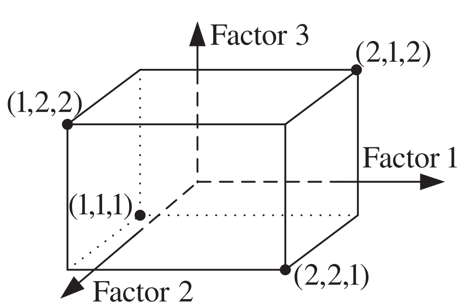

Fig. 1. Illustration of the orthogonal array $L_4(2^3)$.

Fig. 1 shows a three-factor solution space of the example. The four vertices of the cube cover presented in Eq. (1) all facets of the problem and are representative enough to calculate the optimal solution.

Consider an experiment containing two factors x_1 and x_2, assuming each factor has two levels. The two levels of x_1 are 2.4 and 3.6, two levels of x_2 are 60 and 80, and the experimental output data y is obtained from the experiment. A sample of the experimental data is shown in Table 1.

Let Eq. (2) be defined as a transformation of variable for any factor with two levels, where x is the original variable, x_U is the high level and x_L is low level. Then x_N will be a new variable with values +1 or −1, That is

$$x_N = \frac{x - \frac{(x_U + x_L)}{2}}{\frac{(x_U - x_L)}{2}} \quad (2)$$

For convenience of description, a transformation of variable is made for x_1 and x_2 by

$$A = \frac{x_1 - \frac{3.6+2.4}{2}}{\frac{3.6-2.4}{2}} = \frac{x_1 - 3.0}{0.6} \quad (3)$$

$$B = \frac{x_2 - \frac{80+60}{2}}{\frac{80-60}{2}} = \frac{x_2 - 70}{5} \quad (4)$$

A and B are used for Table 1, with another $A \times B$ added as shown in Table 2.

The response value $R_{f,l}$ is calculated from Eq. (5), and the effect of each factor E_f is calculated through Eq. (6). We have

$$R_{f,l} = \text{Mean of } y_i \text{ for factor } f \text{ at level } l \quad (5)$$

where $f = A, B$, and $l = 1, 2$, and the effect

$$E_f = \frac{R_{f,Level\ 2} - R_{f,\text{Level }1}}{2} \quad (6)$$

The related data are listed in Table 3.

With the above arrangement, the relationship between function value y_i and other factors $A, B, A \times B, R_{f,l}$ and E_f can be described by

$$y_i = \text{Average} + \text{Effect}_A \times A + \text{Effect}_B \times B + \text{Effect}_{A\times B} \times A \times B \quad (7)$$

A numerical example of Experiment 1 can be seen by

$$y_1 = 5.4 = 6 + 0.7 \times (-1) - 0.2 \times (-1) - 0.1 \times (-1) \times (-1) \quad (8)$$

Other experimental outputs can be obtained from Eq. (7) similarly. It shows that the function of orthogonal array can use a small amount of experimental data to construct a mathematic model like Eq. (7).

In Eq. (7), $A \times B$ is the "interaction term", if it is removed from Eq. (7), the numerical value of experiment will have an error of 0.1 as

$$y_1 = 6 + 0.7 \times (-1) - 0.2 \times (-1) = 5.5 \quad (9)$$

In OED, if the interaction term $A \times B$ is neglected, Eq. (7) is regarded as an output equation affected by only the *main effect A* and *B*, as shown in Eq. (10).

Table 1
Two factors x_1, x_2 experiments.

Experiment no.	Factors		Function value y_i
	x_1	x_2	
1	2.4	60	5.4
2	2.4	80	5.2
3	3.6	60	7.0
4	3.6	80	6.4

Table 2
Two factors A, B experiments.

Experiment no.	Factors			Function value y_i
	A	B	$A \times B$	
1	−1	−1	1	5.4
2	−1	1	−1	5.2
3	1	−1	−1	7.0
4	1	1	1	6.4

Where −1 is called " Level 1", and +1 is "Level 2".

Table 3
Two factors effect analysis.

Experiment no.		Factors			Function value y_i
		A	B	$A \times B$	
1		−1	−1	1	5.4
2		−1	1	−1	5.2
3		1	−1	−1	7.0
4		1	1	1	6.4
Response	Level 1 $R_{f,1}$	5.3	6.2	6.1	Average of y_i 6.0
	Level 2 $R_{f,2}$	6.7	5.8	5.9	
Effect E_f		0.7	−0.2	−0.1	

$$y_i = \text{Average} + \text{Effect}_A \times A + \text{Effect}_B \times B \tag{10}$$

From Table 3, the optimal level of factors can be obtained based on the response of each level. Take the minimum as an example, Level 1 response of Factor A is 5.3, which is less than the Level 2 response 6.7, so Factor A selects Level 1; Level 2 response of Factor B is 5.8, which is less than the Level 1 response 6.2, so Factor B selects Level 2. The final experimental output is the minimum value 5.2. The above process is called "factor analysis". From Eq. (7), we can also see that

1. OED can approximate the input–output relationship through a simple linear equation, when neglecting the interaction term.
2. We can predict the system output not shown in the orthogonal array based on Eq. (7).
3. We can use OED to search for the extreme values of the system. By selecting a proper factor-level in linear Eq. (7), preferable system output can be obtained.

3. Enhanced Probability Neural Network

EPNN consists of the input, hidden, summation, and output layers. The unknown input vector $X = [x_1, x_2, \ldots, x_i, \ldots, x_n]$, $i = 1, 2, \ldots, n$, is connected to the input layer. The number of hidden nodes H_k, $k = 1, 2, \ldots, K$, is equal to the number of learning data sets, while the number of summation nodes S_j and output nodes O_j, $j = 1, 2, \ldots, m$, are equal to the forecasting points. The weights w_{ki}^{IH} connecting the kth hidden node with the ith input node, and w_{jk}^{HS} connecting the jth summation node with the kth hidden node, are determined by the kth input–output training pairs.

The EPNN structure is shown in Fig. 2 with two stages of "Learning" and "Recalling". Besides, the "OED" process is embedded in the Recalling Stage.

3.1. Learning Stage

Step 1: For each learning data set $X(k) = [x_1(k), \ldots, x_i(k), \ldots, x_n(k)]$ $i = 1, 2, \ldots, n$, $k = 1, 2, \ldots, K$, weight w_{ki}^{IH} is determined by

$$w_{ki}^{IH} = x_i(k) \tag{11}$$

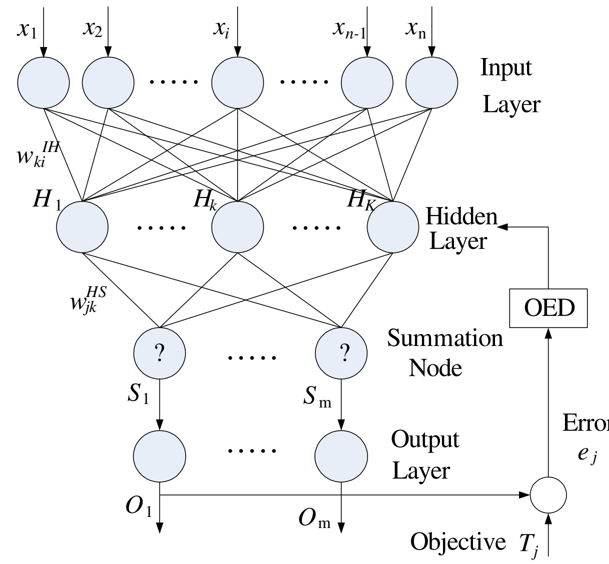

Fig. 2. The EPNN structure.

$$H_k = \exp\left[-\sum_{i=1}^{n} \frac{(x_i - w_{ki}^{IH})^2}{2\sigma_k^2}\right] \tag{12}$$

In this paper, $x_i(k)$ is the ith variable of the kth predicted output. All σ_k are predetermined parameters.

Step 2: Create weights $w_{jk}^{HS}, j = 1, 2, \ldots, m$ between the hidden node H_k and the summation node S_j, where the values of w_{jk}^{HS} is the predicted outputs LMP ($) associated with each stored pattern w_{ki}^{IH}. We also have

$$O_j = \frac{\sum_{k=1}^{K} w_{jk}^{HS} H_k}{\sum_{k=1}^{K} H_k} = \frac{S_j}{\sum_{k=1}^{K} H_k} \tag{13}$$

The optimization method is used to adjust parameter σ_k, and adjusting σ_k would refine the accuracy in the dynamic environment. The optimal σ_k can yield a minimum forecast error based on the testing data.

3.2. Recalling Stage

Step 1: Get network weights w_{ki}^{IH} and w_{jk}^{HS}.
Step 2: Get a recalling data $X = [x_1, x_2, \ldots, x_i, \ldots, x_n]$ from the training data set. Apply input vector X to the input layer. Compute the output of hidden node H_k by Eq. (12), where the parameters σ_k's are all assumed 1.0 initially.
Step 3: Compute output S_j of hidden nodes by Eq. (14), compute the output of node O_j by Eq. (13), compute the error e_j of output by Eq. (15), and then total the square errors by Eq. (16). We have

$$S_j = \sum_{k=1}^{K} w_{jk}^{HS} H_k \tag{14}$$

$$e_j = [T_j - O_j] \tag{15}$$

$$SE_j = \sum_{j=1}^{m} [T_j - O_j]^2 \tag{16}$$

Step 4: Detect Convergence. If the error Eq. (16) converges, end; otherwise go to Step 5.

Step 5: Perform the OED process. By using OED for σ_k, σ_k at each node H_k can be adjusted to minimize forecast error.

3.3. Orthogonal Experimental Design process

Step 1: All σ_k are regarded as factors of OED and each SE_j is regarded as a objective function value.
Step 2: Construct an orthogonal array $L_m(2^M)$,

where M is the number of control factors and m is the experiment number.

$$m = 2^{[\log_2(M+1)]} \tag{17}$$

Step 3: Define the level of σ_k by

$$\begin{cases} \sigma_{k,L1=}\sigma_k - \Delta\sigma \\ \sigma_{k,L2=}\sigma_k + \Delta\sigma \end{cases} \tag{18}$$

where $\Delta\sigma$: predetermined alteration of σ_k, $\sigma_{k,L1}$: level 1 of σ_k, $\sigma_{k,L2}$: level 2 of σ_k.

In this paper, there are K control factors σ_k, $k = 1, 2, \ldots, K$ to be selected to perform OED corresponding to nodes H_k. These σ_k's, corresponding to K control factors, are defined as in Table 4. In Table 4, $\Delta\sigma$ is defined as 0.01.

Step 4: Perform the factor response analysis and get the optimal σ_k.

4. The implementation of EPNN

Daily LMP profiles are similar on weekdays but different on Saturdays and Sundays. The fluctuation of LMP may come from the loads, transfer flow, and temperature changes. EPNN is capable of coping with complicated interactions between those factors and LMPs. Hence, those factors can be taken into account as inputs to the EPNN. In this paper, three-layer network based on EPNN are shown in Fig. 3.

The input layer contains three input variables – loads, temperature, and transfer flow. The hidden layer contains K hidden nodes. The output layer only contains one output variables which is the LMP. EPNN training is done to minimize the fitting error for a sample training set. For a given training data, the objective function is defined as in Eq. (16). After training, input the tth hour forecasting data in tth hour EPNN network to forecast LMP_t.

The price profile presents seasonal characteristics, usually day and week cycles. The training data are classified as weekdays from Monday to Fridays, and weekends of Saturday and Sunday. The selection of similar days for each season is trained in EPNN. For example, eight similar days are selected for training to predict the LMP on Monday or Saturday, etc. One day is taken from the similar days as test data. The sample data of PJM is constructed in EXCEL Workspace. The data analysis and data storage can be easily manipulated with this database. To evaluate the accuracy for EPNN in forecasting, the Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), and Root Mean Square Error (RMSE) are all used in this paper. The MAPE, MAE, and RMSE are defined as

Table 4
Level definition in $L_m(2^K)$ orthogonal array.

Levels	Factor						
	1 σ_1	2 σ_2	... ...	k σ_k	... ...	$K-1$ $\sigma_{K\text{-}1}$	K σ_K
Level 1	$\sigma_{1,L1}$	$\sigma_{2,L1}$	...	$\sigma_{k,L1}$	...	$\sigma_{K\text{-}1,L1}$	$\sigma_{K,L1}$
Level 2	$\sigma_{1,L2}$	$\sigma_{2,L2}$	...	$\sigma_{k,L2}$	...	$\sigma_{K\text{-}1,L2}$	$\sigma_{K,L2}$

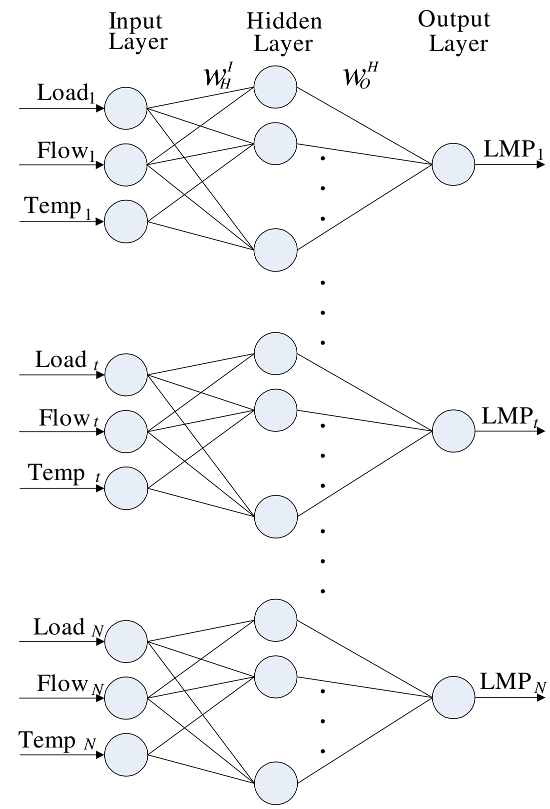

Fig. 3. The proposed tth EPNN network.

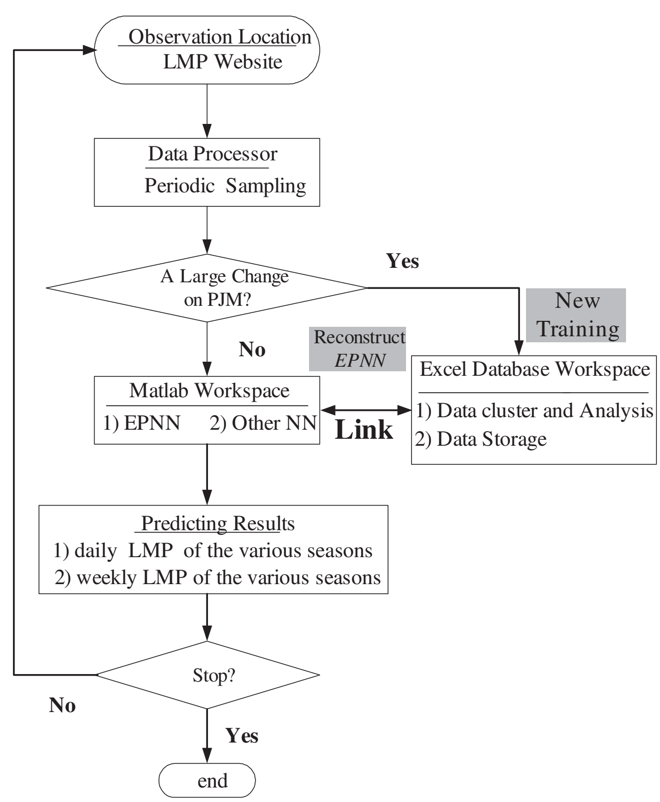

Fig. 4. The forecasting procedure of EPNN.

$$\mathrm{MAPE} = \frac{1}{N}\sum_{t=1}^{N}\frac{|P_t^{ture} - P_t^{predict}|}{P_t^{ture}} \times 100\% \tag{19}$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{t=1}^{N}|P_t^{ture} - P_t^{predict}| \tag{20}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}|P_t^{ture} - P_t^{predict}|^2} \tag{21}$$

where P_t^{ture} is the actual price at hour t and $P_t^{predict}$ is the forecasting price at hour t. N is the number of testing data. N is 24 and 168

Table 5
The data sets for all studies cases.

Case	Training data	Training data (date)	Test data (date)
1	Similar days of Friday in spring	2002.01.04 to 2002.02.22	2002.03.01
2	Similar days of Saturday in summer	2002.04.06 to 2002.05.26	2002.06.01
3	Similar days of Sunday in autumn	2002.07.07 to 2002.08.25	2002.09.01
4	Similar days of weekly price in summer	2002.04.01 (Monday) to 2002.05.26 (Sunday)	2002.06.10 (Monday) to 2002.06.16 (Sunday)
5	Similar days of weekly price in winter	2002.10.07 (Monday) to 2002.11.24 (Sunday)	2002.12.02 (Monday) to 2002.12.08 (Sunday)

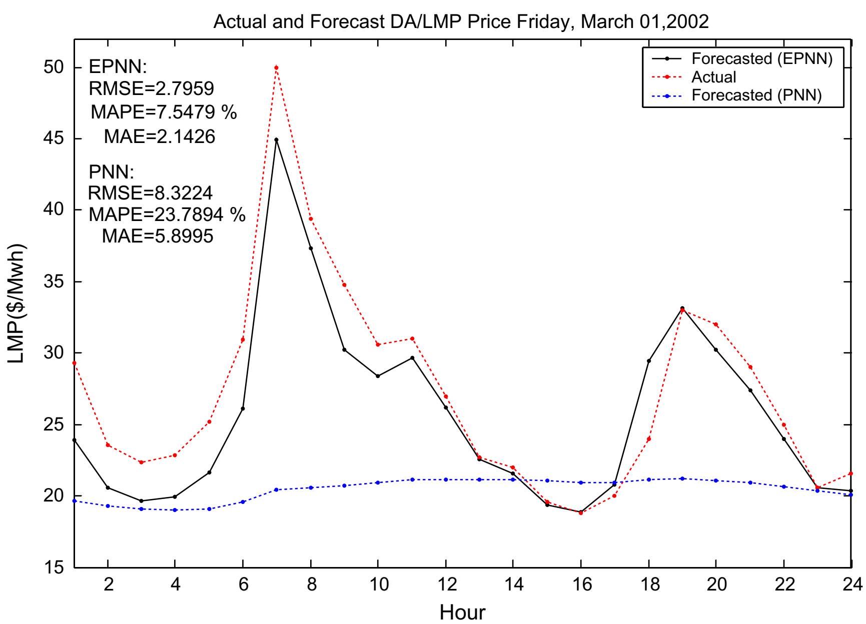

Fig. 5. Actual and forecast day-ahead PJM electricity price of case 1.

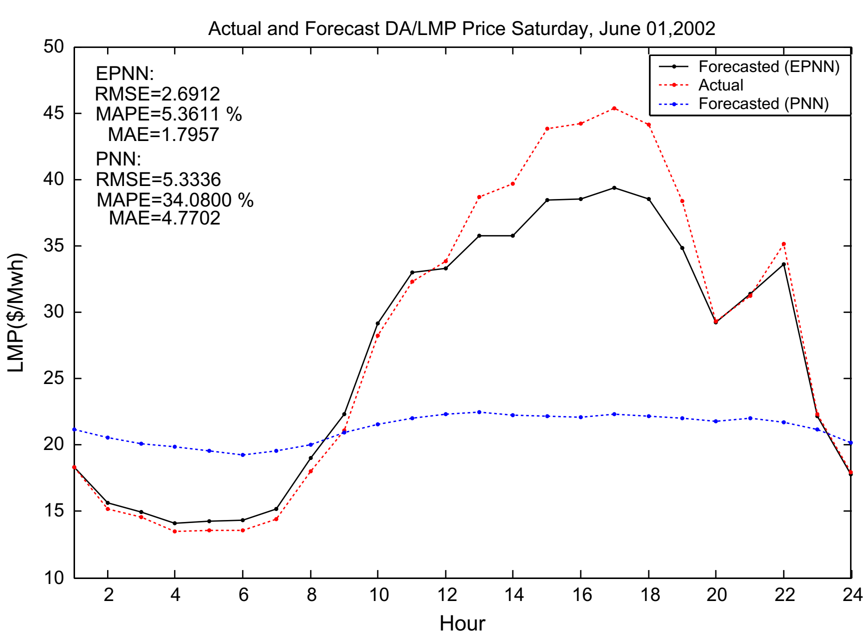

Fig. 6. Actual and forecast day-ahead PJM electricity price of case 2.

(hourly) for daily and weekly sample data. Fig. 4 is the forecasting procedure of EPNN.

5. Results

The data of PJM website [3] were used to train and test the proposed method. For comparison purpose, PNN and Back-progation Neural Network (BPN) were also built for tests. The data set was divided into two parts, the training data and testing data as in Table 5. The training data were used for training and updating the biases and weight. The test data were used to test the proposed methods after training. The simulation was implemented with Matlab on a PIV-2.6 GHz computer with 512 MB RAM

Fig. 5 shows the day-ahead price forecasting for case 1. The forecasting results are compared with the Actual LMP value. Forecasting results of PNN are also chosen to show the differences. The MAPE of EPNN and PNN are 7.55% and 23.79%, respectively. The forecasting results of EPNN can track very closely the actual LMP with OED applied. Note that the algorithm could even track the first spike nicely, which is not easily attainable by other algorithms, and a common neural network such as PNN will not have the capability.

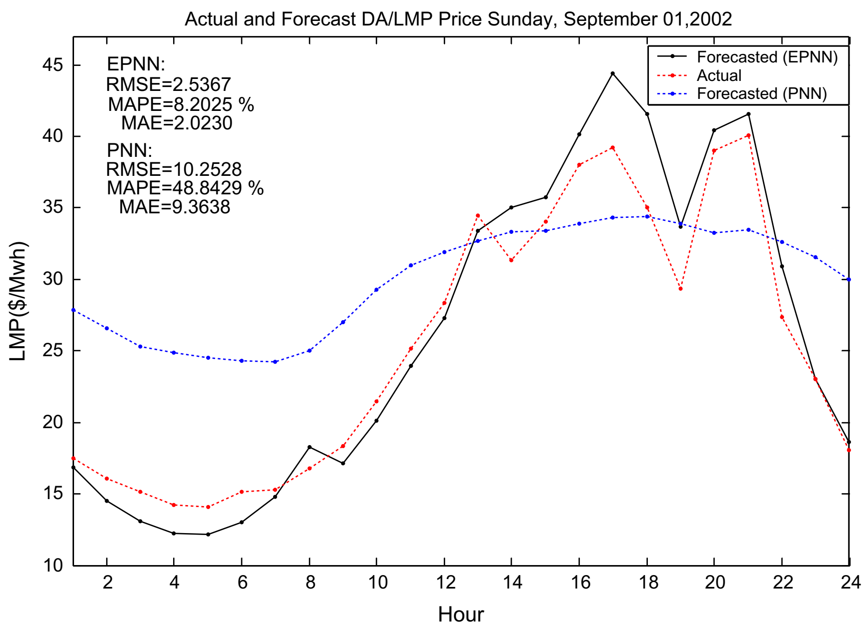

Fig. 7. Actual and forecast day-ahead PJM electricity price of case 3.

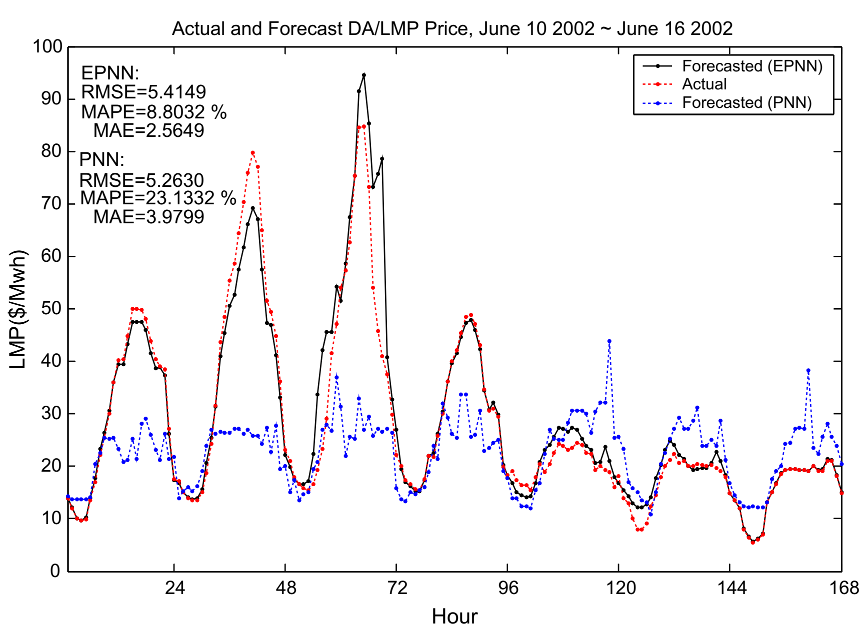

Fig. 8. Actual and forecast day-ahead PJM electricity price of case 4.

Similarly, day-ahead price forecasting for case 2 and case 3 are shown in Figs. 6 and 7, respectively, where MAPE values are about 9% for EPNN Method. It is obvious that EPNN has the ability to find better solutions.

Fig. 8 shows the weekly price forecast for case 4, which is a typical summer week. From Fig. 8, the forecasting price of EPNN is close to the actual LMP values. It can be seen that EPNN has the capability to follow the spikes as shown in the 24th to 72th (Tuesday to Wednesday) hour of this week. MAPE from the EPNN method is only 8.8%, much less than that of PNN method (23.13%). Similarly, Fig. 9 shows the weekly price forecasting results for case 5, which is a typical winter week. In this case, weekly MAPE value of EPNN is lower than that of case 4.

Fig. 10 shows the curves of daily RMSE in a summer. For the tests, the number of sets of training data increases from 48 to 286 (2–12 days) and execution time increases from 5.5 s to 32.828 s. The RMSE of all tests is well below 5% with 192 training data (8 days). It can be seen that it is accurate enough for EPNN with 192 training data sets. This characteristic can be utilized to greatly reduce the training time, and the data storage can be reduced without losing originalities. We can minimize the data storage, shorten the preprocessing needs, and reduce the network

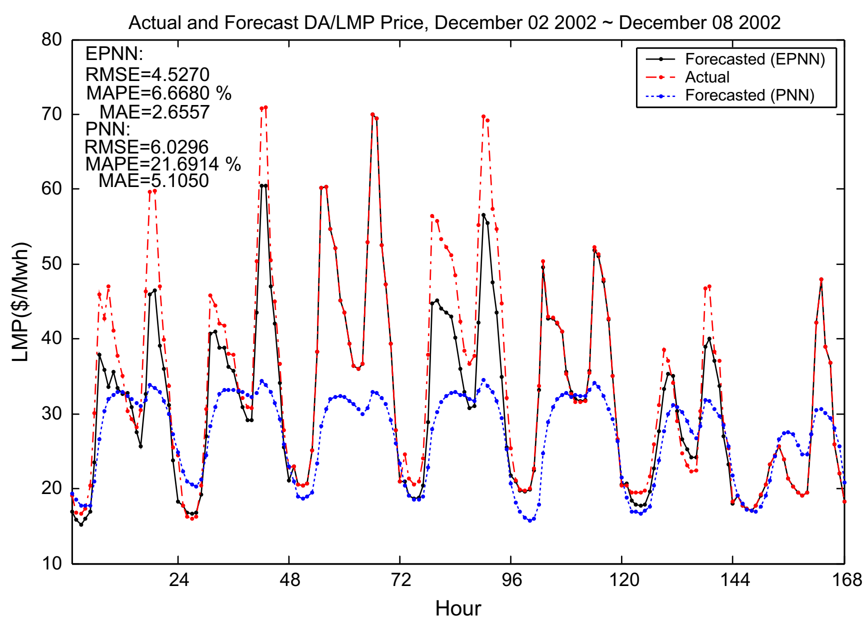

Fig. 9. Actual and forecast day-ahead PJM electricity price of case 5.

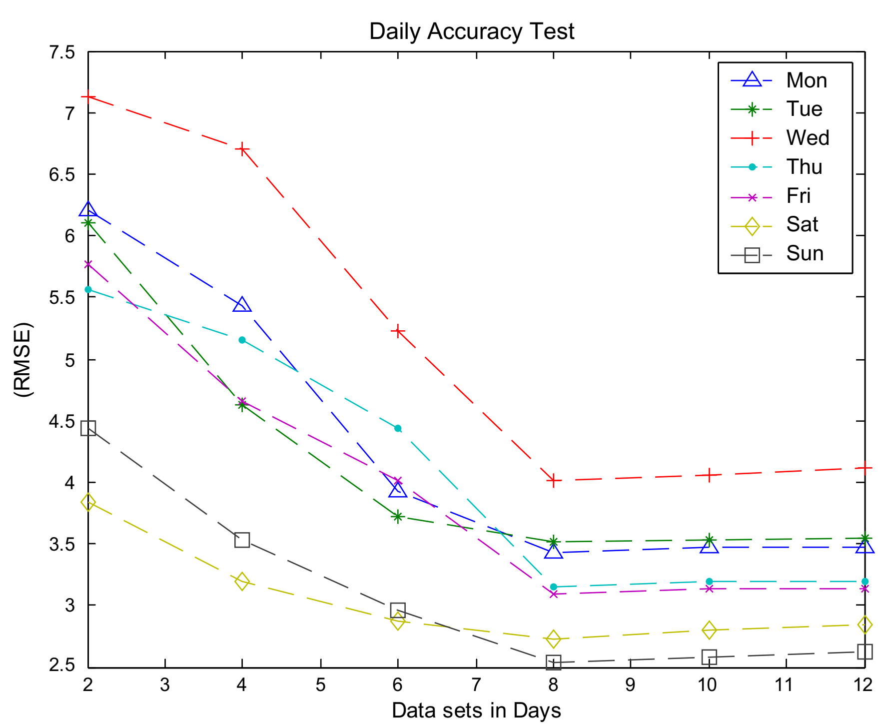

Fig. 10. The RMSE of EPNN for various data sets.

size. From Fig. 8, we can also see that the actual LMP volatiles more on Wednesday (48–72 h) than it is on Sunday. The RMSE on Wednesday is larger as expected form the training data.

6. Conclusions

EPNN integrated the PNN and OED to forecast LMPs based on similar days. PNN has the capability of dealing with varied and complicated relations between input and output data, and OED helps with the appropriated regulation of smoothing parameters to improve the forecasting results. The actual data of PJM were used to demonstrate the performance of EPNN. For selected days and weeks under study, the daily and weekly RMSE values are calculated, and it shows that EPNN can even track the spikes closely, which is not easily attainable with other methods. Compared with EPNN and PNN, the results demonstrate that EPNN is robust, efficient, and accurate. It proved that EPNN has the capability to produce better results for volatile price forecast.

References

[1] Ma X, Sun DI, Cheung KW. Evolution toward standardized market design. IEEE Trans Power Syst 2003;18(2):460–9.
[2] Shahidehpour M, Yamin H, Li Z. Market operations in electric power systems. John Wiley & Sons, Inc; 2002.
[3] PJM Energy Management. <http://www.PJM.com/index.jsp>.
[4] Nogales FJ, Contreras J, Conejo AJ, Espínola R. Forecasting next-day electricity prices by time series models. IEEE Trans Power Syst 2002;17(2):342–8.
[5] Lora AT, Santos JMR, Expósito AG, Ramos JLM, Santos JCR. Electricity market price forecasting based on weighted nearest neighbors techniques. IEEE Trans Power Syst 2007;22(3):1294–301.
[6] Contreras J, Nogales FJ, Conejo AJ. ARIMA models to predict next-day electricity prices. IEEE Trans Power Syst 2003;18(3):1014–20.
[7] Zhou M, Yan Z, Ni YX, Li G, Nile Y. Electricity price forecasting with confidence-interval estimation through an extended ARIMA approach. IEE Proc – Gener Transm Distrib 2006;153(2):187–95.
[8] Conejo AJ, Plazas MA, Espínola R, Molina AB. Day-ahead electricity price forecasting using the wavelet transform and ARIMA models. IEEE Trans Power Syst 2005;20(2):1035–42.
[9] González AM, Roque AMS, González J. Modeling and forecasting electricity prices with input/output hidden Markov models. IEEE Trans Power Syst 2005;20(1):13–24.
[10] Mandal P, Senjyu T, Urasaki N, Funabshi T, Srivastava AK. A novel approach to forecast electricity price for PJM using neural network and similar days method. IEEE Trans Power Syst 2007;22(4):2058–65.
[11] Catalão JPS, Mariano SJPS, Mendes VMF, Ferreira LAFM. An artificial neural network approach for short-term electricity prices forecasting. In: The 14th international conference on intelligent system applications to power systems, ISPA, Kaohsiung, Taiwan; 2007. p. 430–5.
[12] Zhang L, Luh PB, Kasiviswanathan K. Energy clearing price prediction and confidence interval estimation with cascaded neural networks. IEEE Trans Power Syst 2003;18(1):99–105.
[13] Catalão JPS, Mariano SJPS, Mendes VMF, Ferreira LAFM. Short-term electricity prices forecasting in a competitive market: a neural network approach. Electr Power Syst Res 2007;77:1297–304.
[14] Pao HT. Forecasting electricity market pricing using artificial neural networks. Energy Convers Manage 2007;48(3):907–12.
[15] Gareta R, Romeo LM, Gil A. Forecasting of electricity prices with neural networks. Energy Convers Manage 2006;47(13–14):1770–8.
[16] Pino R, Parreno J, Gomez A, Priore P. Forecasting next-day price of electricity in the Spanish energy market using artificial neural networks. Eng Appl Artif Intell 2008;21:53–62.
[17] Lin CH, Tsao MC. Power quality detection with classification enhancible wavelet-probabilistic network in a power system. IEE Proc – Gener Transm Distrib 2005;152(6):969–76.
[18] Specht DF. Probabilities neural network for classification, mapping, or associative memory. In: Proceedings IEEE int. con. neural network; 1991. p. 568–76.
[19] Ross PJ. Taguchi techniques for quality engineering. 2nd ed. The McGraw-Hill Companies, Inc.; 1988. p. 203–43.