J. Indian Inst. Sci., Jan.–Feb. 1997, 77, 107–123.

An intelligent hybrid scheme for time-series forecasting of electric load

P. K. DASH
Centre for Intelligent Systems, Department of Electrical Engineering, Regional Engineering College, Rourkela 769 008, India.

Received on December 15, 1994; Revised on June 6, 1995.

Abstract

This paper presents an intelligent hybrid scheme for short-term electric load forecasting using multilayered perceptrons. The hybrid neural network uses the membership values of the linguistic properties of the past load and weather parameters and the output of the network is defined as the fuzzy class membership values of the forecasted load. A hybrid learning algorithm consisting of unsupervised and supervised learning phases is used for training of the feedforward neural network. In the unsupervised learning phase optimal fuzzy membership values of input/output variables are obtained along with the optimal fuzzy logic rules. Kalman filter is used for the supervised learning phase. Extensive tests have been performed on a two-year utility data for the generation of peak and average load profiles in 24 and 168 hours ahead time frame. Results for typical winter and summer months are given to confirm the effectiveness of the hybrid scheme in comparison to standard ANN approach using backpropagation algorithm.

Key words: Hybrid learning scheme, fuzzy logic, ANN-based architecture, Kalman filters, load forecasting.

1. Introduction

Load forecasting plays a central role in the operation, planning and control of electric power systems. The forecast lead times range from a few minutes ahead for economic operation and load sharing between power plants to over 40 years for economic planning of new generating capacity and transmission networks.

The short-term load forecast (one to twentyfour hours) is of importance in the daily operations of a power utility. It is required for unit commitment, energy-transfer scheduling and load-management strategies, and for utility operations. The development of an accurate, fast and robust short-term load forecasting methodology is of importance to both electric utility and its customers.

A number of algorithms and techniques have been suggested for the solution of load prediction problem. They include statistical techniques, expert system and neural network approaches. The time series and regression techniques are the two major classes of conventional statistical algorithms, and have been applied successfully in this field for many years[1–5]. However, this technique does not produce a sufficiently accurate forecast and the accuracy deteriorates for larger variations of nonstationary load and weather variables. The expert system-based algorithm uses a symbolic computational approach for load forecasting and takes the expert knowledge of the operator which is, however, neither easy to elicit nor to articulate.

Over the past few years, artificial neural networks (ANN) have received a great attention and are now being proposed as a powerful computational tool for short-term load predictions[6–10]. This is because of two key features of the neural networks. First, the neural network does not rely on the explicitly expressed relationship between input variables and load forecasted. When using neural networks for load forecasting, one needs only to consider the selection of variables as the network input variables. The relationship between the input variables and predicted load will be formulated by a training process. Thus this approach avoids the difficulties in the modelling process. The adaptive algorithm is another appealing feature of neural networks. New training cases can be selected and system parameters estimated each time a new forecast is needed. Typically an ANN-based load forecasting problem uses the backpropagation approach during the training phase. The network is presented with a training data set made up of load and weather parameters.

The backpropagation algorithm although yields an accurate load forecast under normal circumstances, is susceptible to inaccurate predictions during fast changes in weather variables. Also extremely slow training or even training failure occurs in many cases due to difficulties in selecting proper structures of the neural network being used, and due to the errors in associated parameters such as learning rates, activation functions, etc., which are fundamental to any backpropagation neural network. The expert system, on the other hand,[11] provides a symbolic approach and emulates human expertise by capturing the knowledge of one or more experts in the form of rules and symbols. The heuristic approach of an expert system in arriving at decisions and/or solutions of a problem makes it unique concerning its performance. An explanation facility can provide the user with the line of reasoning followed by the expert system. The fuzzy logic-based expert system[12] for load forecasting requires a detailed analysis of data and the fuzzy rule base to be developed heuristically for each season. The rules fixed in this way may not always yield the best forecast. The shortcomings of the neural network paradigm can somewhat be remedied by the recognition of the fact that the learning speed and accuracy of an ANN may often be enhanced by integrating a fuzzy expert system into the neural network architecture. Expert networks represent one of the emerging hybrid approaches which combine the attributes of both the expert system and neural networks.

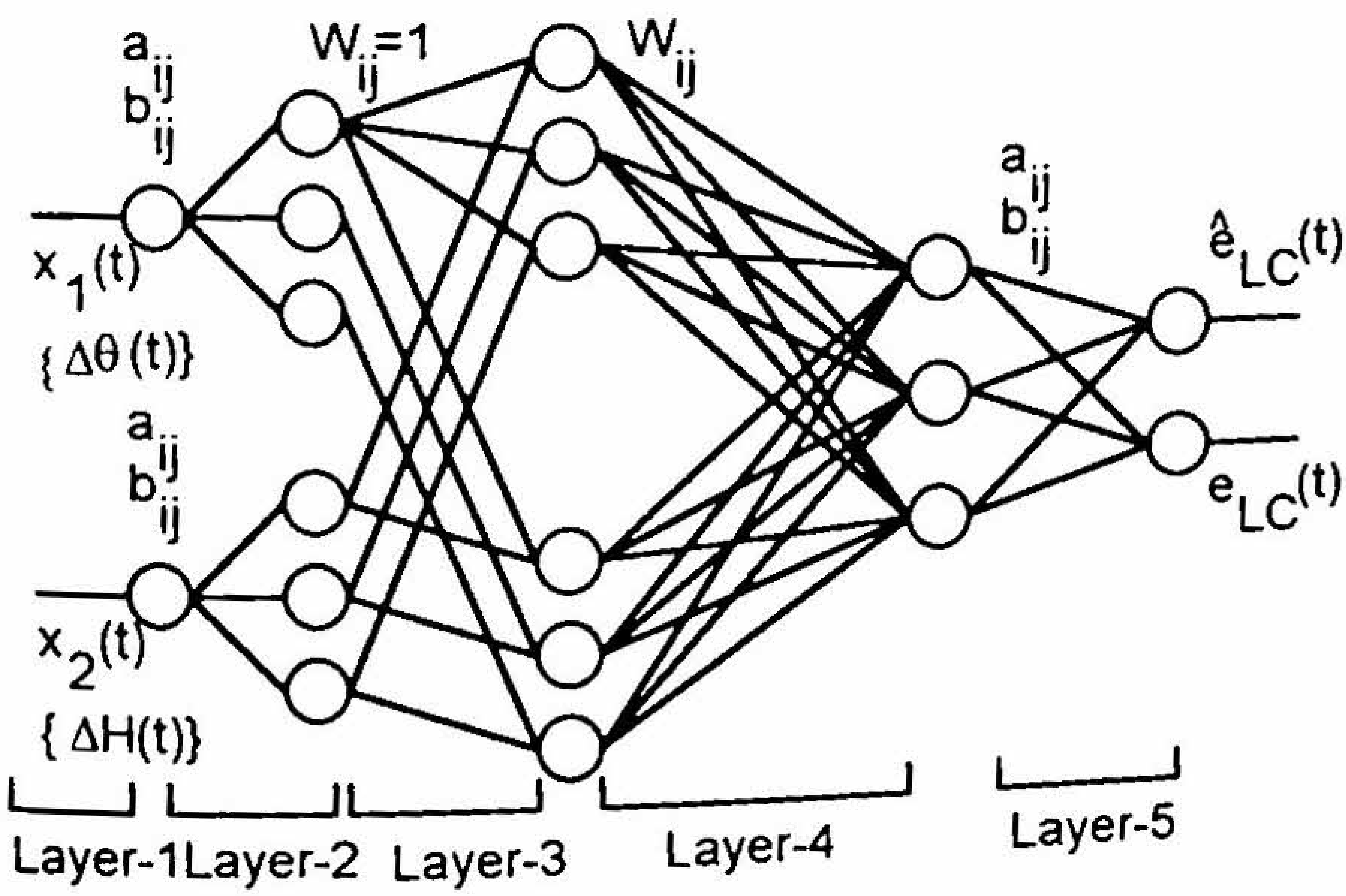
This paper presents a new hybrid approach for load forecasting using both supervised and upsupervised learning paradigms for integrated fuzzy-neural network (FNN model)[13–17]. The input to the hybrid model consists of the membership values of linguistic properties of past load and weather parameters and the output vector is defined in terms of fuzzy class membership values of the load forecasted. The output of the FNN model gives the load corrections which when added to the past load provides the load forecasted. The supervised learning paradigm for the hybrid model consists of a linear Kalman filter[18] with a variable forgetting factor. This method is similar to recursive least squares and produces a very fast convergence in comparison to the standard beckpropagation algorithm. A few examples of peak ad average load forecasts for a typical utility using a 24-hour lead time are presented in this paper to validate this hybrid approach. The accuracy of this model and its faster convergence with regard to the Neural Network model have been highlighted in this paper.

2. Hybrid neural network for time-series forecasting

An alternative to the neural network based load forecast is the expert system approach. A fuzzy expert system for load forecast consists of a collection of fuzzy IF–THEN rules showing the relation between load and weather variables. One of the difficulties with the fuzzy expert system is the rule matching and composition time, apart from the time consuming process of adapting the rules. The neural network eliminates the rule matching process and stores the knowledge in the link weights. The decision signals can be pumped out immediately after the input data are fed in. Figure 1 shows the proposed fuzzy neural network (FNN) to model the fuzzy expert system in the form of FNN using the ANN architecture. The FNN clusters the differential temperatures and humidities of the ith and $i+n$th day into fuzzy terms sets. The output of the FNN is the final crisp load correction ($\hat{e}_{LC}$). Hence the load forecasted on i + nth day ($P_f(i + n)$) is given by:

$$p_f(i+n) = P(i) + \hat{e}_{LC}(i) \tag{1}$$

where, n is the lead time for the forecast.



Layer-1 = Input linguistic nodes, Layer-2 = Input terms
Layer-3 = rule nodes, Layer-4 = Output term nodes
Layer-5 = Output linguistic nodes
$\Delta\theta$-Differential temperature, t-Iteration no
ΔH-Differential humidity, W_{ij}-Weights,
($\hat{e}_{LC}$)-Actual load correctioin, e_{LC}-Desired load correction

FIG. 1. Hybrid neural network for load forecasting.

The FNN has a total of five layers. Nodes at layer one are the input linguistic nodes. Layer 5 is the output layer and consists of two nodes (one for the actual load correction ($\hat{e}_{LC}$) and the other for the desired load correction (e_{LC})). Nodes at layer two and four are term nodes which act as membership functions to represent the term sets of the respective linguistic variable. Each node at layer three represents the preconditions of the rule nodes, and layer four links define the consequence of the rules. The functions of each layer is described as follows :

a) *Layer* 1 : The nodes in this layer just transmit the input feature x_i, $i = 1,2$ to the next layer.

b) *Layer* 2 : Each input feature x_i, $i = 1,2$ is expressed in terms of membership values $\mu_x^j(a_{ij}b_{ij})$, where i corresponds to the input feature and j corresponds to the number of term sets for the linguistic variable x_j. The membership function $\mu_{x_i}^j$ uses the Gaussian membership function given (Fig. 2).

$$\mu_{x_i}^j(a.b) = \exp\left\{\frac{-(x-a)^2}{b}\right\} \tag{2}$$

where a and b are the centre and width of the Gaussian function.

c) *Layer* 3 : The links in this layer are used to perform precondition matching of fuzzy logic rules. Hence the rule nodes perform the product operation (or AND operation).

$$\mu_{R_p} = \prod \mu_{x_i}^j \tag{3}$$

Where $R_p = 1,2,\ldots,n$. R_p corresponds to the rule node and n is the maximum number of rule nodes. However, if the fuzzy AND operation is used

$$\mu_{R_p} = \min\left\{\mu_{x_i}^j\right\} \tag{4}$$

d) *Layer* 4 : The nodes in this layer have two operations, *i.e.*, forward and backward transmission. In forward transmission mode, the nodes perform the fuzzy OR operation to integrate the fired rules which have the same consequence :

$$\mu^4 = \sum_{i=1}^{p} \mu_i^4 \tag{5}$$

where p corresponds to the links terminating at the node. In the backward transmission mode, the links function is exactly the same as the layer 2 nodes.

e) *Layer* 5 : There are two nodes in this layer for obtaining the actual and desired output load correction, respectively. The desired output load correction (e_{LC}) is fed into the hybrid model during learning whereas the actual load correction ($\hat{e}_{LC}$) is obtained by using the centroid defuzzification method[15].

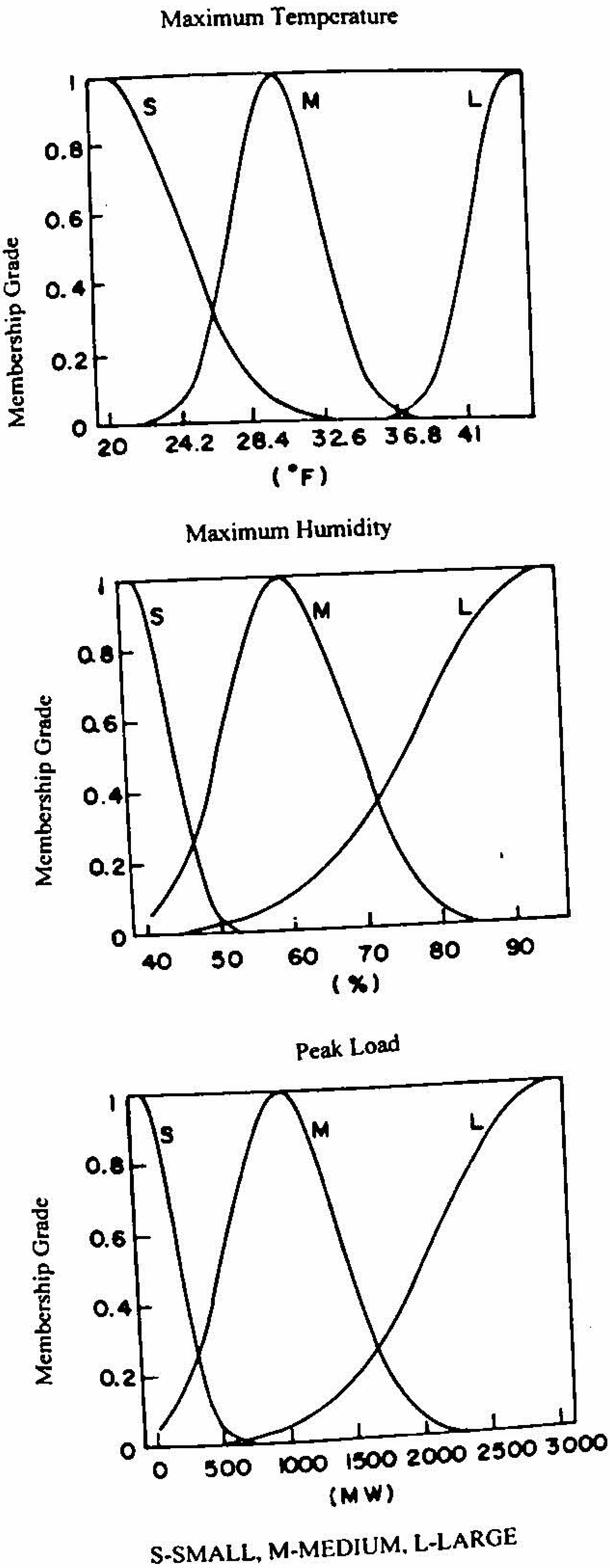


FIG. 2. Peak load, maximum temperature, maximum humidity membership functions (data for winter months are used).

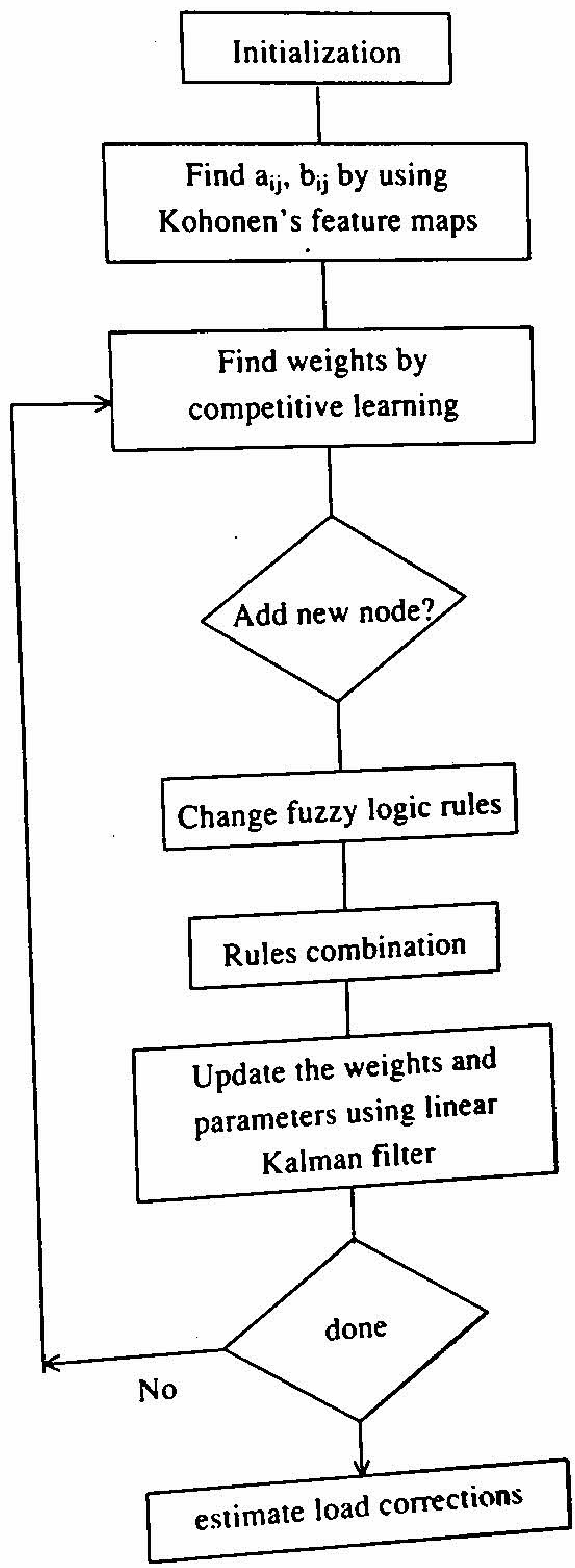


FIG. 3. Hybrid learning procedure.

2.1. *Hybrid learning algorithm for fuzzy neural network*

The hybrid learning scheme consists of unsupervised and supervised learning phases. In the unsupervised phase, the initial membership functions of the input and output linguistic variables are fixed and an initial form of the network is constructed. Then during the learning process, some nodes and links of this initial network are deleted or combined to form the final structure of the network. In the supervised learning phase, the input and output membership functions are optimally adjusted to obtain the desired outputs.

2.1.1. *Unsupervised learning phase*

Given the training input data, $x_i(t)$, $i = 1, 2$, the desired output load correction ($e_{LC}(t)$) and the fuzzy partitions $\left|\mu_{x_i}^{j}\right|$, we want to locate the membership function (*i.e.*, a_{ij} and b_{ij}) and find the fuzzy logic rules.

The Kohonen's feature map algorithm[13] is used to find the values for a_{ij} and b_{ij}.

$$\left\|x(t) - a_{i,\text{closest}}(t)\right\| = \min_{1\le j\le t}\left\{\left\|x_1(t) - a_{ij}(t)\right\|\right\} \tag{6}$$

$$a_{i,\text{closest}}(t+1) = a_{i,\text{closest}}(t) + \eta(t)\left[x_i(t) - a_{i,\text{closest}}(t)\right] \tag{7}$$

$$a_{ij}(t+1) = a_{ij}(t) \text{ for } a_{ij} \neq a_{i,\text{closest}} \tag{8}$$

where $\eta(t)$ is the monotonically decreasing learning rate and t, the number of term set for the linguistic variable x_i.

The width, b_{ij}, is determined heuristically at this stage[13] as follows :

$$b_{ij} = \frac{\left|a_{i,j} - a_{i,\text{closest}}\right|}{r} \tag{9}$$

where r is an overlap parameter. After the parameters of the membership functions have been found, the weights in layer 4 are obtained by using the competitive learning algorithm[6] as follows :

$$W_{ij} = \mathrm{LI}_j^4(\mathrm{LI}_j^3 - W_{ij}) \tag{10}$$

where LI_j^3 serves as the win-loss index of the rule node at layer three and LI_j^4, as the win-loss index of the jth term node at layer four, respectively.

After competitive learning through the whole training data set, the link weights at layer four represent the strength of the existence of the corresponding rule consequence. If a link weight between rule node and the term node of the output linguistic node is very small, then all the corresponding links are deleted, meaning that this rule node has little or no relation to the output.

After the consequences of rule nodes are determined, the rule combination is performed to reduce the number of rules in the following manner. The criterion for the choice of rule nodes is :

(i) they have the same consequences
(ii) some preconditions are common to all the rule nodes in this set
(iii) the union of other preconditions of these rule nodes composed the whole term set of some input linguistic variables.

The rule nodes which satisfy these criteria are replaced by a new rule node with common preconditions.

2.1.2. Supervised learning phase

Once the fuzzy logic rules have been found, supervised learning is used to find the optimum weights and the input and output membership functions.

Referring to Fig. 1, the tuning of Gaussian membership function at layer two and four (a_{ij}, b_{ij}) is similar to the weight update equations at layer three. The supervised learning phase of the FNN model uses the linear Kalman filter equations for updating the weights and the membership function. Unlike the backpropagation technique, this algorithm assumes that the estimated weight matrix is non-stationary and hence will allow the tracking of a time varying data like that of load forecasting. The hybrid learning procedure is summarised in Fig. 3.

This algorithm defines locally at each node a gradient based on present and past data, and updates the weights of each node using the linear Kalman filter equations so as to bring this gradient identically to zero whenever an update is made. Performing the update thus and defining the gradient in this manner ensures that maximum use is made of available information.

The gradient for the linear combiner at each node is defined as

$$G = RW - C \tag{11}$$

Here R is the auto correlation matrix for each layer and is calculated as

$$R = \sum_{np=1}^{NP} ff^{NP-np} x_{np} x_{np}^T \tag{12}$$

and C is the cross-correlation matrix and is given by :

$$C = \sum_{np=1}^{NP} ff^{NP-np} d_{np} x_{np}^T \tag{13}$$

where NP denotes the total number of patterns, and ff, the forgetting factor. d_{np} and x_{np} are the summation output and the output of the nonlinearity (Gaussian membership function) for the nodes of layers two and five, respectively. At layer four nodes contain no non-linearity term, therefore $d_{np} = x_{np}$.

The weight vector which makes $G = RW - C$ zero is the solution to the equations.

The weight update equations for the hybrid model using the linear Kalman filter equations are :

a) The weight update equations for layer four are:

$$W_{ij}(t) = W_{ij}(t) + \eta K_j(t)\left[\frac{-\partial E}{\partial W_{ij}}\right] \tag{14}$$

where, $\frac{\partial E}{\partial W_{ij}}$ is given below and $K_j(t)$ is the Kalman gain.

The error function E is given by

$$E = \frac{1}{2}\left[e_{LC}(t) - \hat{e}_{LC}(t)\right]^2.$$

Since

$$\hat{e}_{LC} = \frac{\sum (a_{ij} b_{ij}) \mu_i^5}{\sum b_{ij} \mu_i^5} \tag{15}$$

and using centroid defuzzification method[15] we get

$$\mu_i^5 = \mu^4 = \sum_{-=1}^{p} \mu_i^4 W_{ij}$$

where $W_{ij} = 1$ for $t = 1$

Therefore,

$$\frac{\partial E}{\partial W_{ij}} = \frac{\partial E}{\partial \hat{e}_{LC}} \; \frac{\partial \hat{e}_{LC}}{\partial \mu_i^5} \; \frac{\partial \mu_i^5}{\partial W_{ij}}. \tag{16}$$

From 16, we obtain

$$\frac{\partial E}{\partial W_{ij}} = \left[e_{LC}(t) - \hat{e}_{LC}(t)\right]\left\{\frac{a_{ij} b_{ij}\left(\sum b_{ij} \mu_i^5\right) - \left(\sum a_{ij} b_{ij} \mu_i^5\right) b_{ij}}{\left(\sum b_{ij} \mu_i^5\right)^2}\right\}$$

The Kalman gain is given by

$$K_j(t) = \left\{\frac{R_j^{-1}(t) x_1(t)}{f_j + X_i^T(t) R_j^{-1}(t) x_1(t)}\right\} \tag{17}$$

where, $x_i(t)$ corresponds to the previous layer.

The forgetting factor f_j and the inverse convariance matrix $R_j^{-1}(t)$ are updated using

$$f_j(t+1) = f_0 f_j(t) + (1 - f_0) \tag{18}$$

$$R_j^{-1}(t+1) = \left[R_j^{-1}(t) - K_j(t) X_i^T(t) R_j^{-1}(t)\right] / f_j \tag{19}$$

b) The update equations for a_{ij} and b_{ij} at layer five are:

$$a_{ij}(t+1) = a_{ij}(t) + \eta_1 K_j(t) \left[\frac{-\partial E}{\partial a_{ij}}\right] \tag{20}$$

where $\frac{\partial E}{\partial a_{ij}}$ is given by

$$\frac{\partial E}{\partial a_{ij}} = \delta_i^2 e^{\left(\frac{-(x_i - a_{ij})^2}{b_{ij}}\right)} \left\{\frac{2(x_i - a_{ij})}{b_{ij}}\right\}$$

$$b_{ij}(t+1) = b_{ij}(t) + \eta_2 K_j(t) \left[\frac{-\partial E}{\partial a_{ij}}\right] \tag{21}$$

where $\frac{\partial E}{\partial b_{ij}}$ is given by

$\frac{\partial E}{a b_{ij}} = \frac{\partial E}{\partial \hat{e}_{LC}} \; \frac{\partial \hat{e}_{LC}}{a b_{ij}}$ and can be computed using eqn (15)

c) The update equations for a_{ij} and b_{ij} at layer two are :

$$a_{ij}(t+1) = a_{ij}(t) + \eta_3 K_j(t) \left[\frac{-\partial E}{\partial a_{ij}}\right]. \tag{22}$$

Similarly,

$$b_{ij}(t+1) = b_{ij}(t) - \eta_4 K_j(t) \delta_i^2 e^{\left(\frac{-(x_i - a_{ij})^2}{b_{ij}}\right)} \left\{\frac{(x_i - a_{ij})^2}{b_{ij}}\right\} \tag{23}$$

where δ_i^2 is given by

$$\begin{aligned} \delta_i^2 &= \sum \delta_i^3 \\ \delta_i^3 &= \sum \delta_i^4 \end{aligned} \tag{24}$$

and

$$\delta_i^4 = \left[e_{LC}(t) - \hat{e}_{LC}(t)\right] \left\{\frac{a_{ij} b_{ij} \left(\sum b_{ij} \mu_i^5\right) - \left(\sum a_{ij} b_{ij} \mu_i^5\right) b_{ij}}{\left(\sum b_{ij} \mu_i^5\right)^2}\right\} \tag{25}$$

where a_{ij} and b_{ij} correspond to the output term set.

5. Implementation results

In order to evaluate the performance of the hybrid models, the load forecasting is performed on a typical utility data. The hybrid model along with the ANN-based model are tested on a two-year utility data for generating peak and average load profiles and some of the results are given in the subsequent subsections. The training sets are formed separately for each of the seven-day types (*i.e.*, Tuesdays through Thursdays, Mondays, Fridays, Saturdays, Sundays, holidays). The selection of training patterns is given in Rahman *et al.*[9].

5.1. *Peak load forecasting*

For peak load forecasting, the following training data are used for the model :

Input pattern : $P_{\max}(i)$, $\theta_{\max}(i)$, $H_{\max}(i)$, $\theta^{f}_{\max}(i)$, $H^{f}_{\max}(i)$

Output pattern : $P_{\max}(i+n)$ and $\mu(P_{\max}(i+n))$ for *ANN* and *FNN*, respectively.

where P, θ, H stand for load, temperature and humidity, respectively. Superscript f denotes the forecasted values for θ and H; n is the lead time for the forecast ($n = 24$ for 24-hours ahead forecast, $n = 168$ for 168-hour ahead forecast).

For the hybrid FNN model, the training patterns used are :
Input pattern : $\Delta\theta_{\max}(i, i+n)$ and $\Delta H_{\max}(i, i+1)$;
Output pattern : e_{LC}, the desired load correction.

Here again the weather variables used for $(i + n)$th day are the forecasted values.

Table I gives the learned membership function using FNN model for 24-hour ahead peak load forecasting in winter. For example, rule 0 is interpreted as :

RO = IF $\Delta\theta_{\max}$ is term 0 and $\Delta H_{\max}$ is 3 THEN $(\hat{e}_{LC})$ is term 7.

Figure 4 gives the learned membership functions for the FNN model after the first phase (unsupervised learning phase) and the second phase (supervised learning phase). Figure 5 gives the plot of mean absolute percentage errors (MAPEs) *versus* the number of iterations for the ANN and the hybrid FNN models, respectively. The results in Figs 4 and 5 were obtained 24-hours ahead of peak load forecasting in winter.

From Fig. 5 we see that the hybrid FNN model gives an extremely fast rate of convergence followed in comparison to the ANN model. The linear Kalman filter equations and the variable forgetting factor used for the training of the FNN model are instrumental in driving the MAPE low during the first few hundred iterations until bias, caused by initial parameters arbitrarily chosen is eliminated.

Figure 6 gives the peak load forecasting results, for both ANN and the hybrid FNN model in the month of June (summer) using 24-hour ahead forecast. The number of iterations and MAPE for ANN and the hybrid models are :

Table I
The learned fuzzy logic rules for 24- hours ahead peak load forecasting using FNN_2 in winter

Rule	*Term sets*		
	Preconditions		*Consequence*
	$\Delta\Theta_{max}(i, i+1)$	$\Delta H_{max}(i, i+1)$	$\hat{e}_{LC}(i)$
0	0	3	7
1	0	4	7
2	1	0	8
3	1	1	7
4	1	2	7
5	1	3	6
6	1	4	6
7	2	0	8
8	2	1	7
9	2	2	7
10	2	3	6
11	2	4	7
12	3	1	2
13	3	2	4
14	3	3	6
15	4	2	5
16	4	3	6
17	4	4	1
18	5	0	3
19	5	1	2
20	5	2	1
21	5	3	1
22	5	4	0
23	6	0	1
24	6	1	1
25	6	2	0

ANN :
Number of iterations = 970
MAPE = 2.45
Hybrid model:
Number of iterations = 440
MAPE = 1.0

From the figure we observe that the FNN gives a very accurate prediction followed by the neural network.

5.2. *Average load forecasting*

For average load forecasting, the following training data are used for ANN model
Input pattern:

$P_{av}(i)$, $\theta_{max}(i)$, $\theta_{min}(i)$, $H_{max}(i)$, $H_{min}(i)$, $\theta^f_{max}(i+n)$, $\theta^f_{min}(i+n)$, $H^f_{max}(i+n)$, $H^f_{min}(i+n)$
Output pattern : $P_{av}(i+n)$ and $\mu(P_{av}(i+n))$ for *ANN* and *FNN*, respectively.

where n is the lead time for the forecast as given in Section 5.1.

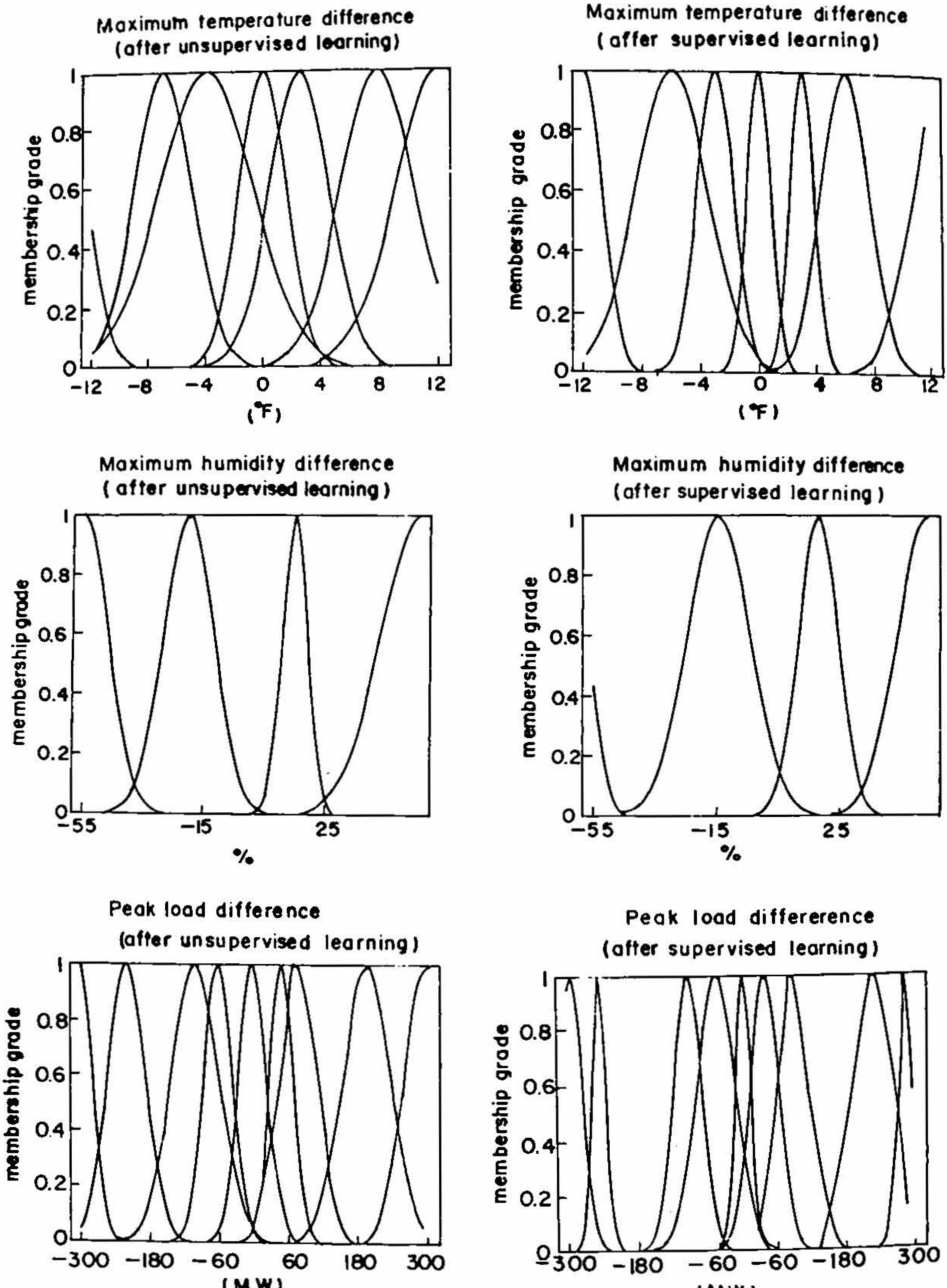


FIG. 4. Learning membership functions for peak load forecasting in winter using the hybrid model.

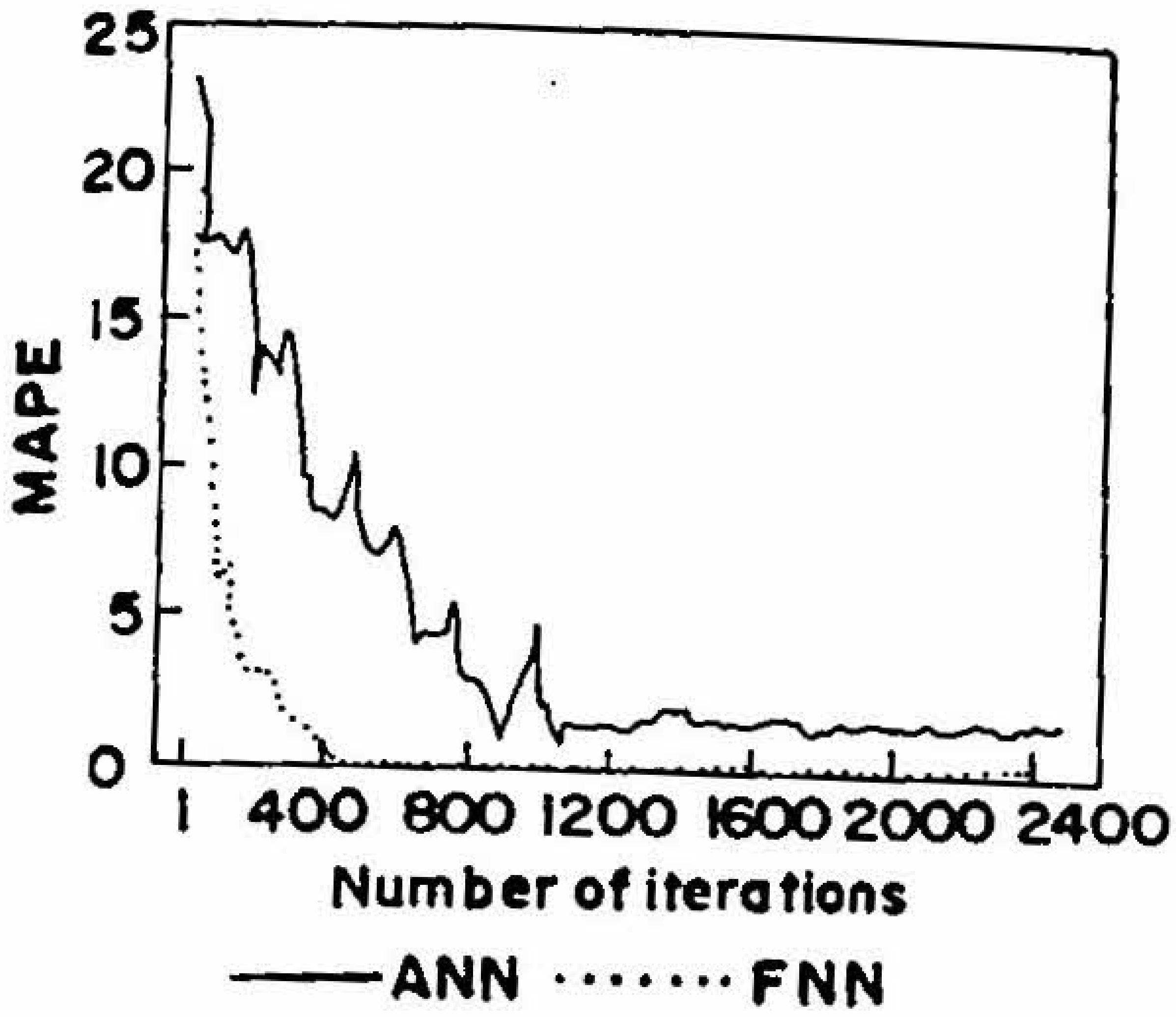


FIG. 5. Mean absolute percentage error for 24-hour ahead peak load forecasting in winter.

For the hybrid model, the training patterns used are :

Input pattern : $\Delta\theta_{max}(i,i+n)$, $\Delta\theta_{min}(i,i+n)$, $\Delta H_{max}(i,i+n)$, $\Delta H_{min}(i,i+n)u$
Output pattern : $e_{LC}(i)$, the desired load correction.

The $P_{av}(i+1)$ for FNN model is obtained using eqn (1).

For the average load forecast also, the forecasted temperature and humidity values are used for the day of the forecast.

Figure 7 presents the average load forecasting results, for both the ANN and the hybrid model, for the month of January (winter) using 24-hour ahead predictions. From these results we note the improved performance of the hybrid model in terms of faster convergence and improved overall accuracy followed by ANN model.

6. Discussion

The proposed hybrid fuzzy neural network model is found to be very powerful in providing an accurate load forecast. Although the results for two seasons of the year are presented in this paper for validating the effectiveness of this approach, extensive tests have been conducted for other seasons, Sundays, holidays and special days of the year. From the results presented in this paper, it can be observed that significant accuracy can be achieved in the case of 24-hour ahead hourly load forecasts and the PEs can be less than 1. However, the PEs increase in the case of peak load forecasts and will remain within 2. If the lead time increases to one weak, the Kalman filter-based hybrid model yields a PE around 2 for average load forecast and around 3 for peak load forecast. Further the results presented in the paper also reveal the superiority of the Kalman filter based hybrid forecasting model over the ordinary neural network model in terms of speed of convergence, MAPE and maximum percentage error.

The accuracy of the hybrid models can be further enhanced by choosing more number of fuzzy overlapping sets for fuzzification of input variables instead of the three used for

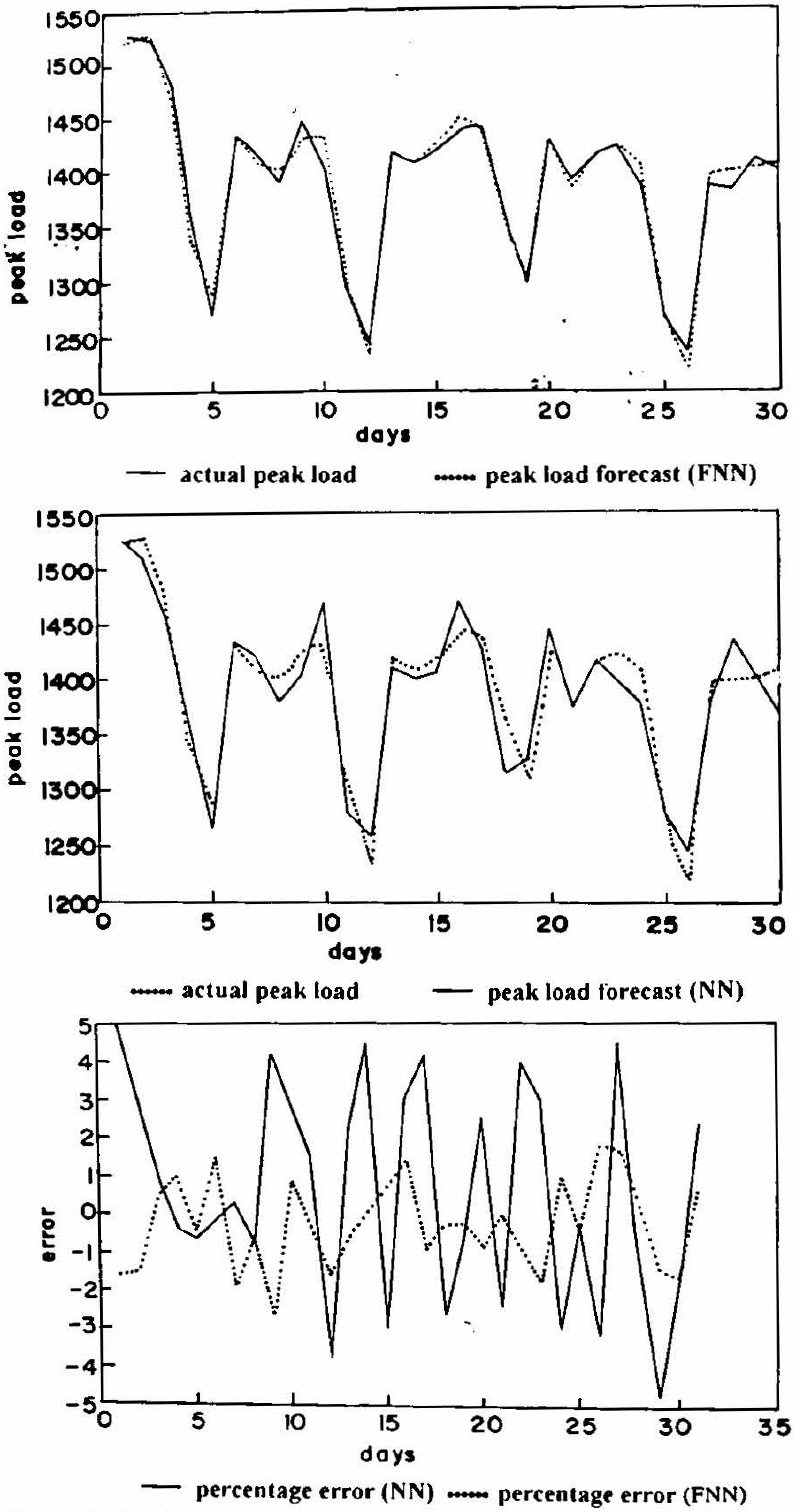


FIG. 6. Peak load forecasting results.

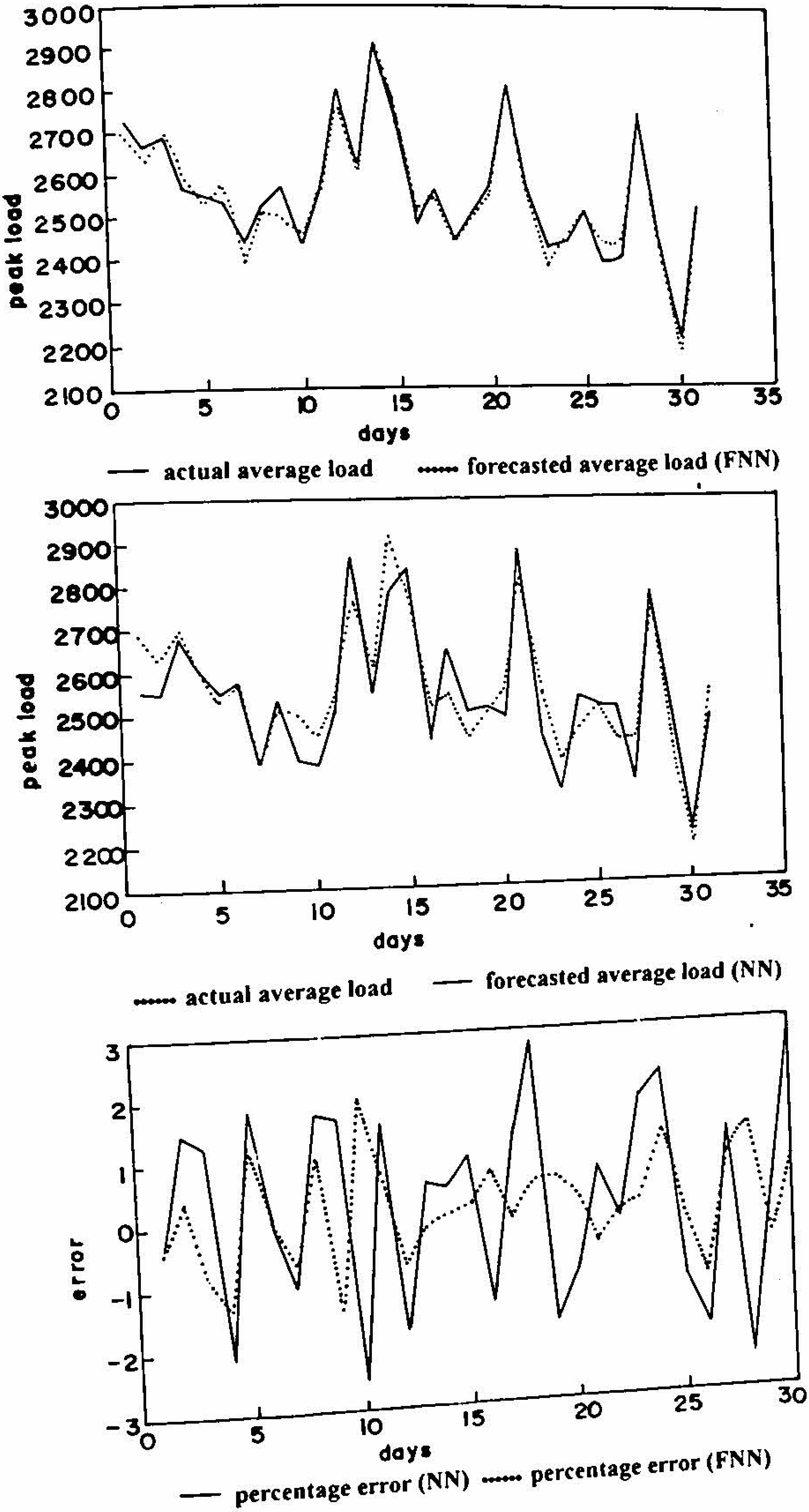


FIG. 7. Average load forecasting result.

this application. Also the choice of membership function is flexible to take into account different seasonal load and weather variables. This increases the number of rules and consequently the rule nodes in the hybrid model. The database used for this study comprises a 14-day period prior to the day of forecast and thus by using a larger database (say 4 weeks) and increased number of load and weather parameters as input variables, a more accurate and robust forecast for one-day to one-week ahead forecast can be obtained.

We have also performed extremely short-term predictions from 1- to 6-hour ahead forecasts. The main features and advantages of the hybrid model are : (i) it provides a general method to combine available numerical and human linguistic information into a common framework ; (ii) requires much less construction time than a comparable neural network, and (iii) significant accuracy in predicting chaotic time series models.

7. Conclusions

This paper presents three fuzzy neural network (FNN) models for time series forecasting of electric load. The proposed hybrid model introduces the low-level learning power of artificial neural network into a fuzzy expert system and provides a high-level human-understandable meaning to the normal neural network. A hybrid learning scheme consisting of self-organized learning phase and supervised learning phase is used for training the network. Also the Kalman filter update equations in the supervised learning phase of FNN give better convergence and accuracy over the gradient-descent backpropagation algorithm in the supervised learning phase of the hybrid model.

Acknowledgements

The author greatfully acknowledges grants from the Ministry of Human Resources, Govt. of India, for undertaking this research.

References

1. GROSS, G. AND GALIANA, F. D., *Short-term load forecasting*, *Proc. IEEE*, 1987, 75, pp. 1558–1572.
2. BOX, G. E. P. AND JENKINS, G. M. *Time series analysis forecasting and control*, Oakland, CA. 1976, Holden-Day.
3. MOGHRAM, I. AND RAHMAN, S., Analysis and evaluation of five short-term load forecasting techniques, *IEEE Trans.*, PWRS-4, 1989, pp. 1484–1490.
4. BUNN, D. W. AND FARMER, E. D. *Comparative models for electric load forecasting*, 1989. Wiley.
5. HAGAN, M. T. AND BEHR, S. M. *The time series approach to short-term load forecasting*, *IEEE Trans.*, PWRS-2, 1987, pp.785–791.
6. EL-SHARKAWI, M. A, OH, S., MARKS, R. J., DAMBOURG, M. J. AND BRACE, C. M. Short term electric load forecasting using an adaptively trained layered perceptron, *First Int. Forum on Applications of Neural Networks to Power Systems*, Seattle, WA July 23-26 1991, pp.3–6.
7. LEE, K. Y., CHA, Y. T. AND PARK, J. H. Short term load forecasting using an artificial neural network, *IEEE Trans.*, PWRS-7, 1992, pp.124–133.

8. PENG, T. M., HUBELE, N. F. AND KARADY, G. G. Advancement in the application of neural networks for short term load forecasting, *IEEE Trans.*, PWRS-7, 1992, pp.250–258.

9. RAHMAN, S., DREZGA, I. AND RAJAGOPALAN, J., Knowledge enhanced connectionist models for short-term electric load forecasting, *Int. Conf. on ANN applications to Power Systems*, Japan, April 1993.

10. PARK, J. H., PARK, T. M. AND LEE, K.Y. Composite models for adaptive short term load forecasitng, *IEEE* Trans., PWRS-1, 1991, pp.450–457.

11. RAHMAN, S. AND BHATNAGAR, R. An expert system based algorithm for short term load forecast, *IEEE Trans.*, PWRS-3, 1988, pp.392–399.

12. DASH, P. K., DASH, S., RAMAKRISHNA, G. AND RAHMAN, S., Forecasting of a load time series using a fuzzy expert system and fuzzy neural networks, *Engng Intell. System J.*, **1**, 1993, pp.103–117.

13. LIN, C. T., GEORGE, C. S., LEE, Neural network-based fuzzy logic control and decision system, *IEEE Trans.*, C-**40**, 1991, pp.1320–1336.

14. HAYASHI, Y., BUCKLEY, J. J. AND Fuzzy expert systems versus neural networks, *Proc. Int. + Conf. on Neural Networks*, Baltimore, June 7–12 1992, **2**, pp.720–726.

15. WANG, L. X. AND MENDEL, J. M. Generating fuzzy rules by learning from examples, *IEEE Trans.*, SMC-22, 1992, pp.1414–1427.

16. PAL, S. K. AND MITRA, S. Multilayer Perceptrons, fuzzy and classifications, *IEEE Trans.*, NN-3, 1992, pp.683–698.

17. BUCKLEY, J. J., HAYASHI, Y. AND CZOGALA, E. On the equivalence of neural nets and fuzzy expert systems, *Fuzzy Sets System*, 1993, **53**, pp.129–134.

18. DASH, P. K., SATPATHY, J. K., RAMA KRISHNA, G. and RAHMAN, S., An improved Kalman filter based neural network approach for short-term load forecasting, *Proc. third Int. Symp. on Electricity Distribution and Energy Management*, 1993, **1**, pp.3–8.