SHORT TERM DAILY AVERAGE AND PEAK LOAD PREDICATIONS USING A HYBRID INTELLIGENT APPRACH

P.K.Dash & H.P.Satpathy
Regional Engineering College
Rourkela, India

S.Rahman
Virginia Polytechnic Institute and State University,
USA

ABSTRACT A fuzzy neural network based on the multilayer perceptron and capable of fuzzy classification of patterns is presented in this paper. A hybrid learning algorithm consisting of unsupervised and supervised learning phases is used for training the network. In the supervised learning phase linear Kalman filter equations are used for tuning the weights and membership functions. Extensive tests have been performed on a two-year-utility data for generation of peak and average load profiles for 24- and 168-hours ahead time frames and results for winter and summer months are given to confirm the effectiveness of the new approach.

Keywords: Fuzzy neural network, membership function, unsupervised learning, supervised learning, load time series, forecasting.

I. INTRODUCTION

One of the most promising application areas of artificial neural network (ANN) is the load forecasting. The neural network is able to perform nonlinear modelling and adaptation and does not rely on the explicitly expressed relationship between the electric load and other variables such as weather conditions. When using neural networks for load forecasting one needs only to consider the selection of variables and previous load patterns as the network input. The relationship between the input variables and predicted load will be formulated by a training process.

Although ANN is a very promising tool for load forecasting, several key issues must be addressed before it can be effectively used[1-3]. The size of the ANN using a multi-layered perceptron is determined experimentally rather than theoretically. Several authors have used the error backpropagation learning algorithm to train the ANNs for forecasting a time series. This method of learning requires a large training time and forecasting accuracy is prone to errors due to non-stationary nature of the load and weather data. The fuzzy expert system approach has also been applied to load forecasting problem whereby the advantage of an operator's expert knowledge is used for load prediction. Although the fuzzy logic based forecaster showed promising results [4], the approach is subjective and somewhat heuristic, and the choice of membership functions and rule base has to be developed heuristically for each season. The rules fixed in this way may not yield the best forecast always and the choice of membership function still depends on trial-and-error. On the other hand hybrid solutions have been proposed for short-term forecasting of electric loads whereby the functionality of the fuzzy expert system and learning ability of the neural network can be merged to yield a forecasting system more powerful than either of its components alone[5]. Such a hybrid model provides human understandable meaning to the normal feed-forward neural network in which the internal units are always opaque to the users. The structure also avoids the rule matching time of the inference engine in the traditional fuzzy logic system and results in enhanced learning speed and prediction accuracy.

The hybrid model presented here is aimed at achieving a robust and accurate load forecast using a fuzzy expert system modelled as a five layer feed-forward neural network (FES)[6]. The input vector to the FNN consists of differences in weather parameters between the present and forecasted instant. The output of the Fuzzy Expert System(FES) gives the load correction which when added to the past load gives the forecasted load. The learning algorithm for the FES combines unsupervised and supervised learning procedures to build the rule nodes and train the membership functions. The supervised learning procedure for FES uses the Kalman filter based algorithm [7,8] for tuning the weights and membership functions. The hybrid algorithm performs superiorly in comparison with the purely supervised learning algorithm due to the apriori classification of training data through an overlapping receptive field before learning.

A few examples of peak and average load forecasting using the above technique for a typical utility with 24- and 168-hours lead times in the months of winter and summer are shown in this paper. A comparison with the standard backpropagation algorithm is presented in the paper.

II.FUZZY EXPERT SYSTEM (FES) FOR LOAD FORECASTING

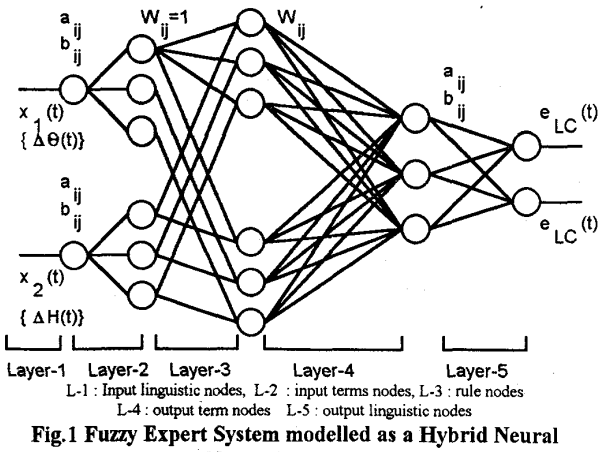


Fig.1 Fuzzy Expert System modelled as a Hybrid Neural Network (5 layers)

Figure 1 shows the proposed FES, to model the fuzzy expert system using the ANN [6] architecture. In many cases it is convenient to express the membership function of a fuzzy subset in terms of a standard nonlinear function. The Gaussian membership function is used for the input and output linguistic parameters of the FNN in this study :

$$\mu_{A_i}(x) = \exp\left\{-(x-a)^2 / b\right\} \qquad (1)$$

here, a and b are the centre and the width of the bell shaped function, respectively. As the values of the parameters change, the bell-shaped functions vary accordingly thus exhibiting various form of membership functions on linguistic label A_i.

The FES clusters the differential temperatures ($\Delta\theta$) and humidities (ΔH) of the (k)th and (k+n)th day into fuzzy term sets using the Gaussian membership function given in (1). Thus, $\Delta\theta = \theta(k+n) - \theta(k)$, $\Delta H = H(k+n) - H(k)$, here, n is the lead time for

IEEE Catalogue No. 95TH8130
0-7803-2981-3/95/$4.00©1995 IEEE

the load forecast (i.e. n=24 for 24-hours ahead forecast). The forecasted load on (k+n)th day (P_f(k+n)) is given by

$$P_f(k+n) = P(k) + e_{LC}(k) \quad (2)$$

where P(k) = load on the kth day and e_{LC} is the output from the FNN (load correction).

The FES 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 is for the actual load correction (e_{LC}) and the other is desired load correction ($\bar{e}_{LC}$)). 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 function 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 $\mu^j_{x_i}(a_{ij}, b_{ij})$, where i corresponds to the input feature and j corresponds to the number of term sets for the linguistic variable x_i . The membership function $\mu^j_{x_i}$ uses the Gaussian membership function (1).

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).

$$\mu_{R_P} = \Pi \mu^j_{x_i} \quad (3)$$

where R_P = 1,2,..,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

$$\mu_{R_P} = \min\{\mu^j_{x_i}\} \quad (4)$$

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.

$$\mu = \sum_{i=1}^{p} \mu^4_i \quad (5)$$

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

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

III. Hybrid Learning Algorithm For FES

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.

A. Unsupervised Learning Phase

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

The Kohonen's feature maps algorithm [6] is used to find the value for a_{ij} and b_{ij}.

$$\left\|x(t) - a_{i,closest}(t)\right\| = \min_{i \le j \le l} \left\{\left\|x_i(t) - a_{ij}(t)\right\|\right\} \quad (6)$$

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

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

where η(t) is the monotonically decreasing learning rate and l is the number of term set for the linguistic variable x_i.

This adaptive formulation runs independently for each input linguistic variable x_i.

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

$$b_{ij} = \left|a_{ij} - a_{i,closest}\right| / r \quad (9)$$

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 :

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

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

After the 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

1) they have the same consequences,
2) some preconditions are common to all the rule nodes in this set,
3) the union of other preconditions of these rule nodes composes 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.

B. Supervised Learning Phase

Once the fuzzy logic rules have been found, phase II learning is used to find the optimum weights and the input and output membership functions by using the Kalman filter based learning algorithm [7].

Unlike the backpropagation technique, this algorithm assumes that the estimated weight matrix is nonstationary and hence will allow the tracking of a time varying data like that of load forecasting. 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 [7] so as to bring this gradient identically to zero whenever an update is made. The gradient for the linear combiner is defined as

$$G = R\,W - C \tag{11}$$

The weight vector W which makes G zero is the solution to equation (11).

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

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

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

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

where, NP denotes the total number of patterns, f is a forgetting factor, and d_{np} and x_{np} are the summation output and the output of the nonlinearity (Bell shaped membership function) for the layer two and layer five nodes, respectively. As the layer four nodes contain no nonlinear term, therefore, $d_{np} = x_{np}$. The weights are updated by using a Kalman filter at each layer with a variable forgetting factor (f). The variable forgetting factor (f) is used to take care of new estimates and gives less weightage to the old estimates. The Kalman gain vector K_j(t) at each layer j is given by :

$$K_j(t) = \left\{R_j^{-1}(t)x_i(t)\right\} \Big/ \left\{f_j + x_i^T(t)R_j^{-1}(t)x_i(t)\right\} \tag{14}$$

where, x_i(t) corresponds to the previous layer, μ_j is the learning rate and t denotes the iteration number. The error, E at each iteration is obtained as

$$E = \tfrac{1}{2}\left(\bar{e}_{LC}(t) - e_{LC}(t)\right)^2 \tag{15}$$

and using centroid defuzzification

$$e_{LC} = \Sigma(a_{ij}b_{ij})\mu_i^5 \Big/ (\Sigma(b_{ij})\mu_i^5) \tag{16}$$

The weight update equation for layer four is :

$$W_{ij}(t) = W_{ij}(t) + \eta K_j(t)\left[-\partial E / \partial W_{ij}\right] \tag{17}$$

where,

$$\partial E \Big/ \partial w_{ij} = \left\{e_{LC}(t) - \hat{e}_{LC}(t)\right\}\beta\mu_i^4 \tag{18}$$

and

$$\beta = \left\{a_{ij}b_{ij}\Sigma b_{ij}\mu_i^5 - (\Sigma a_{ij}b_{ij}\mu_i^5)b_{ij}\right\} / (\Sigma b_{ij}\mu_i^5)^2 \tag{19}$$

IV. Results

In order to evaluate the performance of the FES model, the load forecasting is performed on a typical utility data for generating peak and average load profiles and some of the results are given in the subsequent sub-sections. In [3], it has been shown that ANNs give the best prediction and accuracy compared to conventional approaches. So in this paper the results of FES using either backpropagation or Kalman filter approach are compared to that of the standard ANN approach described in reference [3]. The training sets are formed separately for each of the six day types (Tuesdays through Thursdays, Mondays, Fridays, Saturdays, Sundays, Holidays).

A neural network can make forecast only on the basis as to how it is trained. In power systems, numerous previous load and weather data show divergent patterns and dynamic ranges. For load predictions, the candidates for inputs to the neural network are time of the day; day of the weak; season of year; outdoor temperature in terms of maximum, minimum or average value; wind speed; cloud cover; load at past points in time; and weather forecasts. The following load model is used for forecasting.

$$y(i) = f\left(y(i-n), y(i-n-1), \ldots, y(i-n-n_1)\right.$$

A&M University, VPSU, Blacksburg, where he is currently a professor. He serves on the System Planning and demand Side Management Subcommittees of IEEE Power Engineering group and has authored more than 160 technical papers and reports in

$$\left.z(i-n), z(i-1-n), z(i-n-2)\ldots\right)$$

where y and z are the load and weather variables, respectively. Here n indicates the lead time of the forecast, i.e., n = 24 for 24-hour ahead forecast; n_1 indicates the data length for the load and n_2 indicates the data length for temperature and humidity, respectively. Various lengths of the past historical load and temperature values are used and their effects on the load forecast accuracy are studied. It is found that for n_1>0, there are no marked improvements in the results for the utility data used in the analysis. Also the training time increases considerably with larger values of n_1 and n_2. Therefore, n_1 = 0, and n_2=0 are chosen for obtaining the load forecasts.

A. Peak Load Forecasting

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

Input Pattern :

$$P_{\max}(k), \theta_{\max}(k), H_{\max}(k), \theta_{\max}^f(k+n), H_{\max}^f(k+n)$$

Output Pattern : $P_{\max}(k+n)$ for ANN

where P, θ, H stand for load, temperature and humidity, respectively. Superscript 'f' for θ and H stand for the forecasted values. The forecasted values for temperature and humidity are used to get a more realistic load forecast; n is the lead time for the forecast, i.e., n=24 for 24-hours ahead forecast, k = the day of forecast. The backpropagation learning algorithm is used.

For FNN the training patterns used are :

Input Pattern : $\Delta\theta_{\max}$(k,k+n) and $\Delta H_{\max}$(k,k+n)

Output Pattern : $\bar{e}LC$(k), the desired load correction

Kalman filter parameters

f = 0.99, μ_1 = 40, K(0) = 1.0

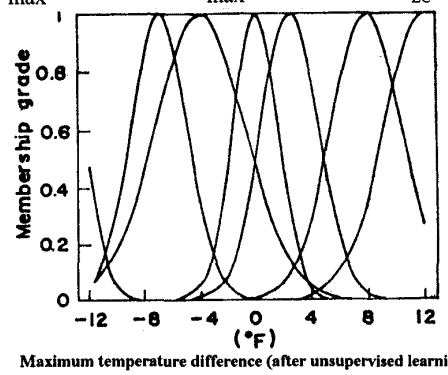
A backpropagation learning algorithm is also used for the FNN to provide a meaningful comparison with the Kalman filter. The peak load forecast ($P_{\max}$(k+n)) is obtained using equation (4) .

Table 1
The learned fuzzy logic rules for peak load forecasting in winter using FNN Term Sets

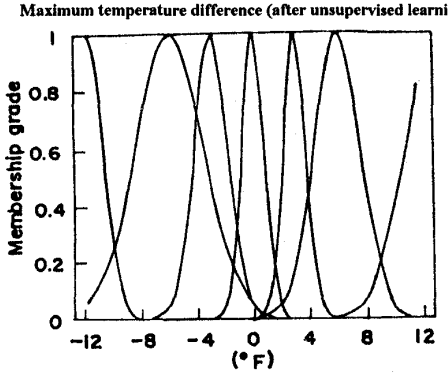
	Preconditions		Consequence
Rule	$\Delta\theta_{\max}(k,k+1)$	$\Delta H_{\max}(k,k+1)$	$e_{LC}(k)$
1	0	3	7
2	0	4	7
3	1	0	8
4	1	1	7
5	1	2	7
6	1	3	6
7	1	4	6
8	2	0	8
9	2	1	7
10	2	2	7
11	2	3	6
12	3	4	7
13	3	1	2
14	3	2	4
15	3	3	6
16	4	2	5
17	4	3	6
18	4	4	1
19	5	0	3
20	5	1	2
21	5	2	1
22	5	3	1
23	5	4	0
24	6	0	1
25	6	1	1
26	6	2	0

Table I gives the learned membership functions using FES for 24-hours ahead peak load forecasting in winter. For example, rule 1 is interpreted as :

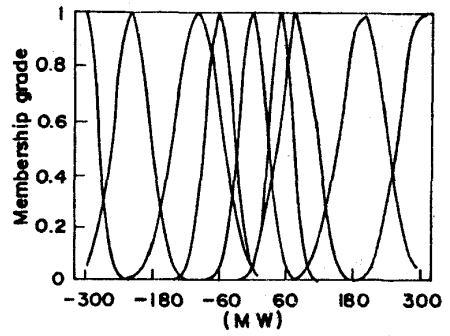
IF $\Delta\theta_{\max}$ is term 0 and $\Delta H_{\max}$ is term 3 THEN e_{LC} is term 7.



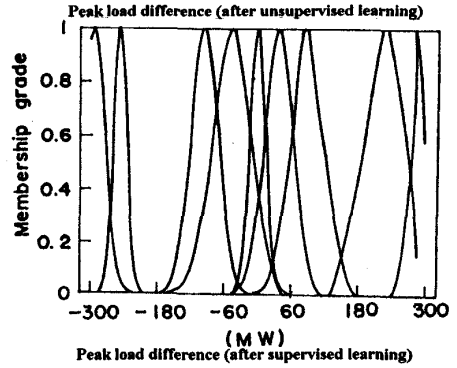
Maximum temperature difference (after unsupervised learning)



Maximum temperature difference (after supervised learning)



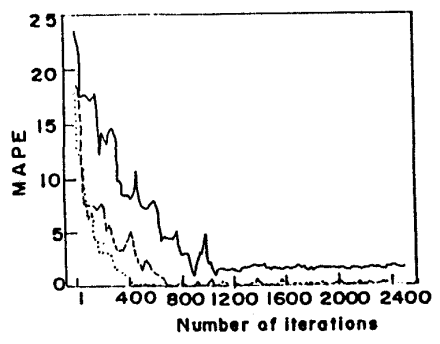
Peak load difference (after unsupervised learning)



Peak load difference (after supervised learning)

Fig.2 Learned membership functions for peak load forecasting in January using FES

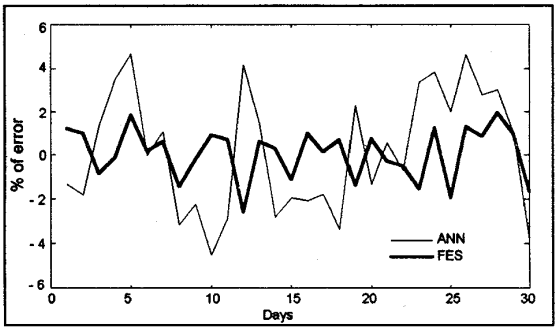
Figure 2 gives the graphs for the learned membership functions for maximum temperature and peak load differences after unsupervised and supervised learning phases for 24-hours ahead peak load forecasting in January. Figure 3 gives the MAPE versus the number of iterations for the three models. From Figure 3 we find that the Kalman filter based FES gives faster convergence in comparison to ANN and the FES using BP algorithm. The speed of convergence the FES is found to be superior because of the linear Kalman filter equations used for updating the weight vector and the error dependant forgetting factor are responsible for driving the PE low during the first few hundred iterations until the bias introduced by the initial conditions is eliminated.



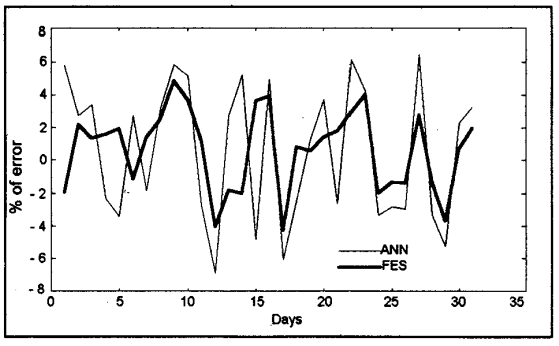
Peak load difference (after supervised learning)

ANN FNN(BP) FNN(KF)

Fig.3 Comparison of the mean absolute percentage error versus the iteration number for 24-hours ahead forecast in January



24 Hours Ahead Peak Load Forecast in June(summer)



168 Hours Ahead Peak Load Forecast in January(winter)

Fig.4 Comparison of PEs between ANN & FES

Fig.4 gives the peak load the percentage error (PEs), for ANN, FES models in the month of June using 24-hour ahead forecast. From the figure we see that the FES using Kalman filter algorithm gives better prediction and accuracy in comparison to the ANN approach. The same trend is observed when the lead time increases to 168 hours (one week ahead) in the month of January.

B. Average Load Forecasting

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

Input Pattern :

$P_{av}(k), \theta_{\max}(k), \theta_{\min}(k), H_{\max}(k), H_{\min}(k),$

$\theta^f_{\max}(k+n), \theta^f_{\min}(k+n), H^f_{\max}(k+n), H^f_{\min}(k+n)$

Output Pattern : P_{av}(k+1)

where, n is the lead time for the forecast as given in section 4.1.

For the FSE, the training patterns used are :

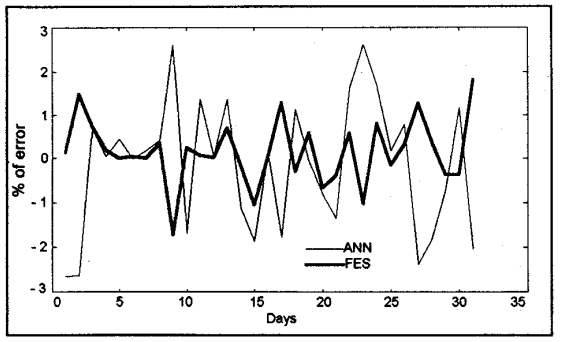
Input Pattern :

$\Delta\theta_{\max}(k,k+n), \Delta\theta_{\min}(k,k+n), \Delta H_{\max}(k,k+n), \Delta H_{\min}(k,k+n)$

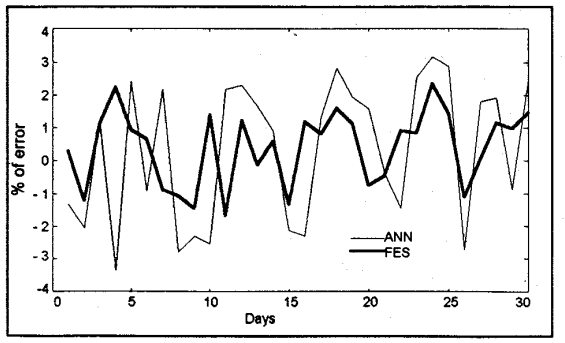
Output Pattern : $\bar{e}LC$(k), the desired load correction

Kalman filter parameters are kept the same as in the case of peak load forecasts.

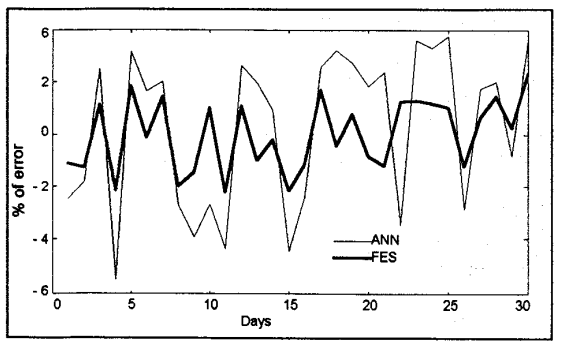
The average load $P_{av}(k+1)$ is obtained using equation(4).



24 Hours Ahead Average Load Forecast in January(winter)



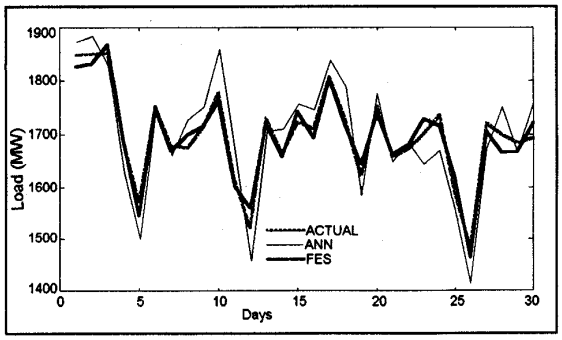
48 Hours Ahead Average Load Forecast in June(summer)



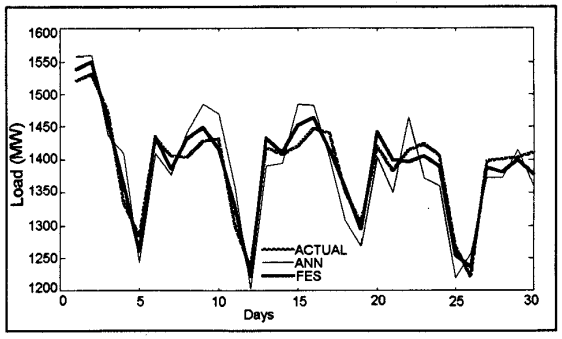
168 Hours Ahead Average Load Forecast in June(summer)

Fig.5 Comparison of PEs between ANN & FES

Fig.5 gives the average load forecasting PEs for the ANN, FES models in the months of January and June using 24-hours, 48 hours, and 168 hours ahead time frame. Again, from the figure we find the improved performance of FES in terms of faster convergence and improved overall accuracy in comparison to the ANN approach. Fig.6 gives the actual and forecasted Peak load for the Hybrid Neural Network and Fuzzy Expert System for the month of June using 24 hours and 168 hours ahead time frame.



24 Hours Ahead Actual vs Predicted Peak Load Forecast in June(summer)



168 Hours Ahead Actual vs Predicted Peak Load Forecast in June(summer)

Fig.6 Actual and forecasted Peak load for the Hybrid Neural Network and Fuzzy Expert System for the month of June

V. Discussion

The proposed hybrid fuzzy neural network model uses the differential weather parameters as input unlike the standard ANN approach using directly the above parameters.

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.

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 this application. Also the choice of membership function is flexible enough 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 can be obtained.

Extremely short-term predictions from 1 to 6-hour ahead over the next 24-hour period using the hybrid models have been also performed and the results reveal significant improvement in accuracy in comparison to 24-hour ahead forecasts. The main features and advantages of the hybrid model are : 1) it provides us with a general method to combine available numerical information and human linguistic information into a common framework; 2) it requires much less construction time than a comparable neural network; 3) significant accuracy in predicting chaotic time series models.

VI. Conclusions

This paper presents a expert system modeled fuzzy neural network a hybrid neural network model for time series forecasting of electric load. The new load forecasting model using FES introduces the low-level learning power of artificial neural network into a fuzzy expert system and provides 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 FES. Also the use of linear Kalman filter update equations in the supervised learning phase of the FES gives better convergence and forecasting accuracy over the gradient-descent backpropagation algorithm. One day and one week ahead forecasting results validate the effectiveness of the FES model.

VII. Acknowledgements

The authors gratefully acknowledge the funds from the National Science Foundation (NSF Grant No.INT-9209103 and INT-9117624), USA for undertaking this research.

VII. References

[1] Gross, George and Galiana F D, "Short-Term Load Forecasting", Proc. of the IEEE, Vol.75, No.12 (1987) pp.1558-1573

[2] M C Brace, "A comparison of the forecasting accuracy of neural networks with other established techniques", First Int. Forum on Applications of neural networks to Power Systems, Seattle, WA (July, 1991) pp.23-26.

[3] T M Peng, N F Hubele and G G Karady, "Advancement in the application of neural networks for short term load forecasting", IEEE Trans. Power Systems, Vol.7, No.1 (1992) pp.250-258.

[4] P K Dash, S Dash, G Ramakrishna and S Rahman, "Forecasting of a load time series using a fuzzy expert system and fuzzy neural networks", Engg. Int. Syst, Vol.1, No.2 (1993), pp.103-117.

[5] K.K.Kim, D.Y.Park and J.K.Park, "A hybrid model of Artificial network and fuzzy expert system for short-term load forecast", Proc. ESAP-IV, Melbourne, (1993), pp.169-175.

[6] Chin-Teng Lin, C S George Lee, "Neural network based fuzzy logic control and decision system", IEEE Trans. Computers, Vol.40, No.12 (1991), pp.1320-1336.

[7] P K Dash, J K Satapathy, G Ramakrishna and S Rahman, "An improved kalman filter based neural network approach for short-term load forecasting", Proc. of the Third Int. Symp. on Electricity Distribution and energy Management, Singapore, Vol. 1 (1993), pp.3-8.

[8] Robert S Scalero and Nazif Tepedelenliouglu, "A fast algorithm for training feedforward neural networks", IEEE Trans. on Signal Processing, Vol.40, No.1(1992), pp.202-210.