

One Hour Ahead Load Forecasting Using Artificial Neural Network for the Western Area of Saudi Arabia

A. J. Al-Shareef, E. A. Mohamed, and E. Al-Judaibi

Abstract—Load forecasting has become in recent years one of the major areas of research in electrical engineering. Most traditional forecasting models and artificial intelligence neural network techniques have been tried out in this task. Artificial neural networks (ANN) have lately received much attention, and a great number of papers have reported successful experiments and practical tests. This article presents the development of an ANN-based short-term load forecasting model with improved generalization technique for the Regional Power Control Center of Saudi Electricity Company, Western Operation Area (SEC-WOA). The proposed ANN is trained with weather-related data and historical electric load-related data using the data from the calendar years 2001, 2002, 2003, and 2004 for training. The model tested for one week at five different seasons, typically, winter, spring, summer, Ramadan and fall seasons, and the mean absolute average error for one hour-ahead load forecasting found 1.12%.

Keywords—Artificial neural networks, short-term load forecasting, back propagation.

I. INTRODUCTION

LOAD forecasting has become in recent years one of the major areas of research in electrical engineering. Load forecasting is however a difficult task. First, because the load series is complex and exhibits several levels of seasonality. Second, the load at a given hour is dependent not only on the load at the previous hour, but also on the load at the same hour on the previous day and because there are many important exogenous variables that must be considered, specially the weather-related variables [1].

Load forecasting plays an important role in power system planning and operation. Basic operating functions such as unit commitment, economic dispatch, fuel scheduling and unit maintenance, can be performed efficiently with an accurate forecast [2, 3-4].

Various statistical forecasting techniques have been applied to short term load forecasting (STLF). Examples of such methods including, Time series [5 -6], Similar-day approach [7], Regression methods [5 &8], expert systems [7, 9-10]. In general, these methods are basically linear models and the load pattern is usually a nonlinear function of the exogenous

variables [1]. On the other hand, ANN has been proved as powerful alternative for STLF that it is not rely on human experience. It has been formally demonstrated that ANN's are able to approximate numerically any continuous function to the desired accuracy and it should be expected to model complex nonlinear relationships much better than the traditional linear models that still form the core of the forecaster's methodology. Also, ANN is data-driven method, in the sense that it is not necessary for the researcher to postulate tentative models and then estimate their parameters. Given a sample of input and output vectors, ANN is able to automatically map the relationship [1, 11 &12].

This paper presents an ANN model of STLF typically, one hour-ahead, for the western area of Saudi Arabia. Time, weather, and load related inputs are considered in this model. Four years of historical dependent data were used. A special design for the forecasting system that cope with the features of Saudi Arabia (SEC-WOA) electrical load, is presented

This article is organized as follow: Section II gives an introduction to ANN in general. Section III shows the load feature of SEC-WOA. The proposed ANN model is described in Section IV. Test results in Section V. Conclusion are drawn in Section VI.

II. INTRODUCTION TO ANN

ANN's are mathematical tools originally inspired by the way the human brain processes information. They are composed of simple elements operating in parallel. These elements are stimulated by biological uneasy systems. As in nature, the network function is determined largely by the connections between elements. You can train an ANN to perform a particular function by adjusting the values of the connections (weights) between elements. Commonly ANN's are adjusted, or trained, so that a particular input leads to a

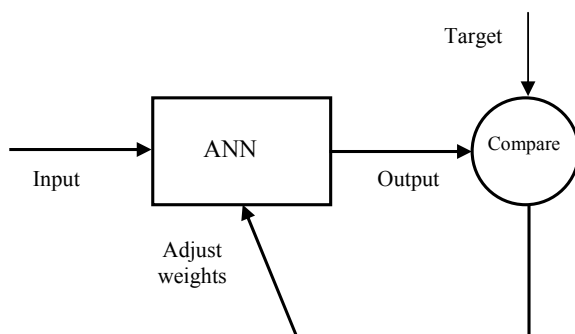


Fig. 1 Simple ANN concept

A. J. Al-Shareef is with King Abdul-Aziz University, Kingdom of Saudi Arabia.

E. A. Mohamed is with Qassim University, Kingdom of Saudi Arabia.

E. Al-Judaibi is with SEC-WOA, Jeddah, Kingdom of Saudi Arabia.

specific target output. Therefore, the network is adjusted, based on a comparison of the output and the target, until the network output matches the target (see Fig. 1). Typically many such input/target pairs are needed to train a network.

Their basic unit is the artificial *neuron*. The neuron receives (numerical) information through a number of input nodes, processes it internally, and puts out a response. The processing is usually done in two stages: first, the input values are linearly combined, and then the result is used as the argument of a nonlinear *activation function*. The combination uses the *weights* attributed to each connection, and a constant *bias* term. Fig. 2 shows one of the most used schemes for a neuron.

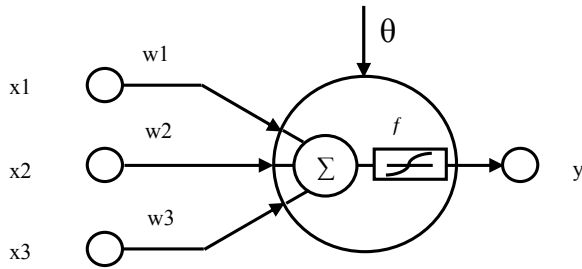


Fig. 2 An artificial neuron

The neuron output is given by:

$$y = f(\sum w_i \cdot x_i - \theta) \quad (1)$$

Where x_i is the neuron input multiplied by weight link w_i , θ is the characteristic neuron offset (bias), and f is the activation function. The most common choice for the activation function in multilayer networks is tan-sigmoid activation function.

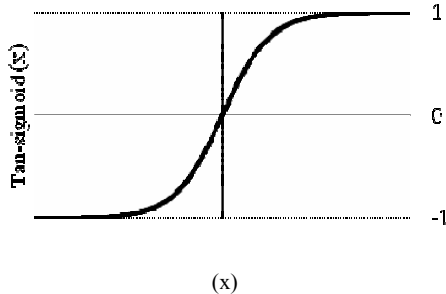


Fig. 3 Tan-sigmoid activation function

As shown in Fig. 3, the tan-sigmoid activation function output is limited between $[-1, 1]$, the output of the function is formulated in (2).

$$f(x) = \frac{2}{1 + \exp(-2x)} - 1 \quad (2)$$

The neurons are organized in a way that defines the network structure. The most concerned structure is the *multilayer perceptron* (MLP) type, in which the neurons are organized in layers. The neurons in each layer may share the same inputs, but are not connected to each other. If the architecture is *feed-forward*, the outputs of one layer are used as the inputs to the following layer. The layers between the input neurons and the output layer are called the *hidden layer*. Fig. 4 shows an

example of a two-layers feed-forward perceptron network, with four input neuron, three neurons in the hidden layer and one neuron in the output layer. Each layer has a specified number of nodes; the interconnections are only between neurons of adjacent layers, and each neuron belonging to a layer is connected to all the neurons of adjacent layers. Note that ANN may contain more than one hidden layer; the number of neuron in each layer should be carefully selected depends on the application requirements.

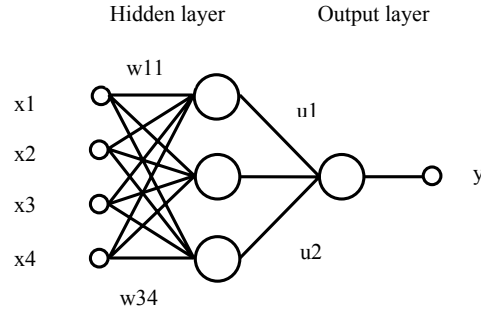


Fig. 4 A two-layer feed-forward neural network

The parameters of this network are the weight matrix, and the bias. The estimation of the parameters is called the training of the network. The most used training algorithm in load forecasting is *back-propagation* one, There are many variations of the back-propagation algorithm. The simplest implementation of back-propagation learning updates the network weights and biases in the direction in which the performance function decreases most rapidly, the negative of the gradient. An iteration of this algorithm can be written in (3).

$$x_{k+1} = x_k - \alpha_k g_k \quad (3)$$

Where x_k is a vector of current weights and biases, g_k is the current gradient, and α_k is the learning rate.

In load forecasting applications, this basic form of *multilayer feed-forward* architecture is still the most popular. Nevertheless, there are a large number of other designs, which might be suitable for other applications. ANN should prove to be particularly useful when one has a large amount of data, but little a prior knowledge about the laws that manage the system that generated the data. [1, 13-14]

III. LOAD FEATURE OF SEC-WOA

Western operational area of Saudi Electricity Company is covering very important cites with special features. It includes the two holy mosques in Makah and Al-Madina; where Muslims come from all over the world to perform religious tasks. Beside, the most economical and tourism city like Jeddah and other small cites such as Taif and Yanbu, Fig. 5 represent the main network of WOA. There are many factors affecting the hourly load of this area, which makes the hourly forecasting unique and challenging.

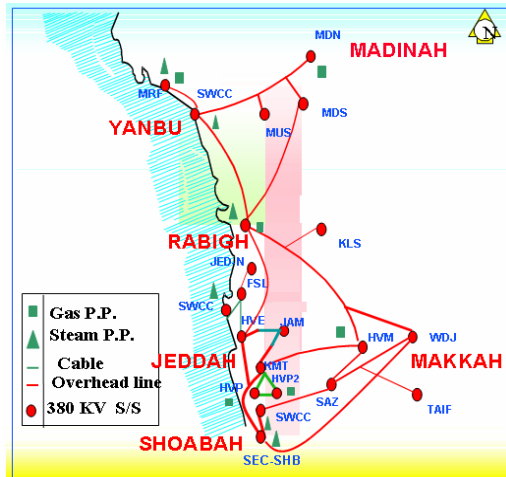


Fig. 5 The main network of WOA

The weather changes greatly affect the load demand due to a huge Air conditioning load in the system. Fig. 6 shows a linearity relationship between system daily peak load and related temperature for the year 2004. Another important factor is the time of the day, as social life and activities of the consumers depends on the time of the day such as; working and schools hours and prayer times, when all public and commercial activities stop five times a day to perform prayers. In addition to the seasonal load behavior factor, which reflect how load draws a changeable load profile, because the impact of seasonality. The effect of weekends and holidays is essential, as well as the large religious events such as month of Ramadan (month of feast) and Hajj. These events, based on the lunar calendar that shifts eleven days a year, will cause unsimilarity in load conditions every year.

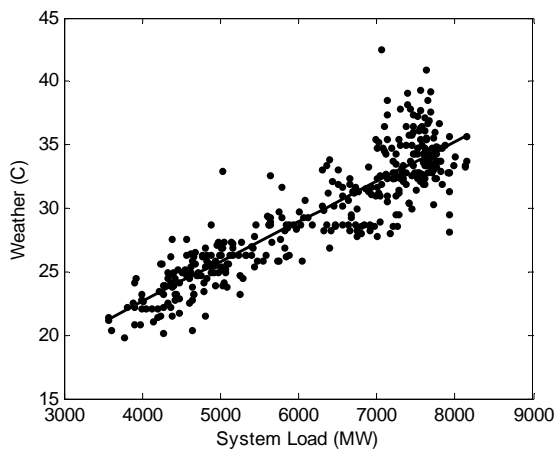


Fig. 6 Relationship between temperature and system daily peak load of 2004

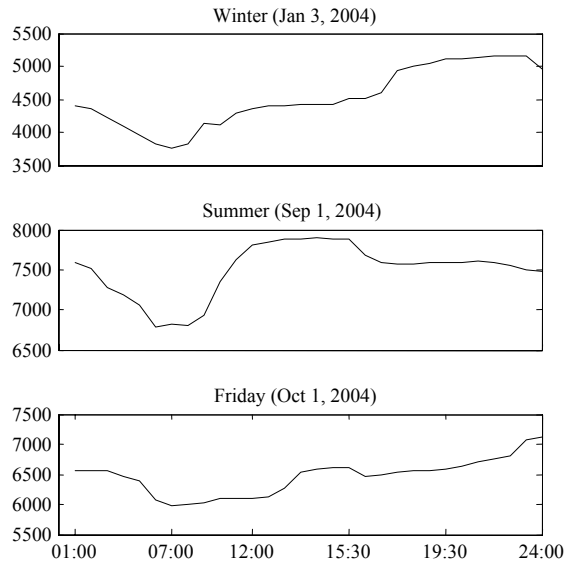


Fig. 7 Examples of daily load consumption profile of a typical Friday, winter and summer days

Examples of a daily load consumption profile of a typical Friday, winter and summer days are shown in Fig. 7. The difference between winter and summer profiles is clear; the effect of hot weather is reflected on the great amount of load consumption at afternoon in the summer day. At Friday a sudden increase in the load demand afternoon is due to Juma'a prayer, and as weekend the load is stable at morning.

IV. THE PROPOSED ANN MODEL

The proposed ANN model is a two layers feed-forward ANN, has eight inputs, one hidden layer with ten neurons, and one neuron in the output layer, with tan-sigmoid activation function in the hidden layer, and linear function in the output layer.

A. Input Vector Configuration

The input vector to the ANN specifies the time and weather data for the hour to be forecasted, and associated historical load data. Typically, it is configured from month, year, hour, day type, temperature, relative humidity and historical load for the previous two hours and the same hour of the last day. Table I depicts the input vector configuration.

As indicated in Table I, numerical indexes were given to represent the inputs for the forecasted hour. Indexes of [1:12] were given to represent the month, [1:24] to represent the hour and [1:7] to represent the day type starting from Saturday to Friday respectively. Regarding to religious events such as Ramadan and Hajj, have a great influence in load profile, an index is added to represent them. Moreover, we have considered half-hourly data for the high load variation periods of the day, typically 13:30, 14:30, 15:30, 18:30, 19:30 and 20:30 which were represented by the fractions (13.5, 14.5, 15.5, 18.5, 19.5 and 20.5) respectively.

TABLE I
INPUT VECTOR CONFIGURATION

Input	Description
Month	1..12
Year	2001...2004
Hour	1..24
Day type	1..7
Temperature	Forecasted temp
Humidity	Forecasted relative humidity
Special events index	0..1..
Historical load	Passed tow hours load Passed 24 hours load

B. Data Range

Since the variety of load shows similarity on the same period each year. Therefore, the limits of one month passing the forecasting day, and two months before and after the same day from each of the previous three years were selected as input data range for the training set (Fig. 8). [3, 4-15]

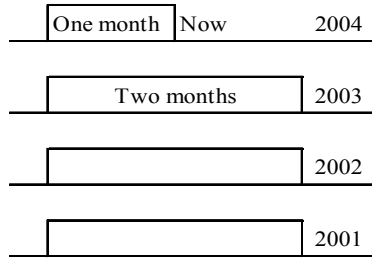


Fig. 8 Limits of Input data range for the training set

C. Hidden Layer

In the proposed ANN model, it is found that one hidden layer is enough for the learning of the network. In fact, networks with more than one hidden layer are more complex and time consuming. The number of neurons in the hidden layer has the greatest effect in the network performance. If there is no enough hidden neurons, the network will find difficulty in the learning. On the other hand, the more hidden neurons, the more capability network to over-fit the trained data. In our experiments, varying the number of hidden neurons has leaded us to the set with the smallest error. Typically, eight neurons in the hidden layer were chosen.

D. Data Preprocessing

Before training, it is useful to scale all the inputs and targets so that they always fall within a specified range. Equation (4) is used for each input and targets independently.

$$y_s = \frac{(x_{\max} - x_{\min}) (x - x_{\min})}{x_{\max} - x_{\min}} + y_{\min} \quad (4)$$

Where y_s is the scaled data element, x is the original data element for each input and target vectors, x_{\max} and x_{\min} are

the maximum and minimum corresponding data element respectively. Due to nature of the sigmoid function, the outputs of the neurons fall in the interval of -1 and +1. Therefore, y_{\max} and y_{\min} are set to 1 and -1, respectively. [13]

E. Training

Back propagation algorithm with automated regularization and fixed learning rate is used in this work. The learning rate is multiplied times the negative of the gradient to determine the changes to the weights and biases. If the learning rate is made large, the algorithm becomes unstable. If the learning rate is set small, the algorithm takes a long time to converge. In this work the learning rate is fixed to 0.05, which show a good performance.

The main problem in the training is called *over-fitting*. The error of training becomes very small, but the network will memorize the input data. The network will not be generalized for new test data, and will show a large error when it is tested. To improve the generalization of the network, an *automated regularization* technique [13] is developed in this model. Typically, the objective of the training process is to reduce the sum of squared error:

$$E_d = \sum_{i=1}^n (t_i - a_i)^2 \quad (5)$$

Where t_i is the target value, and a_i is the network output. However, additional term is added for the regularization, the objective function becomes:

$$f = \beta E_d + \alpha E_w \quad (6)$$

Where E_w is the sum of the squared error of the network weights, and α and β are the objective function parameters. This performance function causes the network to have smaller weights and biases, which make the network response to be smoother and less likely to overfit. One approach to determine the optimal regularization function parameters in an automated fashion is the Bayesian framework [16, 17].

V. TEST RESULTS

The performance of the developed model for short-term load forecasting has been tested for one week in five different seasons using the actual load and weather data (for the year 2001 to 2004) for the training. Data was provided by SEC-WOA (Regional Control Center) and MATLAB 7.4.0 tools, was used for the numerical simulations.

The load forecast was compared to the actual load data and the error is calculated. The mean absolute percentage error (MAPE) is used to evaluate the performance of these models, it is defined as:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|t_i - a_i|}{t_i} \times 100 \quad (7)$$

Where t_i is the actual load, a_i is the forecasted load, n is the number of data points.

The proposed STLF model is to be tested for one week in five different seasons, typically, winter, spring, summer, Ramadan and fall seasons. The obtained training performance of the fall season is shown in Fig. 9:

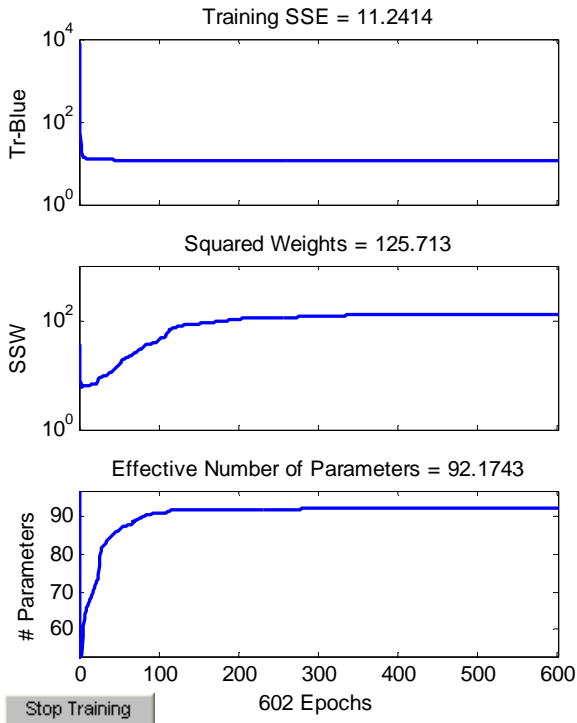


Fig. 9 Training process

The test results are represented in Table II, MAPE for each hour on each day are shown. Comparison of the ANN model output by the actual load represented in Fig. 10 which observes satisfactory forecasting results.

TABLE II
TEST RESULTS (MAPE)

Season	Date	MAPE
Winter	Feb 21- Feb 27	1.23%
Spring	May 8- May 14	1.29%
Summer	Aug 7- Aug 13	0.81%
Ramadan	Oct 23- Oct 29	1.14%
Fall	Nov 27- Dec 3	1.53%
Average		1.12%

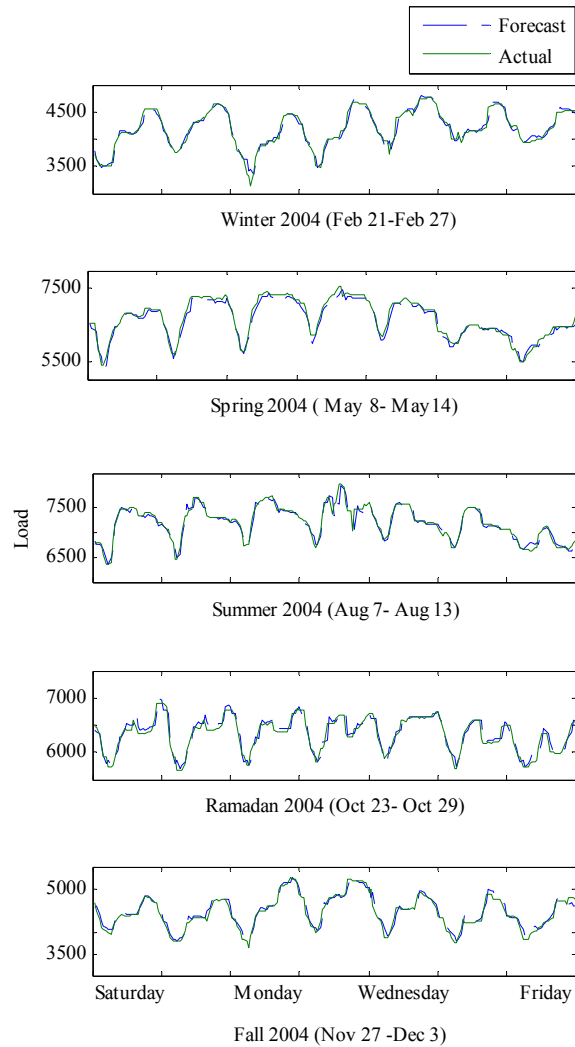


Fig. 10 Comparison of the ANN Model Output by the Actual Load

VI. CONCLUSION

This paper presents a new ANN based STLF application on the load of SEC-WOA network, in the Kingdom of Saudi Arabia. This network has a load pattern with special features. These features are to cope with the special religious activities of the Kingdom, such as Ramadan and Hijj. Moreover, the load pattern is much affected by the time schedule of the five prayers a day. A seasonal ANN structure is also adopted to handle the major contribution of the temperature changes with the four seasons a year. A simple ANN structure based on the back-propagation and the automated regularization is firstly implemented and secondly evaluated with different scenarios. The mean absolute percentage error (MAPE) for one hour-ahead load forecasting was 1.12%.

Further development of ANN models should include more weather data such as rainfall, wind speed and direction and sky condition will have great effect of the load.

Holy cities of Makkah and Madinah host millions of Muslims in different seasons during the year to perform Umra

and other religious activities. In fact, the religious tourism influences the system load profile. Therefore, tracking these seasons every year and assigning special index for them will enhance the forecast accuracy.

ACKNOWLEDGMENT

The authors would like to thank SEC-WOA Co, System operation and control department-west, for providing the electrical load and weather data used in this research.

REFERENCES

- [1] Henrique Steinhert Hippert, Carlos Eduardo Pedreira, and Reinaldo Castro Souza. "Neural Networks for Short-Term Load Forecasting: A Review and Evaluation". IEEE Transactions on Power Systems, Vol. 16, No. 1, February 2001.
- [2] O.A. Alsayegh. "Short-Term Load Forecasting Using Seasonal Artificial Neural Networks". International Journal of Power and Energy Systems, Vol. 23, No. 3, 2003.
- [3] Tomonobu Senjyu, Hitoshi Takara, Katsumi Uezato, and Toshihisa Funabashi. "One Hour-Ahead Load Forecasting Using Neural Network", IEEE Transactions on power systems, Vol. 17, NO. 1, February 2002.
- [4] A.G.Baklitzis, V.Petridis, S.J.Klartzis, M.C.Alexiadis, A.H.Malssis, "A Neural Network Short Term Load Forecasting Model for the Greek Power System", IEEE Transactions on power systems, vol.11, No2, May 1996.
- [5] Ibrahim Moghram, Saifure Rahman, "Analysis Evaluation of Five Short Term Load Forecasting Techniques", IEEE Transactions on power systems, VOL. 4, NO.4, 1989.
- [6] K.Lru, S.Subbarayan, R.R.Shoults, M.T.Manry, C.Kwan, F.L.lewis, J.Naccarino, "Comparison of very short term load forecasting techniques". IEEE Transactions on power systems, VOL. 11, NO.2, 1996.
- [7] Eugene A. Feinberg, Dora Genethliou. "Applied Mathematics for Power Systems: Load Forecasting".
- [8] Alex D, Timothy C. "A Regression-Based Approach to Short Term System Load Forecasting". IEEE transaction on Power Systems,5,4,1535-1550,1990.
- [9] S. Rahman, O. Hazim, "A Generalized Knowledge-Based Short-Term Load Forecasting Technique", IEEE Transactions on power systems, Vol. 8, No. 2, May 1993.
- [10] Kun-Long Ho, Yuan-Yih Hsu, Chih-Chien Liang, Tsau-Shin Lai "Short-Term Load Forecasting of Taiwan Power System Using A Knowledge-Based Expert Systems", IEEE Transactions' on power systems, Vol. 5, No. 4, Nov 1990.
- [11] H.Chen, C.A. Canizares, and A. Singh. "ANN-Based Short-Term Load Forecasting in Electricity Markets". Proceedings of the IEEE Power Engineering Society Transmission and Distribution Conference,2:411-415, 2001.
- [12] D.C. Park, M.A.Elsharqawi, R.J.Marks II. "ELECTRIC Load Forecasting using Artificial Neural Network", IEEE Transaction on Power System, 6, 1991.
- [13] Hagan, M.T., H.B. Demuth, and M.H. Beale, Neural Network Design, Boston, MA: PWS Publishing, 1996.
- [14] Simon Haykin. "Neural Network A Comprehensive Foundation". Prentice Hall International, Second edition.
- [15] Tomonobu Senjyu, Paras Mandal, Katsumi Uezato, and Toshihisa Funabashi. "Next Day Load Curve Forecasting, Using Hybrid Correction Method", IEEE Transactions on power systems, Vol. 20, No. 1, February 2005.
- [16] D.J.C.MacKay, "Bayesian Interpolation," Neural Computation, vol. 4, pp415-447, 1992.
- [17] Dan Foresee, F. Hagan, M.T." Gauss-Newton approximation to Bayesian learning," International Conference on Neural Networks, vol.3, pp1930-1935, 1997.