Stochastic Modeling Compared With Artificial Intelligence Based Approach for Short Term Wind Speed Forecasting

E .M. Abd El-Gawad **, M.A. Mustafa Hassan*† , M. A. M. Hallouda*, O.Y. Abul-Haggag*

* Elec. Power Dept, Faculty of Engineering, Cairo University, Egypt.
**Elec. Power and Machines Dept, Faculty of Engineering, Kafr Elsheikh University, Egypt
† Corresponding E-Mail: mmustafa@eng.cu.edu.eg

***Abstract*:** The sophisticated Application of Artificial Intelligent Approaches was introduced recently in renewable energy in electric power systems. However, these approaches started with introducing Fuzzy Logic (FL) in the last decades of the last century. Furthermore, Artificial Neural Network (ANN) was introduced to solve many problems in electric power systems. Among these problems is forecasting of wind speed. In this proposed article, the application of Adaptive Neuro-Fuzzy Inference System (ANFIS) is used to forecast the coming speed of wind using real data of the past. The ANFIS can be viewed as a combination of fuzzy system and neural network or fuzzy neural network. This paper aims; firstly, to forecast the average value of wind speed via some well known method. Secondly compare between these different method like Autoregressive Integrated Moving Average (ARIMA), Autoregressive Moving Average form (ARMA), Autoregressive Form (AR). The goal of these methods is to search for the best one compared to Adaptive Neuro Fuzzy Inference System (ANFIS).

[E .M. Abd El-Gawad, M.A. Mustafa Hassan, M. A. M. Hallouda, O.Y. Abul-Haggag. **Stochastic Modeling Compared With Artificial Intelligence Based Approach for Short Term Wind Speed Forecasting.** Journal of American Science 2011;7(4):393-399]. (ISSN: 1545-1003). http://www.americanscience.org.

Keywords: Adaptive Neuro Fuzzy Inference System (ANFIS), Autoregressive Integrated Moving Average form (ARIMA), Autoregressive Moving Average form (ARMA), Autoregressive Form (AR), Short-Term Load Forecasting.

1. INTRODUCTION

Short-Term Forecasting (STF) plays an important role in power systems. Accurate short-term load forecasting has a significant influence on the operational efficiency of a power system, such as unit commitment, annual hydro-thermal maintenance scheduling hydro-thermal coordination, demand side management, interchange evaluation, security assessment and others. The recent developments in the areas of Renewable Energies are promising spots in power systems. Improvements in the accuracy of short-term load forecasts can result in significant financial savings for utilities and co-generators. Various forecasting techniques have been proposed in the last few decades. These models include: time series [2,3], Multiple Linear Regression [4] and Auto Regressive Moving Average (ARMA) [5]. Wind speed were forecasted by the stochastic modelling [6]. However, these result in difficulty in taking system variation into account as the rules are fixed. They do not have the ability to adapt dynamically to the system operating conditions, and to make correct decisions if the signals are uncertain. Recently, intelligent soft computational techniques such as Artificial Neural Network (ANN), Fuzzy Inference System (FIS) [1] and (ANFIS) can model superiority of human knowledge features. They also reestablish the process without plenty of analysis. Thus these techniques are attracting great attention in an environment that is obvious with the absence of a simple and well-defined mathematical model. Autocorrelation and Cross correlation, both are used in signals and systems analysis. The concept of autocorrelation and cross correlation play an important role. The autocorrelation function of a random signal describes the general dependence of the values of the samples at one time on the values of the samples at another time.

2. STOCHASTIC MODELLING

A statistical phenomenon that evolves in time according to probabilistic laws is called a stochastic process. A Stochastic “random” process is a collection, or “ensemble” of functions of time, which might be observed on any trial of an experiment. The ensemble may consist of finite or infinite number of functions. It has been observed that unique patterns of energy and demand pertaining to fast-growing areas are difficult to analyse and predict by direct application of time-series methods. However, these methods appear to be among the most popular approaches that have been applied and are still being applied to STF. Using the time-series approach, a model is first developed based on the previous data, then future speed is predicted based on this model. The Stochastic Models "Time Series Models" are used in three main areas of application:

a) Forecasting.
b) T.F. determination from input – output data.
c) Stochastic Controller design.

Some of Stochastic Models used in this study will be presented briefly in the next subsections.

A. *Autoregressive Form (AR):*

An autoregressive model (AR) is also known in the filter design industry as an infinite impulse response filter (IIR) or an all pole filter, and is sometimes known as a maximum entropy model in physics applications. There is "memory" or feedback and therefore the system can generate internal dynamics. However, it has a noise term or residue, which is almost always assumed to be Gaussian white noise. Verbally, the current term of the series can be estimated by a linear weighted sum of previous terms in the series. The weights are the auto regression coefficients. The problem in AR analysis is to derive the "best" values for these coefficients given the series. The majority of methods assume the series is linear and stationary. By convention the series is assumed to be zero mean, if not this is simply introducing another term a_0 in front of the summation.

B. *AutoRegressive Moving Average form (ARMA):*

Taking the AR model and the MA model, will produce the ARMA model. The notation ARMA(p, q) refers to a model with p autoregressive terms and q moving average terms This model subsumes the AR and MA models,

C. *Autoregressive Integrated Moving Average (ARIMA):*

The theoretical behaviour of ARIMA processes [7], will be discussed how to use ARIMA models to observed time series data and make forecasts. Before beginning this work, an obvious question needs to be answered. Why should we assume that some random time series can be adequately modelled by an ARIMA process**.** Many time series are no stationary. The only kind of non stationary supported by the ARIMA model is simple differencing of degree d. In practice, one or two levels of differencing are often enough to reduce a no stationary time series to apparent stationary. The following procedure will be used to model a time series as an ARIMA process and produce future forecasts**:**

1) Identify the appropriate degree of differencing d by differencing the time series until appears to be stationary.
2) Remove any nonzero mean from the differenced time series.
3) Estimate the autocorrelation and PACF of the differenced zero mean time series. Use these to determine the autoregressive order p and the moving average order q.
4) Estimate the coefficients φ_1, …….:, φ_p, $\theta 1$, ….., θ q. This can be done in a variety of ways. One simple approach is to make sure that the resulting autocorrelations match the observed autocorrelations. However, a more robust method is maximum likelihood estimation.
5) Once the model has been fitted, future forecasts could be produced with associated uncertainties.

By SAS package and using Levenberg-Marquardt (LM) method, some tests are done. Other names might be heard of in the arsenal of optimization include Steepest-Descent method (SD), Conjugate-Gradient (CG) method, Newton's Method, Gauss-Newton method and many others. LM is in nature an improvement Gauss-Newton method by incorporating SD into the iterative update scheme. Steepest-Descent method is the most straightforward method in optimization. By computing the gradient direction followed by a 1D search, SD iteratively approaches the minimum point of the object function in parameter space. Mathematically, SD can be expressed as follows:

$$x^{k+1} = x^k - \lambda^k \nabla F(x^k) \qquad \textbf{(1)}$$

where $\lambda^k = \text{argmin } F(x^k - \lambda^k \nabla F(x^k))$. Since only the

first order derivative information is used, SD suffers from the slow convergence. However, it is relatively robust even if the initial guess is far away from the true value. Newton's method goes one step further than SD: in the Taylor's expansion of the object function at the current point, the second order derivative term is now included to compute for the update:

$$x^{k+1} = x^k - [H_F(x^k)]^{-1}\nabla F(x^k) \qquad \textbf{(2)}$$

$H_F(x^k)$ in (2) is the Hessian matrix of function F(x) at x^k denoting the second order derivative. Newton's method converges faster than SD. The price to pay is the reduction in robustness, i.e. it is much more sensitive to initial guess than SD. Another drawback is the requirement of computing Hessian matrix H which could be a big issue in many applications where the analytical form of F(x) is not available. For a specific set of optimization problems - least-square optimization, i.e.

$$\mathrm{Min}F(x)=\|f(x)\|^2_2=f^T(x)f(x) \qquad \textbf{(3)}$$

Gauss-Newton (GN) method is more frequently used. GN is a modified Newton's method by replacing the Hessian matrix [HF (x^k)] by the multiplication of two first order derivative (Jacobian matrix) of function f, so the "pseudo- Hessian" matrix has the form of *[Jf (xk)T Jf (xk)]*. The updating equation is now written as:

$$x^{k+1}=x^k-[J_f(x^k)^TJ_f(x^k)]^{-1}J_f(x^k)^Tf(x) \qquad \textbf{(4)}$$

or

$$[J_f(x^k)^TJ_f(x^k)](\Delta x)^k=-J_f(x^k)^Tf(x) \qquad \textbf{(5)}$$

In GN, one only needs to computer $J_f(x^k)$ which, therefore, leads to the savings in computation. However, it sacrifices the convergence rate (GN has 1-order convergence instead of 2-order as in Newton's method). Both Newton's method and Gauss-Newton method demonstrate oscillatory features during iterations and are not as robust as SD. Levenberg (1944) [8] and Marquardt (1963) [9] provided an hybrid technique of GN and SD. They introduced a steering factor _ to switch between the GN direction and SD direction. The update equation in LM is:

$$[J_f(x^k)^TJ_f(x^k)+\lambda I](\Delta x)^k=-J_f(x^k)^Tf(x) \qquad \textbf{(6)}$$

When $\lambda \rightarrow 0$, LM method is reduced to GN. When $\lambda \rightarrow \infty$, LM approaches SD method. The values of λ during the iterative process are chosen in the following way: at the beginning of the iterations, λ is set to a large value, so the LM method manifests the robustness of SD and the initial guess can be chosen with less caution. In each iteration, if $F(x^k+\Delta x^k) < F(x^{k-1}+\Delta x^{k-1})$, decrease λ by certain amount (say divided by 2) to speed up the convergence; otherwise, increase λ value to enlarge the searching area (trust-region).It has been proven that LM is equivalent to a Gauss-Newton minimization under a inequality constrain.

3. MODELING USING ANFIS

In general, a Fuzzy Logic System (FLS) can be viewed as a non-linear mapping from the input space to the output space. An FLS consists of five main components: fuzzy sets, fuzzifiers, fuzzy rules, an inference engine, and defuzzifiers [6]. However, Fuzzy inference system is limited in its application to only modelling ill defined systems. These systems have rule structure which is essentially predetermined by the user's interpretation of the characteristics of the variables in the model. It has been considered only fixed membership functions that were chosen arbitrarily. However, in some modelling situations, it cannot be discerned what the membership functions should look like simply from looking at data. Rather than choosing the parameters associated with a given membership function arbitrarily, these parameters could be chosen so as to tailor the membership functions to the input/output data in order to account for these types of variations in the data values. In such case the necessity of the Adaptive Neuro Fuzzy Inference System (ANFIS) becomes obvious. Adaptive Neuro-fuzzy networks are enhanced FLSs with learning, generalization, and adaptive capabilities. These networks encode the fuzzy if-then rules into a neural network-like structure and then use appropriate learning algorithms to minimize the output error based on the training/validation data sets. Neuro-adaptive learning techniques provide a method for the fuzzy modelling procedure to learn information about a data set. It computes the membership function parameters that best allow the associated fuzzy inference system to track the given input/output data. A network-type structure similar to that of a Artificial Neural Network (ANN) can be used to interpret the input/output map. Therefore, it maps inputs through input membership functions and associated parameters, and then through output membership functions and associated parameters to outputs,. The parameters associated with the membership functions changes through the learning process. The computation of these parameters (or their adjustment) is facilitated by a gradient vector. This gradient vector provides a measure of how well the fuzzy inference system is modelling the input/output data for a given set of parameters. When the gradient vector is obtained, any of several optimization routines can be applied in order to adjust the parameters to reduce some error criteria. This error criterion is usually defined by the sum of the squared difference between actual and desired outputs. ANFIS in the MATLAB program uses a combination of least squares estimation and back propagation for membership function parameter estimation. Furthermore the used ANFIS is assumed to have the following properties [10]:

- It is zero th order Sugeno-type system.
- It has a single output, obtained using weighted average defuzzification. All output membership functions are constant.
- It has no rule sharing. Different rules do not share the same output membership function, namely the number of output membership functions must be equal to the number of rules.
- It has unity weight for each rule.

Figure (1) shows the architecture of the ANN, while Figure (2) shows the architecture of the ANFIS. ANFIS is comprising by input, fuzzification , inference and defuzzification layers. The network can be visualized as consisting of inputs, with N neurons in the input layer and F input membership functions for each input, with F*N neurons in the fuzzification layer. There are F^N rules with F^N neurons in the inference and defuzzification layers. It is assumed one neuron in the output layer. For simplicity, it is assumed that the fuzzy inference system under consideration has two inputs x and y and one output z as shown in Figure (1). For a zero-order Sugeno fuzzy model, a common rule set with two fuzzy if-then rules is the following Rule Set:

IF (x is A) AND (x is B) THEN *f1= p1 x1+ q1 x2 +r1* (7)

IF (x is A) AND (x is B) THEN f2= p2 x1+ q2 x2+ r2 (8)

ANFIS Characteristics [10]:

L0: State variables are nodes in ANFIS inputs layer
L1: Term sets of each state variable are nodes in ANFIS values layer, computing the membership value
L2: Each rule in FC is a node in ANFIS rules layer using soft-min or product to compute the rule matching factor i
L3: Each i is scaled into the normalization layer
L4: Each weighs the result of its linear regression fi in the function layer, generating the rule output
L5: Each rule output is added in the output layer
Layer 1: Calculate Membership Value for Premise Parameter

- Output O1,i for node i=1,2

$$O_{2,i}= Wi=\mu_{Ai}\ (X1) \quad \textbf{(9)}$$

- Output O1,i for node i=3,4

$$O_{2,i}= Wi=\mu_{Ai}\ (X1) \quad \textbf{(10)}$$

Where
A is a linguistic label (small, large, …)
Node output: membership value of input
Layer 2: Firing Strength of Rule.
Use T-norm (min, product, fuzzy AND)

$$O_{2,i}= Wi=\mu_{Ai}\ (X1)\ \mu Bi\ (X2) \quad \textbf{(11)}$$
(for i=1,2)

Node output: firing strength of rule
Layer 3: Normalize Firing Strength
Ratio of ith rule's firing strength vs. all rules' firing strength (for i=1,2)

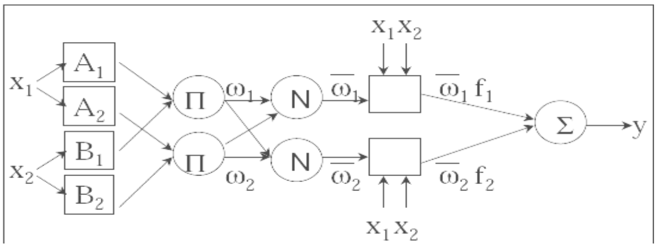

Figure (1) : The Architecture of The ANN

4. Simulation Scheme And Result

It is possible to use a graphics user interface, Command *anfisedit* [11]. It is also possible to use the command line interface or m-file programs. There are functions to generate, train, test and use these systems**.**

Layers: 0 1 2 3 4 5

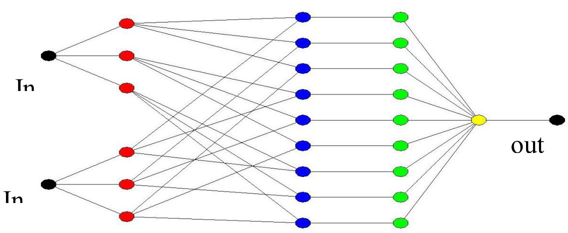

Figure (2) : The Architecture of The ANFIS

Three triangle membership function were used in input by selecting it in ANFIS module then choose output as linear and choose what the number of Epochs as a limiter for the iterations will be used in the training. Based on test data, it is easily to conclude that the result is very good by comparing it with the speed reading in the coming day [12]. Figure (3) illustrates the original data and how these data changed via days. And Figure (4) show the regression forecasting data and ARIMA forecast after regressed by SAS program. Figures

(5) ,(6) show the auto correlation and partial auto correlation of these data respectively. While Figure (7) presents the changes of predict ,lower and upper limit and original data using SAS package [14]. From this model the next value of speed could be found using Matlab [15] M-file as well as using SAS Package [14].

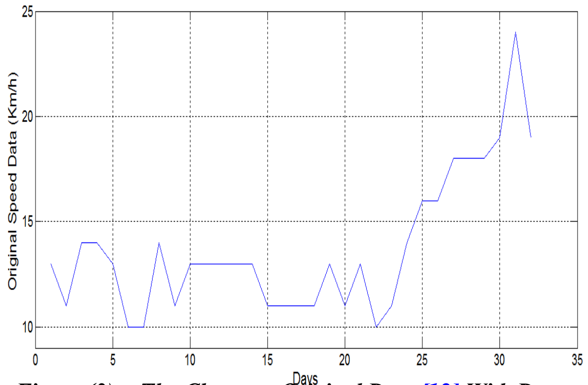

Figure (3) : The Changes Original Data [12] With Days Of One Month

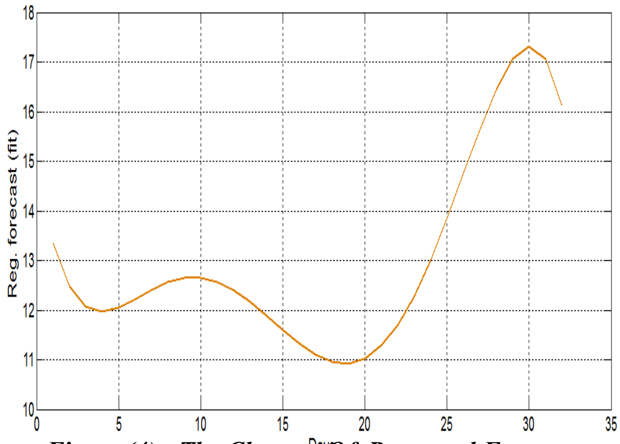

Figure (4) : The Changes Of Regressed Forecast Data Using SAS Package

Table (1) illustrates some of these results based on different ARIMA models using Different packages. Different ANFIS models with different number of inputs (previous days data) were investigated. Also, the effect of shape of membership on wind speed forecasting are studied as well as different numbers Of Membership Functions. The comparison of these ANFIS techniques are illustrated in Table (2). Table (3) illustrates The Comparison **Between Different** Between The Best Time Series Method And The Best ANFIS Method. The used techniques show promising results. However, they are competitive techniques.

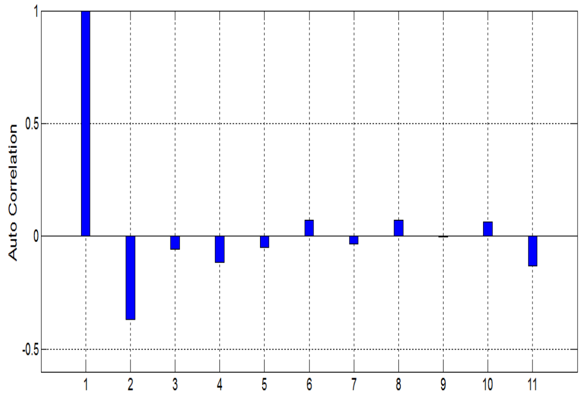

Figure (5) : The Auto Correlation Of The Data Using SAS Package

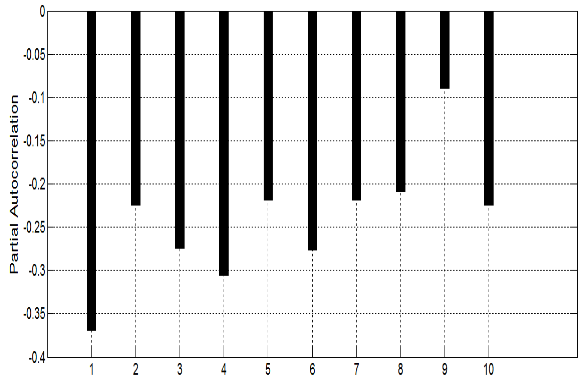

Figure (6) : The Partial Autocorrelation Of The Data

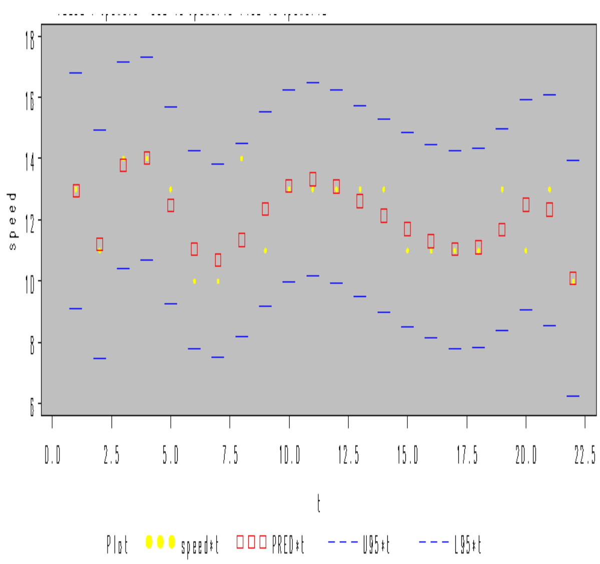

Figure (7) : The Changes Of Predict ,Lower And Upper Limit

Table (1) : The Comparison Time Series Methods (SAS And Matlab Packages)

Date	Original Data	When Matlab Programs used		When SAS Package Used		Error	When Matlab Programs Used		When SAS Package Used	
		ARIMA (1,0,1)	ARIMA (1,1,1)	ARIMA (1,0,1)	ARIMA (1,1,1)		ARIMA (1,0,1)	ARIMA (1,1,1)	ARIMA (1,0,1)	ARIMA (1,1,1)
23/9/2010	11	10.2423	10.6432	10.7505	10.72		6.9%	3.24%	2.268%	2.5%
24/10/2010	14	10.1724	11.0936	14.3397	14.32		27.34%	20.76	2.42%	2.2%
25/10/2010	16	15.1786	12.3433	16.62819	16.7		5.13%	22.85	3.926%	4.37%

Table (2) : The Comparison Between Different Types Of ANFIS Techniques

No Of Membership Function	**2**			**3**		**4**	**Original Reading**	**% Error**	**2**			**3**		**4**
No of input	**Four**	**Five**	**Six**	**Four**	**Five**	**Four**			**Four**	**Five**	**Six**	**Four**	**Five**	**Four**
Tuesday	**11.59**	**11.8**	**12**	**11.859**	**10.514**	**11.2**	**11**		**5.3**	**7.3**	**9.1**	**7.8**	**4.4**	**1.8**
Friday	**12.456**	**11.3**	**12.23**	**12.456**	**12.56**	**13.65**	**14**		**11**	**19.2**	**12.6**	**11**	**10.2**	**2.5**
Saturday	**12.34**	**12**	**11.1**	**11.828**	**16.7**	**13.5**	**16**		**22.8**	**25**	**30.6**	**26**	**4.37**	**15.6**

Table (3) : The Comparison Between The Best Time Series Method And The Best Anfis Method

Date	Original Data	Reg. Forecast	RegARIMA Forecast	ANFIS	Error	
					ANFIS	RegARIMA Forecast
23/9/2010	11	10.784	10.7505	**11.2**	**1.8%**	2.268%
24/10/2010	14	14.377	14.3397	**13.65**	**2.5%**	2.42%
25/10/2010	16	16.66919	16.62819	**13.5**	**15.6%**	3.926%

5. CONCLUSION

The proposed research discussed the use of stochastic modelling as will as ANFIS in wind speed forecasting. ANFIS can forecast very good of the next value (short term forecasting) and time series can forecast of data near of the original data in short term. However, time series can forecast of medium and long forecasting better than ANFIS and by more efficiency. The illustrated results show the effectiveness of both methods in STF for wind speed. These methods will help in utilizing the renewable energy in an efficient way.

Acknowledgement
The authors would like to acknowledge the Cairo University for its financial support for this research.

Corresponding Author:

M. A. Moustafa Hassan
Elec. Power Department, Faculty of Engineering
Cairo University, Giza, Egypt
mmustafa@eng.cu.edu.eg

References:

[1] J.C. Palomares-Salas, J.J. G. de la Rosa, J.G. Ramiro, J.Melgar, A. Agüera and A .Moreno, " Comparison of Models for Wind Speed Forecasting.", ICCS 2009; LSU Center for Computation & Technology, Baton Rouge, Louisiana, U.S.A.; May 25-27, 2009

[2] Research Unit PAIDI-TIC-168. University of Cádiz. Electronics Area. Escuela Politécnica Superior Avda. Ramón Puyol, S/N. E-11202-Algeciras-Cádiz (*Spain*) Moreno

[3] Amit Jain, E. Srinivas, Rasmimayee Rauta , Centre for Power Systems International

Institute of Information Technology Hyderabad - 500 032, INDIA December 2009, Short Term Load Forecasting using Fuzzy Adaptive Inference and Similarity.
[4] Moghram, S. Ruhman, Analysis and evaluation five of load forecasting techniques, IEEE Trans. Power Syst. 4 (4) (1989) 1484–1491.
[5] M.T. Hagan, S.M. Behr, The time series approach to short term load forecasting, IEEE Trans. Power Syst. PWRS-2 (3) (1987) 832–837.
[6] A.D. Papalekopulos, T.C. Hesterberg, A regression-based approach to short-term system load forecasting, IEEE Trans. Power Syst. 5 (4) (1990) 1535–1547.
[7] S. Vemuri, W.L. Huang, D.L. Nelson, On-line algorithms for forecasting hourly loads of an electrical utility, IEEE Trans. Power Apparatus Syst. PAS-100 (8) (1981) 3775–3784.
[8] Gwo-Ching Liao∗, Ta-Peng Tsao, Application of fuzzy neural networks and artificial intelligence for load forecasting, Department of Electrical Engineering, Fortune Institute of Technology, 125-8 Chyi-Wen Road, Chyi-Shan 842, Kaohsiung County, Taiwan,2003
[9] K. Levenberg, "A Method for the Solution of Certain Problems in Least Squares," Quart. Appl. Math. 2, 164-168, 1944.
[10] Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," SIAM J. Appl. Math. 11, 431-441, 1963.
[11] J.S.R. Jang "ANFIS: Adaptive-Network-Based Fuzzy Inference System", IEEE Trans. Systems, Man, Cybernetics, 23(5/6):665-685, 1993.
[12] J.S.R. Jang and N. Gulley, Natick, MA , "The Fuzzy Logic Toolbox for use with MATLAB,: The Math Works Inc., 1995.
[13] http://www.wunderground.com
[14] SAS, The Statistical Analysis System, Version 9.00 (TS M0), by SAS Institute Inc., Cary, NC, USA, 2002.
[15] MATLAB R2008a.

4/5/2011