

# Forecasting of Weekly Electricity Consumption Using Neural Networks

J.V. Ringwood and F.T Murray

Control Systems Group  
School of Electronic Engineering  
Dublin City University  
Glasnevin, Dublin 9

*Abstract: Neural networks have been shown to be effective in modelling time series, with applications in the forecasting of electricity consumption. In applying neural networks to weekly electricity consumption data, several issues, such as selection of network architecture, network structure and input structure need to be addressed. This paper addresses these issues in relation to the current application and also demonstrates that considerable value is to be gained from incorporating the lessons learned from linear time series modelling into the current nonlinear analysis. Results for national Irish weekly electricity data demonstrate the potential improvements which can be obtained using the neural network approach.*

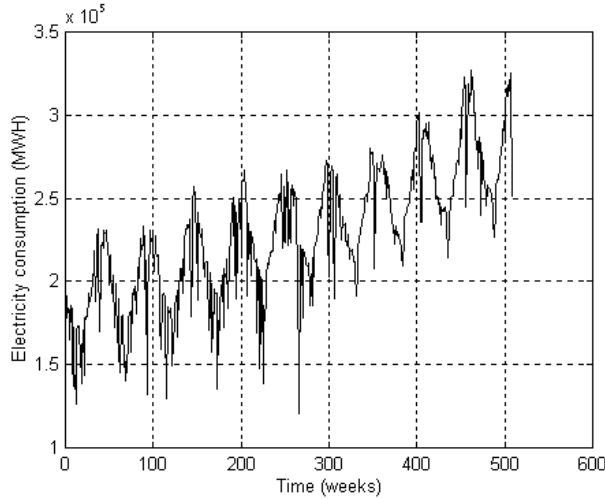
*Keywords: Time series modelling, neural networks, electrical energy consumption.*

## 1. Introduction

Linear time series modelling is a well understood science, ranging from the original formalised techniques of Box and Jenkins [1] to many recent developments, including techniques dealing with nonstationary data, harmonic (sinusoidal) time series and structural component models. One of the common factors is the incorporation of some form of parameter identification technique. These linear time series modelling techniques could therefore be described as black-box modelling techniques, which take into account some structural information about the data series in question. This paper attempts to draw on the structural analysis, while providing a nonlinear modelling tool. The introduction of nonlinear analysis presents a number of difficulties, some of which are additional to those encountered in linear systems analysis. The main issues to be tackled include the choice of nonlinear parameterisation and the algorithm used to determine the parameters. Neural networks provide a very general nonlinear parameterisation [2], where a set of nonlinear basis functions may be scaled and positioned at various points in the input space, using weights and biases. Other possibilities include Volterra [3] and Bilinear [4] representations.

There is considerable economic benefit in obtaining accurate forecasts of electrical energy demand. Given accurate demand forecasts, considerable savings are achieved by arranging to run only just sufficient generating plant to meet customer demand. However, a margin of reserve is provided to allow for forecast inaccuracies and generator breakdown. Further savings are achieved by dispatching the generating sets with the lowest unit costs, given detailed information regarding the daily demand profile. Forecasting analysis is frequently performed on a yearly (strategic planning), weekly (maintenance scheduling) and half-hourly (dispatch) basis. On these different time scales, electricity consumption, as a time series, presents diverse characteristics across different time scales. In addition, different causal inputs are appropriate on different time scales. For example, weather variables provide a significant causal effect on daily and weekly

profiles, but economic variables, such as GDP, are more appropriate when looking at longer trends [5].



**Fig.1: Demand profile for Ireland over a ten year period**

This paper examines the application of neural networks to modelling weekly electrical energy demand, in contrast to the more familiar daily load profile problem [6,7]. Weekly demand exhibits seasonal characteristics (mainly due to lower lighting and heating requirements in Summer) with a rising trend, indicating the progressively larger quantities of electrical energy consumed annually. The appropriate causal inputs on this timescale are weather variables, the primary effect due to temperature [8]. Fig.1 shows the consumption in Ireland over a ten year period approximately.

The technique which will be employed is to utilise the procedures used in linear time series analysis to decompose the series into its component parts. A nonlinear, neural-network based, model will then be fitted in place of the customary linear one.

## 2. Linear Time Series Modelling

For a seasonal time series, such as weekly electrical energy demand, two popular approaches which explicitly take into account the seasonal nature of the data can be adopted. The first (more traditional) approach involves modelling using the Box-Jenkins [1] methodology. This procedure involves the application of transformations which eases the subsequent modelling exercise, which is performed using a model with seasonal and non-seasonal sections. The second method has similarities to the first, but segments the model into three distinct parts, each of which contributes to the overall model output. Such a model is termed a structural model [9]. Models developed using these procedures can be either purely autoregressive (depend only on previous model outputs) or can be causal (driven by appropriate inputs).

### 2.1 Box-Jenkins Methodology

This general linear modelling approach follows the following procedure:

1. Determination of seasonality of time series and application of seasonal differencing.
2. Application of further differencing transformations to make the time series stationary.
3. Investigation of significant inputs to use as causal variables with the model.
4. Determination of orders of seasonal and non-seasonal regressors.
5. Identification of model parameters.

The univariate Box-Jenkins model is derived from the general SARI(p,d)(P,D) (seasonal autoregressive integrated) model which can be written as:

$$\Phi_p(B)\Phi_p(B^L)\nabla_L^D\nabla^d Y_t = a_t \quad (1)$$

where:  $Y_t$  is the time series

$\nabla_L^D \nabla^d = (1 - B^L)^D (1 - B)^d$  is a differencing transformation required if the data is nonstationary.  $d$  is the degree of non seasonal differencing,  $D$  is the degree of seasonal differencing, and  $L$  is the number of seasons in a year,

$a_t$  is the forecast error,

$\Phi_p(B) = (1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p)$  is the non seasonal autoregressive operator of order  $p$ ,

$\Phi_P(B^L) = (1 - \phi_{1L} B^L - \phi_{2L} B^{2L} - \dots - \phi_{pL} B^{pL})$  is the seasonal autoregressive operator of order  $P$ .

The lags  $p$  and  $P$  are determined using correlation analysis, as are the degree of the differencing operators,  $d$  and  $D$ . The seasonality of the data,  $L$ , is usually known *a priori*, or may also be determined using correlation analysis. A variety of methods may be used to determine the model parameters in the  $\Phi(B)$  polynomials, iterative least squares proving a popular approach. Following model construction, t-ratio tests may be used to assess the significance of the model.

## 2.2 Structural State-Space Models

Structural models adopt a different methodology than in Section 2.1 by modelling the trend and seasonal components, rather than removing their effect prior to modelling using transformations. A structural time series model consisting of a trend and a seasonal component may be described by

$$y(k) = t(k) + p(k) + \varepsilon(k) \quad (4)$$

where  $t(k)$  is a linear trend,  $p(k)$  a seasonal component and  $\varepsilon(k)$  a zero mean, serially uncorrelated white noise component. A *generalised random walk* (GRW) model [10] can be used to model the trend behaviour  $t(k)$ . The state-space form of the GRW model is defined by

$$\begin{bmatrix} t(k) \\ d(k) \end{bmatrix} = \begin{bmatrix} \alpha & \beta \\ 0 & \gamma \end{bmatrix} \begin{bmatrix} t(k-1) \\ d(k-1) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \eta_1(k) \\ \eta_2(k) \end{bmatrix} \quad (5)$$

$$t(k) = [1 \quad 0] \mathbf{x}(k)$$

where  $\alpha$ ,  $\beta$  and  $\gamma$  are constant parameters;  $t(k)$  is the trend at sample  $k$ ,  $d(k)$  is a second state variable and  $\eta_1(k)$  and  $\eta_2(k)$  are zero mean, serially uncorrelated discrete white noise inputs. An *integrated random walk* (IRW) is obtained with  $\alpha = \beta = \gamma = 1$ ;  $\eta_1(k) = 0$ .

If the seasonal component is well defined and stationary it can be modelled by a *periodic random walk* (PRW) or a *differenced periodic random walk* (DPRW) model [10]. The DPRW model is defined by:

$$p(k) = - \sum_{i=1}^{s-1} p(k-i) + \eta_p(k-1) \quad (6)$$

where  $s$  is the seasonal period and  $\eta_p(k)$  is a zero mean white noise disturbance input. If the trend component is represented by an IRW model and the seasonal component is represented by a DPRW model, then the complete state-space model is defined by the following:

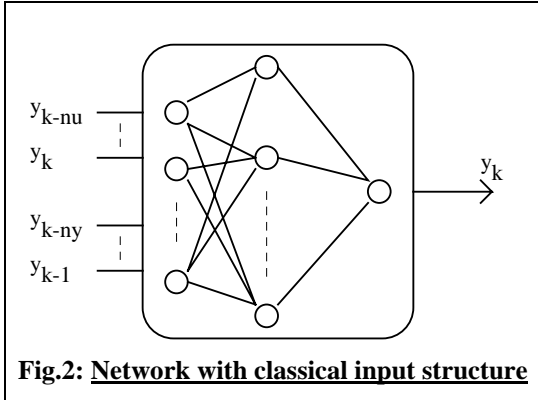
$$\begin{bmatrix} t(k) \\ d(k) \\ \text{-----} \\ p(k) \\ p(k-1) \\ \cdot \\ \cdot \\ p(k-(s-2)) \end{bmatrix} = \begin{bmatrix} 1 & 1 & | & 0 & 0 & \cdot & \cdot & 0 \\ 0 & 1 & | & 0 & 0 & \cdot & \cdot & 0 \\ \text{-----} & \text{-----} & | & \text{-----} & \text{-----} & \text{-----} & \text{-----} & \text{-----} \\ 0 & 0 & | & -1 & -1 & \cdot & \cdot & -1 \\ 0 & 0 & | & 1 & 0 & \cdot & \cdot & 0 \\ \cdot & \cdot & | & \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & | & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & | & 0 & 0 & \cdot & 1 & 0 \end{bmatrix} \begin{bmatrix} t(k-1) \\ d(k-1) \\ \text{-----} \\ p(k-1) \\ p(k-2) \\ \cdot \\ \cdot \\ p(k-(s-1)) \end{bmatrix} + \begin{bmatrix} 0 \\ \eta_2(k-1) \\ \text{-----} \\ \eta_p(k-1) \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \quad (7)$$

$$y(k) = [1 \ 0 \ 1 \ 0 \ \cdot \ \cdot \ 0 \ 0 \ 0]x(k) + \varepsilon(k)$$

In order to perform a prediction, a Kalman filter is used over the identification data set to provide initial state estimates for the model. Covariances for the (process) noise sources  $\eta_1(k)$  and  $\eta_2(k)$  and the measurement noise,  $\varepsilon(k)$ , are determined using maximum likelihood optimisation.

### 3. Time Series Modelling Using Neural Networks

#### 3.1 Network Input Structure



A total black-box approach to neural network modelling of dynamical systems or time series would be to utilise a model of the form shown in Fig.2, with tapped delay lines for input and output variables forming the input to the neural network. Such an approach is common in a variety of time series and model-based control system applications [11,12]. However, such an approach may disregard structural information about the dynamical model available from linear analysis. In the current study, an effort is made to incorporate information on an effective input and model structure suggested by linear time-series modelling techniques

##### 3.1.1 Box-Jenkins Model

For the Box-Jenkins methodology in Section 2.1, two nonlinear options are possible. Expansion of equation (1) gives:

$$(1 - \phi_1 B - \dots - \phi_p B^p - \phi_{1,L} B^L + \phi_1 \phi_{1,L} B^{L+1} + \dots + \phi_p \phi_{1,L} B^{L+p} - \dots - \phi_{p,L} B^{pL} + \phi_1 \phi_{p,L} B^{pL+1} + \dots + \phi_p \phi_{p,L} B^{pL+p}) (1 - B^L)^D (1 - B)^d Y_t = a_t \quad (8)$$

A corresponding nonlinear model for the structure in equation (8) could now be defined as:

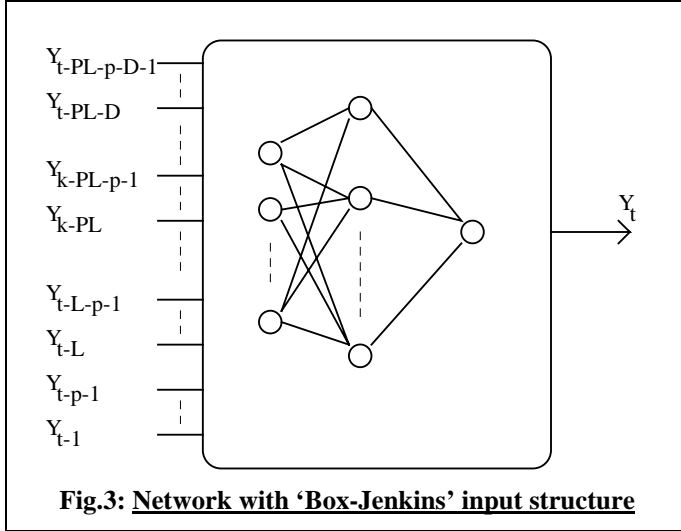
$$f(Y_y, \dots, Y_{t-p-1}, Y_{t-L}, \dots, Y_{t-L-p-1}, \dots, Y_{t-PL}, \dots, Y_{t-PL-p-1}, \dots, Y_{t-PL-D}, \dots, Y_{t-PL-p-D-1}) = a_t \quad (9)$$

This model will be termed a NNBJ type 'A' model. Alternatively, defining  $Z_t$  as:

$$Z_t = (1 - B^L)^D (1 - B)^d Y_t \quad (10)$$

a model (NNBJ type 'B') of the form:

$$g(Z_y, \dots, Z_{t-p}, Z_{t-L}, \dots, Z_{t-L-p}, \dots, Z_{t-PL}, \dots, Z_{t-PL-p}) = a_t \quad (11)$$



results, where the neural network is used to model data which has already been subject to seasonal and one-step differencing. To obtain the final forecast, the output from the neural network must be appropriately integrated, using seasonal and one-step integration. As an example, Fig.3 shows a neural network forecasting model using a form corresponding to equation (9). Compared to Fig.2, the input structure had been modified so that the network is focussed on the most effective inputs. This generally also results in fewer inputs overall, resulting in reductions in training times. For a model of the form of

equation (9), the total number of inputs is  $(P+D)(p+d+1) + p+d$ . For a 'standard' autoregressive (AR) model of the form of Fig.2, it would be usual to choose inputs which span a season i.e.  $L$  inputs. In the current example, this would yield 52 inputs for the AR model, with only 31 and 21 inputs respectively for models based on equations (9) and (11).

### 3.1.2 Structural Model

For the structural state-space model presented in Section 2.2, the approach is to let the linear sections (IRW + DPRW) model the trend and seasonal components, with the neural network used to model the remaining residuals in  $\varepsilon_k$ . This concurs roughly with the model presented in (11). However, in order to ensure a good training set for the network, a state-space smoothing algorithm [13] is employed to back-smooth the state estimates resulting from the Kalman filter, since the estimates from the filter will be poor during initial convergence. The output  $[t(k)+p(k)]$  from the smoothed state estimates are then subtracted from the identification data, with this difference providing the network training set. Since the network is now dealing with (approximately) data which has been de-trended and de-seasonalised, an input structure similar to that in equation (11) can be used.

## 3.2 Neural Network Design

### 3.2.1 Network Architecture and Structure

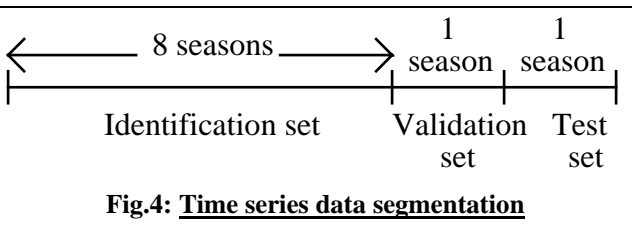
Network architecture requirements are for a network which can operate recurrently (since the time series is autoregressive) and produce a continuous output. In addition the size of the network should not be intractable. The latter condition excludes the utilisation of local approximators, due to the dimension of the input space encountered in the current application and recurrent Multi-Layer Perceptrons (MLP's) were adopted as a suitable network structure [14]. In terms of configuration, a three layer structure was adopted with a linear output neuron (effectively removing any restriction on output range), while the number of neurons in each layer was

determined using optimisation across Monte-Carlo runs, with the mean-squared error across a validation set as a criteria. Since the time series considered are exclusively unipolar, log-sigmoid functions were employed in the nonlinear neurons in layers 1 and 2.

### 3.2.2 Neural Network Training

Two important aspects of network training which must be considered here are the choice of training algorithm and the training cessation point. A standard LMS gradient technique with backpropagation was employed for training, which also included a momentum term and adaptive learning rate. Faster techniques, such as the Lervenberg-Marquardt algorithm [15] were also examined, but found to be extremely sensitive to initial conditions and local minima. This can be overcome, to some extent, by utilising sufficient Monte-Carlo runs, but this extra computation, combined with the slower computational speed of such algorithms was found to more than offset any gains in convergence speed.

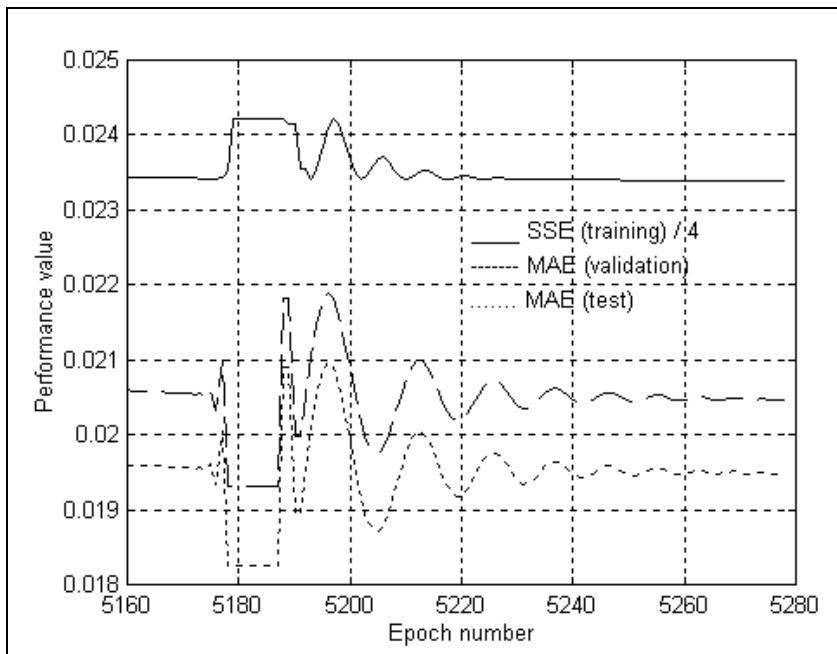
Neural networks trained for time series applications are typically trained using single step prediction criteria. However, this does not always determine the weight set which optimises the multi-step prediction performance of the network. One reason for this is that backpropagation training with multi-step criteria are difficult to design and can be computationally intensive, particularly when the prediction



horizon is long. A compromise is to train the network for single-step performance, but examine the multi-step performance during training. Fig.5 shows the variation in single step sum-squared error (SSE) and the multi-step mean absolute error (MAE) over the validation set and test set (over which the prediction is to be performed, see Fig.4). Note that the variation in the multi-step criteria for the validation and test portions are *consistent*, allowing a stopping point (weight vector) to be chosen based on the validation set which will give good multi-step performance when doing the actual forecast. For example, a choice of weights at epoch 5180 gives a multi-step performance value 0.01825 while the corresponding value at epoch 5188 is 0.02090 (approximately 14% worse), in spite of the fact that the single-step SSE suggests that a choice of weights at epoch 5180 is

horizon is long. A compromise is to train the network for single-step performance, but examine the multi-step performance during training. Fig.5 shows the variation in single step sum-squared error (SSE) and the multi-step mean absolute error (MAE) over the validation set and test set (over which the prediction is to be performed, see Fig.4). Note that the variation in the multi-step

criteria for the validation and test portions are *consistent*, allowing a stopping point (weight vector) to be chosen based on the validation set which will give good multi-step performance when doing the actual forecast. For example, a choice of weights at epoch 5180 gives a multi-step performance value 0.01825 while the corresponding value at epoch 5188 is 0.02090 (approximately 14% worse), in spite of the fact that the single-step SSE suggests that a choice of weights at epoch 5180 is



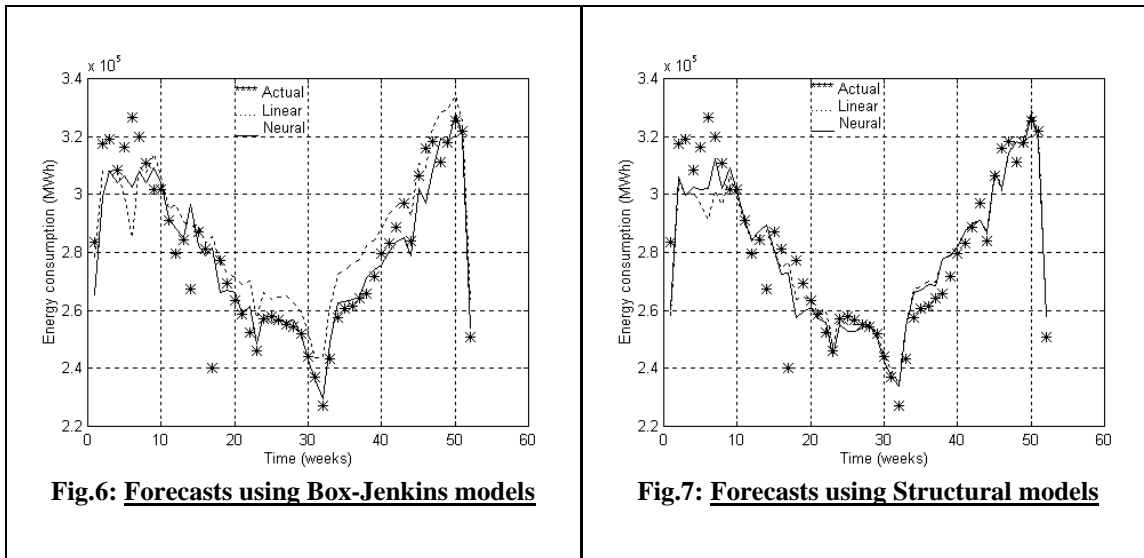
**Fig.5: Variation in single- and multi-step criteria during training**

particularly bad.

#### 4. RESULTS

Model type	MAF Linear	MAF Neural Network
Autoregressive (52 inputs)	$1.1101 \times 10^4$	$1.2481 \times 10^4$
Box-Jenkins	$1.0691 \times 10^4$	$0.9040 \times 10^4$ NNBJ type 'A' $0.6587 \times 10^4$ NNBJ type 'B'
Structural State Space	$0.7687 \times 10^4$	$0.7198 \times 10^4$

**Table 1: Comparative results for linear and neural models**



Ten Monte-Carlo runs were used in the training of the neural networks, in an attempt to decrease the sensitivity to initial conditions. Table 1 compares the multi-step (52 step ahead) forecasting results for linear and neural models. Figs.6 and 7 demonstrate qualitatively the improvement achieved. While it is interesting to note that there is significant improvement in the use of neural networks with Box-Jenkins and Structural models, it is perhaps more dramatic the effect that the choice of input structure has on performance. In particular, note the poor performance of the 'classical' autoregressive (with inputs formed from the past 52 week's demand) model. The neural version of this, which incidentally is inferior to its linear counterpart, has a much poorer performance than (either linear or neural) Box-Jenkins or Structural models.

Initial results for causal models using HDD [8] and average temperature inputs along with the examination of a further data set from a regional power board in New Zealand confirm the benefit of utilising structured inputs when forecasting seasonal weekly electricity demand profiles.

#### 5. Conclusions

This paper addresses the issue of utilising neural networks for forecasting weekly electrical energy demand, which is a seasonal time series. While it is shown that benefits exist in utilising nonlinear modelling tools, such as neural networks, evidenced by the results given, perhaps the most important lesson is to incorporate ideas from linear seasonal time series analysis, which are

well founded. Many open questions still exist in terms of choice of network structure and there is scope for improvement in training algorithms which can hopefully give us algorithms with good convergence speed while having the capability to achieve global minimums, irrespective of initial conditions. However, for time series modelling, which is basically an off-line model calculation, achievement of global minima in the performance surface remains the priority.

## REFERENCES

1. Box, G. and Jenkins, G. *Time Series Analysis, Forecasting and Control*, Holden-Day, San Francisco, 1976.
2. Azoff, M.E. *Neural Network Time Series Forecasting of Financial Markets*, John Wiley and Son, 1994.
3. Karkikeyan, A. and Sarkar, B. *Forecasting by Volterra Type Models*, Journal on Statistics, Computation and Simulation, Vol.28, pp 245-260, 1987.
4. Granger., C.W.J. and Andersen, A.P. *Introduction to Bilinear Time Series Models*, Vandenhoeck and Ruprecht, Gottingen, 1978.
5. Murray, F.T. and Ringwood, J.V. and O'Brien, K. *Yearly Electricity Sales Forecasting Using Artificial Neural Networks*, Neural Computing and Applications, submitted Jan. 1996.
6. Vemuri, V.R. and Rogers, R.D. (Eds.) *Artificial Neural Networks - Forecasting Time Series*, IEEE Computer Society Press, 1994.
7. Azzam-ul-Asar and McDonald, J.R. *A Specification of Neural Network Applications in the Load Forecasting Problem*, IEEE Trans. on Control Systems Technology, Vol.2, No.2, June 1994.
8. Murray, F.T. and Ringwood, J.V. *Improvement of Electricity Consumption Forecasts Using Temperature Inputs*, Simulation Practice and Theory, Vol.2, pp121-139, 1994.
9. Harvey, A.C. *Structural Time Series Models and the Kalman Filter*, Cambridge University Press, 1989.
10. Ng, N.C. and Young, P.C. *Recursive Estimation and Forecasting of Non-Stationary Time Series*, Journal of Forecasting, Vol.9, pp 173-204, 1990.
11. Chakraborty, K., Mehrotra, K., Mohan, C.K. and Ranka, S. *Forecasting the Behaviour of Multivariate Time Series Using Neural Networks*, Neural Networks, Vol.5, pp 961-970, 1992.
12. Kuschewski, J.G., Hui, S. and Zak, S. *Application of Feedforward Neural Networks to Dynamical System Identification and Control*, IEEE Trans. on Control System Tech., Vol.1, No.1, March 1993.
13. Harvey, A.C. *A Unified View of Statistical Forecasting Procedures*, Journal of Forecasting, Vol.3, pp 245-275, 1984.
14. Mitchell, J. *Comparing Feedforward Neural Network Models for Time Series Prediction*, Proc. Conf. on Neural Computing Research and Applications, Belfast, June 1992.
15. Ljung, L. *System Identification: Theory for the User*, Prentice-Hall, 1987.