Title: A strategy for short-term load forecasting in Ireland

Candidate: Damien Fay M.Eng. B.E.

Ph.D. Thesis

University Name: Dublin City University

Supervisor: Marissa Condon

School: Electronic Engineering

Month and Year of submission: July 2004.

Number of volumes: 1

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Ph.D. is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: Janice Foy ID No: 98970394.

Candidate

Date: 30/04/04

Acknowledgements

I would like to express sincere gratitude to my supervisors Prof. John Ringwood and Dr. Marissa Condon for their patience, help and understanding over the past few years. I would also like to thank Michael Kelly and John Kennedy of Eirgrid (ESB) for their sponsorship and for providing the data and expertise to make this project possible. I would also like to thank Prof. Jerome Sheehan for his statistical expertise and advice. Finally, I would like to thank my parents for their endurance and Judith for her constant encouragement.

Contents List

A strategy for short-term load forecasting in Ireland.

Damien Fay

ABSTRACT

Electric utilities require short-term forecasts of electricity demand (load) in order to schedule generating plant up to several days ahead on an hourly basis. Errors in the forecasts may lead to generation plant operation that is not required or sub-optimal scheduling of generation plants. In addition, with the introduction of the Electricity Regulation Act 1999, a deregulated market structure has been introduced, adding increased impetus to reducing forecast error and the associated costs.

This thesis presents a strategy for reducing costs from electrical demand forecast error using models designed specifically for the Irish system. The differences in short-term load forecasting models are examined under three independent categories: how the data is segmented prior to modelling, the modelling technique and the approach taken to minimise the effect of weather forecast errors present in weather inputs to the load forecasting models.

A novel approach is presented to determine whether the data should be segmented by hour of the day prior to modelling. Several segmentation strategies are analysed and the one appropriate for Irish data identified. Furthermore, both linear and non-linear techniques are compared with a view to evaluating the optimal model type. The effect of weather forecast errors on load forecasting models, though significant, has largely been ignored in the literature. Thus, the underlying issues are examined and a novel method is presented which minimises the effect of weather forecast errors.

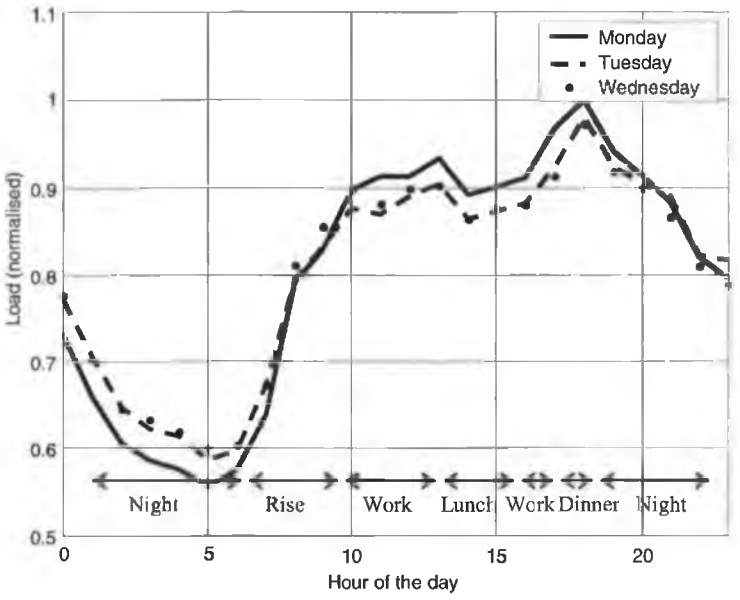

Figure 1.1 Typical working day loads.

The weekly cycle consists of the five working days (Monday to Friday) and the weekend days (Saturday and Sunday) (Figure 1.2). The loads on Saturdays and Sundays have similar load curves to weekdays. However, the load is lower reflecting reduced economic activity.

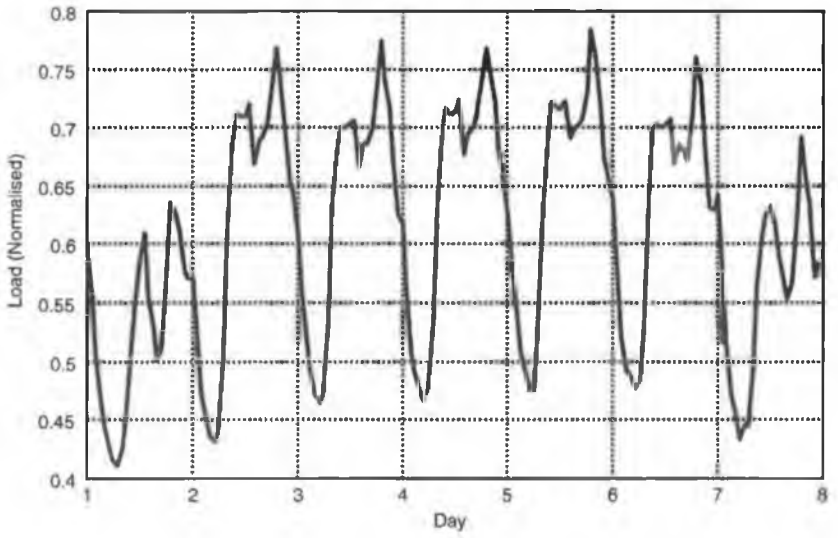

Figure 1.2 A typical weekly load (Day 1=Sunday)

The yearly cycle in Ireland is shown below in Figure 1.3. The load is lower in summer than in winter as there are more heating requirements in winter. However, during Christmas the load shows a sharp drop as economic activity is

reduced. Also, there is a less obvious reduction in load during July and August as economic activity is reduced due to people taking holidays.

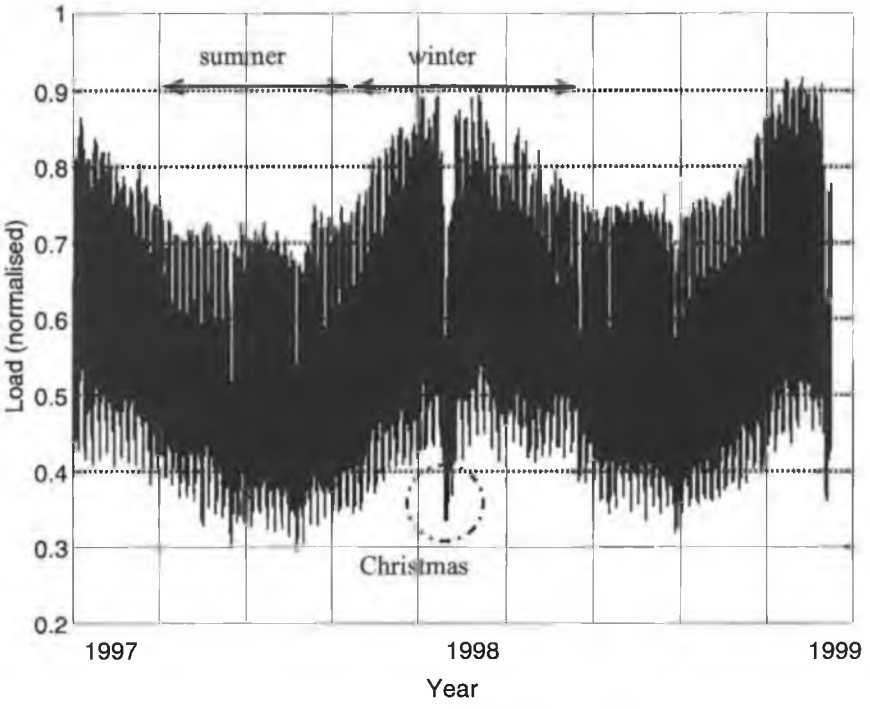

Figure 1.3 The load for 1997 and 1998.

In addition, there exist special periods such as bank holidays, Easter, etc. which result in reduced economic activity and may not occur at the same date each year.

Another important factor in the demand for electricity is the weather. On cold days for example, people will tend to use more electricity for heating than on a day with an average or 'comfortable' temperature. Conversely, a very hot day will result in more electricity being used for air-conditioning. This is reflected in Figure 1.3 as summer days are generally warmer than winter days.

It is important to note that although the factors that effect electricity demand are well known, there is no underlying process that can be measured and so *black box modelling* (i.e. constructing a model from recorded inputs and outputs without knowledge of the intervening process) must be used.

The area of load forecasting in general and the precise area of interest in this thesis are detailed in Section 3.1.1.

1.2 Motivation for Research.

The motivation for forecasting short-term electrical load is to reduce costs. Power plants must be switched on in advance and overestimation of demand results in some generators being operational but not being used. The electrical production must match the demand in order to guarantee supply. The amount of excess electricity production (or *spinning reserve*) required to guarantee supply in the event of an underestimation is also determined by the accuracy of load forecasts. Different power plants have different production costs and take different amounts of time to start up. Forecasting errors can lead to sub-optimal scheduling of power plants (*unit commitment*).

Another factor motivating the use of load forecasts in Ireland came with the introduction of the Electricity Act 1999 which led to a staged deregulation of the energy (gas and electricity, although gas is of no concern here) market in Ireland. Currently, the trading arrangements for this market are regulated by the Commission for Electricity Regulation (CER) (CER, 2000). The market is made up of three types of operators; *generators* (who generate electricity), *suppliers* (who supply the customers with electricity purchased from a generator or a trader) and *traders* (who purchase electricity from generators and sell to suppliers). In addition, there are separate arrangements for sale and purchase of electricity from the Northern Ireland electricity grid.

Although the rules of the energy market are relatively complicated, it is essentially based on a *bilateral agreement* structure (CER, 2000). In this structure, a generator agrees to provide a set amount of electricity and a supplier agrees to purchase that amount at a particular time (hence the term bilateral). Twenty-two days after that load period, the amount of electricity that was required by the supplier is calculated and compared to the amount actually purchased in the bilateral agreement. In the event that the supplier has purchased an excess of electricity (and this excess was used by the grid), this excess is sold

at what is called a *spill price*. In the event that the supplier did not purchase enough electricity, he must then purchase the difference at what is called a *top-up price*. The traders then purchase the excess electricity at the spill-price and sell at the top-up price in order to *balance* the bilateral agreements.

The structure of the market is such that the top-up price is far in excess of what would have been required in the original bilateral agreement, while the spill price does not match the price paid in the original agreement (CER, 2000). A similar situation also faces the electricity generators. Thus, it is important that all parties to a bilateral agreement forecast their expected demand correctly i.e. short-term load forecasting is an integral part of the market.

1.3 Main Thesis Contributions.

The main contributions of this thesis are in the area of short-term load forecasting of overall grid demand. These contributions are as follows:

1. A review of techniques for integrating weather forecast errors into load forecasting models (Chapter 3, Section 3.5). This area has, to a large extent, been ignored in the literature,

2. Identification of the different day-types in *Irish* electrical load data (Chapter 4),

3. Application of parallel models to load forecasting and determining the appropriateness of these in comparison to sequential models (Chapters 5 and 6),

4. Further development of the multi-timescale models proposed by Murray (1996) including a technique for optimising the weights used in that algorithm (Chapter 6), and

5. A novel approach to dealing with the effect of weather forecast errors in load forecasting models. This also includes a novel approach to modelling weather forecast errors (Chapter 7).

Much of the work in this thesis is an extension on the work presented in the following publications:

- Fay, D., Ringwood, J.V., Condon, M., Kelly, M., 2003, 24-hour electrical load data – a sequential or partitioned time series?, *Journal of Neurocomputing,* 55 (3-4), pp 469-498.

- Fay, D., Ringwood, J.V., Condon, M., Kelly, M., 2000, Comparison of linear and neural parallel time series models for short term load forecasting in the Republic of Ireland, *in: Proceedings, 3rd Universities Power Electronics Conference*, September, (not paginated on CD ROM),

- Fay, D., Ringwood, J.V., Condon, M., Kelly, M., 2001, 24-Hour electrical load data - a time series or a set of independent points?, *in: Proceedings, 6th Conference on European Applications of Neural Networks*, June, Cagliari, Italy, (not paginated on CD ROM).

- Fay, D., Ringwood, J.V., Condon, M., Kelly, M., 2001, A data fusion model for Irish electricity load forecasting, *in: Proceedings, Irish Signal and Systems Conference*, Maynooth, Ireland, (not paginated on CD ROM).

- Fay, D., Ringwood, J.V., Condon, M., Kelly, M., 1999, New research developments in half-hourly forecasting methodologies and technologies applied to the Irish electricity market., *in: Proceedings, IIR Conference on New Energy Trading Arrangements*, London, 6th July, pp 301-307.

- Fay, D., Ringwood, J.V., Condon, 2004, On the influence of weather forecast errors in short-term load forecasting models, *in: Proceedings, Control 2004*, University of Bath, U.K., 6-9 September, (not paginated on CD ROM).

1.4 Thesis Structure.

This thesis is organised into eight chapters. The approach here is to first present some preliminary analysis of Irish load data in order to familiarise the reader with the subject (Chapter 2). A literature review is then presented (Chapter 3) followed by analysis and modelling (Chapters 4 to 7). Conclusions are then presented in Chapter 8.

Chapter 2 provides the 'problem setup' for this thesis. Initially, the available data (used throughout this thesis), and the way in which this data is partitioned into sets for model building and testing are documented. The second part of Chapter 2 provides some simple preliminary analysis to highlight the characteristics of Irish load data while attempting not to bias the model building process in later chapters.

Chapter 3 provides a literature review of the area of load forecasting. Initially, the *exact area of interest* is specified with respect to the wider area of load forecasting. However, the primarily purpose of this chapter is to present the *main factors* that differentiate the load forecasting approaches found in the literature. These main factors are:

1. How the data set is *segmented* prior to modelling,
2. The *technique* used to model the data (regardless of the segmentation used), and
3. The approaches taken to deal with *weather forecast errors* in weather inputs.

Chapters 4,5 and 6 investigate point 1, above. Chapters 5, 6 and 7 study point 2. Finally, Chapter 7 examines point 3.

Chapter 4 examines the segmentation of Irish load data in to different types of days (for example, working days, weekend days etc.) called *day-types*. The focus

of this Chapter is to determine *if* different day-types exist and whether there is *sufficient* data within each day-type for consistent model building. The segmentation of the data into the various day-types is then presented.

Chapter 5 constructs the first load forecasting models used in this thesis. These models are based on data segmented by hour of the day (in addition to being segmented by day-type) and are called *parallel models* as there is an array of models, one for each hour of the day. The primary purpose of this chapter is to construct parallel models for comparison with the models in Chapter 6 that do *not* use hour of the day segmentation. In addition, several input selection techniques are examined. Finally, linear and non-linear parallel models are compared to see if non-linear modelling techniques are advantageous for STLF.

Chapter 6 develops load forecasting models for data that is segmented by day-type only. The models used here are called multi-timescale models as they exploit forecasts of the load made at differing timescales. The multi-timescale modelling technique developed by Murray (1996) is developed further. Finally a comparison with the parallel models in Chapter 5 is drawn in order to determine if hour of the day segmentation is advantageous.

Chapter 7 investigates the effect of weather forecast errors on load forecasting models. A novel technique is proposed for producing *pseudo-weather forecast* errors which have the same error statistics as recorded weather forecasts. A model is then proposed which minimises the effect of these weather forecast errors.

Chapter 8 presents the conclusions and assesses the models presented in this thesis. Recommendations for practical implementation of these models are made to Eirgrid (thesis sponsor) and future work in this area is discussed.

Chapter 2

Characteristics of Irish Electrical Load Data

2.1 Introduction

This chapter describes preliminary analysis of Irish electrical load data, which is important for model building. As will be seen, electrical load is a complex time series which among other things changes over time and has embedded cycles.

The amount, timescale and type of data available have obvious importance in model building. This information is detailed in Section 2.2.

In the long term, load is an evolving process. With increased economic activity over the last few years, electrical demand has increased considerably. It is important to quantify the rate of increase, the effect it may have on the shape of the load and to ascertain if these characteristics are deterministic. Such characteristics are examined in Section 2.3.1.

The embedded cycles in the load occur at daily, weekly and yearly timescales. The shape of the daily load curve as well as the level of the load is affected by these cycles. Classification of the shapes is examined in Section 2.3.2. The level of the load is effected mainly by temperature and this relationship is examined in Section 2.3.3 to determine if it is non-linear or can be linearised.

2.2 Data Availability

There are two sources for the data used in this thesis: load data is supplied by the national grid (Eirgrid), while the Meteorological Office of Ireland (MOI) provides weather data gathered at their station at Dublin airport*. Eirgrid also supplies historical weather data gathered at its station in Dublin; however, without an accompanying weather forecasting facility at that location, this data is of limited use and is not used in this project.

* Dublin is the largest population centre in Ireland with approx. ½ the population.

2.2.1 Range and Timescale of Data

The range and timescale of the available electrical demand data is given in Table 2.1.

Table 2.1 Eirgrid data timescale and range.

Range	29th December 1986 – 31st March 2000
Timescale	Hourly
No. of data points	4842 Days (116208 hours)

Two categories of historical weather data are available from the MOI: *readings* (or *actual* weather) and *forecasts*. Both sets of data are for Dublin airport, the closest and most relevant weather station to Dublin. The ranges, timescales and types of data are given in Table 2.2.

Table 2.2 MOI data, time-scales and ranges.

Type	Range	Time scale
Dry bulb temperature *readings*	29th December 1986 – 31st March 2000	Hourly
Humidity *readings*	29th December 1986 – 31st March 2000	Hourly
Wind speed & Direction *readings*	29th December 1986 – 31st March 2000	Hourly
Cloud cover *readings*	29th December 1986 – 31st March 2000	Hourly
Dry bulb temperature *forecasts*	1st February 2000 – 1st March 2000	Hourly
Wind speed & Direction *forecasts*	1st February 2000 – 1st March 2000	Hourly
Cloud cover *forecasts*	1st February 2000 – 1st March 2000	Hourly

The data is subdivided into three sets in order to train and test the load forecasting models (Table 2.3):

- *The training set* is used to calculate model parameters,
- *The validation set* is used to aid in model structure determination and
- *The novelty set* is used to evaluate model performance. As the validation and training sets have significantly influenced the model, a novelty set is used to evaluate model performance with previously unseen data.

Table 2.3. Division of data set (all dates inclusive).

Set	Training	Validation	Novelty
Range	1987-1996	1997-1998	1999-2000

The techniques used for input selection (Section 5.5.2) utilise different training and validation sets where the dates vary due to the use of a bootstrapping technique (van Giersbergen and Kiviet, 2002), which allows a statistical evaluation of the different input selections. In this case, eight bootstraps are constructed, where the validation set occupies a different range for each set (Figure 2.1 and Table 2.4) within data form 1987 to 1998 inclusive.

Table 2.4. Segmentation of data set for input selection (Section 4.5.1).

Set	Training	Validation
Range	Variable	Variable

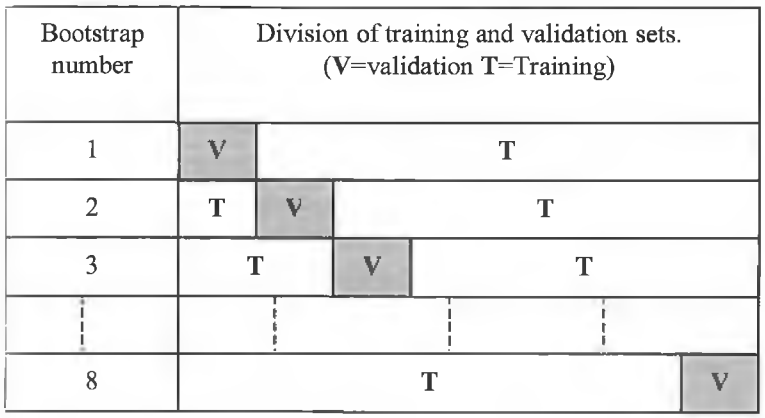

Figure 2.1. Selection of training and validation sets for input selection (Section 4.5.1) .

2.3 Characteristics of Irish Electrical Load Data

2.3.1 Trend and Variability

Figure 2.2 (below) shows the growth in electrical demand in Ireland in recent years. To gain initial insight into the nature of the underlying trend a quadratic curve of the form (Equation 2.1) is fitted to the data:

$$d(t) = at^2 + bt + c + \varepsilon(t) \qquad (2.1)$$

where t is the time in hours since the start of the data (1986), $d(t)$ is the trend at time t, $\varepsilon(t)$ is an error term, and a, b, c are coefficients calculated via least squares. Note that a,b,c are positive (Table 2.5).

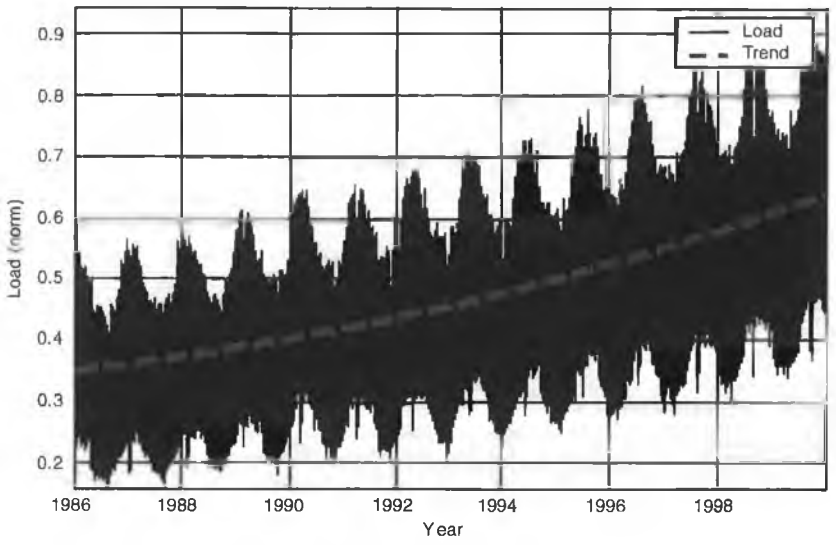

Figure 2.2 Load with approximate trend curve.

Table 2.5 Co-efficients of trend curve (normalised).

Co-efficient	Value
a	1.1360×10^{-11}
b	1.1640×10^{-6}
c	0.3498

In addition to an underlying trend there is also a growing *variability*, v. The variability of load for year j, v_j, is defined as:

$$v_j = y_{\max,j} - y_{\min,j} \tag{2.2}$$

where $y_{\max,j}$ and $y_{\min,j}$ are the maximum and minimum loads in year j, respectively. This rising variability is shown in Figure 2.3(b).

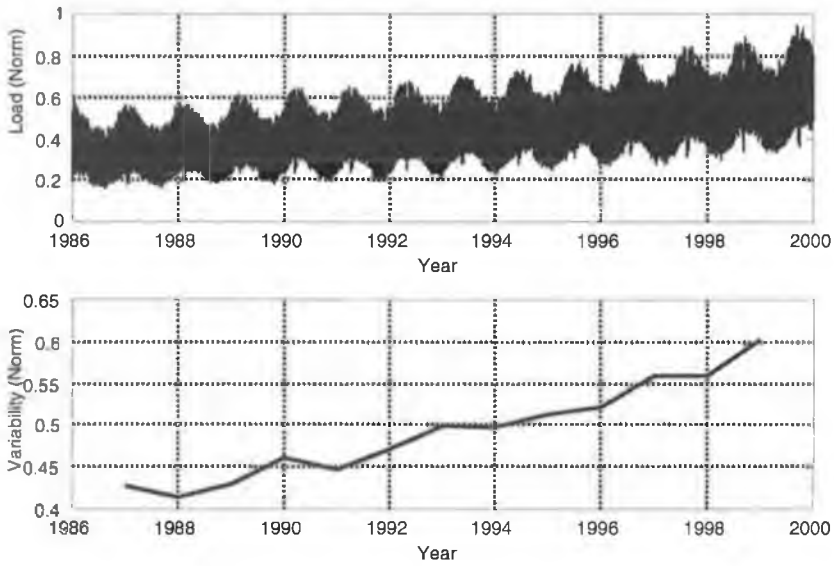

Figure 2.3 (a) Hourly load data and (b) its variability

The trend and the variability can be related by the use of the Box Cox transform (Box and Cox, 1964). They proposed a transform for *equalising* (to make constant) the variance of a time-series where the variance, at a given time *t,* is related to the value of the series, at that time *t*. This transform can also be applied to equalise the variability of time-series whose variability is a function of the trend (Franses and Koehler, 1998).

The Box-Cox transform has been used in many time-series applications; for example, Osula and Adebisi (2001) applied the transform to travel expenditures in Nigeria. This time-series exhibited an increasing trend and variability over time while the main causal variable (the price of oil) did not. When the travel series had been equalised and de-trended, the correlation between it and the oil price could be seen. Pere (2000) in a study of height and weight of adolescents applied the transform to remove differences due to age.

The data is transformed according to a transform parameter λ (Box and Cox, 1964) as:

$$y'(t) = \begin{cases} (y(t)^{\lambda} - 1)/\lambda & \text{if } \lambda \neq 0 \\ \log(y(t)) & \text{if } \lambda = 0 \end{cases} \qquad (2.3)$$

where $y'(t)$ is the transformed load t hours from the start of the data (1986) and $y(t)$ is the original load.

In the current study electrical load data is transformed using the Box-Cox transform with $\lambda = 0.3$ (Figure 2.4). As the purpose of this section is merely to establish that a relationship between trend and variability *exists*, λ is determined empirically (as suggested by Cryer, 1986). The transformed load data exhibits an approximately equalised variability (Figure 2.4), showing that a definite relationship exists between the trend and variability of the load.

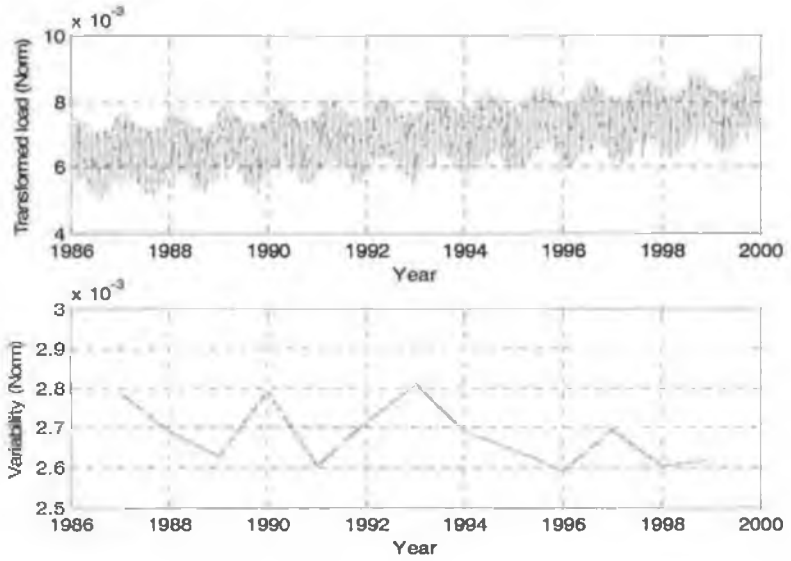

Figure 2.4 (a) Transformed load and (b) variability of transformed load

2.3.2 Day-Types

Daily load data can be disaggregated into distinct groups (called *day-types*) each of which have common characteristics. As can be seen in Figure 2.5 there is, for example, an obvious difference between the *shape* of the load on a typical Sunday and Monday due to decreased economic activity on a Sunday. Furthermore, there is a distinct difference between the *shape* of a typical winter day and summer day (Figure 2.6). A typical winter day exhibits a higher peak at 6pm relative to a summer day, due to increased lighting needs in winter, among other things.

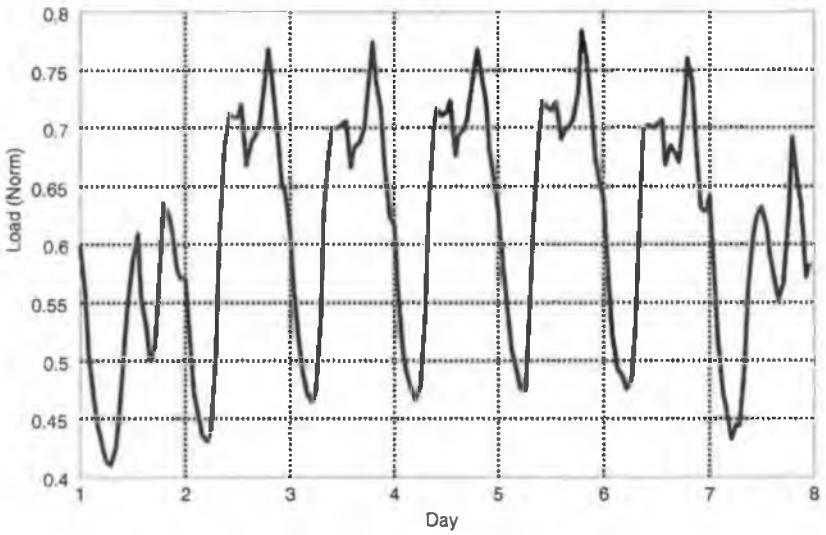

Figure 2.5 A typical weekly load (Day 1=Sunday)

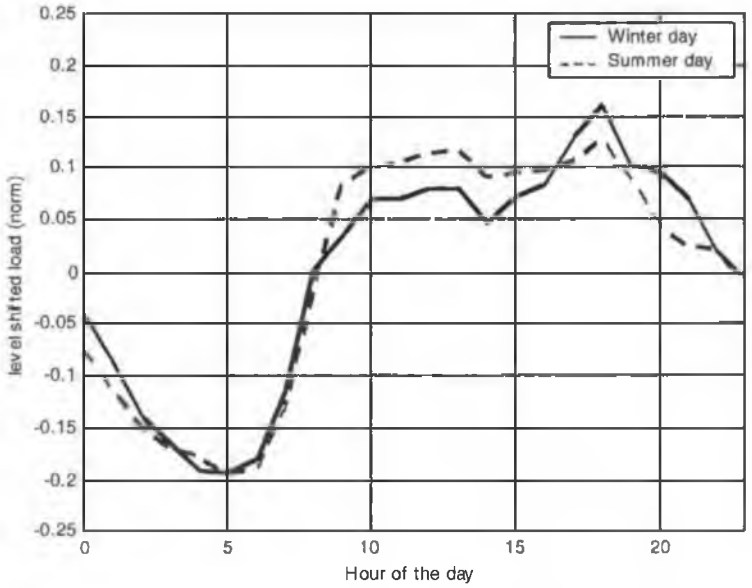

Figure 2.6 Typical shape of a Winter (18/11/1998) and Summer working day (17/06/1998) (The mean load for each day has been subtracted)

Techniques for day-type identification are discussed in the literature review chapter (Section 3.2.1) and day-type identification for the current research is the topic of Chapter 4.

2.3.3 Temperature-Load Relationship

This section examines the relationship between load and temperature, the dominant causal variable for load (as pointed out by Murray, 1996, Hyde and Hodnett, 1997b for Irish load data and Lu *et. al.*, 1989, Chen and Kao, 1996, and Hara et. al., 1997, for other systems to mention but a few).

As pointed out in Section 2.3.1, Irish load data has an underlying trend and rising variability. These characteristics are not present in the temperature readings for the same period (1986-2000) (Figure 2.7) and so the relationship between load and temperature is obscured. Section's 2.3.3.1 and 2.3.3.2 present the techniques used to pre-process the load and temperature so that the relationship between the two can be examined (Section 2.3.3.3).

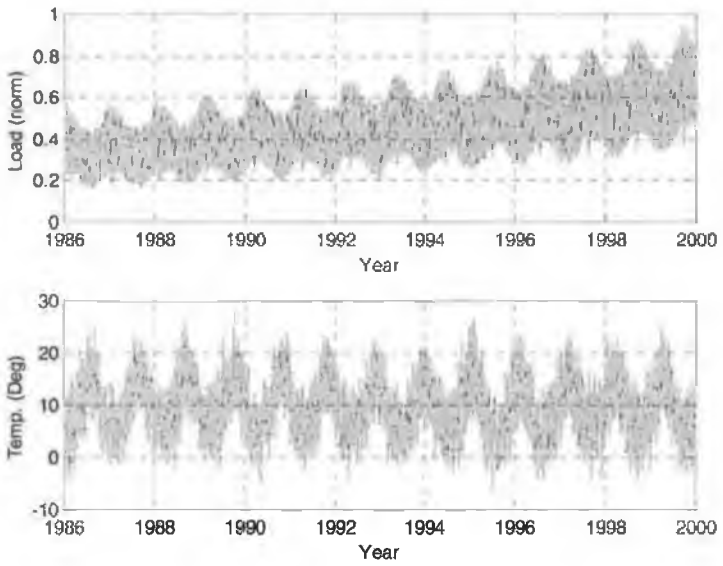

Figure 2.7 Load and temperature.

2.3.3.1 Pre-Processing of Electrical Data to Remove Trend and Variability

This section details how the growing trend and variability are removed from the data, so that the fundamental relationship between the load and temperature is revealed. This is achieved in three steps:

1. As pointed out in Section 2.3.1 the Box-Cox transform can be used to remove the variability of a time series if the variability is related to trend. This is the case with Irish electric load, so the Box-Cox transform is applied as in

Section 2.3.1. Franses and Koehler (1998) show that there are however, problems with the Box-Cox transform; primarily that a series with increasing trend (as is the case with Irish electrical demand) results in a transformed series with a trend that is *increasing* at a *decreasing* rate. As the transformed series in this study is not used for model building this is not an issue,

2. Secondly, the trend of the transformed load is removed using a quadratic curve of the form Equation (2.1). This results in the transformed and *de-trended* (the trend is removed) load series shown in Figure 2.8 (note the temperature in this figure has not been transformed in any way), and

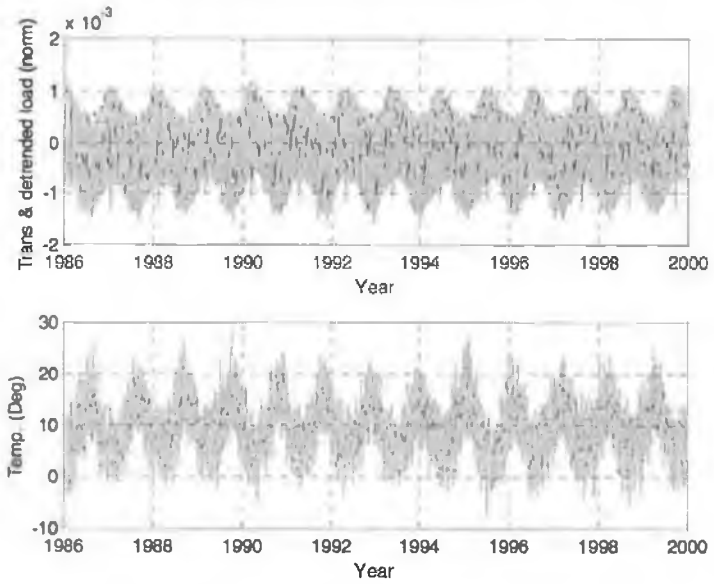

Figure 2.8 Temperature and transformed and de-trended load.

3. Finally, the transformed trend value at the end of year 2000 (the trend value of prime importance in future forecasts) is reintroduced to all the data. This is known as *level shifting*. The resultant series now has the equalised variability and the trend value of the year 2000 (Figure 2.9).

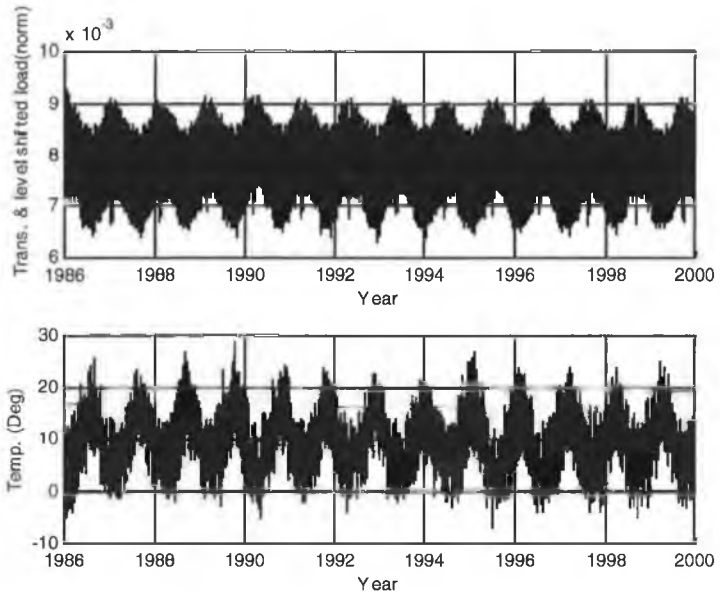

Figure 2.9 Temperature and transformed re-trended load.

The inverse Box-Cox transform (Equation 2.4) is then applied to the load series in Figure 2.9 to produce an equalised, level shifted and de-trended load series (Figure 2.10, below). The inverse Box-Cox transform can be easily derived from Equation (2.3) as:

$$z'(t) = \begin{cases} (\lambda z(t)+1)^{1/\lambda} & \text{if } \lambda \neq 0 \\ \exp(z(t)) & \text{if } \lambda = 0 \end{cases} \qquad (2.4)$$

where $z'(t)$ is the inverse-transformed series at time t and $z(t)$ is the transformed series which has been further manipulated (de-trended and level shifted).

It should be noted that the *pre-processed* (equalised and de-trended) load in Figure 2.10 has the same value as the original load at the last data point (31st March 2000). The pre-processed load is an approximation to Irish load in a system without growth. Thus, the relationship between this and temperature is not obscured.

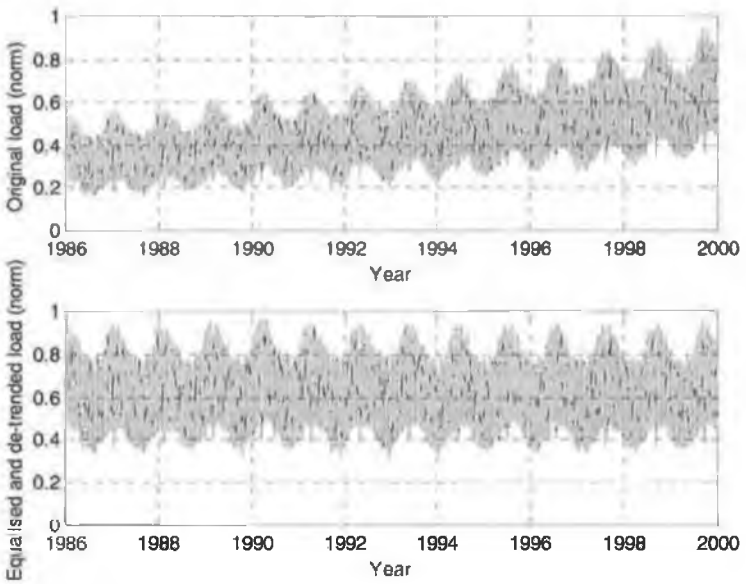

Figure 2.10 Original and equalised de-trended load (normalised).

2.3.3.2 Pre-Processing of Temperature and Load

The temperature at a particular hour is rarely the only temperature input used when forecasting the load at that hour (Dash *et. al.*, 1997, is an example of one exception). The load is typically aggregated in some fashion. Arahal and Camacho (1999), for example, use the average temperature for the day in question while Hyde and Hodnett (1997b) use several different non-linear transformations of the current and previous temperatures, to name but a few. Before explaining the reason for using more than one temperature in predicting the load, a measure called *coherence* must be defined.

The coherence (also known as the *squared coherency function*) between a time series $x(t)$ and $y(t)$ is the correlation of the power in series x at frequency f with the power in series y at frequency f (Brockwell and Davis, 1987):

$$C_{xy}(f) = \frac{\left|\left[P_{xy}(f)\right]\right|^2}{P_{xx}(f)P_{yy}(f)} \tag{2.5}$$

where $C_{xy}(f)$ is the coherence of x with y at frequency f, $P_{xx}(f)$ and $P_{yy}(f)$ are the power spectral densities of x and y respectively, $P_{xy}(f)$ is the cross power spectrum of x and y and $|\bullet|$ denotes the absolute value operator. The coherence

can be used to indicate if certain frequencies in one series are related to those frequencies in another.

The coherence is approximated in this body of work using Welch's averaged periodogram method (Brockwell and Davis, 1987). First the data is divided into non-overlapping segments each of size N_{fft}. Each segment is then smoothed using a Hanning window (Brockwell and Davis, 1987) to avoid spectral leakage. The periodogram of the smoothed segments is used to calculate the spectral densities and cross spectrum of x and y. These are then substituted directly into Equation (2.5). In this study N_{fft} was chosen to be 2,000 owing to the large size of the data set and the resolution required at the high frequency end of the spectrum.

The coherence function (Equation 2.5) of temperature with pre-processed load demonstrates that pre-processed load is only correlated to temperature at daily and yearly frequencies (Figure 2.11). The high coherence at a 12-hourly period is due to the *twin peak shape* of the daily load curve caused by the daily maximum and minimum (Figure 2.4) as noted by Moutter *et. al.* (1986). The *high frequency* (time periods less than 12 Hours) components of temperature are not correlated with high frequency components of load. This indicates that high frequency components of temperature are not particularly useful in forecasting load on any (including hourly) basis, and justifies the use of some form of low-pass filtering or aggregation of temperature data.

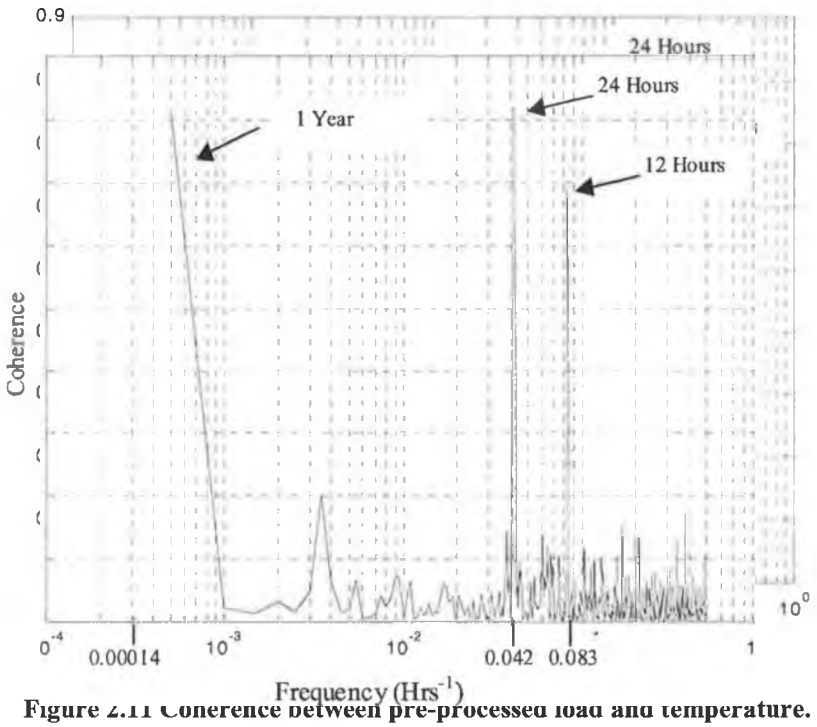

Figure 2.11 Coherence between pre-processed load and temperature.

The pre-processed temperature used in the following analysis is the average temperature, $t_{av}(k)$, defined as:

$$t_{av}(k) = \frac{1}{24}\sum_{i=0}^{23} t_{hr}(i,k) \tag{2.6}$$

where $t_{hr}(i,k)$ is the temperature at hour i on day k.

Although the focus of this research is forecasting the hourly load, the question of examining the load-temperature relationship for each hour of the day is not dealt with until Chapter 4. In the current analysis the load is aggregated as:

$$z_{av}(k) = \frac{1}{24}\sum_{i=0}^{23} z'(i,k) \tag{2.7}$$

where $z_{av}(k)$ is the average load for day k and $z'(i,k)$ is the pre-processed load for hour i on day k.

2.3.3.3 Characterisation of the Temperature-Load Relationship

A scatter plot of the pre-processed load and average temperature is shown in Figure 2.12 below. Typically, for an electrical system, the relationship between load and temperature is similar to that shown in Figure 2.12 (Fan and McDonald,

1994, Lu *et. al.*, 1989). At low temperatures (below 18 degrees for Ireland as discovered by Murray, 1996*) the correlation between load and temperature is negative due to heating requirements. Above 18 degrees there is a *deadband* or comfort zone in which the load is largely unresponsive to the temperature (Fan and McDonald, 1994). Beyond this deadband, as the temperature rises, the load again increases due to air-conditioning (Fan and McDonald, 1994). As the temperature in Ireland rarely exceeds the mid-twenties, Murray, (1996) did not identify any cooling effect.

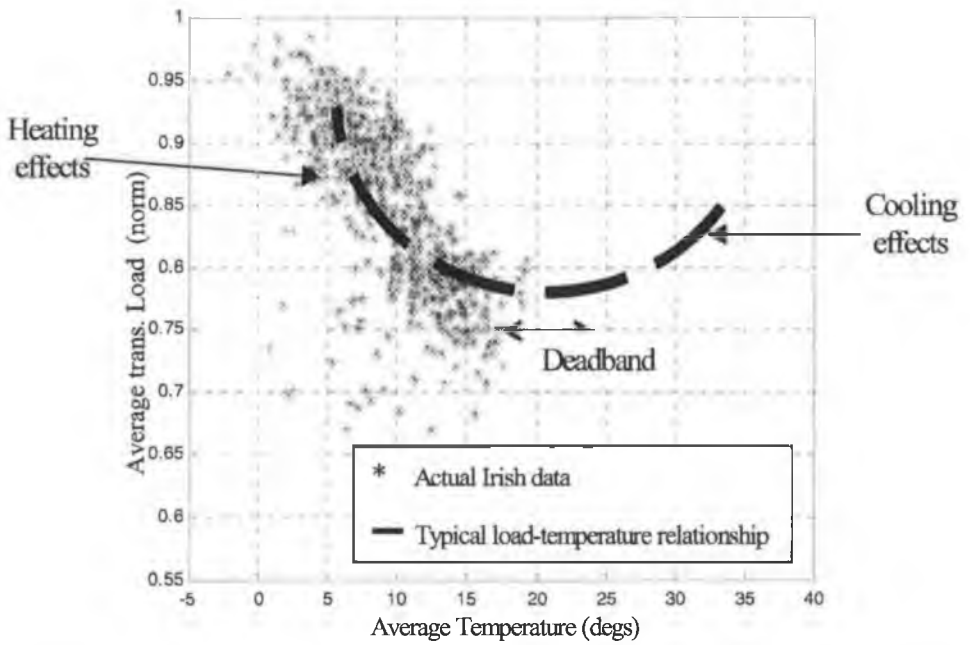

Figure 2.12 Typical and actual scatter plot of temperature-load relationship (Working days).

It should, however, be noted that the temperature-load relationship is not the same for all electrical systems. Murray (1996) pointed out that the relationship was significantly different for a regional power board in Northern New Zealand where temperature was a non-dominant input.

* Note: Murray used temperature readings from Eirgrid which although very similar to the MOI readings do have a bias of +2 degrees.

2.4 Conclusion

This chapter shows that Irish electrical load data has distinct characteristics, which have implications for model building. The following characteristics were highlighted:

- **Trend and variability**. Firstly, the data has a trend that increases at an increasing rate. Also, the variability of the load increases and is *related* to the trend. This indicates that load is a dynamic series which changes over time but in a *predictable* way. Indeed, the simple methods used to remove the trend and variability of the load in Section 2.3.3.1, although not perfect, are quite effective. The forecast horizon that is the objective of this body of research is three days. Relative to this, the rate of change of the approximated trend is very low at 1.13×10^{-11} Mw/hour* (Table 2.5) and removing the trend and variability (as is required in Chapters 5,6 and 7) can be achieved with a high level of accuracy,

- **Day-types**. The *shape* of the load on different days was examined in Section 2.3.2 and it was found that load can be classified into different day-types. The fact that the daily load shape is not consistent is important for modelling as it requires that the model must incorporate shape information, and

- **Temperature-load relationship**. The relationship of temperature to load is examined in Section 2.3.3. The results (Figure 2.12) show that the load is very responsive to the load at temperatures below 16 degrees but thereafter a deadband exists. This implies that temperature may not be a significant variable in forecasting the load for some summer months. This topic is further examined in Section 4.2.2.

* This figure is normalised relative to a maximum load of 1Mw.

Chapter 3

A Historical Review of Approaches to Electrical Load Forecasting.

3.1 Introduction.

This chapter reviews the approaches taken to load forecasting in the literature. There are many ways in which load forecasting approaches may be categorised. However, the aim here is to present the *significant* differences in these approaches. To this end the chapter is divided into the following five sections:

1. The extent of the load forecasting field (Section 3.1.1). Load forecasting is not confined solely to the short-term. Section 3.1.1 describes the wider area of load forecasting and the issues that concern electrical utilities with respect to forecasting in general. Short-term load forecasting, which is the focus of this work, is subsequently defined in more depth,

2. Disaggregation approaches in load forecasting (Section 3.2). Rather than using a single model to forecast the load for all day-types and hours of the day, many approaches segment or *disaggregate* the load and model each part separately. That is, different models may be used to forecast the load in different day-types and even at different hours,

3. Linear load forecasting techniques (Section 3.3). Regardless of the disaggregation approach, load forecasting models are broadly based on two techniques; linear and non-linear,

4. Non-linear load forecasting techniques (Section 3.4). There is a wide variety of techniques in the forecasting literature and the purpose of Sections 3.3 and 3.4 is to categorise these techniques and detail the advantages and disadvantages of each, and

5. Section 3.5 deals with the treatment of *weather* forecast errors. Many of the load forecasting techniques discussed in Sections 3.3 and 3.4 use weather inputs. Typically load forecasting models are trained with actual weather *readings* but online operation requires weather *forecasts*. The errors in these weather forecasts can however, have a disproportionate influence on the load forecast. Section 3.5 discusses the approaches taken to reduce the influence of these errors.

3.1.1 Introduction to the Load Forecasting Area.

The thesis is focused on short-term load forecasting. However, this is a subset of the larger load forecasting area. This section details the extent of the wider load forecasting area, issues that are common to all load forecasting areas and the relevance of each to STLF. Load forecasting can be broken into three areas, long-term (Section 3.1.1.1), medium-term (Section 3.1.1.2) and short-term (Section 3.1.1.3) load forecasting.

3.1.1.1 Long-Term Load Forecasting.

Long-term load forecasts refer to forecasts on a yearly time scale. These are typically total yearly electrical consumption, sales or the peak yearly demand. These forecasts are required for financial planning, transmission network and generation network expansion (Youssef, 2000). As the timescale is yearly, the data has no seasonality or cyclical behaviour.

As noted by Youssef (2000) the amount of historical data available for long term load forecasting is restricted as there is only one data point per year. The number of causal variables for long-term forecasting can be quite large in relation to the typical size of the data set. Youssef and El-Alayly (2000), for example, list sixteen causal variables for total yearly consumption with a database of only thirty-six years (data points) of historical consumption. Murray's (1996) data set consists of twenty-nine years of data with ten causal variables. A small data set also limits the techniques that can be applied and thus non-linear techniques are rarely applied (Youssef, 2000).

3.1.1.2 Medium-Term Load Forecasting.

Medium term load forecasting refers to forecasting on a timescale less than a year but greater than a day. Examples are forecasts of total weekly electrical demand (Murray, 1996 and Cloarec *et. al.*, 1998) and the monthly peak load (Barakat and Eissa, 1989, Train *et. al.*, 1984). These are required for fuel procurement and to schedule plant maintenance. In the case of total electrical weekly demand this area is quite similar to Short-Term Load Forecasting (STLF) in that the data exhibits a yearly cycle and also has special periods such as Christmas and summer holiday seasons (Cloarec *et. al.*, 1998). Thus the techniques applied in this area must deal with similar issues to those in STLF.

3.1.1.3 Short-Term Load Forecasting.

The term *short-term load forecasting* typically refers to forecasting overall system demand on an hourly basis up to seven days ahead, which has not been adjusted by demand side management*. However, the term is sometimes interpreted differently. For example, the following topics are considered part of the STLF field but are not relevant to this thesis:

- Direct Load Control (DLC), in which the operator may intervene to change the level of demand (for examples see Bhattacharyya and Crow, 1996 and Yu, 1996),

- Changing customer base, in which the system size changes in the short term due to competition from other electricity suppliers (for example Morrissey and Van Toai, 1988),

- Bus load forecasting, in which the load in a local region is forecast (for examples Kassaei *et. al.*, 1999 and Handschin and Dörnemann, 1988),

- Residential and industrial load forecasting, which is concerned with forecasting the loads of individual customers (for examples see Morrissey

* For example large industrial customers can be instructed to lower usage by the grid operator.

and Van Toai, 1988, Ihara *et. al.*, 1994, Capasso *et. al.*, 1994 and Harris and Liu, 1993), and

- Very Short-Term Load Forecasting (VSTLF), (also frequently referred to as short-term load forecasting) which refers to forecasting the load for horizons up to 30 minutes ahead on a minute by minute basis (for example Liu *et. al.*, 1996).

In contrast, the area of daily peak forecasting (for example Dash *et. al.*, 1995) is relevant as the peak daily load is merely a point in the daily load curve. The majority of short-term load forecasting literature is relevant to the current research.

The main forecasting accuracy measure used in STLF is the Mean Absolute Percentage Error (MAPE), defined as:

$$MAPE = \sum_{i=1}^{N} \frac{|y(i)-\hat{y}(i)|}{y(i)} \frac{1}{N} \tag{3.1}$$

where $y(i)$ is the actual load at time i, $\hat{y}(i)$ is the load forecast and N is the number of points used. The reason this measure is used is that it allows comparison between different electrical systems without revealing the size of the system or the variance of the forecasting errors (unlike for example, the mean squared error), which are often considered confidential information.

However, electrical utilities, including Eirgrid, consider large forecast errors to be proportionately more costly than small errors. Thus, a measure such as the Mean Squared Error (MSE) would seem more appropriate as it involves the squared error which penalises large errors more heavily than small errors:

$$MSE = \sum_{i=1}^{N} (y(i)-\hat{y}(i))^2 \frac{1}{N} \tag{3.2}$$

It is therefore common that models are trained to minimise the MSE but are evaluated using the MAPE. Although minimising the MSE of a model does not

guarantee a minimum MAPE this is the convention in this area and will be followed here.

3.2 Disaggregation approaches in Load Forecasting.

Disaggregation of electrical load data refers to the segmentation of the data set into several distinct disaggregated sets, each of which is modelled separately. This approach may be described as a 'divide and conquer' approach. It is proposed here that disaggregation is advantageous if the disaggregated sets are sufficiently different. Load data is typically disaggregated at two levels:

1. Day-type (Section 3.2.1), and

2. Hour of the day (Section 3.2.2).

These disaggregations are not mutually exclusive and the load may be disaggregated both by day-type and hour of the day or at one level only.

3.2.1 Day-Type Disaggregation.

The existence of several different day-types has been shown by several researchers (Muller and Schatzel,1999, Muller and Petrisch,1998, Ho *et al.*, 1990, Bretschneider *et al.*, 1999, Hsu and Yang, 1991, to mention a few). Muller and Schatzel (1999) for example, identified five primary classes:

- Summer days,
- Cold, early and late summer days,
- Spring and autumn days,
- Early and late winter days, and
- Winter days.

Each of these primary classes contains seven secondary classes:

- Mondays and working days after a holiday,
- Tuesdays, Wednesdays and Thursdays,
- Fridays and working days before a holiday,
- Saturdays,
- Sundays,
- Holidays, and
- Special days.

The total number of day-types is thus thirty-five. In an earlier study, Muller and Petrisch (1998) identified only summer and winter primary classes for the same data. This shows how the level of disaggregation in day-type selection is, to a large extent, subjective and dependant on the judgement of the forecaster.

As pointed out by Hubele and Cheng (1990), the application of a separate load forecasting model for different seasons (i.e. summer, autumn, winter and spring) has the advantage that the models need not incorporate seasonal information. Further disaggregation of the load by day of the week (for example summer Sunday, winter Sunday, summer Monday etc.) reduces further the amount of information that a model need incorporate. Such approaches have been implemented successfully by Srinivasan *et al* (1999) and Mastorocostas *et al* (1999), to mention but a few.

Where a single model is used for all the data, the day-type information is often incorporated as an additional input (two examples are Chen *et al*, 1992 and Lertpalangsunti and Chan, 1998.). In either case the day-types must be identified.

3.2.1.1 Techniques for Day-Type Identification.

The selection of day-types can be guided by analytical techniques. Three candidate techniques were considered for day-type identification:

1. Interviews with system operators (for example Ho *et al.*, 1990),
2. Clustering algorithms (for example Bretschneider *et al.*, 1999), and
3. Self-organising feature maps (for example Hsu and Yang, 1991 and Pelikan *et. al.*, 1996).

Lonergan (1994) presented interviews with Irish system operators. The interviews indicate that there were considered to be only two seasons in the year; winter and summer. There is no disaggregation by day of the week and so the total number of day-types identified is two.

Clustering algorithms group data together into *clusters* (sets). Often a cluster can be assigned to a single unusual data point while another cluster represents two close but distinct groups (Jain and Dubes, 1998). These algorithms are best applied when the *a-priori* existence of the number of clusters is known.

The self-organising feature map or Kohonen map (Kohonen, 1990) is a technique which maps a continuous input space to a discrete output space. The discrete output space provides an approximation to the input space (Haykin, 1999). The output space is typically organised as a two dimensional grid (Figure 3.1, below) in which similar inputs will be mapped onto the same area of the grid. In terms of day-type identification this means that similar days will be mapped to a particular area of the grid. These areas can then be identified as day-types. Unlike cluster algorithms, the number of day-types need not be pre-specified and the proximity of the identified day-types is known.

The Kohonen map can be implemented for day-type identification in several different ways, (examples are Bretschneider *et al.*, 1999, Muller and Petrisch, 1998, Hsu and Yang, 1991); however differences in the results are insignificant in most cases. The algorithm used by Hsu and Yang (1991) is now presented as an example. The Kohonen map structure is diagrammatically shown in Figure 3.1 below.

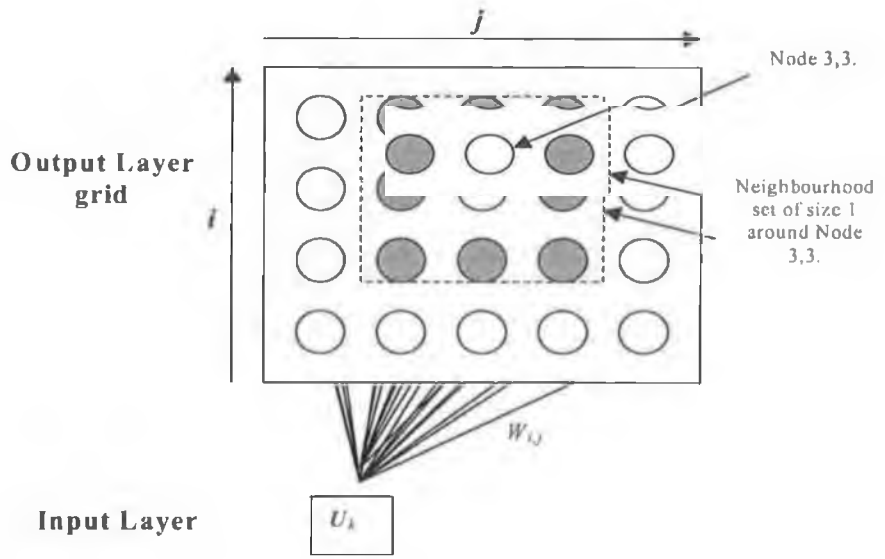

Figure 3.1 Kohonen map structure.

The network consists of a grid of output nodes connected to the inputs via a set of weights. When presented with the k^{th} input vector $\boldsymbol{U}_k$, $\in R^{1\times n}$, the network calculates the *activation* of each node by $\boldsymbol{U}_k$ as:

$$a_{i,j,k} = \boldsymbol{W}_{i,j}\boldsymbol{U}_k \tag{3.3}$$

where $a_{i,j,k}$ and $\boldsymbol{W}_{i,j}$ are the activation of, and weight ($\in R^{n\times 1}$) connecting $\boldsymbol{U}_k$ to, node *i,j* respectively. $\boldsymbol{U}_k$ is said to be *mapped* onto the node with the highest activation. After several inputs have been presented, similar inputs are mapped to the same or adjacent nodes, i.e. within a small *neighbourhood.* A neighbourhood of size *Nc* around node *i,j* is defined as nodes $i \pm Nc$ to $j \pm Nc$.

Hsu and Yang (1991) construct the input, $\boldsymbol{U}_k$, for load forecasting in two steps. Initially, the daily load curve is extracted from each day to give a set of load curves that have a minimum value of zero and a maximum value of one (Hsu and Yang, 1991):

$$\boldsymbol{Y}'(i)_k = \frac{\boldsymbol{Y}(i)_k - \min(\boldsymbol{Y}_k)}{\max(\boldsymbol{Y}_k) - \min(\boldsymbol{Y}_k)} \qquad i = 1,2,3...24 \tag{3.4}$$

where $\boldsymbol{Y}'(i)_k$ and $\boldsymbol{Y}(i)_k$ are the i^{th} elements (hour) of the load curve $\boldsymbol{Y}'_k$, $\in R^{1\times 24}$, and actual load $\boldsymbol{Y}_k$, $\in R^{1\times 24}$, of day *k* respectively. The load curves are then normalised to give them unity length:

$$\boldsymbol{U}(i)_k = \left\|\boldsymbol{Y}'(i)_k\right\| = \boldsymbol{Y}'(i)_k \Big/ \left(\sum_{j=1}^{24} \boldsymbol{Y}'(j)_k^2\right)^{1/2} \qquad i = 1,2,3,...,24 \tag{3.5}$$

where $\boldsymbol{U}(i)_k$ is the i^{th} element of $\boldsymbol{U}_k$, and $\|\bullet\|$ is the normalisation operator. The weights are initialised as (Hsu and Yang, 1991):

$$\boldsymbol{W}_{i,j} = \left\| \left[\mu_u(1), \mu_u(2), ..., \mu_u(24)\right] + 5\Lambda\left[\sigma_u(1), \sigma_u(2), ...\sigma_u(24)\right] \right\| \tag{3.6}$$

where $\mu_u(i)$ and $\sigma_u(i)$ are the mean and standard deviation of $\boldsymbol{U}(i)$ over all *k*, Λ is a uniformly distributed random number in the range -0.5 to 0.5 and $\boldsymbol{W}_{i,j}$ is normalised to unit length, in a manner similar to Equation (3.5). The weights are not initialised randomly but initialised around the mean of the inputs so that the

corresponding outputs are well distributed in the output grid (Hsu and Yang, 1991).

During training the inputs are presented one by one and the weights of the *triggered* node (the node to which the inputs is mapped) and nodes in its neighbourhood are updated (Hsu and Yang, 1991) via:

$$W_{i,j}(m+1) = W_{i,j}(m) + \alpha(m)[U_k - W_{i,j}(m)] \qquad (3.7)$$

Where α is the adaptation gain, with $0 < \alpha < 1$, and m is the iteration number. This has the effect of increasing the activation of the triggered node and it's neighbours. In a single iteration all the inputs are presented and the weights adapted. After several iterations, the neighbourhood size is reduced by one and so on until zero, i.e. at this point the triggered node only is adapted.

3.2.1.2 Techniques for Determining the Transition between Day-Types.

Cloarec *et al.* (1998) demonstrated that for Irish weekly demand there is a transitional period between winter and Christmas. Analytical techniques, such as clustering algorithms (for example Imai *et al.,* 1998) and Kohonen maps (for example Muller and Petrisch 1998), have been used to determine the transitions between day-types. However, Kohonen maps are not ideal for this task. Consider a day lying in the transition between summer and winter. This day will trigger a node in between the summer and winter regions of the output nodes. However, as pointed out by Song and Hopke (1996) the relative position of two nodes in a Kohonen Map is not an exact measure of the proximity between the inputs that trigger those nodes.

Cloarec *et al.* (1998) and Rahman and Bhatnagar (1988) determine the transition between the loads in two seasons using model forecast errors. The load in question (weekly in Cloarec's study and hourly in Rahman and Bhatnagar's) is forecast using both seasonal models and the ratio of model errors is used to determine the transition. This technique is desirable for any online system, as it tracks the errors of that system. However, it is dependant on the performance of load forecasting models and not the characteristics of the data.

Cluster analysis methods are used mainly as a first step in designing a fuzzy logic system based on the data as opposed to *a priori* information (Jain and Dubes, 1998). There are several types of clustering algorithms such as *K-means clustering*, *nearest neighbour*, *polynomial interpolants* and *adaptive vector* (for an overview see Harris *et al.*, 1993). Other algorithms such as the adaptive fuzzy classification algorithm suggested by Bretschneider *et al.* (1998) can be adaptive and change with the dynamic behaviour of the data. By far, the most popular clustering algorithm is the *Fuzzy C-Means* (FCM) algorithm (Dunn, 1974). As pointed out by Jain and Dubes (1998), there are no guidelines for *a-priori* choice of the correct algorithm based on the data. As the FCM algorithm is the most popular it is now presented.

The FCM algorithm seeks to break the data into N clusters. Each cluster is characterised by a *cluster centre* $\boldsymbol{C}_i \in \mathbf{R}^{1\times Q}$, which represents a point at the centre of the cluster and Q is the dimension of the inputs. Given an input vector $\boldsymbol{U}_k \in \mathbf{R}^{1\times Q}$, instead of assigning it to a single cluster its proximity to all clusters is quantified by a *potential* defined as:

$$\mu_{j,k} = L(\boldsymbol{U}_k,\boldsymbol{C}_j)^{\frac{-2}{\kappa-1}} \Big/ \sum_{i=1}^{N} L(\boldsymbol{U}_k,\boldsymbol{C}_i)^{\frac{-2}{\kappa-1}} \qquad j=1...N \qquad (3.8)$$

where $\mu_{j,k} \in [0,1]$, is the potential of $\boldsymbol{U}_k$ to cluster centre $\boldsymbol{C}_j$, κ is a scalar which determines the level of *fuzziness* (explained below) of the resulting potentials, and $L(\boldsymbol{U}_k,\boldsymbol{C}_j)$ is the distance between $\boldsymbol{U}_k$ and $\boldsymbol{C}_j$ defined as:

$$L(\boldsymbol{U}_k,\boldsymbol{C}_j) = \sqrt{\sum_{i=1}^{Q}\left(\boldsymbol{U}_k(i)-\boldsymbol{C}_j(i)\right)^2} \qquad (3.9)$$

where $\boldsymbol{U}_k(i)$ and $\boldsymbol{C}_j(i)$ are the i^{th} elements of $\boldsymbol{U}_k$ and $\boldsymbol{C}_j$ respectively.

The level of *fuzziness* determines the degree of overlap between clusters. A value of one for κ leads to no overlap. As κ is increased above one the overlap is increased.

In addition to Equations (3.8) and (3.9), it is required that the sum of potentials for $\boldsymbol{U}_k$ equals unity:

$$\sum_{j=1}^{N} \mu_{j,k} = 1 \tag{3.10}$$

It should be noted that although $\mu_{j,k}$ has similar characteristics as the probability that $\boldsymbol{U}_k$ is a member of cluster $\boldsymbol{C}_j$ (i.e. $\mu_{j,k}$, $\in [0,1]$ and the sum of the memberships equals one), the two measures are not the same (Jain and Dubes, 1998). The potential represents the *membership* (degree to which an input can be classified as a member of a cluster) of an input to a cluster. While an input can have membership of two or more clusters, the probability that it is a member of a cluster implies that it belongs to one cluster alone. Thus, the potential is an ideal measure of the membership of a day to a certain day-type.

To apply the algorithm, the cluster centres, $\boldsymbol{C}_j$, and level of fuzziness, κ, must be determined. The aim is to minimise overall proximity of the data to the cluster centres. This is achieved by minimising the cost function:

$$J = \sum_{l=1}^{M} \sum_{i=1}^{N} (\mu_{i,l})^{\kappa} L(\boldsymbol{U}_l, \boldsymbol{C}_i) \tag{3.11}$$

where J is the cost function to be minimised and M is the number of input vectors. The algorithm is initialised using random potentials. The cluster centres are then calculated via:

$$C_j = \sum_{i=1}^{M} \mu_{j,i}^{\kappa} \boldsymbol{U}_i \Big/ \sum_{i=1}^{M} \mu_{j,i}^{\kappa} \tag{3.12}$$

The potentials are then updated via Equation (3.8) and the new cluster centres and value of J are calculated via Equations (3.11) and (3.12). This process continues until the value of J reaches a pre-specified level or the number of iterations reaches a pre-specified maximum.

3.2.2 Hour of the Day Disaggregation.

STLF forecasting models fall broadly into two categories; *sequential* models (examples of which can be found in Choueiki *et. al.*, 1997, Papadakis *et. al.*, 1999) and *parallel* models (examples of which can be found in Infield and Hill, 1998 and Gupta, 1985). The sequential approach models short-term (hourly) load as a single series while the parallel approach models each hour of the day as a separate series. The difference in approaches is shown diagrammatically in Figure 3.2 below. A daily series composed of the loads at a single hour is called a *partitioned series*. Figure 3.2 shows the construction of, for example, the partitioned series for 1 p.m. and 6 p.m.

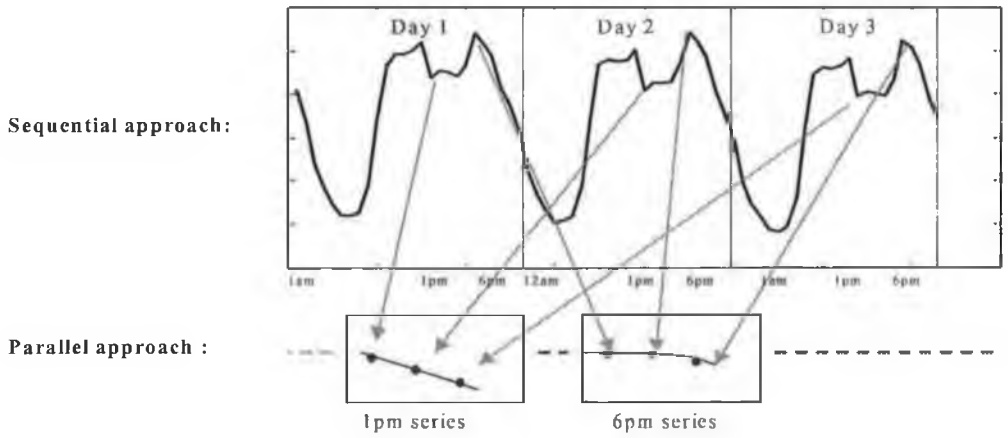

Figure 3.2 Constructing partitioned data series from electrical load data.

A general equation for the load on hour i of day k, $y_i(k)$ may be expressed as a function, f, of previous loads, current and previous inputs and an error term as:

$$\begin{aligned} y_i(k) = f(&y_{i-1}(k),\ldots, y_{i-N}(k), y_{i-1}(k-1),\ldots, \quad ,\ldots, y_{i-N}(k-P),\\ &\boldsymbol{U}_{i-1}(k),\ldots,\boldsymbol{U}_{i-M}(k),\boldsymbol{U}_{i-1}(k-1),\ldots, \quad ,\ldots,\boldsymbol{U}_{i-M}(k-Q),\\ &i,k) + \varepsilon_i(k) \end{aligned} \tag{3.13}$$

where $\boldsymbol{U}_i(k)$ is a vector of causal variables on hour i of day k, $\varepsilon_i(k)$ is an error term, N, M are the orders of the *hourly regressors* (N,M <24) and P,Q are the orders of the *daily regressors*. Note that k is included as a factor in Equation (3.13) to reflect that, due to the long-term trend, load is a non-stationary process. Indexing the load by both hour and day, though cumbersome, is useful in pointing out the difference between the parallel and sequential approaches to load forecasting.

The *sequential* approach uses just one function, f_s, relating the current load to previous loads and inputs and so Equation (3.13) becomes:

$$y_i(k) = f_s(y_{i-1}(k),..., y_{i-N}(k), y_{i-1}(k-1),..., \quad ,..., y_{i-N}(k-P), \\ \boldsymbol{U}_{i-1}(k),..., \boldsymbol{U}_{i-M}(k), \boldsymbol{U}_{i-1}(k-1),..., \quad ,..., \boldsymbol{U}_{i-M}(k-Q)) + \varepsilon_i(k) \quad (3.14)$$

However, in the *parallel* approach, the data is partitioned so that each hour of the day is modelled by a separate function and so Equation (3.13) becomes:

$$y_0(k) = f_0\left(y_0(k-1),..., y_0(k-P_0), \boldsymbol{U}_0^*(k-1),..., \boldsymbol{U}_0^*(k-Q_0)\right) + \varepsilon_0(k)$$
$$y_1(k) = f_1\left(y_1(k-1),..., y_1(k-P_1), \boldsymbol{U}_1^*(k-1),..., \boldsymbol{U}_1^*(k-Q_1)\right) + \varepsilon_1(k)$$
$$\vdots$$
$$y_{23}(k) = f_{23}\left(y_{23}(k-1),..., y_{23}(k-P_{23}), \boldsymbol{U}_{23}^*(k-1),..., \boldsymbol{U}_{23}^*(k-Q_{23})\right) + \varepsilon_{23}(k) \quad (3.15)$$

where $\boldsymbol{U}_i^*(k)$ is the input vector for hour i of day k (which may now include the load at previous hours), $y_i(k)$ is the load at hour i on day k (for example $y_1(k)$ is the load at 1 hrs or 1 a.m. on day k etc.), f_i is the *parallel model* for hour i. P_i and Q_i are the orders of the regressors for *partitioned series i*.

Note that, in the case of a sequential approach, the order of the regressors for each hour of the day is fixed while in the parallel approach the regressor order is variable if required. Similarly, in the parallel case f_i can vary from hour to hour. However, for the sequential approach, the same function applies to all hours of the day.

To produce a multi-step ahead forecast using a sequential approach requires that the forecasts be produced *iteratively* (Figure 3.3). This is because actual values of the time series are not available past the forecasting origin and forecast values must be used instead. As pointed out by Bretschneider *et. al.* (1998) the error in the first and subsequent forecasts is propagated and tends to build up. This phenomenon is known as *propagation error*. However, parallel models have the advantage that they may exclude the hourly regressors, which are not available at the forecast origin and thus 24 hour ahead forecasts can be produced without propagation errors.

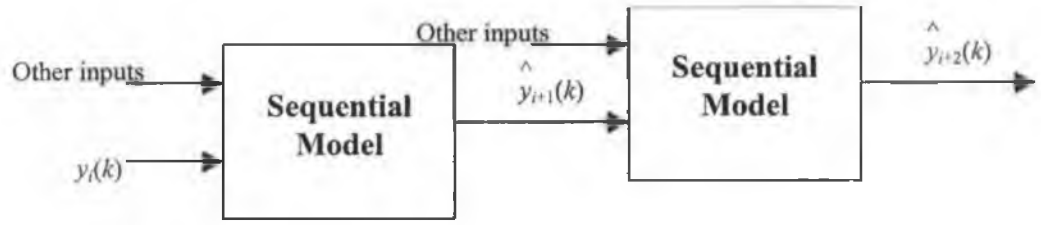

Figure 3.3 Producing a multi-step ahead forecast using a sequential model.

The parallel approach may make the modelling task more difficult as:

- Although $f_0 \ldots, f_{23}$ no have the hour of the day as an input (i.e. i is excluded from Equation (3.15)), this may not result in $f_0, \ldots, f_{23}$ being any less complex than f_s. The partitioned series are created by daily sampling of load (Figure 3.1). As shown by Harvey (1981), a sub-sampled (i.e. taking every p^{th} sample) AR (Auto-Regressive) process is itself an AR process of equal or *higher* order. That is, it is likely to be a more complex process. ARMA processes, of which AR processes are a subset, are explained in Section 3.3.1. However, at this stage it is sufficient to say that ARMA processes covers a large class of linear processes. Short term load has been modelled by several authors as an ARMA process with varying degrees of success by several authors (examples are Vemuri *et. al.* 1981, Barakat *et. al.* (1990) and Elkateb *et. al.* (1998)),

- The number of parameters that need to be estimated in the parallel approach (24 sets of parameters) exceeds that of the sequential approach, where only one set of parameters needs to be estimated (Lee *et. al.*, 1992),

- The data set is partitioned into 24 separate time-series (Figure 3.2), reducing the number of input-output pairs for training of the model, and

- Training 24 separate models can be overly computationally expensive.

Examples of a sequential approach can be found in Barakat *et. al.* (1990), Connor *et. al.* (1992), Darbellay and Slama (2000) and Vermuri *et. al.* (1981) to mention but a few. As observed by Connor *et. al.* (1992), this approach can lead

to excellent results if the hour of the day dependence is not a dominant factor for the electrical system being modelled. In that study, Connor *et. al.* (1992) observed that sequential approaches which ignored hour of the day dependence were superior to those that did not (in this case two types were examined; a recurrent neural network (Section 3.4.3.2) and a *Multi Layer Perceptron* (MLP, see Section 3.4.3.1). This was reported as being due to the increased complexity of the modelling task. Darbellay and Slama (2000) similarly observed that an ARMA model, which ignores hour of the day dependence, was superior to a feedforward neural network (Section 3.4.3.1), which used the hour of the day as an additional input. However, differences in the type of model employed, prevent a genuine evaluation of the effect of including this time dependence.

The parallel approach has also been used by many authors (Chow and Leung, 1996 and Ramanthan *et. al.*, 1997 to mention a few). In the study by Connor *et. al.* (1992), it was found that the parallel approach vastly improved the forecasting performance, where recurrent neural networks (Section 3.4.3.2) were used as the modelling tool. In fact, this was found to be the optimal technique. In contrast, Lee *et. al.* (1992) found the performance of parallel and sequential models which used MLPs (Section 3.4.3.1), indistinguishable. Interestingly, although not explicitly stated in the paper, the parallel model gave superior results for some hours of the day. In a similar study, also using MLP neural networks, Lu *et. al.* (1993) found that the sequential approach was superior to the parallel approach.

The only consistent conclusion to be drawn from the literature is that the choice of sequential or parallel modelling is highly dependent on the particular power system being analysed.

A number of other studies (Gupta, 1985, Khotanzad *et. al.*, 1996, Murray *et. al.*, 2000) have examined combining the sequential and parallel approaches. This combined approach is known as the Multi-TimeScale (MTS) approach and adjusts the forecasts of a sequential model with those of parallel models. For example, Gupta (1985) first forecasts the load curve for the following day using a sequential model. A parallel model is then used to forecast the load at 6 p.m. (the daily peak load). The difference between the load curve forecast at 6 p.m. and the

parallel model forecast for 6 p.m. is then used to adjust the whole load curve forecast. The topic of MTS models is further examined in the section on linear models (specifically Section 3.3.4).

3.3 Linear Techniques for Load Forecasting.

A linear system is one which satisfies the principal of *superposition* (Sinha, 1991). Superposition is composed of two concepts, *additivity* and *homogenity* (Sinha, 1991). Additivity means that the output of a linear system given the sum of two inputs is equal to the sum of the outputs from each input individually:

$$f(u_1(t)+u_2(t)) = f(u_1(t)) + f(u_2(t)) \tag{3.16}$$

where f is a linear function and $u_{1,2}$ are inputs. Homogenity means that multiplying an input by a constant results in the output being multiplied by the same constant:

$$f(cu_1) = cf(u_1(t)) \tag{3.17}$$

where c is a constant. A simple exception to this is an *affine* (a linear function with an offset) function:

$$f(u(t)) = au(t) + c \tag{3.18}$$

where a and c are constants. In this case neither additivity nor homogenity apply, though the function still falls into the realm of linear systems (Seber and Wild, 1989). The definition of a linear system used in this text is that adopted by Seber and Wild (1989), in which a system is linear if the partial derivatives of the system outputs to the input variables are constant in form.

The various linear techniques differ in their form of representation of a linear system. The Box-Jenkins techniques of Section 3.3.1, for example, use difference equations, state space form is used in Sections 3.3.2 and 3.3.4, while the Bayesian techniques explained in Section 3.3.3 can be applied to both representations.

3.3.1 Box-Jenkins Techniques.

Box and Jenkins (1970) laid many of the foundations of classical time-series analysis (as noted by Harvey, 1981, among others). Their techniques have been used in several load forecasting applications (Di Caprio *et. al.*,1985, Rajukar and Newill, 1985 to mention a few) and are often used as a baseline for comparison with other techniques (examples are Ho *et. al.*,1990, Murray, 1996, Infield and Hill, 1998, Di Caprio *et. al.*,1985, Charytoniuk *et. al.*,1999 to mention but a few).

Box-Jenkins techniques involve modelling a time-series as a function of previous outputs, output errors and external inputs (three commonly used textbooks on the subject are, Bowerman and O'Connell, 1987, Brockwell and Davis, 1987 and Cryer, 1986). However, a *stationary* time series is required in the later stages of Box-Jenkins techniques (Box and Jenkins, 1970). A stationary time series is one for which the statistical properties are invariant to a shift in the origin (Papoulis, 1991). However, this definition is quite strict as it includes all the statistical properties of the series and the alternative definition of *wide sense stationarity* is used (Brockwell and Davis, 1987). A series is wide sense stationary if its mean and covariance are invariant to a shift in the origin (Papoulis, 1991).

Box and Jenkins (1970) suggested that *prior* to modelling, the time series should be transformed into a stationary time series with a *stationarity transform*, explained below.

3.3.1.1 Stationarity Transformations.

A stationarity transform is one which produces a stationary time series from a non-stationary time series. As pointed out by Priestly, (1981) any linear time series can be transformed into a stationary times series by means of an appropriate stationarity transform. Many stationarity transformation techniques exist but the appropriate one to use depends on the time series being transformed.

Box and Jenkins (1970) proposed *differencing* as a stationarity transform. Differencing is based upon the delay and differencing operators, q^{-1} and ∇,

respectively. The general differencing stationarity transform for a seasonal series can be composed of both seasonal and non-seasonal differencing (Bowerman and O'Connell, 1987) as:

$$z(k) = \nabla_s^D \nabla^d x(k) = (1 - q^{-s})^D (1 - q^{-1})^d x(k) \tag{3.19}$$

where $z(k)$ is the transformed series, $x(k)$ is the original series, ∇_s^D is a seasonal differencing operator with a season s and D is called *the order of seasonal differencing*, ∇^d is a non-seasonal differencing operator of order d. D and d are determined by use of the *Sample Auto Correlation Function* (SACF) which is explained below.

The SACF of a time series $x(k)$ represents the linear dependence between observations separated by a lag, and may be expressed (Bowerman and O'Connell, 1987) as:

$$\hat{r}_x(\tau) = \frac{\sum_{\tau=1}^{n-\tau} (x(\tau) - \bar{x})(x(k+\tau) - \bar{x})}{\sum_{\tau=1}^{n} (x(\tau) - \bar{x})^2} \tag{3.20}$$

where $\hat{r}_x(\tau)$ is the SACF value for a lag of τ, n is the number of observations used and $\bar{x}$ is the average value of $x(k)$. Note that the SACF is an estimate of the auto-correlation function, as sample data is used.

The orders of differencing, D and d, typically lie in the range zero to two (Bowerman and O'Connell, 1987). To determine if the transformed series is stationary Box and Jenkins (1970) suggest repeatedly applying differencing, increasing D or d at each iteration. At each iteration the series is tested to see if it is stationary using the SACF. Box and Jenkins (1970) suggest that a process may be considered non-stationary if the SACF *dies away slowly*, which is determined subjectively with experience. Ng and Young (1990), however, point out that differencing can only be used with time series that are non-stationary and not, for example, with time-series which have non-constant parameters. Another problem with differencing is that subtracting one value of a time-series from another (i.e.

differencing) results in the noise at both values being present in the differenced value.

Other stationarity transforms include the Box-Cox transform (Section 2.3.1), removal of a linear trend (Ljung, 1987), removing a polynomial trend (Brockwell and Davis, 1987) and smoothing by means of a moving average trend (Brockwell and Davis, 1987) to mention but a few.

3.3.1.2 Model Structures.

The second stage of the Box-Jenkins methodology involves modelling a stationary time series. Box and Jenkins suggest several types of models for this purpose. The general form of the Box-Jenkins model is known as the *transfer function model* and can be expressed as a function of previous values of the time series, external inputs and previous model errors (Box and Jenkins, 1970) as:

$$x(k) = \frac{B(q)}{F(q)} u(k-\tau) + \frac{C(q)}{D(q)} \varepsilon(k) \tag{3.21}$$

where

- $x(k)$ is the stationary time series to be modelled at time k (assuming $q \leq 1$),
- $u(k)$ is an external input with delay τ between input and $x(k)$,
- $\varepsilon(k)$ is a white noise term at time k, and
- B,F,C,D are polynomials in the delay operator q^{-1} such that, for example, B may be expanded as:

$$B(q)x(k) = (1 - b_1 q^{-1} - b_2 q^{-2} \cdots - b_{na} q^{-nb}) x(k) \tag{3.22}$$

where $b_{1,\ldots,nb}$ are the coefficients of the polynomial and nb is the order. F,C and D can be similarly expanded with coefficients $f_{1,\ldots,nf}$, $c_{1,\ldots,nc}$, $d_{1,\ldots,nd}$ and orders of nf, nc and nd, respectively. The other models proposed by Box and Jenkins (1970) can be derived from Equation (3.21) by letting F and D equal one and adding a regressor term $A(q)$ which acts on $x(k)$ as shown in Table 3.1 below.

Table 3.1 Box-Jenkins models and associated polynomials (F=1, D=1).

Model name	A	B	C
AR (Auto Regressive)	$1\text{-}a_1q^{-1},\ldots,a_{na}q^{-na}$	0	1
ARX (AR eXogeneous)	$1\text{-}a_1q^{-1},\ldots,a_{na}q^{-na}$	$b_1+b_2q^{-1},\ldots,b_{nb}q^{-nb}$	1
MA (Moving Average)	1	0	$1+c_1q^{-1},\ldots,c_{nc}q^{-nc}$
MAX (MA eXogeneous)	1	$b_1+b_2q^{-1},\ldots,b_{nb}q^{-nb}$	$1+c_1q^{-1},\ldots c_{nc}q^{-nc}$
ARMA (AR – MA)	$1\text{-}a_1q^{-1},\ldots,a_{na}q^{-na}$	0	$1+c_1q^{-1},\ldots c_{nc}q^{-nc}$
ARMAX (ARMA eXogeneous)	$1\text{-}a_1q^{-1},\ldots,a_{na}q^{-na}$	$b_1+b_2q^{-1},\ldots,b_{nb}q^{-nb}$	$1+c_1q^{-1},\ldots c_{nc}q^{-nc}$

In addition, if $x(k)$ has been produced using a differencing transform on an original series, then the model is called an *Integrated* model and the letter 'I' may be added to the model name. For example, an ARMAX model becomes an ARIMAX model, an AR model becomes an ARI model, etc. In this case the time series in Equation (3.22), $x(k)$, represents a stationarity transformed series and Equations (3.19) and (3.22) may be combined (with F and D equal to 1) (Bowerman and O'Connell, 1987) to give:

$$A(q)\nabla_s^D\nabla^d(x(k)) = B(q)u(k-\tau) + C(q)\varepsilon(k) \tag{3.23}$$

Box and Jenkins (1970) also examined the case where the series to be forecast is seasonal. In order to include the seasonal aspects of the data, Box and Jenkins (1970) used the idea of *seasonal* and *non-seasonal operators* to adjust Equation (3.23) (Bowerman and O'Connell, 1987) as:

$$A_s(q)A_{ns}(q)\nabla_s^D\nabla^d(x(k)) = B(q)u(k-\tau) + C_s(q)C_{ns}(q)\varepsilon(k) \tag{3.24}$$

where

- A_s is the seasonal AR operator defined as:

$$A_s(q) = (1 - a_{s,1}q^{-s},\ldots,-a_{s,nsa}q^{-nsa*s}) \tag{3.25}$$

  where $a_{s,1},\ldots,a_{s,nsa}$ are the seasonal coefficients of order nsa,

- A_{ns} is the non-seasonal AR operator defined as:

$$A_{ns}(q) = (1 - a_1q^{-1},\ldots,-a_{na}q^{-na}) \tag{3.26}$$

  where $a_1,\ldots,a_{na}$ are the non-seasonal coefficients of order na,

- C_s is the seasonal MA operator defined as:

$$C_s(q) = (1 - c_{s,1}q^{-s}, \ldots, -c_{s,nsc}q^{-nsc*s}) \tag{3.27}$$

where $c_{s,1},\ldots,c_{s,nsc}$ are the seasonal coefficients of order *nsc*, and

- C_{ns} is the non-seasonal MA operator defined as:

$$C_{ns}(q) = (1 - c_1 q^{-1}, \ldots, -c_{nc}q^{-nc}) \tag{3.28}$$

where $c_1,\ldots,c_{nc}$ are the non-seasonal coefficients of order *nc*.

The model described by Equation (3.24) is called a Seasonal Auto Regressive Integrated Moving Average eXogenous (SARIMAX) model.

3.3.1.3 Model Identification.

The next step in the Box-Jenkins procedure involves calculating the orders of the seasonal and non-seasonal AR and MA operators. Before explaining the procedure, the *Sample Partial Auto Correlation Function* (SPACF) must first be defined.

The SPACF of a time series is defined (Bowerman and O'Connell, 1987) as:

$$\hat{r}_x(\tau,\tau) = \begin{cases} \hat{r}_x(1) & \text{if } \tau = 1 \\ \dfrac{\hat{r}_x(\tau) - \sum_{j=1}^{\tau-1} \hat{r}_x(\tau-1,j)\hat{r}_x(\tau-j)}{1 - \sum_{j=1}^{\tau-1} \hat{r}_x(\tau-1,j)\hat{r}_x(j)} & \text{if } \tau = 2,3,\ldots. \end{cases} \tag{3.29}$$

where $\hat{r}_x(\tau,\tau)$ is the SPACF at a lag of τ, $\hat{r}_x(\tau)$ is the SACF at a lag of τ, and $\hat{r}_x(\tau,j)$, for $j \neq \tau$, is defined (Bowerman and O'Connell, 1987) as:

$$\hat{r}_x(\tau,j) = \hat{r}_x(\tau-1,j) - \hat{r}_x(\tau,\tau)\hat{r}_x(\tau-1,\tau-j) \quad \text{for } j = 1,2,\ldots,\tau-1 \tag{3.30}$$

If an AR model is fitted to the data, $\hat{r}_x(\tau,\tau)$ can be interpreted as a plot of the coefficients of the AR model versus the lag, τ, associated with that coefficient (Harvey, 1981, for example see Figure 3.4 below). For an AR process of order τ, an AR model of order τ is sufficient. Thus the PACF of an AR process of order τ will theoretically have zero elements after τ and the coefficients of the model up to τ (Harvey, 1981). Also it can be shown that an MA process (which has no roots at zero) has an equivalent AR process which has an infinite order (Brockwell and Davis, 1987). Thus for an MA process the SPACF is non-zero for all τ. However, it does die away with increasing τ (Harvey, 1981).

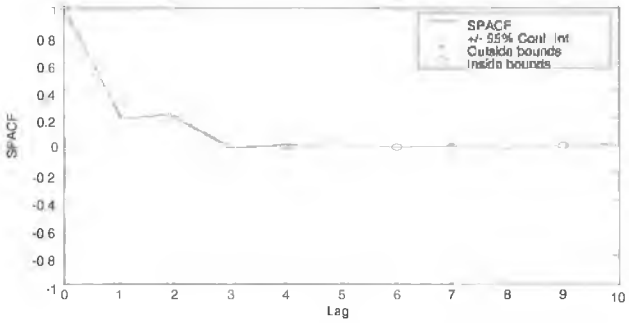

Figure 3.4 The SPACF of an AR(2) process.

Conversely, the SACF of an AR process is non-zero for all τ, while the SACF of an MA process of order τ, is zero after τ (Harvey, 1981).

Thus, the SACF and SPACF can be used to identify the orders of an MA or AR model. However, with an ARMA process the AR part of the data is present in the SACF as a component which dies away (Harvey, 1981). This makes detection of the order of the MA component of the model difficult. Conversely the MA component of the model similarly *contaminates* the SPACF. In addition, as noted by Harvey (1981), the SACF and SPACF of an ARMA process with errors can be highly distorted.

Calculating the orders of an SARMA model involves identifying, in addition to the above, the orders of the seasonal operators. The approach is similar to that above and Bowerman and O'Connell (1987) give guidelines for their selection.

For an integrated model (ARI, SARI, SARIMA, ARIMA, etc.) the transformed series is assumed to be stationary so the model identification is the same as that of a non-integrated model.

For a model with an exogeneous input, *nb* is determined using the Sample Cross Correlation Function (SCCF) defined (Papoulis, 1991) as:

$$\hat{r}_{xu}(\tau) = \frac{\sum_{k=1}^{n-\tau}(x(k)-\bar{x})(u(k+\tau)-\bar{u})}{\left(\sum_{k=1}^{n}(x(k)-\bar{x})^2\right)^{1/2}\left(\sum_{k=1}^{n}(u(k)-\bar{u})^2\right)^{1/2}} \tag{3.31}$$

Where $\hat{r}_{xu}(\tau)$ is the SCCF between $x(k)$ and $u(k)$ at a lag of τ and $\bar{u}$ is the mean value of $u(k)$. The SCCF is zero up to $\tau = k-1$ and non-zero from $\tau = k$ to $\tau = k+nb$, thus allowing determination of the order of B and the delay between the exogeneous variable and the time series.

It should be noted that if $x(k)$ represents a stationarity transformed time-series then the relationship between the original series and $u(k)$ may be distorted (Harvey, 1981). Remedying this situation is dependant on the time series in question and may require transforming $u(k)$ also (Harvey, 1981).

3.3.1.4 Model Estimation and Diagnostic Checking.

The next step in the Box-Jenkins technique is to estimate the coefficients of the model. There are several techniques used to estimate the parameters of Box-Jenkins models some of which are model dependant. These include *maximum likelihood estimation* (Harvey, 1981) and *least squares estimation* (Harvey, 1981) among others. Brockwell and Davis (1987), Cryer (1986) and Bowerman and O'Connell (1987) provide a comprehensive list of the methods used.

Due to the distortions in the SPACF and ACF (mentioned in Section 3.3.1.3), the correct model orders may not have been identified. Thus the next step is *diagnostic checking* in which the model is validated to ensure that it is a good representation of the time-series. The *residuals* (forecast errors, $\varepsilon(k)$) of an

optimal model are taken from a random population (Papoulis, 1991) and thus the first step in diagnostic checking is to examine a plot of the residuals against time (Box and Jenkins, 1970). The second step involves examining the SACF of the residuals to ensure that there is no *serial correlation* ($\varepsilon(k)$ is correlated with $\varepsilon(k-1)$, $\varepsilon(k-2)$, etc.). The ACF of a random process is 1 for a delay of, $\tau = 0$, and zero elsewhere. The SACF of the residuals of an ARMA model may, however, be misleading as they tend to be underestimated (Harvey, 1981). Alternative approaches include that of Box and Pierce (1970), who proposed using a test statistic to determine the randomness of the residuals from an ARMA process. Other tests for randomness rely on examination of the residuals in the frequency domain. The power spectrum of a random series is equal at all frequencies. In order to test this the *periodogram* (Brockwell and Davis, 1987) is often used to estimate the power spectrum. The *periodogram ordinate* (point) for frequency j, $\hat{\pi}_j$, is defined (Harvey, 1981) as:

$$\hat{\pi}_j = a_j^2 + b_j^2 \qquad j = 1,...,n \tag{3.32}$$

where a_j^2 and b_j^2 are the coefficients of the Fourier transform of the residuals defined (Harvey, 1981) as:

$$b_j = (2/N)^{1/2} \sum_{k=1}^{N} \varepsilon(k) \sin(2\pi j / N)k \qquad j = 1,...,n-1 \tag{3.33}$$

and

$$a_j = (2/N)^{1/2} \sum_{k=1}^{N} \varepsilon(k) \cos(2\pi j/N)k \qquad j = 1,...,n-1 \tag{3.34}$$

where N is the number of points used in the calculation.

However, as the periodogram is not consistent (the variance of the ordinates do not go to zero), it is often a bad indicator of randomness. The *cumulative periodogram* is often used instead used to determine randomness. This is defined (Harvey, 1981) as:

$$\hat{\Pi}_i = \sum_{j=1}^{i} \hat{\pi}_j^2 \Big/ \sum_{j=1}^{n} \hat{\pi}_j^2 \qquad i = 1,...,n \tag{3.35}$$

where $\hat{\Pi}_i$ is the cumulative periodogram ordinate at frequency i and n is the maximum frequency (this is ½ the sampling frequency, Harvey, 1981). A plot of

$\hat{\Pi}_i$ against i is called the cumulative periodogram and is a better indicator of randomness than the periodogram as the fluctuations in the periodogram ordinates are smoothed in the summation.

Another measure of the randomness of the residuals is the Ljung-Box test statistic (Bowerman and O'Connell, 1987), defined as:

$$Q(\sqrt{N}) = \frac{N}{N+2} \sum_{j=1}^{\sqrt{N}} \frac{\hat{r}_x(j)^2}{(N-j)} \tag{3.36}$$

Where N is the number of residuals, $Q(N)$ is the Ljung Box test statistic and $\hat{r}_x(j)$ is the SACF of the residuals at lag j. The distribution of the Ljung Box test statistic is approximately Chi-Squared with $\sqrt{N}$ -np degrees of freedom, where np is the complexity of the model. Given a level of significance α, the residuals may be tested using:

$$Q(\sqrt{N}) > \chi_\alpha^2(\sqrt{N} - np) \tag{3.37}$$

where $\chi_\alpha^2(\bullet)$ is the Chi-squared distribution with a significance level of α.

If the model residuals are found to be random then the next step is to check that the model is not *over-fitting* the data. As pointed out by Brockwell and Davis (1987) the mean squared forecast error (in the training set) falls monotonically as the orders of an ARMA model increase. However, the mean squared forecast error outside the training set (i.e. out of sample) does not follow this pattern, and will rise after the complexity of the model surpasses that of the process generating the time series. This phenomenon is known as *over-fitting*. A model whose complexity matches that of the data is known as a *parsimonious* model. Several criteria exist for penalising the complexity of a model relative to the errors generated by that model. An example of one criterion is *Akaike's Information Criterion* (AIC), which may be expressed (Harvey, 1981), as:

$$AIC = -2 \log L(\psi) + 2n \tag{3.38}$$

where $L(\psi)$ is the maximised value of the likelihood function (see Section 3.3.2.4) and n is the number of parameters in the model. Given several candidate models, the one with the lowest AIC is selected. Other criteria are listed in

Harvey (1981), Brockwell and Davis (1987), Bowerman and O'Connell, (1987) and Cryer, (1986).

3.3.1.5 Box-Jenkins Models for Short Term Load Forecasting.

Examples of Box-Jenkins techniques applied to STLF can be found in Kodogiannis and Agnostakis (1999), Mbamalu and El-Hawary (1993), Moghram and Rahman (1989), Barakat *et. al.* (1990) and Elkateb *et. al.* (1998), to mention but a few.

However as pointed out by Mohamad *et. al.* (1996) Box-Jenkins techniques have the disadvantage that they require a large database for training and are susceptible to errors in that database because of differencing. In addition, as Box-Jenkins techniques assume that the load curve is static they can give large errors when the load curve changes rapidly (Mohamad *et. al.*, 1996, Fan and McDonald, 1994). Rajurkar and Newill (1985) present an ARMAX model in which the coefficients of the model are allowed to change dynamically which allows the load shape to change more rapidly.

Another problem with the Box-Jenkins approach is the *subjective* selection of the model orders, which can result in the wrong model structure being chosen (Chen and Kao, 1996). Chen and Kao, to overcome this drawback, propose an automated approach based on repetitively applying a *gentle difference algorithm.* The gentle differencing algorithm essentially replaces the difference operator ∇ (i.e. $1-q^{-1}$) with $1-a_g q^{-1,}$ where a_g is a coefficient great than 1. This is equivalent to fitting a non-stationary AR model as opposed to the use of differencing. After each application the resultant time series is tested for stationarity. The performance of their models is found to be comparable with manually selected ARIMAX models. Parzen (1982) proposed a similar approach called the Auto-Regressive Auto-Regressive Moving Average (ARARMA) model. This differs from the ARIMA model as the integrated part of the model is now created using an AR differencing technique.

In an application of Box-Jenkins models to the STLF problem in the Czech Republic, Darbellay and Slama (2000) apply non-seasonal differencing of 1 hour, seasonal differencing of one day (24 hours) followed by further seasonal differencing of one week (168 hours) thus giving a differenced load defined as:

$$z(k) = \nabla^{168}\nabla^{24}\nabla y(k) \tag{3.39}$$

where $z(k)$ is the differenced load for hour k and $y(k)$ is the load for hour k. With regards to the stationarity of $z(k)$, Darbellay and Slama (2000) note that it is essentially stationary with significant lags in the SACF at one, six, seven and eight days. A non-linear auto-correlation function is also used by Darbellay and Slama (2000), based on information theory and a method developed previously by Darbellay (1999), to show that there is also a non-linear auto-correlation in $z(k)$ at a lag of one hour. From the SACF an ARIMA model is then constructed of the form:

$$(1-a_1q^{-1})(1-a_2q^{-24})(1-a_3q^{-168})z(k) = (1-c_1q^{-1})(1-c_2q^{-24})(1-c_3q^{-168})\varepsilon(k) \tag{3.40}$$

where a_i are the AR coefficients and c_i are the MA coefficients of the model. The ARIMA model was then compared with several non-linear models utilising neural networks (Section 3.4.3). The first of the neural networks was an MLP (see Section 3.4.3.1) using the delayed elements of $z(k)$ on the left hand side of Equation (3.40) as inputs. The second is a recurrent neural network (Section 3.4.3.2) allowing dynamic modelling of the series. This is perhaps a better choice for comparison with the ARIMA model described by Equation (3.40) as delayed errors (the MA part or right hand side of Equation (3.40)) are also modelled. It was found that there was little difference in the results. Thus the non-linear correlation detected at a lag of 1 hour was deemed to be of little extra use in forecasting the load, in this case.

Darbellay and Slama (2000) also compared an ARIMAX model with a neural network. In this case temperature was the exogenous input for both the ARIMAX and neural network models. The results show that the ARIMAX model is superior to the neural network although it should be noted that the time series to be forecast in this case is the *total daily load* and *not* the hourly load as in the previous case above.

3.3.2 Linear State Space Techniques.

Linear State Space (SS) techniques formulate the modelling problem in terms of *states,* which represents the underlying process generating a time series. This approach differs from difference equation approaches (such as those in Section 3.3.1) although the two forms are interchangeable.

The states are related to the time series via a *measurement equation* defined (Gelb, 1984) as:

$$\boldsymbol{x}(k) = \boldsymbol{H\theta}(k) + \boldsymbol{\varepsilon}(k) \tag{3.41}$$

where $\boldsymbol{x}(k)$ is the time series value at time k (which can be multi-variate), $\boldsymbol{\theta}(k)$ is a vector of states (called the *state vector*) at time k, $\boldsymbol{H}$ is called the *observation matrix* and $\boldsymbol{\varepsilon}(k) \sim \mathrm{N}(0, \boldsymbol{R}_x)$ is a vector of white noise error terms, known as *the measurement noise*, and $\sim \mathrm{N}(0, \boldsymbol{R}_x)$ denotes white noise with a normal distribution, mean of zero and covariance matrix $\boldsymbol{R}_x$. The states are propagated from time k to time k+1 via the *state transition equation* (Gelb, 1984) as:

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\Phi\theta}(k) + \boldsymbol{\eta}(k) \tag{3.42}$$

where $\boldsymbol{\Phi}$ is called the *state transition matrix* and $\boldsymbol{\eta}(k) \sim \mathrm{N}(0, \boldsymbol{Q}_\theta)$ is a vector of white noise error terms known as *the process noise.*

For a time series with exogeneous inputs the SS equation can be expressed (Harvey, 1981) as:

$$\boldsymbol{x}(k) = \boldsymbol{H\theta}(k) + \boldsymbol{Gu}(k) + \boldsymbol{\varepsilon}(k) \tag{3.43}$$

where $\boldsymbol{u}(k)$ is a vector of exogenous inputs at time k and $\boldsymbol{G}$ is the matrix of associated parameters. In many cases the value of $\boldsymbol{G}$ is unknown. However, it can *augmented* into the state vector and calculated recursively (thus $\boldsymbol{G}$ becomes $\boldsymbol{G}(k)$).

The measurement equation then becomes (Harvey, 1981):

$$\boldsymbol{x}(k) = \begin{bmatrix} \boldsymbol{H} & \boldsymbol{u}(k) \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}(k) \\ \boldsymbol{G}(k) \end{bmatrix} + \boldsymbol{\varepsilon}(k) = \boldsymbol{H}^{'} \boldsymbol{\theta}^{'}(k) + \boldsymbol{\varepsilon}(k) \tag{3.44}$$

where $\boldsymbol{H}^{'}$ is the augmented measurement equation and $\boldsymbol{\theta}^{'}$ is the augmented state vector. Similarly, the state transition equation (3.42) can be augmented (Harvey, 1981) as:

$$\begin{bmatrix} \boldsymbol{\theta}(k+1) \\ \boldsymbol{G}(k+1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}(k) \\ \boldsymbol{G}(k) \end{bmatrix} + \begin{bmatrix} \boldsymbol{\eta}(k) \\ \boldsymbol{0} \end{bmatrix} = \boldsymbol{\theta}^{'}(k+1) = \boldsymbol{\Phi}^{'} \boldsymbol{\theta}^{'}(k) + \boldsymbol{\eta}^{'}(k) \tag{3.45}$$

where $\boldsymbol{\Phi}^{'}$ is the augmented state transition matrix. As can be seen, the augmented SS equations (3.44) and (3.45) are of the same format as Equations (3.41) and (3.42) thus the following analysis for *univariate* (no exogeneous inputs) SS models applies also to exogeneous SS models.

3.3.2.1 Kalman Filtering and State Estimation.

An *optimal* technique (in the least squares sense) exists for estimating the states called the *Kalman filter* (Harvey, 1981). The Kalman filter is a recursive algorithm, which is implemented in several steps (a derivation may be found in Gelb, 1984):

1. An estimate of $\boldsymbol{\theta}(k)$ at time k+1, is produced using an equation similar to Equation (3.42) (Harvey, 1984) as:

$$\hat{\boldsymbol{\theta}}^{-}(k+1) = \boldsymbol{\Phi} \hat{\boldsymbol{\theta}}^{+}(k) \tag{3.46}$$

   where $\hat{\boldsymbol{\theta}}^{-}(k+1)$ is called the *a-priori* estimate of the state vector at time k+1. The estimate is an *a-priori* estimate, as it does not include information about the value of $\boldsymbol{x}(k+1)$. $\hat{\boldsymbol{\theta}}^{+}(k)$ is the *a-posteriori* estimate of $\boldsymbol{\theta}(k)$ as it includes information about the value of $\boldsymbol{x}(k)$. As the technique is recursive $\hat{\boldsymbol{\theta}}^{+}(k)$ is generated from step 3 (below) in the previous iteration,

2. Next an *a-priori* estimate of the *error covariance matrix* of the state vector is calculated. The error covariance matrix of the state vector is a measure of the accuracy of the state vector estimate. This is defined (Gelb, 1984) as:

$$\boldsymbol{P}^{+}(k) = E\left[\left(\boldsymbol{\theta}(k) - \hat{\boldsymbol{\theta}}^{+}(k)\right)\left(\boldsymbol{\theta}(k) - \hat{\boldsymbol{\theta}}^{+}(k)\right)^{\mathrm{T}}\right] \tag{3.47}$$

and

$$\boldsymbol{P}^{-}(k) = E\left[\left(\boldsymbol{\theta}(k) - \hat{\boldsymbol{\theta}}^{-}(k)\right)\left(\boldsymbol{\theta}(k) - \hat{\boldsymbol{\theta}}^{-}(k)\right)^{\mathrm{T}}\right] \tag{3.48}$$

where $\boldsymbol{P}^{+}(k)$ is the *a-posteriori* error covariance matrix of $\hat{\boldsymbol{\theta}}^{+}(k)$, $\boldsymbol{P}^{-}(k)$ is the *a-priori* error covariance matrix of $\hat{\boldsymbol{\theta}}^{-}(k)$, $E[\bullet]$ is the expectation operator and T denotes the transpose operator. The error covariance matrix of $\hat{\boldsymbol{\theta}}^{-}(k+1)$ is generated using $\boldsymbol{P}^{+}(k)$ from the previous iteration (Harvey, 1984) as:

$$\boldsymbol{P}^{-}(k+1) = \boldsymbol{\Phi}\boldsymbol{P}^{+}(k)\boldsymbol{\Phi}^{\mathrm{T}} + \boldsymbol{Q}_{\theta} \tag{3.49}$$

3. Once $x(k+1)$ becomes available this is integrated into the estimate of $\hat{\boldsymbol{\theta}}^{-}(k+1)$ (Harvey, 1984) via:

$$\hat{\boldsymbol{\theta}}^{+}(k+1) = \hat{\boldsymbol{\theta}}^{-}(k) + \boldsymbol{K}(k)\left[x(k+1) - \boldsymbol{H}\hat{\boldsymbol{\theta}}^{-}(k+1)\right] \tag{3.50}$$

where $\boldsymbol{K}(k)$ is called the *Kalman gain* and is calculated (Gelb, 1984) as:

$$\boldsymbol{K}(k+1) = \boldsymbol{P}^{-}(k+1)\boldsymbol{H}^{\mathrm{T}}\left[\boldsymbol{H}\boldsymbol{P}^{-}(k+1)\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R}_{x}\right]^{-1} \tag{3.51}$$

4. Finally, $\boldsymbol{P}^{+}(k+1)$ is calculated (Gelb, 1984) using:

$$\boldsymbol{P}^{+}(k+1) = \left[\boldsymbol{I} - \boldsymbol{K}(k)\boldsymbol{H}\right]\boldsymbol{P}^{-}(k+1) \tag{3.52}$$

where $\boldsymbol{I}$ is the identity matrix.

Equations (3.46) and (3.49) from steps 1 and 2 are known as the *prediction equations* as they propagate the state vector and associated error covariance matrix. Equations (3.50) and (3.52) are known as the *updating equations* as they update the states and associated error covariance matrix. In terms of electrical load there is usually no measurement error as such, but rather an unpredictable part of the load.

A forecast of $\boldsymbol{x}(k)$ is produced from $\hat{\boldsymbol{\theta}}^{-}(k)$ using the measurement equation (Equation 3.41). Also the variance of the forecast, $\sigma_{\hat{x}}^{2}(k)$, can be calculated from $\boldsymbol{P}^{-}(k)$ as (Harvey, 1984):

$$\sigma_{\hat{x}}^{2}(k) = \boldsymbol{H}\boldsymbol{P}^{-}(k)\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R}_{x} \tag{3.53}$$

The Kalman filter is shown diagrammatically in Figure 3.5 below.

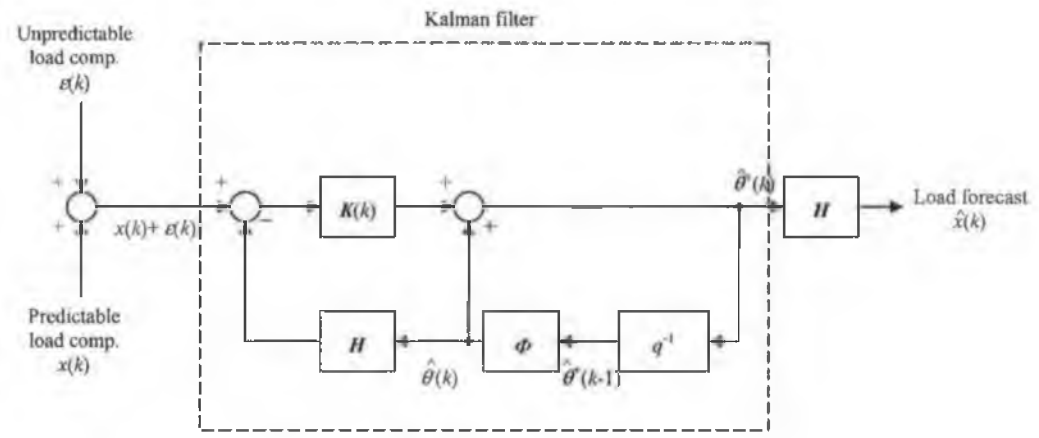

Figure 3.5 Block diagram of Kalman filter operation.

The recursive algorithm described above requires initial values for the state vector, $\hat{\boldsymbol{\theta}}^{+}(0)$, and the error covariance matrix of the state vector, $\boldsymbol{P}^{+}(0)$. There are several ways to achieve this:

Approach 1.

In the event that no *a-priori* knowledge exists regarding these values, $\boldsymbol{P}^{+}(0)$ is typically set to a finite large value to indicate that $\hat{\boldsymbol{\theta}}^{+}(0)$ is *not* a good estimate of $\boldsymbol{\theta}(0)$. In addition $\hat{\boldsymbol{\theta}}^{+}(0)$ is set to zero or randomly initialised (Harvey, 1981). With these initial conditions several iterations, denoted m_{θ}, are required until $\hat{\boldsymbol{\theta}}^{+}(k)$ is a good estimate of $\boldsymbol{\theta}(k)$. Harvey (1994) suggests that m_{θ} is equal to the number of states in the state vector. However m_{θ} can also be determined *subjectively*, by

examining $\boldsymbol{P}^{+}(k)$ which has a transient for $k<m_{\theta}$ (Harvey, 1994). Forecasts for $k<m_{\theta}$ should be ignored in any subsequent analysis, however this does not present a problem for large data sets (Harvey, 1994),

Approach 2.

For a stationary time series the state estimates reach a *steady state* (they are approximately time invariant) and can be approximated as a random variable with $\boldsymbol{\theta}(k)\sim\mathrm{N}(\overline{\boldsymbol{\theta}}(k),\overline{\boldsymbol{P}}(k))$ where $\overline{\boldsymbol{\theta}}(k)$ is the average value of the state vector and $\overline{\boldsymbol{P}}(k)$ is the average value of the associated error covariance matrix. The initial conditions can then be set to $\boldsymbol{\theta}(0)=\overline{\boldsymbol{\theta}}(k)$ and $\boldsymbol{P}(0)=\overline{\boldsymbol{P}}(k)$ (Harvey, 1984),

Approach 3.

In the case where the state vector has stationary and non-stationary states a mix of approaches 1 and 2 may be used (Harvey, 1981). The stationary elements of $\boldsymbol{\theta}(0)$ and $\boldsymbol{P}(0)$ are determined as in approach two while the non-stationary elements are determined as in approach 1 (Harvey, 1981). The number of iterations required before the state vector transient has dissipated is now equal to the number of non-stationary states (Harvey, 1981), and

Approach 4.

For some SS models the states represent physical attributes of the time series, for example, the trend. In this case the initial states can be set using estimates of these physical attributes, for example a trend may be estimated by the first value of the time series.

3.3.2.2 Linear State Space Model Structures.

There are an infinite number of state space models; the most commonly used models are listed here. The observation and transition matrices fully define a SS model structure. The most basic SS model is the *Generalised Random Walk* (GRW) defined (Ng and Young, 1990) as:

$$\boldsymbol{\Phi}_{grw} = \begin{bmatrix} \alpha & \beta \\ 0 & \gamma \end{bmatrix} \qquad \boldsymbol{H}_{grw} = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{3.54}$$

where $\boldsymbol{\Phi}_{grw}$ is the state vector, $\boldsymbol{H}_{grw}$ is the observation matrix, α,β and γ are the coefficients of the model. Several variations of the model exist such as the *random walk* model ($\alpha=1,\beta=0$ and $\gamma=0$), the *smoothed walk* model ($\alpha\in(0,1),\beta=1$ and $\gamma=1$) and the *Integrated Random Walk* (IRW) model ($\alpha=1,\beta=1$ and $\gamma=1$) (Ng and Young, 1990).

The IRW is often used to model a trend with only two states (Ng and Young, 1990) as:

$$\begin{bmatrix} d(k) \\ \dot{d}(k) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} d(k-1) \\ \dot{d}(k-1) \end{bmatrix} + \begin{bmatrix} \eta_d(k) \\ \eta_{\dot{d}}(k) \end{bmatrix} \tag{3.55}$$

where $d(k)$ is the trend at time k, $\dot{d}(k)$ is the rate of change of the trend at time k (this is also called the velocity or slope of the trend), $\eta_d(k)$ and $\eta_{\dot{d}}(k)$ are white noise error terms for the trend and slope respectively. This representation of the trend has been found useful by Ng and Young for modelling the trend in several time series (Ng and Young, 1990) and by Infield and Hill (1998) for removing the trend in STLF. It has the advantage that the trend can be modelled using only two states, both of which have physical interpretation.

SS models have also been used to model *seasonal* time series. Three common approaches are:

1. The *Periodic Random Walk* (PRW) model. This can be expressed (Young, 1988) as:

$$x(k) = x(k-s) + \eta_{\psi}(k) \tag{3.56}$$

where s is the seasonal length and $\eta_{\psi}(k)$ is a white noise error term. A problem with the PRW is that it can be *unobservable* (the seasonal states cannot be distinguished from the trend states) when used in conjunction with the IRW model (Young, 1988),

2. The *Differenced Periodic Random Walk* (DPRW) model. This can be expressed (Young, 1988) as:

$$x(k) = -\sum_{j=1}^{s} x(k-j) + \eta_{\psi}(k) \tag{3.57}$$

where s is the seasonal length and $\eta_{\psi}(k)$ is a white noise error term. By taking the sum on the right-hand side of Equation (3.57) to the left-hand side gives:

$$\sum_{j=0}^{s} x(k-j) = \eta_{\psi}(k) \tag{3.58}$$

Equation (3.58) requires that the sum of the time series, over a season, is equal a white noise term. This is an important as it allows the seasonal shape to change if required, given that the *expected* (the expected value of a white noise term is zero) sum is zero. The DPRW may be expressed in state space notation (Young, 1988) as:

$$\boldsymbol{\Phi}_{dprw} = \begin{bmatrix} -1 & \cdots & \cdots & -1 \\ 1 & 0 & 0 & 0 \\ 0 & \ddots & 0 & \vdots \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad \boldsymbol{H}_{dprw} = \begin{bmatrix} 1 & 0 \cdots 0 \end{bmatrix} \tag{3.59}$$

where $\boldsymbol{\Phi}_{dprw} \in \mathbf{R}^{s-1 \times s-1}$ and $\boldsymbol{H}_{dprw} \in \mathbf{R}^{1 \times s-1}$ are the state transition matrix and observation matrix for the DPRW, respectively. The advantage of the DPRW over the PRW is that it does not have the observability problems of the PRW when used in conjunction with the IRW model (Ng and Young, 1990), and

3. The *Dynamic Harmonic Regression* (DHR) model. This can be expressed as the sum of sinusoidal signals at frequencies up to half the seasonal length (Ng and Young, 1990) as:

$$x(k) = \sum_{j=1}^{s/2} \left[\theta_{1,j}(k) \cos(2\pi \frac{j}{s} k) + \theta_{2,j}(k) \sin(2\pi \frac{j}{s} k) \right] + \eta_{\psi}(k) \qquad (3.60)$$

where $\theta_{1,j}$ and $\theta_{2,j}$ are the states. The corresponding SS representation given by Ng and Young (1990) calculates $\theta_{1,j}$ and $\theta_{2,j}$ as the states of the system, each of which is assumed to follow a GRW (see Ng and Young, 1990 for more details). Note that a GRW model requires three states to be calculated. The sinusoidal elements of Equation (3.60) are introduced *via* the measurement equation (Ng and Young, 1990) as:

$$\boldsymbol{H}_{dhr} = \left[\cos(2\pi \frac{1}{s} k) \quad 0 \quad \sin(2\pi \frac{1}{s} k) \quad 0 \quad \cdots \quad \cos(2\pi \frac{s/2}{s} k) \quad 0 \quad \sin(2\pi \frac{s/2}{s} k) \quad 0 \right] \qquad (3.61)$$

Alternative forms exist for the DHR in which the sinusoidal element is introduced into the state transition matrix as opposed to the observation matrix (Harvey, 1984). Ng and Young (1990) believe that their form is superior to the DPRW for time series where the seasonal component is growing over time. However, there are significantly more states in the DHR model than the DPRW model due to the incorporated GRW models.

In addition to the three models above, Harvey (1984) gives the SS representation for SARIMAX models, exponentially weighted moving average models, and others. One particular result of interest for this thesis (see Section 5.4.1) is that the gentle

differencing (Section 3.3.1.5) approach may be expressed in SS form (Harvey, 1984) as an IRW.

3.3.2.3 Linear State Space Models for Short-Term Load Forecasting.

A structural model is a state space model in which the states of the model represent physical decompositions of the underlying time series. For example, electrical load has been decomposed by many authors into a trend and cyclical component (Chen and Kao, 1996, Ramanathan *et. al.*, 1997, Haida and Muto, 1994 and Moutter *et. al.*, 1986 to mention but a few). The presence of a trend and cyclical component in Irish load has also been examined in Chapter 2. This type of structural model is thus ideal for electrical load and is called a *Basic Structural Model* (BSM).

Specifically, the BSM is a state space model which represents a time series as a sum of a trend, $d(k)$, a seasonal, $\psi(k)$, and a random white noise component, $\varepsilon(k)$, (Harvey, 1994) as:

$$x(k) = d(k) + \psi(k) + \varepsilon(k) \tag{3.62}$$

The BSM is composed of an IRW model for the trend component used in conjunction with a seasonal model, either that described by Equation (3.59) or Equation (3.61). It is assumed that the trend states and seasonal states are independent and so for example, the SS matrices of a BSM with a DPRW (Harvey, 1994) may be expressed as:

$$\boldsymbol{\Phi}_{bsm} = \begin{bmatrix} \boldsymbol{\Phi}_{irw} & 0 \\ 0 & \boldsymbol{\Phi}_{dprw} \end{bmatrix} \qquad \boldsymbol{H}_{bsm} = \begin{bmatrix} \boldsymbol{H}_{irw} & \boldsymbol{H}_{dprw} \end{bmatrix} \tag{3.63}$$

where $\boldsymbol{\Phi}_{bsm}$ and $\boldsymbol{H}_{bsm}$ are the state transition matrix and observation matrix for the BSM respectively. The BSM with a DHR for the cyclical component is similar; the transition and observation matrices for the DPRW replacing $\boldsymbol{\Phi}_{dhr}$ and $\boldsymbol{H}_{dhr}$ respectively.

3.3.2.4 Variance Parameter Estimation.

In many cases, the covariances of the measurement and process noise, $\boldsymbol{R}_x$ and $\boldsymbol{Q}_\theta$, are unknown and form part of the *parametric description* of the state space model. As the BSM is the most relevant state space model to STLF, the current discussion is limited to estimation of the parameters for this model. The measurement equation for the BSM is Equation (3.51) and the state transition equation for a BSM with a DPRW can be expressed (Harvey, 1994) as:

$$\boldsymbol{\theta}_{bsm}(k+1) = \begin{bmatrix} d(k+1) \\ \dot{d}(k+1) \\ \psi_1(k+1) \\ \psi_2(k+1) \\ \vdots \\ \psi_{s-1}(k+1) \end{bmatrix} = \boldsymbol{\Phi}_{bsm}\boldsymbol{\theta}_{bsm}(k) + \begin{bmatrix} \eta_d(k+1) \\ \eta_{\dot{d}}(k+1) \\ \eta_\psi(k+1) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{3.64}$$

where $\psi_{1,\ldots,s-1}$ are the seasonal states. From Equations (3.64) and (3.41) there are four white noise components in the model (i.e. $\eta_d(k)$, $\eta_{\dot{d}}(k)$, $\eta_\psi(k)$ and $\eta_\varepsilon(k)$). The noise components are assumed to be independent and so $\boldsymbol{R}_x$ and $\boldsymbol{Q}_\theta$ may be expressed (Harvey, 1994) as:

$$\boldsymbol{R}_x = \sigma_\varepsilon^2 \quad \text{and} \quad \boldsymbol{Q}_\theta = \begin{bmatrix} \sigma_d^2 & & & & 0 \\ & \sigma_{\dot{d}}^2 & & & \\ & & \sigma_\psi^2 & & \\ & & & 0 & \\ 0 & & & & \ddots \\ & & & & & 0 \end{bmatrix} \tag{3.65}$$

where σ_ε^2 is the variance of $\varepsilon(k)$, σ_d^2 is the variance of $\eta_d(k)$, $\sigma_{\dot{d}}^2$ is the variance of $\eta_{\dot{d}}(k)$ and σ_ψ^2 is the variance of $\eta_\psi(k)$. σ_ε^2 is the variance of the unpredictable part of the load and a high value will cause the Kalman filter to apply a lower weight to the measurements, $y(k)$ (see Equations (3.51) and (3.52)). As σ_d^2 increases, the Kalman filter allows the trend level to vary more. As $\sigma_{\dot{d}}^2$ increases, the Kalman filter

allows the trend slope to vary more. As σ_s^2 increases, the Kalman filter allows the seasonal component to change shape more rapidly.

Theoretically, the optimal values of $\boldsymbol{R}_x$ and $\boldsymbol{Q}_\theta$ can be calculated by *maximum likelihood estimation* (explained below) which depends on the one-step ahead prediction errors. However, as pointed out by Harvey (1994), the prediction errors in turn depend on the way the state vector is initialised (see Section 3.3.2.1). The different initialisation procedures lead to different forms of the likelihood function (explained below). These forms are, however, equivalent for sufficiently large data sets (Harvey, 1994).

Given a process and a set of parameters, the data produced by that process can be described by some probability distribution. For example, if the process is Gaussian the parameters are the mean and standard deviation. However, it is more typical that the data is given, a model class is assumed and the parameters are required. In this case the probability distribution may be expressed in terms of the known data with the parameters as dependent variables. This is known as the *likelihood function* and expresses the probability* that the data observed came from a process as a function of the parameters. Thus, the parameters that maximise the likelihood function are those that most likely produced the data (for the assumed process) (Papoulis, 1991). In this case, the process is the BSM, the data is the recorded load and the parameters are σ_ε^2, σ_d^2, $\sigma_{\dot{d}}^2$ and σ_ψ^2. Typically, by taking the log of the likelihood function (known as the *log likelihood function*) a more convenient expression is achieved (Harvey, 1984). By minimising the log likelihood function, the likelihood function is also maximised. This is known as maximum likelihood estimation (see Papoulis, 1991, Harvey, 1981 or Murray, 1996 for more details).

* Strictly speaking a probability interval should be taken.

In the case where the state vector is initialised to zero and $\boldsymbol{P}^{+}(0)$ is set to a large finite number (i.e. approach 1 in Section 3.3.2.1), the *log likelihood function* is (Harvey, 1981):

$$\log L\left[\hat{x}(1),\hat{x}(2),\ldots;\sigma_{\varepsilon}^{2},\sigma_{d}^{2},\sigma_{\dot{d}}^{2},\sigma_{\psi}^{2}\right]= \frac{-(N-m_{\theta})}{2}\log 2\pi-\frac{1}{2}\sum_{j=m_{\theta}+1}^{N}\log\sigma_{\hat{x}}^{2}(j)-\frac{1}{2}\sum_{j=m_{\theta}+1}^{N}\log\frac{\varepsilon^{2}(j)}{\sigma_{\hat{x}}^{2}(j)} \quad (3.66)$$

where $L[x;\sigma]$ denotes the likelihood function, N is number of points in the training set, m_{θ} is the point past which the initial transient has passed (Section 3.3.2.1), $\varepsilon(k)$ are the errors in estimates of $x(k)$ and $\sigma_{\hat{x}}^{2}(k)$ is the estimated variance of those errors. As can be seen Equation (3.66) requires $\sigma_{\hat{x}}^{2}(k)$ and $\varepsilon(k)$ to be estimated, however these are dependant on σ_{ε}^{2}, σ_{d}^{2}, $\sigma_{\dot{d}}^{2}$ and σ_{ψ}^{2}. Thus, the log likelihood function must be maximised with the aid of a function maximisation routine. As this technique depends on the prediction errors it is often called *Prediction Error Decomposition* (PED).

As pointed out by Ng and Young (1990) and Harvey (1994), $\boldsymbol{R}_{x}$ and $\boldsymbol{Q}_{\theta}$ are directly related. Specifically a model with a particular value of $\boldsymbol{R}_{x}$ and $\boldsymbol{Q}_{\theta}$ is identical to a model with $\boldsymbol{R}_{x}/\xi$ and $\boldsymbol{Q}_{\theta}\xi$, where ξ is any scalar and is called the *noise variance ratio*. Thus, $\boldsymbol{R}_{x}$ is typically set to unity and removed from the parameters to be optimised (Ng and Young, 1990 and Harvey, 1994).

Ng and Young (1990) point out, however, that estimating the optimal parameters by PED can be complex and may not lead to optimal parameters. The technique of *Sequential Spectral Decomposition* (SSD) is instead proposed. This technique estimates the parameters by relating them to their spectral qualities. Specifically, the trend component acts as a low-pass filter while the seasonal component acts as a band-pass filter. The steps involved in the technique are:

1. R_x is set to unity. σ_d^2 is set to zero and the trend component allowed to evolve via the variance of the slope, $\sigma_{\dot{d}}^2$.

2. The periodogram (Section 3.3.1.4) of the time series is examined and the *50% cut-off frequency*, f_{50}, of the low frequency component identified. The 50% cut-off frequency is that frequency at which the amplitude of the low frequency component is a ½ the maximum (Figure 3.6). Alternatively a specific cut-off frequency may be specified (for example, the trend component in electrical load will act at periods less than a year).

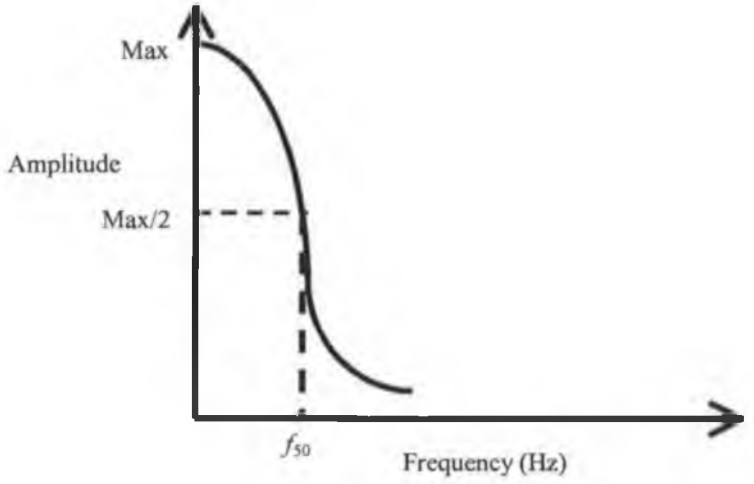

Figure 3.6 Diagram of the 50% cut-off frequency.

3. The value of $\sigma_{\dot{d}}^2$ is related to f_{50} (Ng and Young, 1990) via:

$$\sigma_{\dot{d}}^2 = 1605(f_{50})^4 \tag{3.67}$$

4. An IRW model with $\sigma_{\dot{d}}^2$ calculated as in step 3 is used to de-trend the time series.

5. R_x, σ_d^2 and $\sigma_{\dot{d}}^2$ have now been calculated and the last remaining parameter, σ_ψ^2, is now calculated using PED (Young, 1988).

3.3.3 Bayesian Techniques

Classical statistical approaches describe modelling a data set using a parameter space, Ω, a sample space, Ξ, and family of distributions defined on the sample space, *P*. The parameters are viewed as fixed constants, which must be estimated. In the Bayesian approach however, Ω is viewed not as a fixed set but as random with associated probability distributions (Geweke, 2001).

Let Θ be a set of unknown parameters, and $\boldsymbol{x}$ be a set of observed data. In the Bayesian approach, any prior knowledge or ignorance about Θ is first expressed as a prior distribution, or *prior*, $P(\Theta)$. A model is then proposed which may be expressed as $P(\boldsymbol{x}|\ \Theta)$. This model may be an ARIMA, neural network model, etc. but simply expresses the probability of observing a set of data given a set of parameters for the model. Now, given an observed data set, $\boldsymbol{x}$, the prior may be updated (Geweke, 2001), as:

$$P(\Theta \mid \boldsymbol{x}) = \frac{P(\boldsymbol{x} \mid \Theta)P(\Theta)}{P(\boldsymbol{x})} \tag{3.68}$$

which is an extension of Baye's theorem (and thus the name), to give a *posterior* estimate of the distribution of Θ, $P(\Theta \mid \boldsymbol{x})$. Equation (3.68) is often expressed as:

$$posterior = \frac{likelihood \times prior}{evidence} \tag{3.69}$$

The Bayesian approach is similar to the Kalman filter which begins with an estimate of the parameters and updates that estimate as new information arrives. Indeed, Harvey, (1984) shows that when the Bayesian approach is used to calculate the states of a SS model, the resulting algorithm is equivalent to the Kalman filter.

Bayesian techniques can also be applied to structure determination of any type of model (Broemling and Shaarawy, 1985). In this case the model structure is expressed as a random variable and can be updated if the probability distributions above can be expressed in terms of known quantities. However, this may be difficult in some cases (Broemling and Shaarawy, 1985).

Broemling and Shaarawy (1985) for example, use the Bayesian approach instead of the Box-Jenkins approach (Section 3.2.1) for selecting the orders and parameters of an ARMA model. Kiartzis *et. al.* (1997) uses the Bayesian approach to weight the forecasts of three different candidate STLF models (two AR models and a neural network). Interestingly, as the Bayesian approach allows the distribution of the forecast errors to be calculated, the weights can become forecast horizon dependant (Kiartzis *et. al.*, 1997).

3.3.4 Multi-Timescale Techniques

Sequential approaches model data on an hourly timescale while parallel approaches are on a daily timescale (Section 3.2.2). However, Multi TimeScale (MTS) techniques differ in that the data is modelled using a *combination* of forecasts made at different timescales.

There are two types of multi-timescale technique examined in this section the *adaptive scaling techniques* of Khotanzad *et. al.* (1996) and Gupta (1985), and the MTS technique of Murray *et. al.* (2000).

3.3.4.1 Adaptive Scaling Techniques

Khotanzad *et. al.*, (1996) developed a technique for forecasting hourly *temperatures* for an electricity utility given that only daily minimum and maximum temperature forecasts from the meteorological office are available. The first part of the technique uses a sequential model to produce an hourly forecast of the temperatures up to 24 hours ahead. This hourly forecast is then adjusted using temperature forecasts from parallel models*.

* Note that forecasting electrical load is not the aim of this technique.

The Sequential Model (SM), in this case is an MLP (see Section 3.4.3.1). This model uses the current and previous temperatures as inputs (Khotanzad *et. al.*, 1996).

The Parallel Models (PM's) are those of the Meteorological Office (MO) which produce forecasts of the temperature at only two hours in the day (the hours at which the maximum and minimum temperature occur). Thus, parallel model forecasts for every hour of the day are *not* required.

The SM forecast is adjusted (Khotanzad *et. al.*, 1996) via:

$$\hat{t}'_{hr}(i,k) = m_{as}(k)\hat{t}_{hr}(i,k) + b_{as}(k) \tag{3.70}$$

where $\hat{t}'_{hr}(i,k)$ and $\hat{t}_{hr}(i,k)$ are the adjusted and SM temperature forecasts for hour i of day k. $m_{as}(k)$ and $b_{as}(k)$ are the adaptive scaling parameters calculated such that the adjusted forecast must agree with the PM forecasts at the appropriate hours (Figure 3.7).

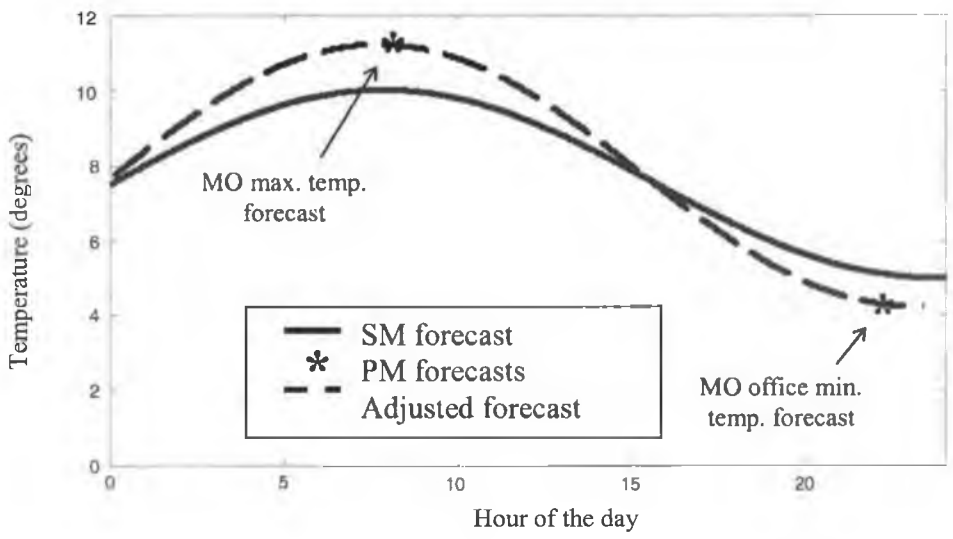

Figure 3.7 Adaptive scaling of a 24 hour temperature forecast using minumum and maximum temperature forecasts.

The main advantage of this technique above a SM is that propagation errors in the SM forecasts are reduced. Disadvantages of this technique are that:

1. The scaling is linear, which distorts the shape of the SM forecast (for example the adjusted forecast shown in Figure 3.7 above),
2. Forcing the MTS forecast to pass through the PM forecasts assumes that these forecasts are far more accurate than the SM forecasts which may not be true, and
3. The number of PM forecasts that can be used is restricted to two points in the day.

Gupta (1985) proposes a similar method, which is applied to load forecasting. In this method, a SM forecast of the load (in this case a linear model) is combined with a daily peak load forecast (this is the PM forecast in this case). The covariance matrix of SM forecast errors, $\boldsymbol{Q}_{sm}$, and the variance of the PM forecast errors, σ^2_{pm}, is used to form the adjusted forecast (Gupta, 1985), via:

$$\hat{y}_{mts}(i,k) = \hat{y}_{sm}(i,k) + \frac{\left[\hat{y}_{pm}(k) - \hat{y}_{sm}(p,k)\right] \boldsymbol{Q}_{sm}(i,p)}{\boldsymbol{Q}_{sm}(p,p) + \sigma^2_{pm}} \tag{3.71}$$

where $\hat{y}_{mts}(i,k)$ and $\hat{y}_{sm}(i,k)$ are the adjusted and SM forecasts at hour i of day k respectively, $\hat{y}_{pm}(k)$ is the PM (peak) forecast for day k, p is the hour at which the peak occurs and $\boldsymbol{Q}_{sm}(i,j)$ is element i,j of $\boldsymbol{Q}_{sm}$. With this technique, the adjustment is not linear as in the adaptive scaling technique, but rather depends on the covariance between the SM errors at different hours and the variance of the PM model forecasts. Additionally, the MTS forecast is not forced to pass through the PM forecast as in the adaptive scaling technique. Instead, the MTS forecast is a weighted combination of both. The disadvantage of this technique is that only a single PM forecast (in this case the peak load) may be used.

The next method proposed by Murray *et. al.* (2000) again attempts to use the longer timescale data for similar reasons but is more flexible in that numerous PM forecasts can be used. Maintaining the shape of the SM forecast is also taken into account.

3.3.4.2 Murray's Multi-Timescale Technique

The MTS technique proposed by Murray *et. al.* (2000) combines PM forecasts with a SM forecast and, additionally, a forecast of the total daily consumption. The total daily consumption is called the *daily-sum*.

The number of PM forecasts is not restricted although for the STLF problem Murray *et. al.* (2000) suggests the use of PM forecasts at:

- The load at 6pm,
- The overnight minimum at 5am,
- The lunchtime peak at 1pm, and
- The load at 2pm.

An additional cardinal point known as the *end-point* is used. This is a PM forecast 24 hours ahead of the forecasting origin. Unlike the techniques in Section 3.3.4.3, Murray *et. al.*'s (2000) MTS technique (furthermore known as the MTS technique) can combine a SM forecast for several days ahead with PM and multiple end-sum forecasts. This technique is shown diagrammatically in Figure 3.8 (notation explained below).

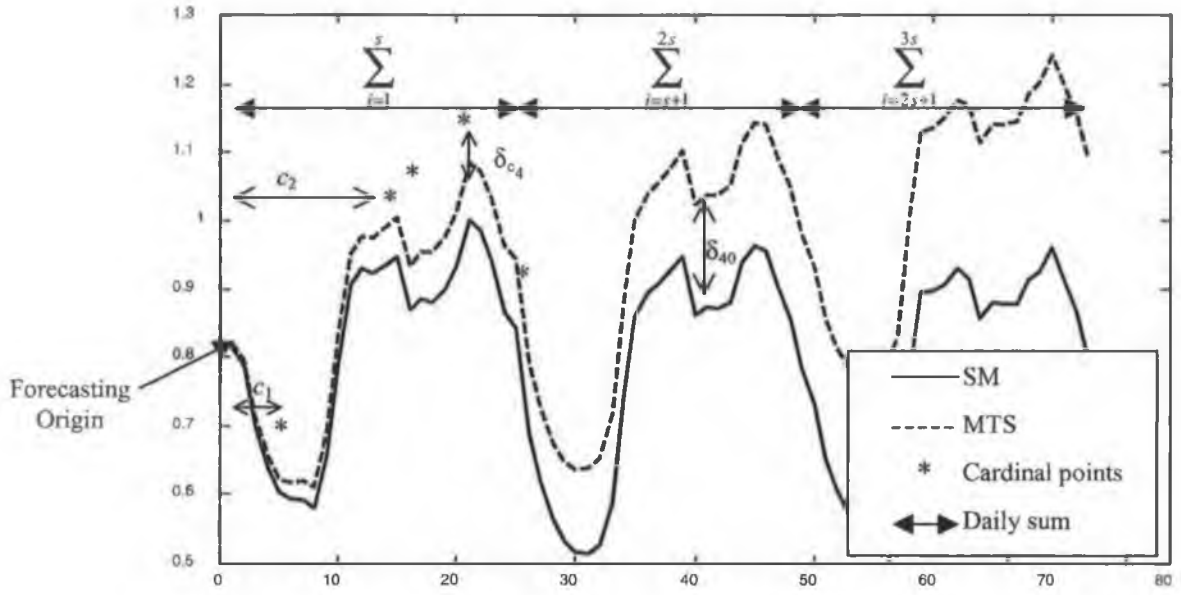

Figure 3.8 Diagram showing MTS technique.

The SM is restricted to a state space model but the PM's and daily-sum models are unrestricted. Let the state vector of the SM at the forecast *origin* be $\boldsymbol{\theta}(k)$. The MTS technique partitions the state vector into states which are *fixed* at the forecasting origin, while allowing others to be *varied* (*adjustable* or *freed* states) by the PM and daily-sum forecasts. Thus, the end result of the MTS technique is a new state vector, $\boldsymbol{\theta}^*(k)$, in which the freed states have been varied. This state vector may then be propagated forward to produce load forecasts.

If the dimension of the state vector is n and the number of adjustable states, r, then there are n-r fixed states. The state transition equation (Equation 3.42) of a state space model may be partitioned in terms of the fixed and adjustable states (Murray *et. al.*, 2000), as:

$$\boldsymbol{\theta}(k+1) = \begin{bmatrix} \boldsymbol{\theta}_1(k+1) \\ \boldsymbol{\theta}_2(k+1) \end{bmatrix} = \boldsymbol{\Phi}\boldsymbol{\theta}(k) = \begin{bmatrix} \boldsymbol{\Phi}_1 & \boldsymbol{\Phi}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}_1(k) \\ \boldsymbol{\theta}_2(k) \end{bmatrix} \tag{3.72}$$

where $\boldsymbol{\theta}_1(k)$ and $\boldsymbol{\theta}_2(k)$ are vectors of fixed and adjustable states at the forecasting origin, k, respectively and $\boldsymbol{\Phi}_1$ and $\boldsymbol{\Phi}_2$ are the partitions of the state transition matrix

associated with $\theta_1(k)$ and $\theta_2(k)$. In addition the observation matrix may also be partitioned in terms of the fixed and freed states (see Equation 3.41), as:

$$x(k) = \boldsymbol{H}\boldsymbol{\theta}(k) + \varepsilon(k) = \begin{bmatrix} \boldsymbol{H}_1 & \boldsymbol{H}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}_1(k+1) \\ \boldsymbol{\theta}_2(k+1) \end{bmatrix} + \varepsilon(k) \tag{3.73}$$

where $\boldsymbol{H}$ is the observation matrix (Section 3.3.2), $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ are the partitions of $\boldsymbol{H}$ associated with $\boldsymbol{\theta}_1(k)$ and $\boldsymbol{\theta}_2(k)$. The dimensions of these variables are given below in Table 3.2.

Table 3.2 A list of variables used in the MTS technique (Part I).

Variable	Dimension	Description
n	Scalar	Length of the state vector $\boldsymbol{\theta}(k)$
r	Scalar	Number of freed states of $\boldsymbol{\theta}(k)$
$x(k)$	Scalar	Load at time k
$\boldsymbol{\theta}(k)$	$n\times 1$	State vector at time k
$\boldsymbol{H}$	$1\times n$	Observation matrix
$\boldsymbol{\Phi}$	$n\times n$	Transition matrix
$\boldsymbol{\theta}_1(k)$	$(n\text{-}r)\times 1$	Vector of fixed states at time k
$\boldsymbol{\theta}_2(k)$	$n\times 1$	Vector of freed states at time k
$\boldsymbol{\Phi}_1$	$(n\text{-}r)\times(n\text{-}r)$	Partition of $\boldsymbol{\Phi}$ for fixed states
$\boldsymbol{\Phi}_2$	$r\times r$	Partition of $\boldsymbol{\Phi}$ for freed states
$\boldsymbol{H}_1$	$1\times(n\text{-}r)$	Partition of $\boldsymbol{H}$ for fixed states
$\boldsymbol{H}_2$	$1\times r$	Partition of $\boldsymbol{H}$ for freed states

The SM may be used to generate forecasts of the state vector i-steps ahead by repeated use of Equation (3.72) (Murray *et. al.*, 2000) as:

$$\boldsymbol{\theta}(k+i) = \begin{bmatrix} \boldsymbol{\theta}_1(k+i) \\ \boldsymbol{\theta}_2(k+i) \end{bmatrix} = \boldsymbol{\Phi}^i\boldsymbol{\theta}(k) = \begin{bmatrix} \boldsymbol{\Phi}_1^i & \boldsymbol{\Phi}_2^i \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}_1(k) \\ \boldsymbol{\theta}_2(k) \end{bmatrix} \tag{3.74}$$

where $\boldsymbol{\Phi}_1^i$ and are $\boldsymbol{\Phi}_2^i$ the partitions of $\boldsymbol{\Phi}^p$ associated with $\theta_1(k)$ and $\theta_2(k)$ (note: $\boldsymbol{\Phi}_2^i$ is not the same as $(\boldsymbol{\Phi}_2)^i$, which is in general does not exist as $\boldsymbol{\Phi}_2$ is non-square) .

For the MTS model, an i-step ahead forecast of the state vector is generated by the altered state vector, $\boldsymbol{\theta}^*(k)$, (Murray *et. al.*, 2000) as:

$$\boldsymbol{\theta}^*(k+i)=\begin{bmatrix}\boldsymbol{\theta}_1(k+i)\\ \boldsymbol{\theta}_2^*(k+i)\end{bmatrix}=\boldsymbol{\Phi}^p\boldsymbol{\theta}^*(k)=\begin{bmatrix}\boldsymbol{\Phi}_1^i & \boldsymbol{\Phi}_2^i\end{bmatrix}\begin{bmatrix}\boldsymbol{\theta}_1(k)\\ \boldsymbol{\theta}_2^*(k)\end{bmatrix} \tag{3.75}$$

and $\boldsymbol{\theta}_2^*(k)$ is the vector of freed states which have been altered by the PM and end-sum forecasts. A corresponding forecast of the load at $k+i$ may then be made from $\boldsymbol{\theta}_2^*(k+i)$ by use of the observation equation (Equation 3.41), as:

$$\hat{y}_{mts}(k+i)=\boldsymbol{H}\boldsymbol{\theta}^*(k+i) \tag{3.76}$$

where $\hat{y}_{mts}(k+i)$ is the MTS forecast of the load, y, at time $k+i$. The MTS technique calculates $\boldsymbol{\theta}_2^*(k)$ by means of a solution to an over-determined equation set in $\boldsymbol{\theta}_2^*(k)$ (Murray, *et. al.*, 2000) as:

$$\begin{matrix}\text{Constraint 1}\\ \\ \\ \text{Constraint 2}\\ \\ \\ \text{Constraint 3}\\ \\ \\ \end{matrix}\left\{\begin{bmatrix}\boldsymbol{H}\boldsymbol{\Phi}\boldsymbol{\theta}(k)-\boldsymbol{H}_1\boldsymbol{\Phi}_1^1\boldsymbol{\theta}_1(k)\\ \vdots\\ \boldsymbol{H}\boldsymbol{\Phi}^N\boldsymbol{\theta}(k)-\boldsymbol{H}_1\boldsymbol{\Phi}_1^N\boldsymbol{\theta}_1(k)\\ \hat{y}_{c_1}-\boldsymbol{H}_1\boldsymbol{\Phi}_1^{c_1}\boldsymbol{\theta}_1(k)\\ \vdots\\ \hat{y}_{c_M}-\boldsymbol{H}_1\boldsymbol{\Phi}_1^{c_M}\boldsymbol{\theta}_1(k)\\ \hat{y}_{s_1}-\sum_{j=1}^{S}\boldsymbol{H}_1\boldsymbol{\Phi}_1^j\boldsymbol{\theta}_1(k)\\ \vdots\\ \hat{y}_{s_P}-\sum_{j=(P-1)S}^{PS}\boldsymbol{H}_1\boldsymbol{\Phi}_1^j\boldsymbol{\theta}_1(k)\end{bmatrix}\right.=\begin{bmatrix}\boldsymbol{H}_2\boldsymbol{\Phi}_2^1\\ \vdots\\ \boldsymbol{H}_2\boldsymbol{\Phi}_2^N\\ \boldsymbol{H}_2\boldsymbol{\Phi}_2^{c_1}\\ \vdots\\ \boldsymbol{H}_2\boldsymbol{\Phi}_2^{c_M}\\ \sum_{j=1}^{S}\boldsymbol{H}_2\boldsymbol{\Phi}_2^j\\ \vdots\\ \sum_{j=(P-1)S}^{PS}\boldsymbol{H}_2\boldsymbol{\Phi}_2^j\end{bmatrix}\boldsymbol{\theta}_2^*(k)+\begin{bmatrix}\delta_1\\ \vdots\\ \delta_N\\ \delta_{c_1}\\ \vdots\\ \delta_{c_M}\\ \delta_{s_1}\\ \vdots\\ \delta_{s_P}\end{bmatrix} \tag{3.77}$$

where S is the length of the season (in this case one day or 24 points) and the other notation is explained below. This equation is made up of three types of soft constraint (Murray *et. al.*, 2000) as:

1. A *smoothing constraint* in which the MTS forecast from $k+1$ to $k+N$ (generated by Equation 3.76) deviates from the SM estimate by $\delta_1,\ldots,\delta_N$ (Figure 3.8 shows δ_{40} for example),

2. A *PM constraint* in which M parallel model forecasts, $\hat{y}_{c_1},\ldots,\hat{y}_{c_M}$, of the load at times $k+c_1, \ldots, k+c_M$ deviate from the SM forecasts at those times by $\delta_{c_1},\ldots,\delta_{c_M}$ (Figure 3.8 shows δ_{c_4} for example; the deviation of the 4th cardinal point), and
3. A daily-sum constraint in which P forecasts of the daily-sum, denoted $\hat{y}_{s_1},\ldots,\hat{y}_{s_P}$, for days 1 to P deviate from the sum of the SM forecasts over those days by $\delta_{s_1},\ldots,\delta_{s_P}$.

The matrices in Equation (3.77) may be renamed as:

$$\boldsymbol{A}(k) = \boldsymbol{B}\boldsymbol{\theta}_2^{*}(k) + \boldsymbol{\Lambda}(k) \tag{3.78}$$

where $\boldsymbol{A}(k)$ is the left hand side of Equation (3.77), $\boldsymbol{B}$ is the first term on the right hand side and $\boldsymbol{\Lambda}(k)$ is the last term on the right hand side of Equation (3.77). Murray *et. al.* (2000) then proposes solving Equation (3.78) for $\boldsymbol{\theta}_2^{*}(k)$ by using weighted least squares to give:

$$\boldsymbol{\theta}_2^{*} = \left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{A} \tag{3.79}$$

where $\boldsymbol{W}$ is a diagonal matrix of weights which allows Equation (3.77) to recognise that, for example, the deviation of the PM from the SM solution may be more important than other deviations.. Table 3.3, below, summarises the variables defined in the above analysis and their dimensions.

Table 3.3 A list of variables used in the MTS technique (Part II).

Variable	Dimension	Description
N	Scalar	Number of points used in the smoothing constraint
M	Scalar	Number of cardinal point forecasts used.
P	Scalar	Number of end-sum forecasts used.
$\hat{y}_{c_i}$	Scalar	i^{th} cardinal point forecast
c_i	Scalar	The distance from the forecasting origin to the i^{th} cardinal point.
δ_{c_i}	Scalar	The deviation of the i^{th} cardinal point from the MTS forecast.
$\hat{y}_{s_i}$	Scalar	i^{th} end-sum forecast.
S	Scalar	Length of the summations for end-sum model.
δ_{s_i}	Scalar	The deviation of $\hat{y}_{s_i}$ from the summation of the MTS forecasts from $k = S(i-1)$ to Si
δ_i	Scalar	The deviation of the state space model forecast at $k+i$ from the MTS forecast at $k+i$.
$\boldsymbol{A}(k)$	$(N+M+P)\times 1$	Left hand side of Equation (3.76).
$\boldsymbol{B}(k)$	$(N+M+P)\times r$	1^{st} term on the right hand side of Equation (3.76).
$\boldsymbol{W}$	$(N+M+P)\times (N+M+P)$	Diagonal weight matrix.
$\Lambda(k)$	$(N+M+P)\times 1$	Vector of deviations.

The disadvantages of this technique are:

1. A linear state space model is required for the SM,
2. The linear state space model does not include exogeneous inputs,
3. A technique does not exist for optimising the weight matrix, instead Murray *et. al.* (2000) provide guidelines for the values of the weights, and
4. There are no general guidelines for the selection of freed states.

However the advantages of this technique are:

1. The shape of the SM forecast is preserved *via* constraint 1 in Equation (3.77),
2. The number of PM forecasts is not restricted,
3. The adjustment of the SM forecast may be controlled via $\boldsymbol{W}$,
4. A forecast of the daily consumption can be combined with the SM forecast, and
5. An MTS forecast can be produced for horizons greater than one day. This is important considering that propagation errors in the SM forecast grow over time.

3.4 Non-Linear Techniques for Load Forecasting

A non-linear system is one for which the principle of superposition (Section 3.3) does not apply (Sinha, 1991). There are three factors typically present in short-term electrical load which make non-linear forecasting techniques appealing:

1. The non-linear relationship between load and temperature mentioned in Section 2.3.3,

2. The presence of a non-linear auto-regressive relationship in load (as pointed out by Fay *et. al.*, 2000, Darbellay and Slama, 2000 and Mori and Kobayashi, 1996), and

3. The non-stationarity of the load due to the trend (As examined in Section 2.3.1).

The non-linear techniques examined in this section differ in the means by which non-linearities are modelled. Parametric techniques (Section 3.4.1) model a system using a combination of linear transforms, multiplications and delays. Fuzzy logic techniques (Section 3.4.2) model a system using several models which operate in overlapping regions. Neural networks model a system using a combination of weighted non-linear mappings (Section 3.4.3).

3.4.1 Parametric Non-Linear Techniques

Parametric non-linear techniques are based on the Volterra series (Schetzen, 1980) which describes a time *invariant* (a system which does not alter with time) non-linear system in terms of the input at time t[%], $u(t)$ and the output $x(t)$ as a sequence of integrals (Schetzen, 1980) as:

$$x(t) = \int_{-\infty}^{\infty} h_1(\tau_1)u(t-\tau_1)d\tau_1 + \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_2(\tau_1,\tau_2)u(t-\tau_1)u(t-\tau_2)d\tau_1 d\tau_2 + \cdots + \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} h_n(\tau_1,\tau_2,\ldots,\tau_n)u(t-\tau_1)u(t-\tau_2)\ldots u(t-\tau_n)d\tau_1 d\tau_2 \ldots d\tau_n \quad (3.80)$$

where $h_i(\bullet)$ is called the i^{th} order *Volterra kernel*, and n is the order of the non-linear system. Note that for a 1st order system Equation (3.80) reduces (Schetzen, 1980) to:

$$x(t) = \int_{-\infty}^{\infty} h_1(\tau_1)u(t-\tau_1)d\tau_1 \quad (3.81)$$

which is the well known input-output relationship of a linear system and where $h_1(t)$ is known as the impulse response. Just as the linear system in Equation (3.81) may be expressed in terms of the Laplace transform of $h_1(t)$, a Volterra series may be expressed in terms of the Laplace transform of the Volterra kernels $h_i(\bullet)$. For example (Schetzen, 1980) a 2nd order Volterra system with:

$$h_1(\tau_1) = 0$$
$$h_2(\tau_1,\tau_2) = \begin{cases} e^{-(a\tau_1+b\tau_2)} & \text{for } 0 \le \tau_1 \le c\tau_2 \\ 0 & \text{otherwise} \end{cases} \quad (3.82)$$

where a, b and c are constants greater than zero, has the Laplace transform:

$$H_1(s_1) = 0$$
$$H_2(s_1,s_2) = \frac{1}{(b+s_2)(b+a+s_1+s_2)} \quad (3.83)$$

where s_1 and s_2 are Laplace operators. The block diagram for this system is shown in Figure 3.9 below.

[%] Note t is used instead of k for time as it is continuous here.

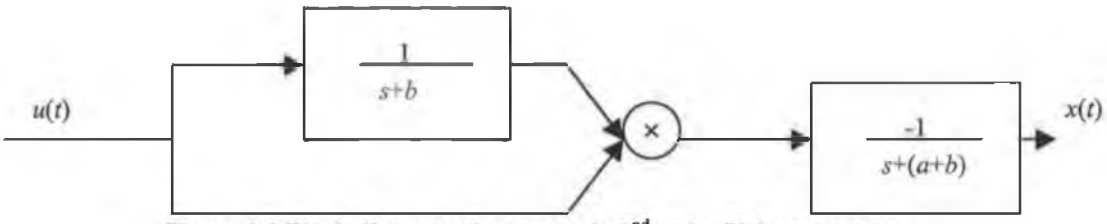

Figure 3.9 Block diagram of an example 2nd order Volterra system.

As can be seen, the system can be reduced to two linear transforms and a multiplication. For a second order system there can be an infinite number of multiplications involved in the Laplace representation (Schetzen, 1980). This applies equally for higher order systems. Modelling a non-linear system with a Volterra series has three disadvantages (Mars *et. al.*, 1996):

1. Evaluating the Volterra kernels can be difficult, even when the order is known (Bai, 2002),

2. The series may not converge, i.e. there may an infinite number of multiplications as mentioned above, and

3. The Volterra series may only be used to model time invariant processes.

As a result of the last point above, the Wiener and Hammerstein models were developed to model non-linear time variant processes.

3.4.1.1 Wiener Models

The Wiener model consists of a linear model, with memory, followed by a non-linear transform with a similar structure to a Volterra series (Figure 3.10).

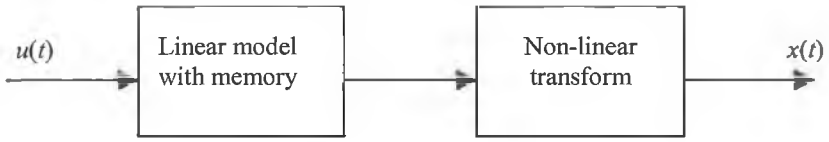

Figure 3.10 Block diagram of a Wiener model.

The Wiener model suffers from the same disadvantages as the Volterra model except that the linear model allows a large class of time varying processes to be modelled (Schetzen, 1980). The Hammerstein model is similar to the Wiener model except that the order of the non-linear transform is reversed as shown in the next section.

3.4.1.2 Hammerstein Models

The Hammerstein model consists of a non-linear transform followed by a linear model (Figure 3.11).

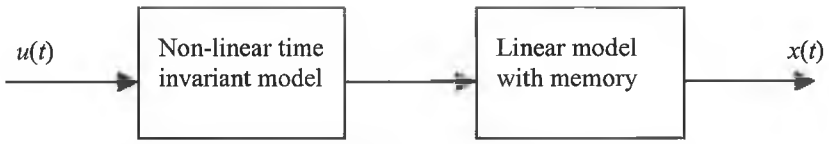

Figure 3.11 Block diagram of a Hammerstein model.

This parametric non-linear modelling technique is attractive for short-term load forecasting due to the nature of the load-temperature relationship (Section 2.3.3). As discussed in Section 2.3.3.3 the load-temperature relationship typically follows a curve, as shown in Figure 2.12 (shown below for convenience).

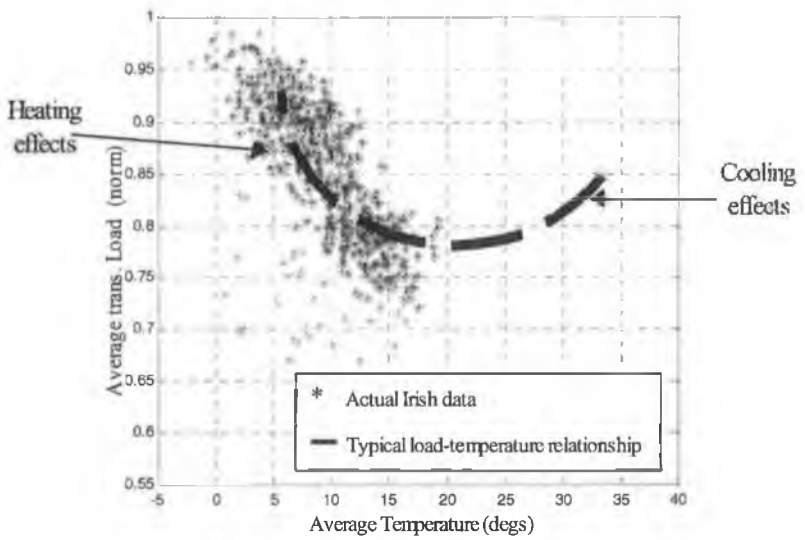

Figure 3.12 (2.12) Typical and actual scatter plot of temperature-load relationship (Working days).

Several short term load forecasting applications (Haida and Muto, 1994, Rahman and Bhatnagar, 1988, Sforna, 1995, Lu *et. al.*, 1989 and Hyde and Hodnett, 1997a) apply a non-linear transform to the temperature prior to modelling. These non-linear transforms all have in common that they seek to linearise the relationship between load and temperature.

For example, Haida and Muto (1994) linearise the load-temperature relationship (for their system) using polynomial transformations. Specifically, Haida and Muto (1994) first transform the daily maximum temperature, $t_{max}(k)$, using transforms of the form:

$$t'_{max}(k) = f(t_{max}(k)) = \sum_{i=0}^{n} a_i (t_{max}(k))^i \tag{3.84}$$

where $t'_{max}(k)$ is the transformed daily maximum temperature, $f(\bullet)$ is the polynomial transform, n is the order of the polynomial and a_i are the associated coefficients. Several of these transforms are then used to linearise the load-temperature relationship as:

$$y_{peak}(k) = c + \sum_{i=1}^{m} f_i(t_{max}(k)) + \varepsilon(k) \tag{3.85}$$

where $y_{peak}(k)$ is the peak load to be forecast, c is a constant, $f_i(\bullet)$ is polynomial transform i (Equation 3.84), m is the number of polynomial transforms used and $\varepsilon(k)$ is the modelling error. In terms of the Hammerstein technique, Equation (3.84) represents the non-linear transform and Equation (3.85) represents the linear model.

The Hammerstein load forecasting models used by Haida and Muto (1994), Rahman and Bhatnagar (1988), Sforna (1995), and Lu *et. al.* (1989) give no procedure for determining the form of the non-linear transform. In addition the techniques do not model any non-linear relationship between the load and autoregressive elements of the load (a non-linear factor mentioned in Section 3.4).

3.4.2 Fuzzy Logic Techniques

Fuzzy logic techniques perform function approximation. This is achieved by first partitioning the input and output space into several *overlapping* regions (Kosko, 1997). A separate model is then used to approximate the function within each region. The final output is a function of these model outputs and the degree to which each region is relevant.

As an example, consider a simplistic peak load forecasting model which has a single input, the average temperature on day k, $t_{av}(k)$, and a single output, the peak load forecast for day k, $\hat{y}(k)$ (Figure 3.13). In this model the input space is divided into two regions; one for a *hot* temperature and the other for a *cold* temperature. Two linear models are used within each region; a model reflecting the heating effect at cold temperatures and a model reflecting the cooling effect at hot temperatures (Section 2.3.3.3). For a particular input, for instance $t_{av}(0)$ (Figure 3.13), the output of the models for the hot and cold regions are $\hat{y}_1(k)$ and $\hat{y}_2(k)$ respectively.

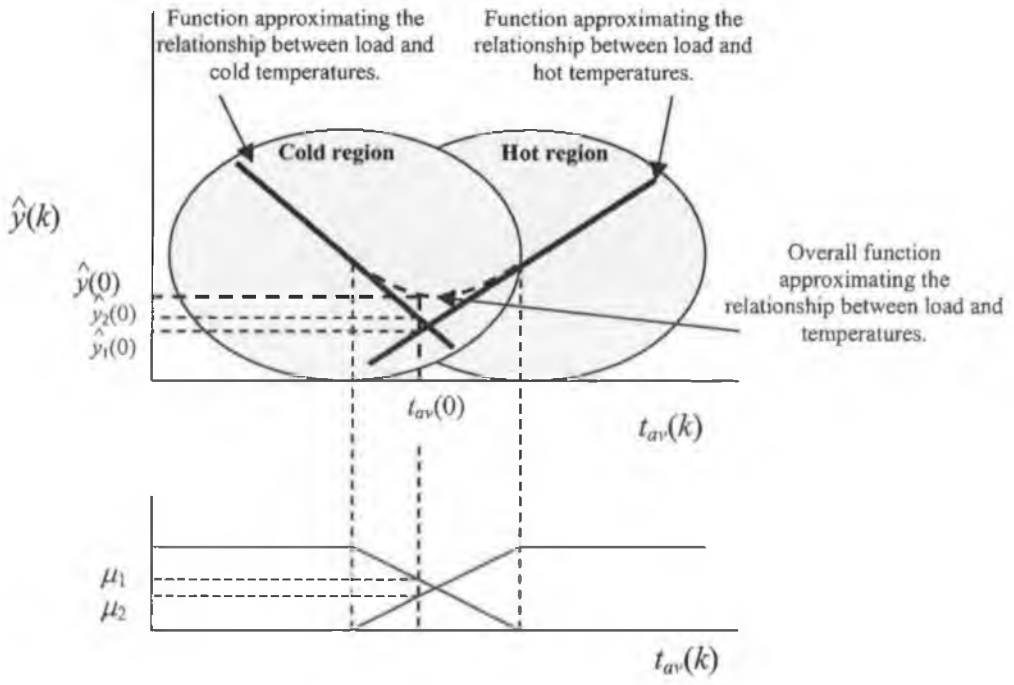

Figure 3.13 A fuzzy approximation for the relationship between load and temperature and the associated membership function.

The relationship between the inputs and outputs can be more easily expressed using a *rule base*, for this example the rule base is:

$$\begin{aligned} R_1&: \text{ IF } (t_{av}(k) \text{ is COLD) THEN } \hat{y}_1(k) = f_1(t_{av}(k)) \\ R_2&: \text{ IF } (t_{av}(k) \text{ is HOT) THEN } \hat{y}_2(k) = f_2(t_{av}(k)) \end{aligned} \tag{3.86}$$

where R_1 and R_2 are the rules for the cold and hot regions respectively and $f_1(\bullet)$ and $f_2(\bullet)$ are the linear models for the cold and hot regions respectively. As can be seen in Figure 3.13, $t_{av}(0)$ lies in both regions and thus it is neither fully cold nor fully hot. The degree to which a region is relevant is called the membership* and is defined by the membership function (An example is shown in Figure 3.13). In the current example $t_{av}(0)$ has a membership value of μ_1 for the cold region and μ_2 for the hot region. The final output, $\hat{y}(k)$, is a formed by a function of $\hat{y}_1(k)$, $\hat{y}_2(k)$, μ_1 and μ_2. This function is known as a *defuzzifier*. A *centre of gravity* defuzzifier (Kosko, 1997), for example, forms the output as:

$$\hat{y}(k) = \frac{\sum_{i=1}^{N} \mu_i \hat{y}_i(k)}{\sum_{i=1}^{N} \mu_i} \tag{3.87}$$

where N is the number of regions.

For a fuzzy model with N regions per input, Q inputs and M outputs there is N^{M+Q+1} rules required (Kosko, 1997). Thus as the number of inputs increases the number of rules increases exponentially. Thus the number of *inputs* that may be used with a fuzzy model is *restricted*. This is the greatest setback of fuzzy models and known as *the curse of dimensionality* (Kosko, 1997).

* This definition of membership is consistent with that in Section 3.2.1.2 where cluster analysis was used to determine membership functions.

There are several types of Fuzzy logic models which differ in the models used to approximate the function within the regions, among the most important are:

- The Fuzzy Inference Engine (FIE), which uses the expected value of the outputs for each region (Kosko, 1997),

- The Takagi-Sugueno-Kang (TSK) model which uses a linear regression model (Kosko, 1997), and

- The Fuzzy ARMAX (FARMAX) model which uses an ARMAX model (Yang and Huang, 1998).

In addition, the shape of the membership functions, the location of the regions, and the type of defuzzifier must be decided. The following section details the approaches taken in the field of Short-Term Load Forecasting (STLF).

3.4.2.1 Fuzzy Logic Models for Short-Term Load Forecasting

There have been several approaches taken in STLF using fuzzy techniques, these can be classed as:

1. **Input fuzzification**. In this case the membership of the inputs to the fuzzy sets, μ_i, are calculated and used as inputs for a separate non-fuzzy model. As noted by Bitzer and Rößer, (1998) human sensitivity to weather is not exact and is based on fuzzy measures like 'cold' and 'warm'. For example, their model incorporates five fuzzy regions for the temperature ranging from 'very cold' to 'hot'. Given a certain temperature there is a corresponding membership assigned to each of these regions. These memberships are then used as inputs for a neural network. Similar approaches are taken by Elkateb *et. al.* (1998), Müller and Petrisch (1998) and Tamimi and Egbert (2000). There is a trade-off involved in this approach; although the membership variables are *presumably* more

correlated with the output, there are more of them. This is important as the studies mentioned above (Bitzer and Rößer, 1998, Elkateb *et. al.*, 1998, Müller and Petrisch, 1998 and Tamimi and Egbert, 2000) use neural networks as the non-fuzzy model. As will be seen in Section 3.4.3, training becomes more difficult for neural networks as the number of inputs increases.

2. **Pattern matching techniques**. This is based on the assumption that if the load on day n+1 has the same *history* (or pattern, an example is given below) as the load on day n+1-i then these loads will be equal. For example, Otto and Schunk (1999) use the rule base:

$$R_i\text{: IF } (\boldsymbol{y}_n = \boldsymbol{y}_{n-i}) \text{ AND } (\boldsymbol{y}_{n-1} = \boldsymbol{y}_{n-i-1}) \ldots \text{ AND } (\boldsymbol{y}_{n-N} = \boldsymbol{y}_{n-i-N}) \text{ THEN } \boldsymbol{y}_{n+1} = \boldsymbol{y}_{n-i+1} \quad (3.88)$$

where R_i is the i^{th} rule, $\boldsymbol{y}_n$ is a vector containing the 24 loads on day n, the average temperature on day n and the average solar radiation on day n. N is the *window* size (i.e. the number of days that are compared to each other) and n+1 is the day to be forecast. In this case $\boldsymbol{y}_n$ to $\boldsymbol{y}_{n\ N}$ is the history or pattern of the day to be forecast and $\boldsymbol{y}_{n-i}$ to $\boldsymbol{y}_{n-i-N}$ is the pattern that it is being compared to. Similar approaches have been used by Dash *et. al.* (1995b) and Papadakis *et. al.* (1999). This approach is advantageous in that it mirrors the manual approach taken by many system operators (Jabbour *et. al.*, 1988, Lonergan and Ringwood, 1995) and can thus be understood intuitively by the user.

3. **Function approximation techniques**. These techniques seek to approximate the function between a set of inputs and the load to be forecast. For example, Yang and Huang (1998) model load as a function of the previous 24 hours of load and weather inputs using a FARMAX model. In this case Yang and Huang note that the forecasting accuracy of the FARMAX model is not as sensitive to model structure as a traditional ARMAX model (Section 3.3.1.2). Similarly, Mastorocostas *et. al.* (1999) model load as a function of the loads, the average, the maximum and the minimum daily temperatures in the previous seven days

using a TSK type fuzzy model, trained using the gradient descent algorithm. Mori and Kobayashi (1996) model load using only auto-regressive terms as:

$$y(k) = f\left(y(k-1), y(k-1) - y_{mean}(k-1), y_{mean}(k) - y_{mean}(k-1) \right) + \varepsilon(k) \qquad (3.89)$$

where $y(k)$ is the load at time k, $\varepsilon(k)$ is the modelling error, $f(\bullet)$ is the function to be approximated and $y_{mean}(k)$ is an average defined as:

$$y_{mean}(k) = \frac{1}{3}\left(y(k-7\times 24) + y(k-14\times 24) + y(k-21\times 24) \right) \qquad (3.90)$$

The function $f(\bullet)$ is approximated by use of an FIE. Interestingly the third term in $f(\bullet)$, $y_{mean}(k)$- $y_{mean}(k\text{-}1)$, is found to be highly non-linear and requires ten membership functions while the first two terms require only two membership functions each. Other studies that have used this approach are Srinivasan *et. al.* (1995), Raanaweera *et. al.* (1996), Dash *et. al.* (1997), Dash *et. al.* (1994), Abdelaziz and Gouda, 1998, and Bretschneider *et. al.* (1999).

4. **Residual modelling techniques**. This approach first uses a non-fuzzy model to produce a forecast of the load. The error in this forecast is then estimated using a fuzzy model. An example of this is the method used on Irish data by Lonergan and Ringwood (1995) in which the load is first forecast using a *standard day*. The standard day is simply the load curve for the same day on the previous week. This load curve is then adjusted using a FIE with temperature, and other weather variables as inputs. In a later study Commarmond and Ringwood (1997) used a similar technique employing a TSK fuzzy model. Similarly Kim *et. al.* (2000) use a neural network to generate a forecast of a daily load curve which is altered using a FIE fuzzy model. Similar approaches are used by Liang and Cheng (2000) for a linear regression model and Srinivasan *et. al.* (1999) with a Kohonen map.

5. **Fuzzy post-processor**. Sharaf and Lie (1995) use a FIE to determine error bands and variance of a load forecast produced by a neural network. Thus the

FIE is not explicitly used to forecast the load but rather to calculate statistics of the forecast.

3.4.2.2 Radial Basis Function Neural Networks

Radial Basis Function (RBF) neural networks are a special case of fuzzy models (Kosko, 1997) and are thus included in this section as opposed to the section on neural networks (Section 3.4.3). The RBF may be used to approximate the function relating an input vector at time k, $\boldsymbol{u}(k) \in \mathbf{R}^{1\times Q}$, to an output at time k, $x(k)$, (Mars *et. al.*, 1996) as:

$$x(k) = \sum_{j=1}^{N} \alpha_j f_j(\boldsymbol{u}(k) - \boldsymbol{C}_j) \tag{3.91}$$

where α_j is a weight, $f_j(\bullet)$ is a *radial basis function* (explained below), N is the number of radial basis functions used and $\boldsymbol{C}_j \in \mathbf{R}^{1\times Q}$ is called the *centre* of $f_j(\bullet)$. The radial basis functions may take on several forms, one example (more may be found in Mars *et. al.*, 1996) is the *Gaussian form* (Mars *et. al.*, 1996), which may be expressed as:

$$f_j(\boldsymbol{u}(k) - \boldsymbol{C}_j) = \exp\left(\frac{-\sum_{i=1}^{Q} u_i(k) - c_{i,j}}{\sigma_j^2} \right) \tag{3.92}$$

where Q is the dimension of the input, u_i is the i^{th} element of $\boldsymbol{u}(k)$, $c_{i,j}$ is the i^{th} element of $\boldsymbol{C}_j$ and σ_j is the standard deviation of the Gaussian radial basis function.

In terms of fuzzy logic models, the radial basis functions cover a region of the input space centred on $\boldsymbol{C}_j$ and the width of the region is determined by σ_j. Thus $f_j(\bullet)$ represent the models and the membership functions. De-fuzzification is accomplished by means of a weighted average using α_j.

Mitchell (1992) has questioned the use of radial basis function neural networks as time-series models for several reasons:

- As the network performs an interpolation between prior values of the series it is assumed that the series is not changing with time,

- As the dimension of the input variables increases (for example, five causal variables are used instead of four) the number of prior examples required to maintain the density of examples increases geometrically; i.e. the *curse of dimensionality* (Section 3.4.2). Mitchell (1992), thus notes that more than six inputs would seem impractical. Kodogannias and Anagnostakis (1999) similarly noted that the number of inputs is restricted in an application of RBF networks to short-term load forecasting. However, Gontar and Hatziargyriou (2004) found that an RBF with nineteen inputs was superior to a feed forward neural network (Section 3.4.3). This was not a valid comparison of the two methods however, as the feed forward neural network used only a single layer which is not typically the preferred option (Section 3.4.3.1), and

- Small amounts of noise lead to large gaps between the training examples in *n*-dimensional space and thus the network is very sensitive to input noise (Mitchell, 1992).

3.4.3 Neural Network Techniques

Neural network techniques are *black-box* modelling techniques. That is, they require no understanding of the physical process underlying the data (Aussem, 1999). The techniques are based on approximating a function via a set of basic processing units called *neurons* or *nodes* (Mars *et. al.* 1996). The structure of a typical neuron is shown in Figure 3.14 below. The inputs u_1, ..., u_N are multiplied by weights w_1, ..., w_N and the sum of these and a bias, b, is transformed using a non-linear *activation* function $f(\bullet)$ to form the output x as (Mars *et. al.*, 1996):

$$x = f(w_1 u_1 + w_2 u_2 + ... + w_N u_N + b) \tag{3.93}$$

where N is the number of inputs to the neuron. Note also that some of the outputs may be fed back as inputs.

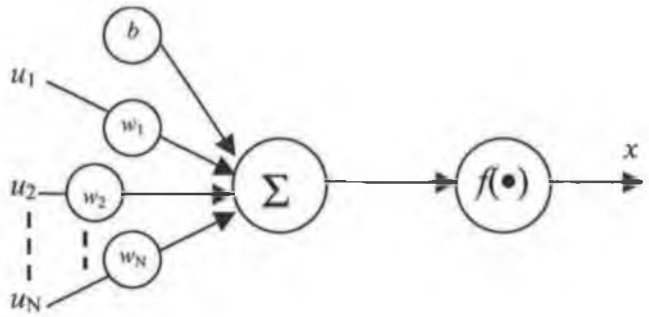

Figure 3.14 Structure of a neuron.

The neural networks examined in this section differ:

- In the way in which these neurons are arranged (*topology*),
- The activation functions,
- The means by which the weights are determined (*training algorithm*), and
- Whether feedback is used.

3.4.3.1 Feed Forward Neural Networks

In a typical feed forward neural network, also known as a Multi Layer Perceptron (MLP), the neurons are arranged in layers. An example of a 3 layer MLP is shown in Figure 3.15 (next page). The network has N inputs, *the input layer**, which are first fed into a layer of neurons called *hidden layer* 1. The output of hidden layer 1 is fed forward to hidden layer 2 and then to the final layer of neurons called the *output layer* (Figure 3.15). A *fully connected* MLP is one in which each node is connected to *every* node in the following layer. However, this is not always the case, as will be seen in the next section. In this example, there are four and two neurons in the first and second hidden layers, respectively, and one in the output layer thus the network structure is denoted as a 4×2×1 network. The output of *node j* in layer k, $x_{j,k}$, may be expressed in a similar way to Equation (3.93) as (Haykin, 1999):

$$x_{j,k} = f_{j,k}\left(\sum_{i=1}^{N_{k-1}} w_{i,j,k} x_{i,k-1} + b_{j,k}\right) \tag{3.94}$$

where $f_{j,k}(\bullet)$is the activation function used, $w_{i,j,k}$ is the weight connecting node i in layer k-1 to node j in layer k, $b_{j,k}$ is the bias on node j in layer k and N_{k-1} is the number nodes in layer k-1.

The activation functions which may be used can be *sigmoidal* (see Schalkoff, 1997), *sinusiodal* (see Choueiki *et. al.*, 1997) or *linear* (see Schalkoff, 1997) among others. Typically the activation functions in the hidden layers are sigmoidal, while the activation functions in the output layer are linear (Hippert *et. al.*, 2001). Choueiki *et. al.* (1997) examined the most important factors in designing neural networks for STLF and found that the choice of activation function in the output layer was the most important factor followed by the activation function in the hidden layers.

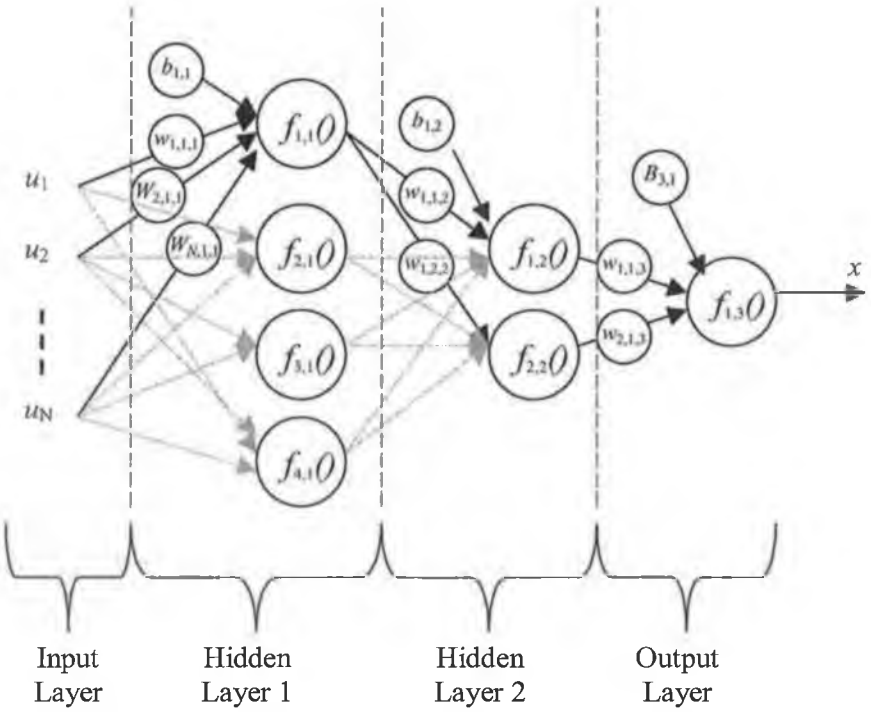

Figure 3.15 Feed forward neural network with 4×2×1 structure[†].

[*] The input layer is typically not counted as a layer as it just presents the inputs to the network.
[†] Some weights and biases have been suppressed for clarity.

The MLP maps inputs to outputs and has no memory. When the inputs represent previous values of a time series and external inputs it can be considered as a non-linear equivalent of an ARX model (Connor *et. al.*, 1992). Theoretically it has been shown that a feed forward neural network is a *universal approximator* (Funahashi, 1989), i.e. it can approximate any continuous function to any degree of accuracy, given a sufficient number of nodes. Although this result may seem appealing, selecting the correct number of nodes and training the network to approximate the function is a difficult task (Mars *et. al.*) due to two problems:

1. The network topology must be determined, and
2. The weights and biases for this topology must be estimated.

Topology Determination

The number of hidden layers and nodes and the connections between these nodes form the topology of a network. The ability of an MLP to model complex functions rises with the number of hidden layers, nodes and connections. However, its ability to overfit the data also increases. There is no one agreed approach for topology determination for MLP's. Primarily the number of hidden layers must be chosen.

With respect to the field of STLF, studies which use a single hidden layer can be found in Chen *et. al.* (1992), Mohamad *et. al.* (1996), Drezga and Rahman (1998) and Lu and Vemuri (1993). Although a single hidden layer is the most popular configuration (Reinschmidt, 1995), Lee *et. al.* (1992) found that an MLP with a single hidden layer required a large number of nodes to model the load. This large number of nodes was subsequently difficult to train. Thus two hidden layers where used and found to give superior performance. Other examples of authors that use two hidden layers have been found in Hsu and Yang (1991b) and Kalaitzakis *et. al.* (2002) to mention but a few. Using three or more hidden layers for STLF is rare. Mizukami and Nishimori (1993) specifically studied the merit of using three or more hidden layers for STLF and found no benefit.

The next step is to determine the number of nodes in each hidden layer and the number of connections. The most common approach for STLF is trial and error. This involves testing several different topologies and selecting the best one (for example, Mohammed *et. al.*, 1996, Drezga and Rahman, 1999, Lee *et. al.*, 1992, Gross and Wagner, 1996, Kiartzis *et. al.*, 1995, Darbellay and Slama, 2000 and Peng *et. al.*, 1992). Another method is *network pruning* (Weigend *et. al.*, 1991, Hassibi *et. al.*, 1992, Le Cun *et. al.*, 1990). A non-fully connected network is less complex than a fully connected network and the method proposed by Weigend *et. al.* (1991) for example, advocates eliminating connections with insignificant weights as a method of reducing the network complexity. Additionally, if all the weights to a node have been eliminated then the node itself can be removed from the network. Examples of this method of topology determination applied to STLF can be found in Lu *et. al.* (1993) and Chen *et. al.* (1992) among others.

Training Algorithms

The most common training algorithm used for the MLP is called the Back Propagation (BP) algorithm (Rumelhart *et. al.*, 1986). This algorithm can be implemented in several steps (further details may be found in Chihocki and Unbehauen, 1993):

1. The training data (previous inputs and associated known outputs or *targets*) are presented to the network,

2. The error between the network outputs and the targets is calculated,

3. The error is used to estimate the derivatives of the weights and biases with respect to the errors,

4. The weights are adjusted, using the derivatives, in the direction of fastest decent of the errors, and

5. The whole process is repeated until the error has reached a desired level or the maximum number of *epochs* (iterations) has been exceeded. Both the desired error level and maximum number of epochs are user specified.

This algorithm is a *calculus based search algorithm*; it attempts to find the optimum values of the weights and biases starting in a neighbourhood around an initial starting point (Mars *et. al.*, 1996). However, as this type of algorithm is based on a local search it is susceptible to converge on a local minimum instead of the global minimum required (Mars *et. al.*, 1996). Training several networks each with different initial conditions can help to alleviate this problem (Schalkoff, 1997). However, as the search involves a multi-dimensional search, for example a 4×2×1 network (Figure 3.15) with 3 inputs has 34* weights and biases to estimate, a good local minimum is often the best that can be achieved. *Global* search algorithms such as Genetic Algorithms (GA) have been used for training neural networks (Mars *et. al.*), however they can be slow to converge as the dimension of search space is so large (Mars *et. al.*). For STLF the BP algorithm is the most popular algorithm used (Hippert *et. al.*, 2001, examples are Chiu *et. al.*, 1997, Dash *et. al.*, 1995a) although GA have been used by several authors (Srinivasan, 1998 and Yang and Huang, 1996 to mention a few). Alternatively, Park et. al., (1991a), suggest adaptively training a neural network so that the weights can be updated as the network proceeds through the data.

* This number is derived from 12 weights and 4 biases from the input to 1st hidden layer, 15 weights and biases from 1st to 2nd hidden layers and 3 weights and biases from the 2nd hidden layer to the output layer.

3.4.3.2 Recurrent Neural Networks

Unlike feed forward neural networks, recurrent neural networks involve some form of feedback in their structure. This means that recurrent neural networks incorporate memory into the structure as is also the case with an MA model (Section 3.3.1.2). When the inputs represent previous values of a time series and external inputs it is often considered as a non-linear equivalent of an ARMAX model (Aussem, 1999, Connor *et. al.*, 1992). The Dynamic Recurrent Neural Network (DRNN) presented by Aussem (1999) is more general and is the non-linear equivalent of a state space model. Figure 3.16 below, shows an MLP in which the output is fed back to the input to form an *Infinite Impulse Response* (IIR) recurrent neural network.

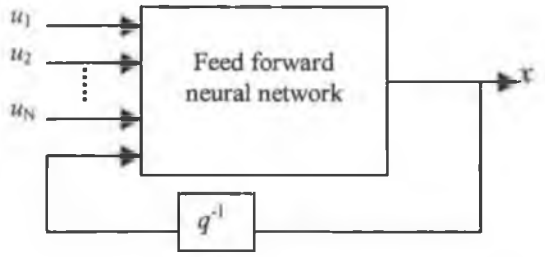

Figure 3.16 IIR recurrent neural network with 4×2×1 structure.

Alternative recurrent neural network representations can be found in Chihocki and Unbehauen (1993) and Schalkoff (1997).

Recurrent neural networks can be trained in several ways. The IIR network shown above may be trained using the same methods as employed with the feed forward network (Section 3.4.3.1). In this case the outputs that are fed back to the inputs (x in Figure 3.16) are replaced with the target values, i.e. open-loop operation (Schalkoff, 1997). The network can be also be trained in *recurrent mode*, i.e. feeding the outputs back to the inputs.

Although a recurrent neural network has the ability to model processes with memory this increased capacity (above an MLP) leads to several disadvantages:

1. Network behaviour can be difficult to predict due to the feedback in the structure (Choueiki *et. al.*, 1997). Specifically the network output is liable to oscillate, be unstable or fail to converge to a stable state (Schalkoff, 1997),

2. Training can be more computationally expensive than for an MLP. This is because the training forecast errors of a recurrent neural network are dependent on the memory in the network which is in turn dependent on the parameters (Schalkoff, 1997), and

3. Larger data sets are required than for an MLP as a recurrent neural network can model more complex mappings (Choueiki *et. al.*, 1997).

Examples of recurrent neural networks applied to load forecasting can be found in Vermaak and Botha, (1998) and Hippert *et. al.* (2001) among others.

3.5 Techniques for Integrating Weather Forecast Errors into Load Forecasts

Miyake (1995) breaks the area of short term load forecasting into the consideration of three problems:

1. The load is affected by factors which cannot be included in any model, for example the effect of a football match,

2. The relationship between the load and causal factors, that can be included, needs to be determined *via* a load forecasting model, and

3. Some factors are forecasted values, which have their own associated forecast errors, for example temperature forecasts.

The first problem is usually considered as a random effect (Miyake, 1995). The second problem relates to the forecasting model as discussed in Sections 3.1 to 3.4. The third problem refers to the use of weather variables in load forecasting models. The load on the day to be forecast is among other things affected by the weather on that day. The only source for this weather is *via* a weather forecast, which has associated weather forecast errors. Identifying and minimising the effect of weather forecast error is the focus of this section.

However, Miyake (1995) states that the third problem has not been addressed in the literature. Although this is not strictly true, many authors have to a large extent ignored weather forecast error. For example, Lu *et. al.* (1989), Haida and Muto (1994), Gupta (1985), Park *et. al.* (1991b) and Tamimi and Egbert (2000) remark that weather forecasts are required to produce load forecasts in their models, however the effect of the weather forecast error is not considered. Alternatively, Kodogannis and Anagnostakis (1999), Park *et. al.* (1991c), Mohammed *et. al.* (1995) and Rahman and Bhatnagar (1988) remark that weather forecast error

degrades the performance of a load forecasting model significantly but do not attempt to limit this effect or quantify it.

Although the effect of weather forecast error on a load forecast is highly dependent on the electrical system, the accuracy of the weather forecast and the load forecasting model, several sources have indicated that the effect is *significant.* An IEEE committee report (1985) indicates that between 60-70% of online load forecasting errors may be attributed to weather forecast error. Douglas *et. al.* (1998b) calculate that 30-50% of load forecast errors for their model may be attributed to weather forecast error during the summer months but the effect is negligible in the winter.

Typically a load forecasting model is trained using actual weather readings (for example Rahman and Bhatnagar, 1988, Lu *et. al.*, 1989, Haida and Muto, 1994, Gupta, 1985, Chen *et. al.*, 1992 to mention but a few) rather than with historical weather forecasts (exceptions are Feng *et. al.*, 1998 and Papalexopolous and Hesterburg, 1990). There are several reasons for this:

1. Using weather forecasts in place of actual weather readings effectively introduces noise (i.e. weather forecast error) into the training set making parameter estimation more difficult. Additionally, it has been shown that there is no improvement in load forecasting accuracy by using weather forecasts during training (Papalexopolous and Hesterburg, 1990),

2. Historical weather forecasts are often not available for the whole training set (for example, Lu *et. al.* 1989, Gupta, 1985, Tamimi and Egbert, 2000), and

3. The accuracy of weather forecasts is improving over time due to the introduction of better meteorological modelling techniques (Fuller and Harris, 1999). Therefore the weather forecast errors present in the earlier

part of the training set do not represent the level of error expected in current weather forecasts.

However, weather forecast errors not present in the training set can have a disproportionate influence on the load in models as illustrated by Yoo and Pimmel, (1999) and Dash *et. al.*, (1997). Changing the load model parameters to account for this can be impossible in many conventional models once training is completed. As the effect of weather forecast error cannot be reduced during model training the following question arises as to how to quantify this effect on load forecasting accuracy (Section 3.5.1) and strategies for minimising it (Section 3.5.2).

3.5.1 Techniques for Quantifying the Effect of Weather Forecast Error

The increased load forecast error variance due to weather forecast errors can be calculated by two approaches, either:

Approach 1. Observing the load forecasts *with* and *without* weather forecast error and calculating the variance of the resultant errors (for example
Park *et. al.*, 1993a and Chen *et. al.*, 1992), or

Approach 2. Explicitly calculating the increase in load forecast error statistics from weather forecast error statistics (Douglas *et. al.*, 1998b, Ranaweera *et. al.*, 1995).

In the first case both Park *et. al.*, (1993a) and Chen *et. al.*, (1992) are restricted by a lack of historical weather forecasts. In order to overcome this, Gaussian noise is added to the actual weather readings to produce *pseudo*-weather forecasts. Park *et. al.*, (1993a) uses Gaussian noise with zero mean and a standard deviation of 5% of the actual temperature. The MAPE for the test set, using actual weather, was 1.41%. When the pseudo-weather forecasts where used this rose to 1.70%, or a rise in the MAPE of 20%. However these results are based on two assumptions:

1. The weather forecast error has a Gaussian distribution, and

2. The standard deviation of the Gaussian distribution is 5% of the temperature.

Chen *et. al.*, (1992) does not detail the Gaussian noise and so the results are difficult to evaluate.

Douglas *et. al.* (1998b) approach the problem by relating the variance of the load forecast error to the variance of the temperature forecast error (temperature is the only variable examined in this study). The load forecasting model used is a BSM (Section 3.3.2.3) which is trained using a Bayesian approach (Section 3.3.3). The effect of the temperature forecast may be calculated (Douglas *et. al.*, 1998b) with the aid of the following relationship:

$$E\left[y^2\right] = E\left[\left(y - E\left[y \mid x = X\right]\right)^2\right] + E\left[E^2\left[y \mid x\right]\right] \tag{3.95}$$

Where x and y are random variables, X is a realisation of x and $E[\bullet]$ denotes the expectation operator. In terms of the current discussion Equation (3.95) may be expressed (Douglas *et. al.*, 1998b) as:

$$E\left[\varepsilon(k)^2\right] = E\left[\left(\varepsilon(k) - E\left[\varepsilon(k) \mid \hat{t}_{hr}(k) = t_{hr}(k)\right]\right)^2\right] + E\left[E^2\left[\varepsilon(k) \mid \hat{t}_{hr}(k)\right]\right] \tag{3.96}$$

Where $\varepsilon(k)$ is the load forecasting model error, $\hat{t}_{hr}(k)$ is the temperature forecast and $t_{hr}(k)$ is the actual temperature at hour k. The first term on the right hand side of Equation (3.96) represents the error in the load forecast due to modelling error and the random component of the load, given that the actual temperature is used. The second term, $E\left[E^2\left[\varepsilon(k) \mid t_{hr}(k)\right]\right]$, represents the increase in load forecasting error due to inaccuracies in the temperature forecast. The disadvantage of this technique is that in order to evaluate Equation (3.96) the *variance* of the temperature forecasts must be *known*.

Ranaweera *et. al.* (1995) again relates the variance of the load forecasting error to the variance of the weather forecast errors. This method differs from that of Douglas *et. al.* (1998b) however, in that a Taylor series expansion is used. Consider a function of several variables:

$$x = f(u_1, u_2, ..., u_n) \tag{3.97}$$

Where $f(\bullet)$ is the function, x is the output and $u_{1,...,n}$ are the inputs and n is the number of inputs. Given the variance in one or more of the inputs, the variance in the output may be calculated via a Taylor series expansion of f (Ranaweera *et. al.*, 1995) as[%]:

$$\sigma_x^2 = \sum_{i=1}^{n} \left(\frac{\partial f}{\partial u_i} \right) \sigma_{u_i}^2 \tag{3.98}$$

Where σ_x^2 is the variance of the output and $\sigma_{u_i}^2$ is the variance of input i. In terms of load forecast error variance, σ_x^2 represents the load forecast error variance and $\sigma_{u_i}^2$ the variances of the weather inputs.

Ranaweera *et. al.* (1995) apply this method to two types of model; an MLP (Section 3.4.3.1) and an RBF neural network (Section 3.4.3.3). The forecasted weather variables used are the daily maximum and daily minimum temperatures. As the standard deviation (and thus variance) of the weather forecast errors are unknown a series of values ranging from 5% to 20% of the actual temperature values are used. With the standard deviation of both the weather variables set at 5% the load forecast error standard deviation is increased by .65%. Given that the MAPE of the load forecasts produced using the MLP and RBF networks is 2.05% and 2.08% respectively this represents approximately a 17% increase in the load forecast error MAPE which is significant[*].

[%] This equation is only valid around the operating point at which the partial derivatives are taken.
[*] This figure assumes that a 0.65% increase in the standard deviation of the load forecast error results in a 0.65% increase in the MAPE of the load forecast error which is not strictly true.

This technique has several disadvantages however:

1. Equation (3.98) is valid only if the errors in the inputs are independent. Thus in the study presented by Ranaweera *et. al.* (1995) it is assumed that the error in the forecast of maximum daily temperature is independent of the error in the minimum forecast,

2. The variances of the inputs must be known or approximated,

3. The error in the load forecast due to factors other than weather forecast error are not considered, and

4. Both MLP and RBF neural networks have no memory (see Sections 3.4.3.3 and 3.4.3.3). There is no indication how this technique may be extended to a load forecasting technique with memory.

Having explored the techniques for quantifying the increase in load forecast error due to weather forecast error the next section investigates techniques for limiting that error.

3.5.2 Techniques for Minimising the Effect of Weather Forecast Error

Four techniques have been identified in the literature for minimising the effect of weather forecast errors:

1. The first technique involves fuzzifying the inputs (Section 3.4.2.1). According to Bitzer and Rößer (1998) fuzzification of forecasted temperature inputs reduces the effect of the temperature forecast error on the load forecast. However, the reduction is not quantified,

2. The second technique is implemented by Rahman and Hazim (1993). In this case confidence limits for the weather forecast are made available by the weather service used. The load forecasting model uses a pattern matching technique similar to the one used by Otto and Schunk (1999) (discussed in Section 3.4.2.1). In this case the load forecast is produced using historical data of days with a similar temperature and load profile history. As explained by Rahman and Hazim (1993) a load forecasting model is vulnerable to weather forecast error. To overcome this Rahman and Hazim (1993) select days with not only temperatures similar to the expected temperature of the forecast day but also days in which *the temperature is within the confidence limits of the weather forecast*. Unfortunately the improvement in forecasting accuracy using this technique is not investigated,

3. Taylor and Buizza (2003) propose a technique which employs weather forecasts which are given as probability density functions as opposed to a normal weather forecast which is just a single figure. These weather forecasts are then used to produce ensemble predictions of the load. Each prediction has an associated probability and it is left to the system operator to choose the best course of action,

4. The fourth technique is based on the principle that the optimum load forecasting model is, among other things, dependent on the weather forecast error inherent in some inputs (Miyake *et. al.*, 1995). That is, the advantages of including a weather variable that is correlated to the load may be eroded when weather forecast error is introduced. In order to determine the best model Miyake *et. al.*, (1995) constructs a set of candidate models for each forecast day and the best model is selected. The load forecasting models are similar to the Hammerstein models described in Section 3.4.1.2, specifically the form of the models (Miyake *et. al.*, 1995) is:

$$y(k+i) = a_{0,i} + \sum_{l=1}^{N}\sum_{j=1}^{M} a_{l,j,i}\{x_l(k+i)\}^j \qquad (3.99)$$

Where $y(k)$ is the load at time k, i is the forecast horizon, $a_{l,j,i}$ are the model parameters, N and M are the orders of the model and $x_l(k)$ is input l at time k. To construct the candidate models the orders and inputs for the models are varied.

In order to select the optimum model, an adjustment to a model selection criterion known as the Final Prediction Error (FPE) (Brockwell and Davis, 1987) is used. The FPE is defined (Brockwell and Davis, 1987) as:

$$J_{fpe} = \frac{n+q}{n-q}\hat{\sigma}_{\varepsilon}^{2} \tag{3.100}$$

Where J_{fpe} is the final prediction error, n is the number of data points used, q is the number of parameters in the model and $\hat{\sigma}_{\varepsilon}^{2}$ is the variance of the errors in the training set. The FPE reflects the fact that as the complexity of the model (represented by q in Equation 3.100) increases, the *in-sample* (forecasts made on the training set) forecasts improve, while the *out of sample* (validation set) forecasts may deteriorate due to over fitting. Thus the inclusion of q in Equation (3.100) effectively penalises increased complexity in the model. Miyake *et. al.*, (1995) adjusts the FPE to include for errors in the weather forecast variables as:

$$J_{fpeev} = \frac{n+q-tr\left\{\left(\boldsymbol{U}^{T}\boldsymbol{W}\boldsymbol{U}\right)^{-1}\boldsymbol{Q}_{u}\right\}}{n-q}\sigma_{\varepsilon}^{2} + \frac{1}{n}\hat{\boldsymbol{a}}^{T}\boldsymbol{Q}_{u}\hat{\boldsymbol{a}} \tag{3.101}$$

Where J_{fpeev} is the adjusted final prediction error or Final Prediction Error in Explanatory Variables (FPEEV), $\boldsymbol{U}$ is a vector of inputs, $\boldsymbol{W}$ is a set of weights used to calculate the parameters of the model, $\hat{\boldsymbol{a}}$, and $\boldsymbol{Q}_u$ is the error covariance matrix of the inputs (i.e. the covariance matrix of the weather forecast errors). J_{fpeev} is then calculated for each candidate model and the model with the lowest J_{fpeev} is then chosen. Further details on the construction of $\boldsymbol{U}$, $\boldsymbol{W}$, $\hat{\boldsymbol{a}}$, and $\boldsymbol{Q}_u$ may be found in Miyake *et. al.*, (1995). However, it is sufficient to note here that Equation (3.101) is *specific* to the models described by Equation (3.99).

The advantages of this technique are:

1. Unlike the technique used by Ranaweera *et. al.* (1995) the covariance matrix of the weather forecast errors is used, thus recognising that the errors in one forecast weather variable may be correlated to the errors in another, and

2. The error in the load forecast due to factors other than weather forecast error is considered *via* the FPE.

However the technique is restricted by two factors:

1. The FPEEV in Equation (3.101) is model specific and calculating an FPEEV for a different model type may be difficult, and

2. Calculating the covariance matrix of weather forecast errors may be difficult.

3.6 Conclusion

The literature reviewed in this chapter demonstrates that the area of STLF has been tackled using many different approaches. The reason for the multitude of approaches lies with the fact that each electricity system has different characteristics. Additionally, it is impossible to attempt all of the approaches in order to ascertain the optimum one. Also in many cases the differences in performance may be insignificant (Hippert *et. al.* 2001). However, there remain several topics of discussion in this area which require investigation:

1. A parallel approach versus a sequential approach. The advantages and disadvantages of both approaches have been examined in Section 3.2, however there are no guidelines for determining which approach may be superior for a given electricity system,

2. Choosing the optimum forecasting technique. The techniques explained in Section 3.2 and 3.3 differ in their ability to model complex behaviour. As the complexity of the technique increases however, determining the optimum parameters of the model becomes more difficult. Additionally, different techniques may be more appropriate at different times, or a combination of techniques may be superior. There have been several studies to investigate if certain characteristics of a time series can be used to determine what the optimum model for that time series may be (Arinze *et. al.*, 1997). However the results are inconclusive at this stage and the choice of technique remains a subjective choice to be made by the forecaster, and

3. Integrating the effect of weather forecast error into a load-forecasting model. In Section 3.5 it was pointed out that a significant percentage of load forecasting error may be due to weather forecast error. This topic has however been largely ignored in the literature.

Chapter 4

Day-Type Identification

4.1 Introduction

This chapter describes the techniques used to identify day-types in Irish load data (as discussed in Sections 2.3.2 and 3.2.1). The primary purpose of day-type identification is to disaggregate the data prior to modelling (Section 4.2). Once the data has been disaggregated into the respective day-types, it is then important to know if there is sufficient data within each day-type to allow consistent model building (Section 4.2.1). In addition, the relationship between the load and the dominant exogenous variable, temperature, within each day-type is important for model building and is examined in Section 4.2.2. Finally, the transitions between the day-types (see Section 3.2.1.2) are identified in Section 4.3.

4.2 Day-Type Identification

In Section 3.2.1.1 several techniques for day-type identification were discussed. Of the three candidate techniques proposed in Section 3.2.1.1, the Kohonen map is best suited to this task, as operator interviews can be subjective and cluster algorithms require *a-priori* knowledge of the day-types (for example, see Section 4.3).

The parameters used in the Kohonen map are:

- Initial neighbourhood size: 4,
- Adaptation gain: .002,
- Number of iterations after which *Nc* is reduced: 10, and
- Number of output nodes: 18×18 (324 in total).

These values are similar to those used by Hsu and Yang (1991a). The value of α affects the rate at which the network adapts to each input; if the value is too large, the network over-reacts to each input, while the network will not converge if the value is too small. The value of α may be adapted from iteration to

iteration, however in simulations conducted by the author little difference was found; this was also the case reported by Hsu and Yang (1991a). As was found by Hsu and Yang (1991a), a wide range of values from .001 to .007 was found to give satisfactory results.

For the present study, the trials used all of the last 2 full years of data, 1998 and 1999. Figures 4.1 to 4.6 below show which nodes are triggered and the number of inputs that mapped to that node.

Although the weights are initialised to aid in spreading the activation across the complete output grid, after a single iteration the inputs map to several nodes in close proximity to each other (Figure 4.1). This is because the neighbourhood is large and the adaptation is acting *globally* (the weights of many of the output nodes are adapted by each input).

After fifteen iterations the outputs have segregated into four groups, the groups with the lower number of mappings representing mostly Sundays and Saturdays and the other groups representing mainly weekdays (Figure 4.2). At this point the neighbourhood size has reduced to three.

After thirty iterations the data has been spread across the grid of output nodes with different day-types occupying different parts of this grid (Figure 4.3). The neighbourhood size is now two and the weights are being adapted *locally* (only the weights associated with nodes a distance of two away from the mapped node are being adapted). This causes similar day-types in the four groups to subdivide and trigger adjacent nodes. Graphically, this means that new parts of the grid are being triggered in Figure 4.3 relative to Figure 4.2. An example is indicated in Figure 4.2 and Figure 4.3 of the subdivision of the winter weekday group.

Iteration fifty is the last iteration as the neighbourhood size has now reduced to zero. The final iterations from thirty to fifty refine the day-types already identified (Figure 4.4).

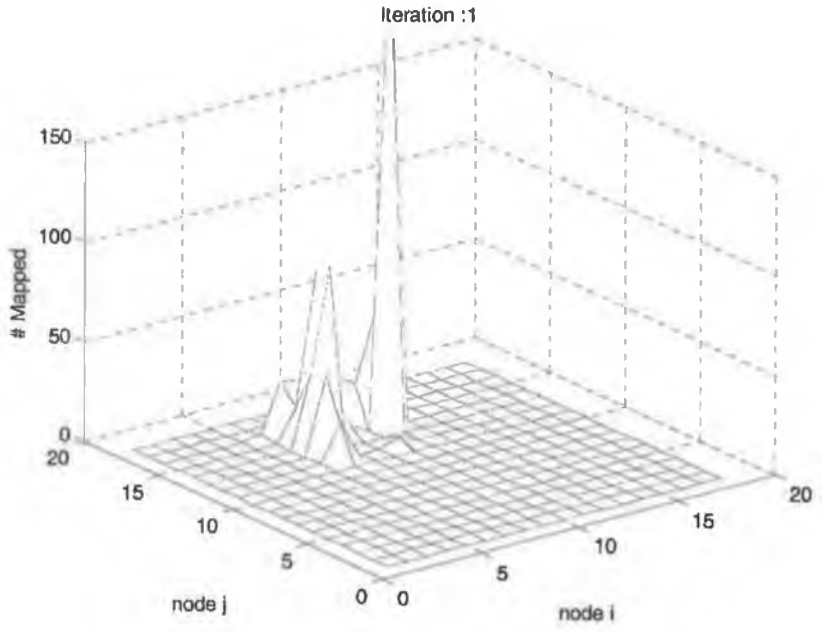

Figure 4.1 Number of inputs that map to each node (iterations:1, *Nc*: 4).

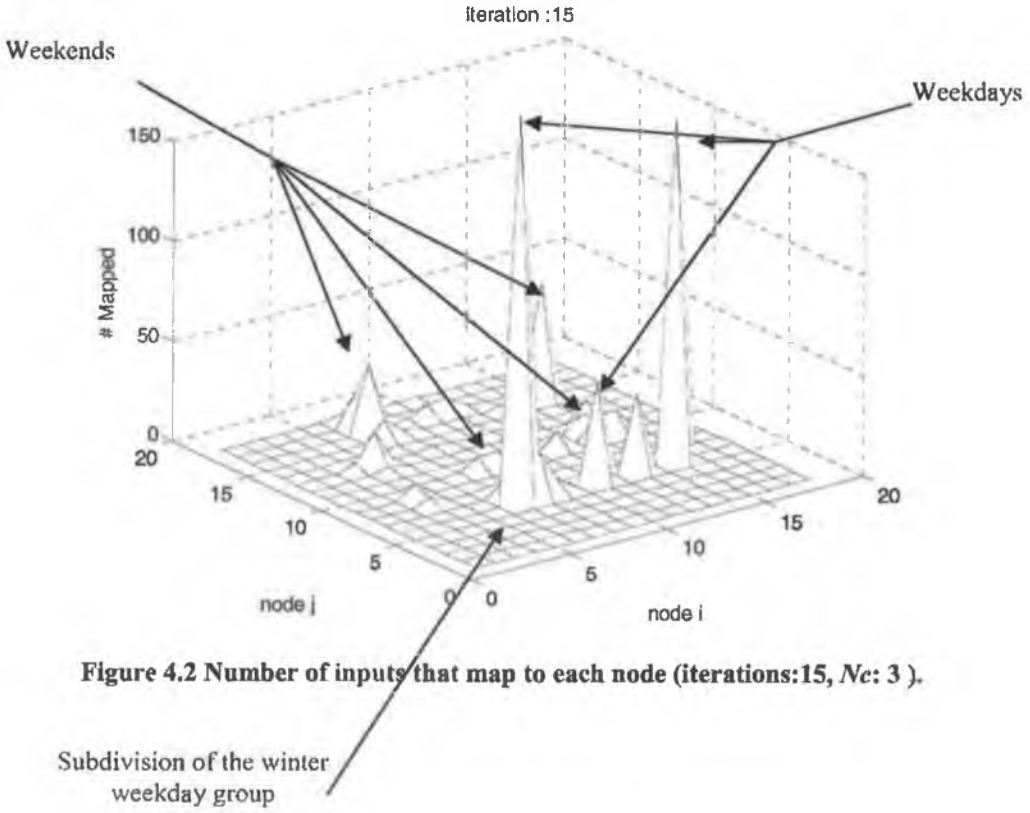

Figure 4.2 Number of inputs that map to each node (iterations:15, *Nc*: 3).

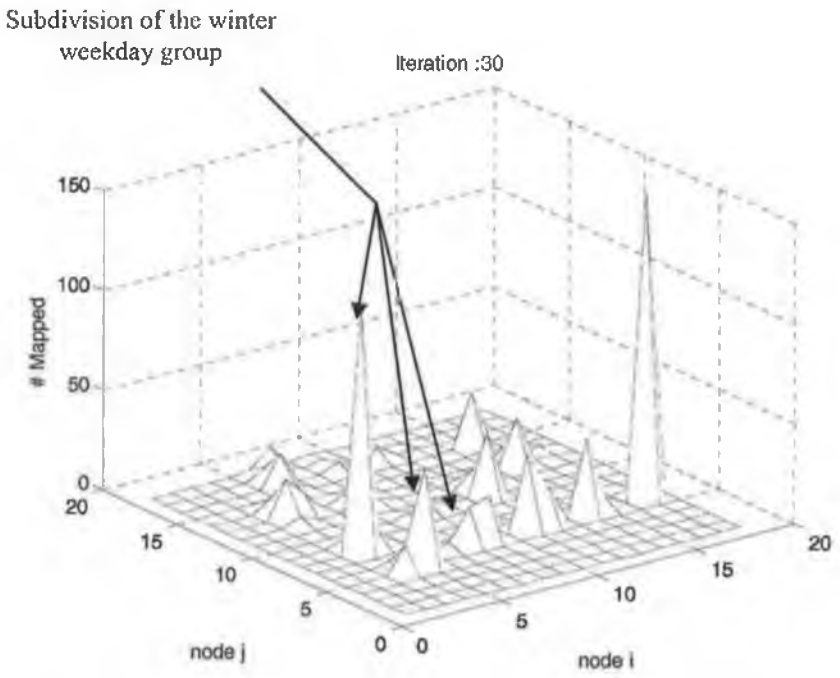

Figure 4.3 Number of inputs that map to each node (iterations:30, *Nc*: 2).

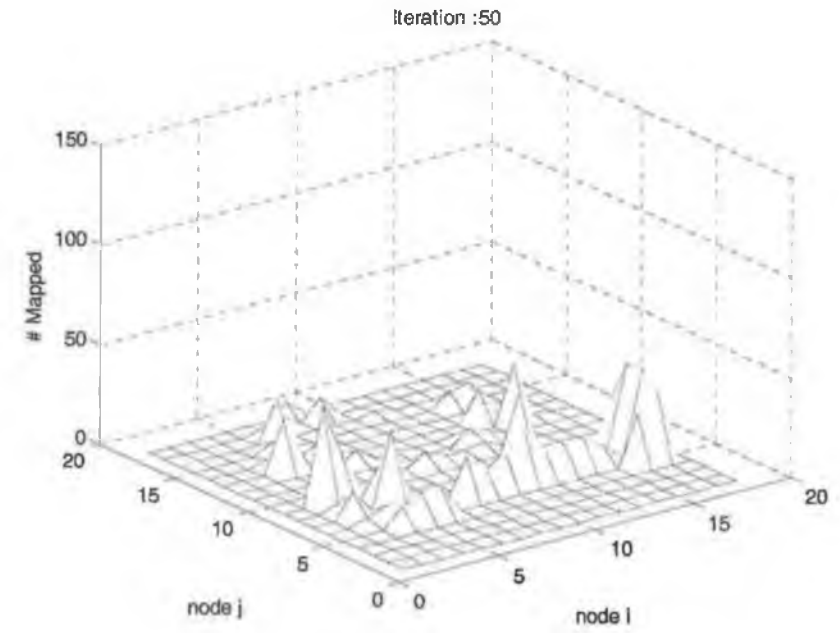

Figure 4.4 Number of inputs that map to each node (iterations:50, *Nc*: 0).

From the above analysis, it can be seen that the algorithm has operated as expected. Initially, the inputs are spread across the output grid according to large differences in the inputs. Subsequently each of the large groups is refined locally into any sub-groups that may exist.

The next step is to assign day-types to the triggered nodes in Figure 4.4. Figure 4.5 shows the nodes that are triggered from Monday to Friday. As can be seen the nodes that are triggered for these days occupy the same parts of the grid. Note the only exception, that several Monday loads trigger nodes around (i=13, j=13). This is explained later in this section.

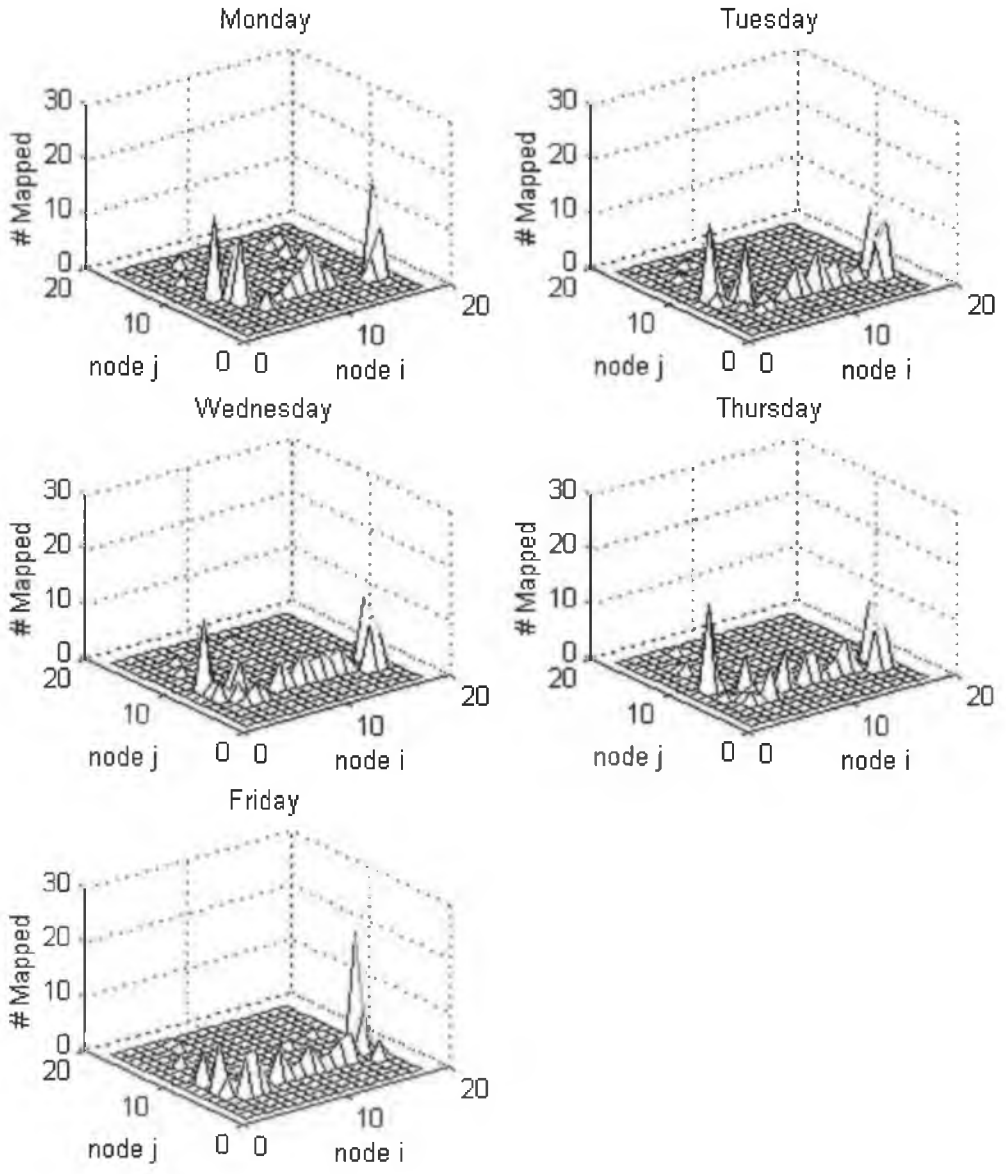

Figure 4.5 Nodes triggered by Monday-Friday loads (iterations:50, *Nc*: 0).

Sunday, Saturday and Monday loads trigger different parts of the grid showing the difference between these days (Figure 4.6). The one exception as pointed out

previously is that several Monday inputs trigger the nodes around (i=13,j=13). The inputs responsible for this are Monday bank holidays and it is interesting to note that they are mapped to the same nodes as some of the Sunday loads (Figure 4.6).

On closer inspection, it is seen that the summer months (May to September) are mapped to the right-hand side of the grid while the winter months (excluding Christmas) are mapped to the left-hand side. Christmas loads with the exception of Saturdays occupy a separate node indicating that Christmas loads are different to others (Figure 4.6). It is by co-incidence that the triggered nodes are aligned on the grid such that loads on different days of the week changes with the y-axis and the time of year with the x-axis (Figure 4.6). Other tests, performed by this author, have shown similar results with the triggered nodes aligned in other directions e.g. diagonally. Finally, with the exception of Summer bank holidays, the spikes in Figure 4.5 for working day loads can be assigned similar day-types to those for Mondays (Figure 4.6).

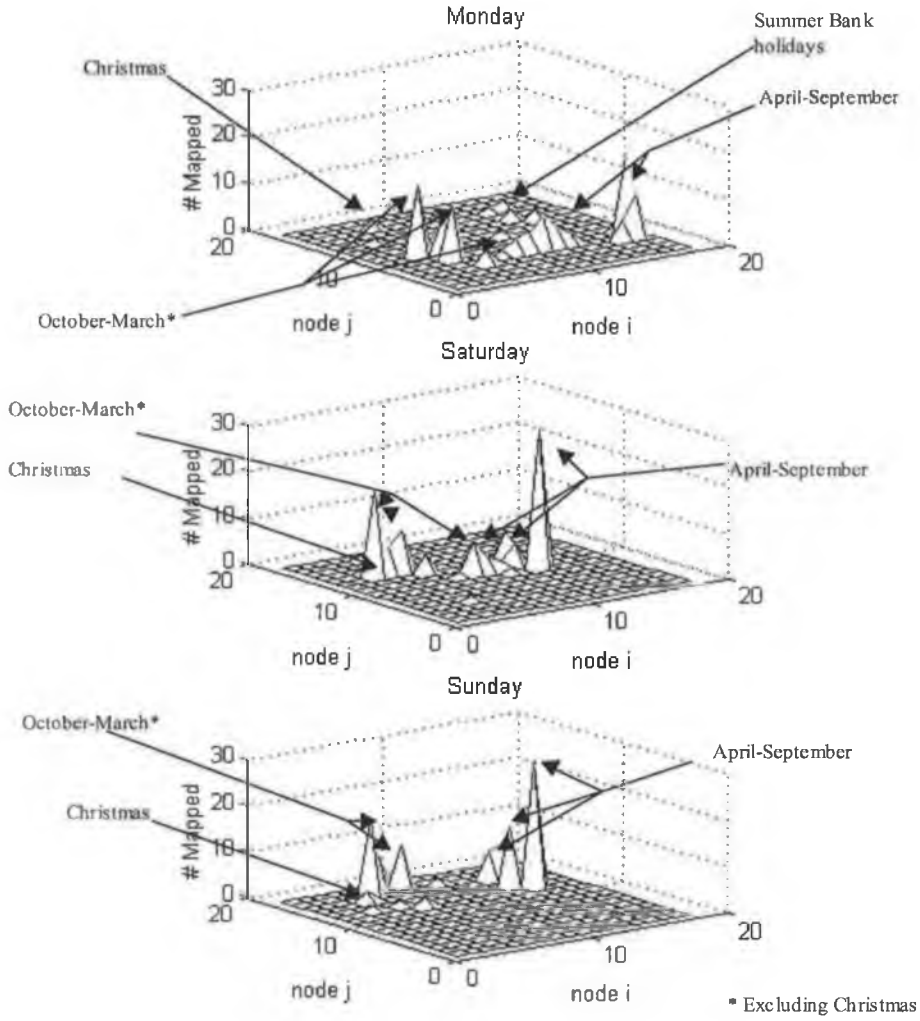

Figure 4.6 Nodes triggered by Monday, Saturday and Sunday loads. (iterations:50, *Nc*: 0).

The day-types identified are collated in Table 4.1. The October to March day-types are split into two groups (early and late winter) to reflect the fact that Christmas lies at the centre of this range.

Table 4.1 Day-types identified for Irish load.

Day-type	Number	Range
Early winter Sundays	1	October-Christmas
Summer Sundays & bank holidays	2	April-September & bank holidays
Late winter Sundays	3	January-March
Early winter working days	4	October-Christmas
Summer working days	5	April-September
Late winter workings days	6	January-March
Early winter Saturdays	7	October-Christmas
Summer Saturdays	8	April-September
Late winter Saturdays	9	January-March
Christmas days	10	Christmas

The bank holidays in the Irish calendar are:

- St. Patrick's day March 17th (or closest Monday),
- Good Friday (Lunar calendar),
- Easter Monday (Lunar calendar),
- May day, 1st Monday in May,
- 1st Monday in June,
- 1st Monday in August and
- Last Monday in October.

The transitions between these day-types are considered in Section 4.3 after the relationship of temperature to the load in each day-type has been examined.

4.2.1 Segmentation of Data for Model Building

This section lays out the blueprint for a load-forecasting *package* (a set of models that can be used to forecast all day-types) in light of the analysis in Section 4.2. As pointed out in Section 3.2.1, by modelling each day-type separately the information that each model must incorporate is reduced. This aids in the modelling task. There is however a trade-off. As pointed out by Hippert *et al.* (2001) in a study of neural networks applied to the load-forecasting problem, if the data is subdivided into too many day-types then the resulting data sets are too small to permit adequate model training.

The segmentation of the data set (by the day-types chosen in Section 4.1) is shown in Figure 4.7 with the number of days in each set shown (note: some days are not classified as they occur at the boundary between day-types).

The amount of data required to allow modelling is difficult to quantify as it is influenced by the type of model used and the amount of noise in the data, among other things. To determine if the amount of data in each subset is sufficient to allow modelling, comparisons are drawn with other studies. Hippert *et al.* (2001) gives a list of the data set sizes used in twenty-two studies by various authors. The size of the *total* data sets used in each study varies from thirty-four days to four years. The *total* size of the data set in the current study is thirteen years, which is relatively large. This allows a greater level of segmentation, if desired.

The amount of data in *each day-type*, for the current study, is given in Figure 4.7 and shows that for the day-types chosen, the amount of data is similar to that used by Sharaf and Lie (1995), Srinivasan *et al.*(1999) to mention but a few. Sharaf and Lie (1995) used three months of data for each of their day-types and Srinivasan *et al.* (1999) used two years of data broken into 3 day-types with approximately three hundred days in the working and Saturday day-type and fifty days in the Sunday day-type.

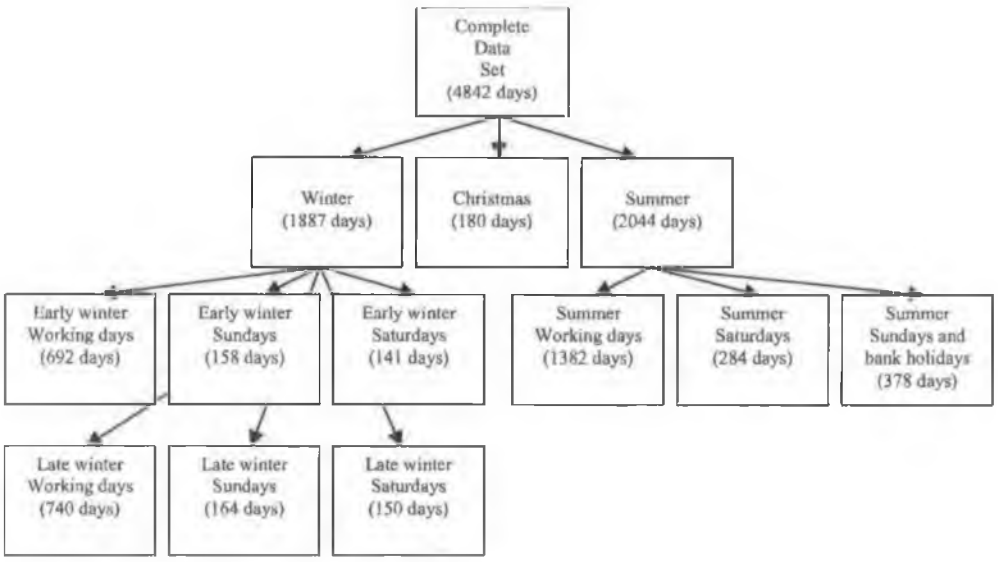

Figure 4.7 Segmentation of data set with approximate number of days in each subset.

4.2.2 Temperature-Load Relationship within Day-Types

The following analysis uses the pre-processed average load, $z_{av}(k)$, and average temperature, $t_{av}(k)$ as defined in Section 2.3.3.2. Figure 4.8 shows the load-temperature relationship for working days with data points for each month of the

year highlighted. The data points for each month group into clusters with a few exceptions due to Christmas, bank holidays and exceptional days (Figure 4.8).

In order to compare the load-temperature relationship at different times of the year regression lines are employed (Figure 4.8). As pointed out by Murray (1996), Hyde and Hodnett (1997b) and Fan and McDonald (1994), among others, the temperature-load relationship is non-linear and so it should be noted that the use of regression lines is an approximation. However, the approximation is considered sufficient to identify significant differences in the load-temperature relationship for different months of the year.

In calculating the regression lines, Christmas, bank holidays and exceptional days are excluded to avoid them influencing the line parameters. Several algorithms exist for selecting which points to include and exclude for calculating regression lines (Wisnowski *et. al.*, 2001, lists several techniques). The points to be excluded in the current analysis are easily identified (as can be seen in Figure 4.8) and so the selection of the exclusion technique is not critical. The BACON (Blocked Adaptive Computationally efficient Outlier Nominators) method proposed by Billor *et. al* (2000) was chosen for its fast convergence. This technique may be broken down into several steps:

1. The technique first requires that a subset of the data, safely assumed to be free of outliers, be chosen. This subset is called the *basic set,* and is formed from chosen loads, $\boldsymbol{Z}_b$, and corresponding temperatures, $\boldsymbol{T}_b$. In this analysis, the basic set is determined by first calculating a regression line using all the data:

$$z_{av}(k) = \beta_1 t_{av}(k) + \beta_2 + \varepsilon(k) \tag{4.1}$$

where β_1, β_2 are parameters of the regression line calculated *via* least squares (i.e. minimum $\sum_k \varepsilon^2(k)$) and $\varepsilon(k)$ is an error term. The basic set then contains the top ten percent of points with the lowest $\varepsilon(k)$,

2. Next, each point is tested to see if it is significantly different from the line. The test statistic, called the *discrepancy*, is calculated (Billor *et. al,* 2000) as:

$$\chi(k) = \begin{cases} \dfrac{\varepsilon(k)}{\hat{\sigma}\sqrt{1 - t_{av}(k)(\boldsymbol{T}_b^T \boldsymbol{T}_b)^{-1} t_{av}(k)}} & \text{if day } k \in \boldsymbol{T}_b \\ \dfrac{\varepsilon(k)}{\hat{\sigma}\sqrt{1 + t_{av}(k)(\boldsymbol{T}_b^T \boldsymbol{T}_b)^{-1} t_{av}(k)}} & \text{if day } k \notin \boldsymbol{T}_b \end{cases} \quad (4.2)$$

where $\chi(k)$ is the discrepancy for day k, and $\hat{\sigma}$ is the standard deviation of the errors in the basic set,

3. The basic set is then increased to include points for which (Billor *et. al,* 2000):

$$\chi(k) < t_{(\alpha_b / 2(r+1), r-1)} \quad (4.3)$$

where r is the number of points in the basic set, α_b is the significance level and $t_{(\alpha_b, r-1)}$ is the 1-α_b percentile of the t-distribution with r-1 degrees of freedom,

4. Following the expansion of the basic set, the regression line parameters are recalculated (Equation 4.1) using only the data points in the basic set. Steps 2 and 3 are then repeated, and

5. The process continues until the basic set no longer increases in size. The points excluded from the basic set are then designated as outliers. Finally, the points in the basic set are used to calculate the parameters of the regression line.

The value of α_b has to be adjusted so that only true outliers are identified. The outliers of interest in this study are Christmas, bank holidays and exceptional days. There may however, be little correlation between the load and temperature in the remaining points. An example of this can be seen for August working days (Figure 4.8). Thus the discrepancy values calculated by Equation (4.2) are relatively high. Choosing a small value for α would thus result in many of the non-exceptional points being excluded. A significance level of 0.4 was found to

be sufficient in this study to identify all the outliers while retaining most of the other points.

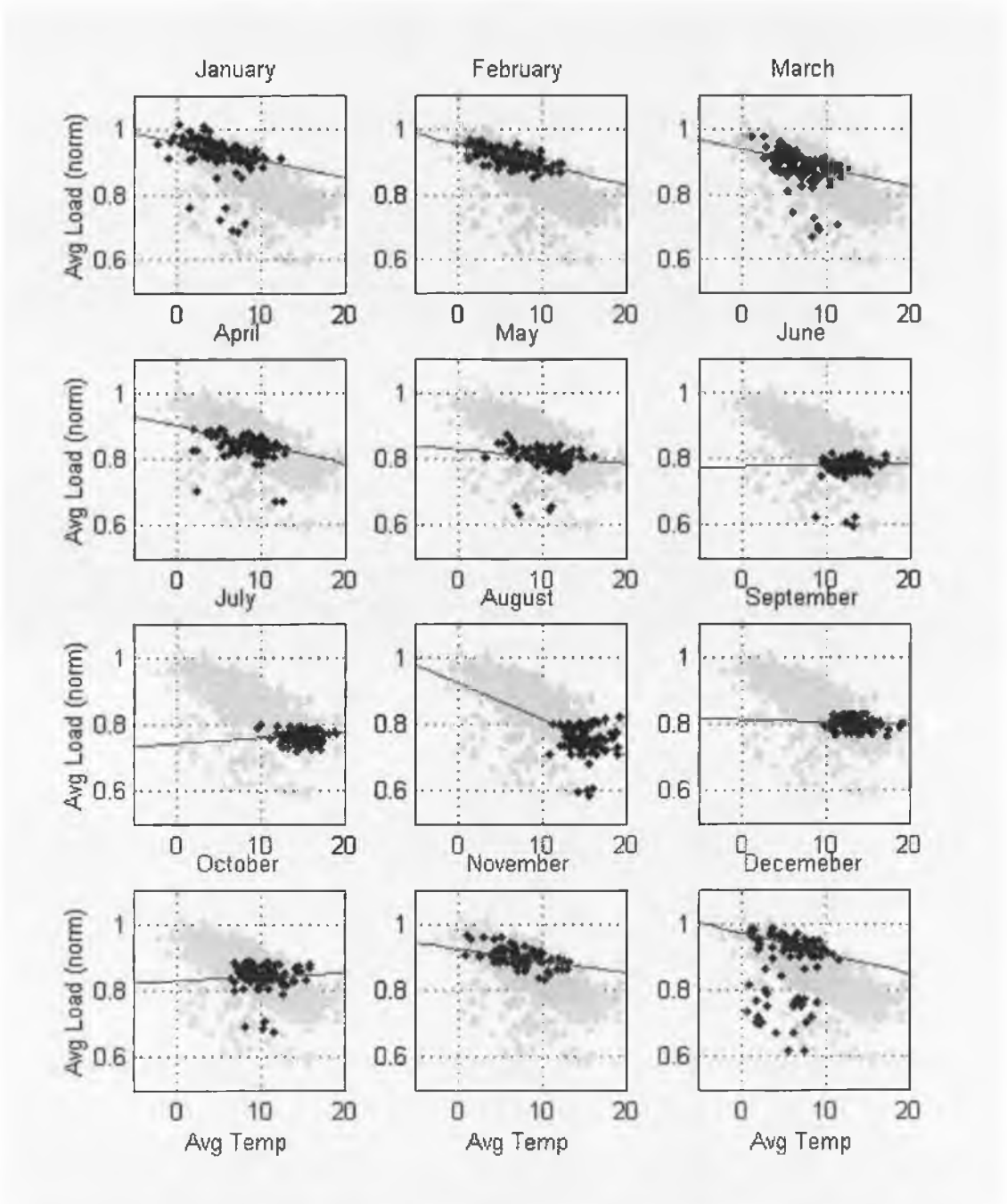

Figure 4.8 Scatter plot with regression lines for temperature-load relationship by month (working days, selected points in black, complete data set in grey).

The load-temperature relationship is different for each month of the year (Table 2.5). For January to April, the slopes of the regression lines are negative and similar. From May to June the slopes approach zero. The low correlation co-efficients for July to October indicate very little correlation between temperature and load in those months. The regression lines and correlation co-efficents for

November and December are similar to those in January. With respect to the day-types identified in Section 2.3.2, this indicates that there is no significant difference in the temperature-load relationship within the periods spanned by the late winter working day day-type (October to March). The summer working day day-type (spanning April to September), however, has a significantly different temperature-load *sensitivity* (defined as the slope of the regression line) in April and May than in the period June-September. The early winter working day day-type (spanning October to December, excluding Christmas) also has a different sensitivity in October than in November-December.

The load-temperature sensitivities for Saturday and Sunday day-types show a similar result to that of working days (Table 4.2).

Table 4.2 Temperature-load regression line co-efficients and correlations.

		Jan.	Feb.	Mar.	Apr.	May	June	July	Aug.	Sept.	Oct.	Nov.	Dec.
Working Days	Slope (β_1)*	-5.42 E-03	-6.20 E-03	-5.72 E-03	-5.69 E-03	-2.19 E-03	0.559 E-03	1.81 E-03	-10.9 E-03	-0.823 E-03	1.15 E-03	-3.60 E-03	-6.13 E-03
	Intercept (β_2)*	9.63 E-01	9.55 E-01	9.39 E-01	9.00 E-01	8.29 E-01	7.75 E-01	7.43 E-01	9.23 E-01	8.13 E-01	8.31 E-01	9.26 E-01	9.72 E-01
	Correlation between temperature and load	-0.53	-0.49	-0.65	-0.48	-0.14	0.25	-0.20	0.02	-0.11	-0.07	-0.63	-0.49
Sundays	Slope (β_1)*	-5.94 E-03	-4.39 E-03	-6.77 E-03	-5.02 E-03	-2.40 E-03	5.94 E-03	1.87 E-03	-1.33 E-03	1.18 E-03	-2.29 E-03	-4.37 E-03	-3.78 E-03
	Intercept (β_2)*	8.00 E-01	7.67 E-01	7.70 E-01	7.30 E-01	6.87 E-01	5.56 E-01	5.86 E-01	6.47 E-01	6.34 E-01	7.02 E-01	7.80 E-01	8.05 E-01
	Correlation between temperature and load	-0.28	-0.65	-0.37	-0.04	-0.56	-0.38	0.26	-0.18	-0.35	-0.15	-0.42	-0.39
Saturdays	Slope (β_1)*	-2.74 E-03	-4.40 E-03	-4.86 E-03	-3.04 E-03	-2.71 E-03	-1.55 E-03	1.95 E-03	-1.69 E-03	5.34 E-04	-1.36 E-03	-2.06 E-03	-2.83 E-03
	Intercept (β_2)*	8.47 E-01	8.48 E-01	8.32 E-01	7.59 E-01	7.58 E-01	7.25 E-01	6.56 E-01	7.18 E-01	7.07 E-01	7.68 E-01	8.31 E-01	8.66 E-01
	Correlation between temperature and load	-0.32	-0.38	-0.45	-0.61	-0.83	-0.19	-0.64	0.20	0.46	-0.18	-0.73	-0.68

* For normalised data.

4.3 Transitions between Day-Types

Two candidate techniques for identifying day-type transitions were examined in Section 3.2.1.2; cluster algorithms and Kohonen neural networks. The FCM cluster algorithm is chosen as Kohonen neural networks are not suited to this task (Section 3.2.1.2). In the current study, the inputs, $\boldsymbol{U}_k$, are composed of the twenty-four hours of data, normalised as described in Section 3.2.1.2 (Equation 3.6). For the current analysis, only data from the years 1999 and 2000 are used. This is to avoid the clustering algorithm identifying the differences between years as opposed to day-types. The desired level of fuzziness depends on the accuracy of load-forecasting models. However, at this stage, a value of 2.0 for κ was found to give a good indication of the transitions between the day-types.

As pointed out in Section 3.2.1.2, clustering algorithms are best used when the existence of clusters is known *a-priori* and it is the transitions that are desired. As an example of this, working day data was analysed to determine if FCM could allocate the clusters and transitions between the five day-types for working days. As can be seen, the algorithm identified five clusters which do not correspond to the day-types identified in Section 4.2 and the Christmas period is not assigned to a cluster at all (Figure 4.9). For example, two clusters are identified for summer while a single cluster was desired.

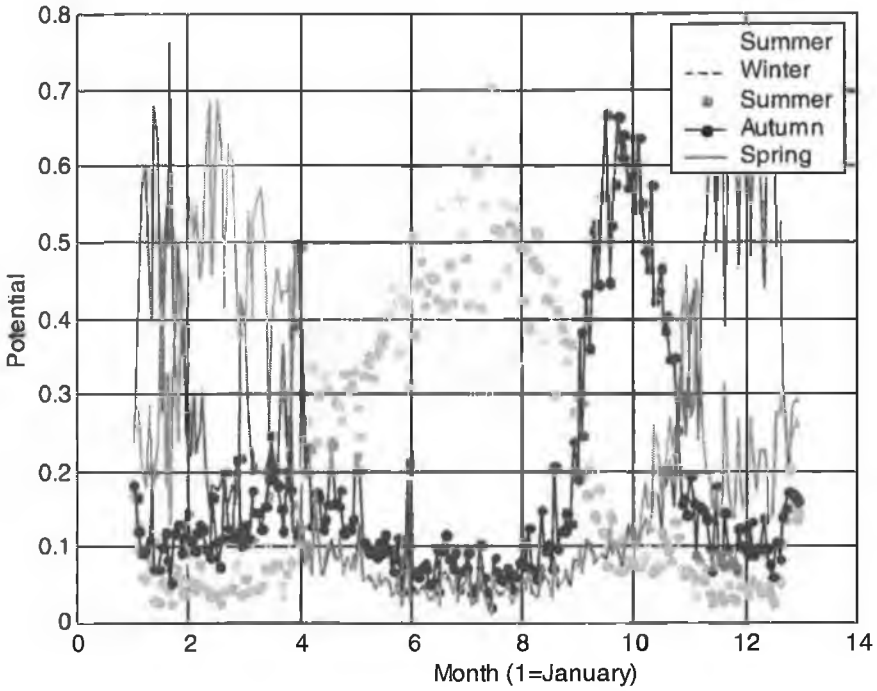

Figure 4.9 Segmentation of data set using FCM with five clusters.

However, as the data has already been segmented in Section 4.2 (Table 4.1), the existence of the clusters is already known. The data was segmented into pairs of known clusters, as only two day-types overlap at a time. FCM was then used to determine the transition between the two. These segmented pairs are:

- Early winter working day and Christmas day-types,
- Christmas and late winter working day day-types,
- Late winter and summer working day day-types and
- Summer and early Winter working day day-types.

The transition between early winter working day day-type and Christmas day-types occurs over the period of several days, however the two periods are distinct (Figure 4.10).

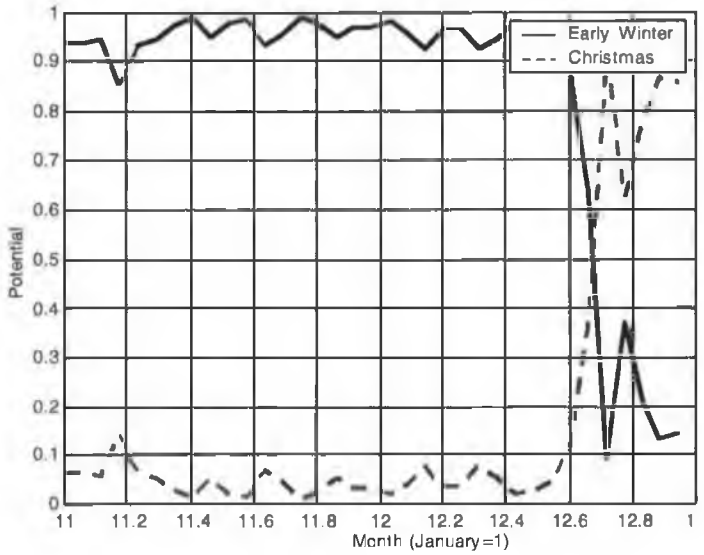

Figure 4.10 Early winter (working day) day-type to Christmas day-type transition.

Recall that a winter day-type is identified in Section 4.2. However, as the winter day-type is disjoint due to the Christmas period, the early and late winter day-types were created. Figure 4.11 shows the transitions between *winter day-type* data and Christmas. It is interesting to note, as expected, that the transition into Christmas corresponds to that in Figure 4.10. The transition between Christmas and late winter working day day-types also occurs over several days (Figure 4.11).

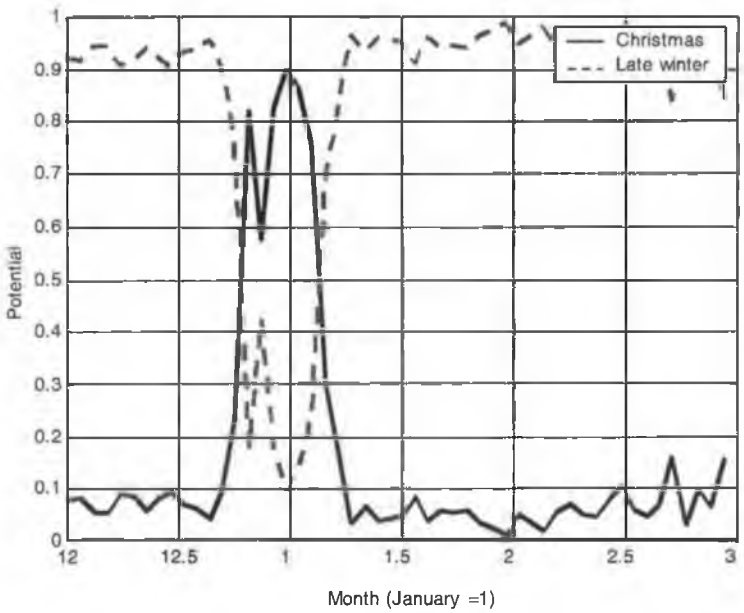

Figure 4.11 Christmas day-type to late winter (working day) day-type transition.

The transition from late winter to summer occurs over the hourly changes for daylight savings. Figure 4.12 shows the load for Friday 25th March 1999 (before the change) and Tuesday the 30th March (after the change). The load profiles for these days are significantly different, due to the fact that there is a lower lighting requirement on Tuesday 30th March.

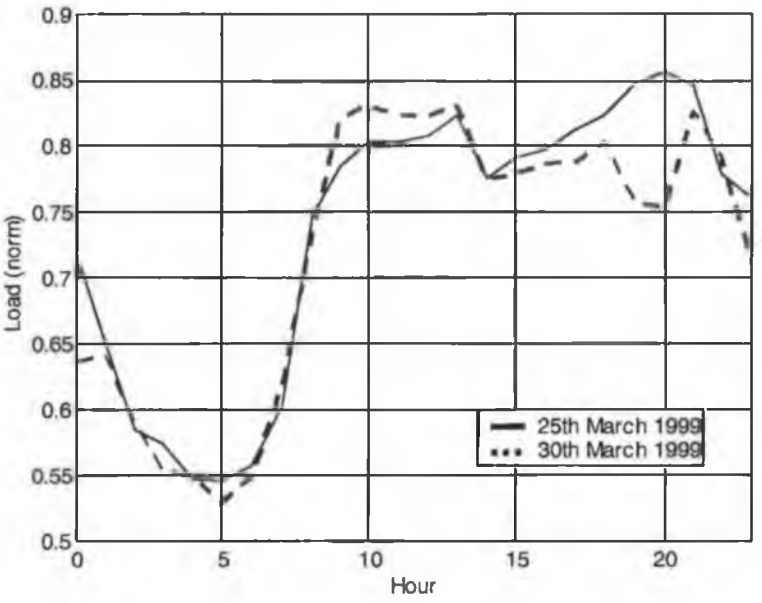

Figure 4.12 Differing load profiles due to day-light savings changes.

As a consequence of the changes for day-light savings, the transition between late winter working day day-types and summer working day day-types is hard (Figure 4.13).

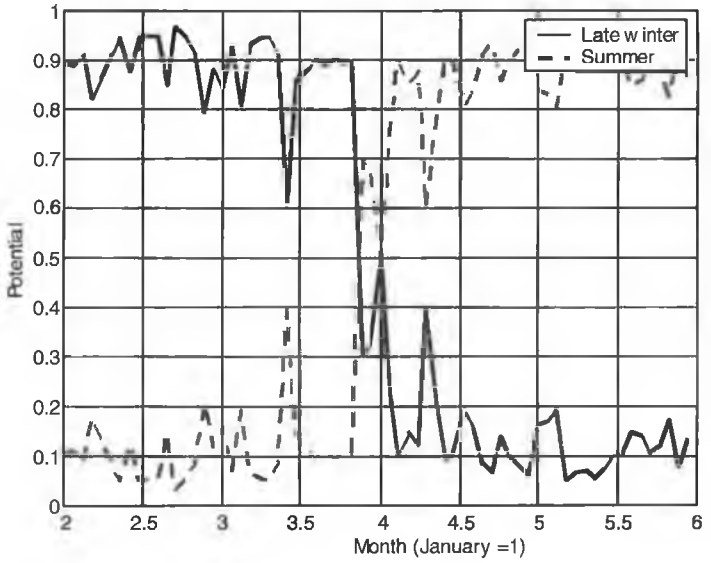

Figure 4.13 Late winter to summer (working day) day-type transition.

Finally the transition between summer and early winter working day day-types is shown in Figure 4.14. This transition is fuzzy demonstrating that there is a significant overlap between the two periods. The day-light savings change does not have the same effect here as lighting requirements are not influential in summer.

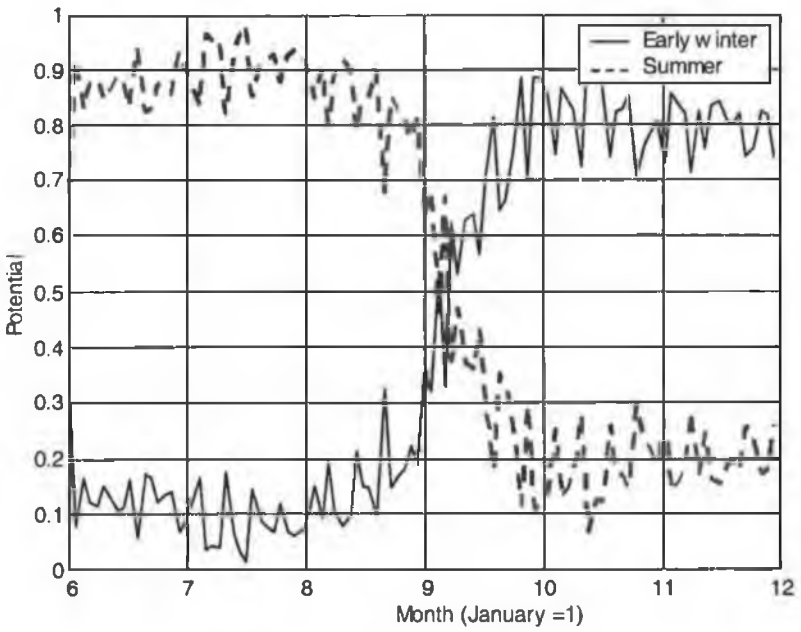

Figure 4.14 Summer to early winter (working day) day-type transition.

4.4 Conclusion

This chapter investigated the disaggregation of Irish electrical load data by day-type. The blueprint for a load-forecasting package is laid out in consideration of these characteristics without biasing the choice of load forecasting models used.

It was found in Section 4.2 that the day-types correspond well with the expected segmentation by working days, weekend days and by summer and winter. Exceptional days, such as bank holidays, are found to have a similar shape to Sundays while the Christmas period is treated separately. The fact that the daily load shape is not consistent is important for modelling as it requires that the model must incorporate shape information.

Section 4.2.1 outlines the consequences (in terms of reduced data sets) of applying a different model to each day-type. While this reduces complexity, in that the shape of the load in each day-type is consistent, the data is more segmented, resulting in several smaller data sets. In this respect, it is the only modelling decision made in this chapter. However, similar studies (see Section 4.2.1) have shown that the segmented data set sizes are not too small to restrict the type of model.

The relationship of temperature (the dominant causal variable to load) to load is examined in Section 4.2.2. The results (Table 4.2) show that the load-temperature relationship for early winter day-types is constant within the range of those day-types. Thus, when forecasting the load, the operating point on the load-temperature curve is not significant. In contrast, this is not true for the other day-types. For these other day-types, the operating point on the load-temperature curve is significant and must be considered when modelling these loads.

Finally, the transitions between the different day-types is highlighted in Section 4.3. For some day-types there is a significant overlap indicating that some days are members of two day-types. As the data has already been segmented and the number of models decided, the load forecast in these overlapping days is required to be composed of the output of the two corresponding models.

Chapter 5

Parallel Models for Short Term Load Forecasting

5.1 Introduction

The relative performance of parallel and sequential models is unclear (Section 3.2.2), thus both are examined in this thesis. Parallel models are presented in this chapter and are compared with sequential models which are presented in Chapter 6. As discussed in Section 3.2.2, parallel models model the load at each hour of the day separately. In addition, for reasons discussed in Chapter 4 (Section 4.2.1), the data is disaggregated by day-type, with separate models being employed for each different day-type.

This chapter begins by demonstrating how the data sets are constructed prior to modelling (Section 5.2). A novel technique is then presented which examines the data to determine whether parallel models *could* be an appropriate approach (Section 5.3).

As mentioned in Section 2.3.1, Irish load has a trend and (yearly) seasonal component which varies little from day to day and so can be easily predicted. The approach taken here is to first remove the trend and seasonal component (Section 5.4) and then model the residual (Sections 5.5, 5.6 and 5.7). An important step in model building is to determine the appropriate inputs to use. This topic is examined in depth in Section 5.5, where several methods are compared. Using the inputs determined in Section 5.5, Sections 5.6 and 5.7 present linear and non-linear parallel models which are compared (Section 5.8), to determine whether the use of non-linear models in STLF is justified.

Practical note: As there are nine day-types (excluding Christmas), and the results of the analysis for each day-type are similar, only one day-type is presented in detail. The results for the other day-types are presented in summarised form, except where large differences in results are present. The late winter working day-type is used as the indicator as it represents a greater

forecasting challenge since the late winter load has a greater variability than the other day-types.

5.2 Construction of Partitioned Series for Parallel Model Building.

In Section 4.2.1, ten day-types were identified in Irish electrical load data. Thus the first step in constructing the partitioned series (Section 3.2.2) is to divide the data into these day-types (Note: these data sets are used for the models in Chapter 6). Within each day-type, the data is then partitioned by hour of the day to form the partitioned series. Let $y_{i,j}$ denote the load at hour i in day-type j. Then, Figure 5.1 below shows an example of how the partitioned series for 00hrs in the late winter working day-type (day-type 6), $y_{00,6}$, is constructed.

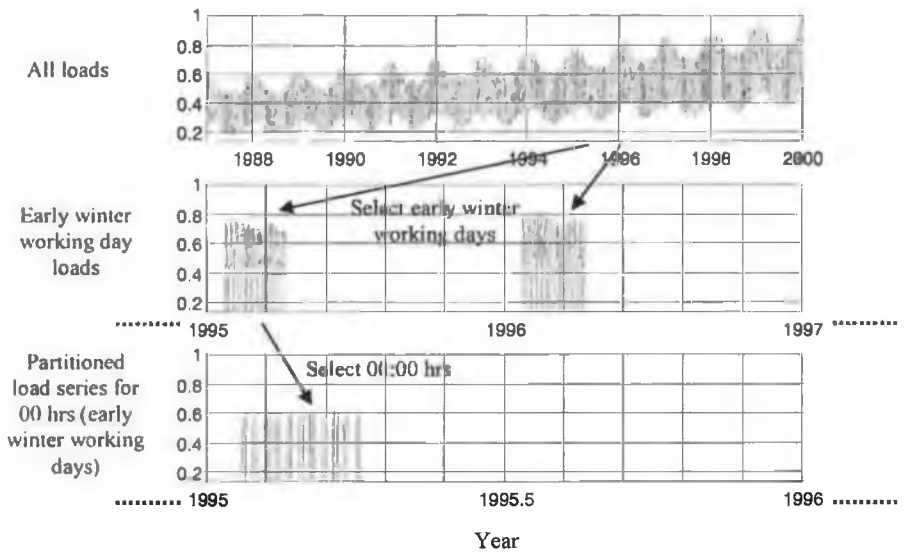

Figure 5.1 Construction of the partitioned series $y_{00,6}$ from complete data set.

$y_{00,6}$ is shown without the intermediate gaps in Figure 5.2, below.

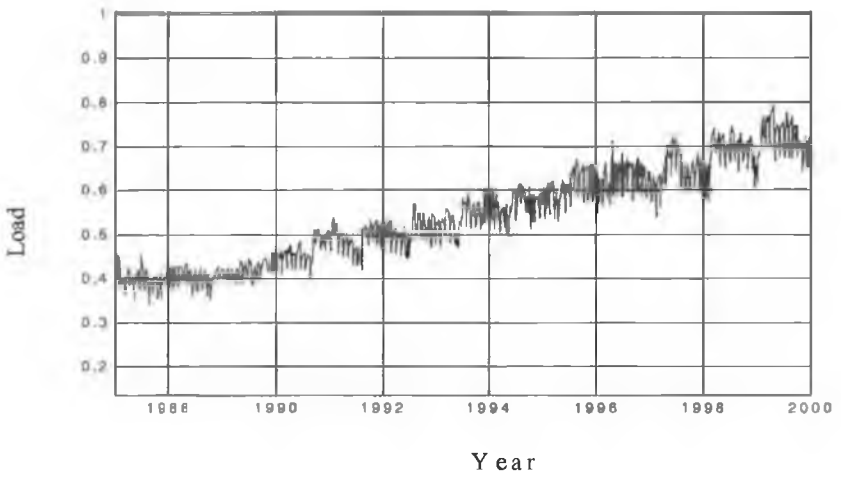

Figure 5.2 Partitioned series $y_{00,6}$ (Intermediate gaps removed).

Partitioning the load data by day-type, as above, is not the only alternative that is considered. In Section 5.4.2.3 two *alternative partitions* (this term will be used strictly to distinguish the *alternative partitioned series* from the *partitioned series*) are examined.

5.3 Examining Partitioned Series for Independence.

This section examines whether Irish electrical data has the characteristics of a sequential time series (defined in Section 3.2.2). This is achieved by examining the partitioned series within each day-type, to see whether they are independent from each other.

Consider the partitioned series $y_{0,j},\ldots,y_{23,j}$ which are the partitioned series for hours 00:00 hrs to 23:00 hrs in day-type j. If electricity demand is hour of the day *independent* * (which is the underlying assumption of the sequential approach), then the cross-correlation between any two adjacent partitioned series should be independent of which two hours are chosen. For example, the correlation between $y_{1,j}$ and $y_{2,j}$ should equal the correlation between $y_{4,j}$ and $y_{5,j}$. If the parallel models for hours i =0,…,23 on day-type j, $f_{0,j}(),\ldots,f_{23,j}()$, (Section 3.2.2) are linear functions then this hypothesis can be tested using the linear cross-correlation coefficient $r_{i,l}$ between parallel series i and l (the day-type j has been suppressed for clarity). Even if $f_{0,j}(),\ldots,f_{23,j}()$ are non linear, the linearising assumption of

* As defined in Section 3.2.2

using linear cross-correlation analysis is sufficient to either confirm or reject the hypothesis.

The cross-correlation coefficient is defined (Papoulis, 1991) as:

$$r_{i,l} = \frac{E[(\bar{y}_{i,j} - y_{i,j})(\bar{y}_{l,j} - y_{l,j})]}{\sqrt{E[(\bar{y}_{i,j} - y_{i,j})^2]}\sqrt{E[(\bar{y}_{l,j} - y_{l,j})^2]}} \quad (5.1)$$

where $\bar{y}_{i,j}$ is the average of $y_{i,j}$ and $E[\bullet]$ denotes the expectation operator. An example of the cross-correlation matrix between the 1pm, 2pm and 3pm series is shown in Table 5.1 below.

Table 5.1. Cross-correlation matrix of $y_{13:00\text{-}0.6}$, $y_{14:00\text{-}0.6}$ and $y_{15:00\text{-}0.6}$.

Hour	1 p.m. (13 hrs)	2 p.m. (14 hrs)	3 p.m. (15 hrs)
1 p.m. (13 hrs)	1	.9958	.9924
2 p.m. (14 hrs)	.9958	1	.9934
3 p.m. (15 hrs)	.9924	.9934	1

As can be seen, the cross-correlations are very high (Table 5.1). This is not surprising; the load curves within any day-type are very similar, thus a large component of the data is highly correlated (Section 2.3.2). Also note that the correlation between the load at 1 p.m. and 3 p.m. is less than that between the load at 1 p.m. and 2 p.m. This is to be expected, as a larger gap between the times leads to a lower correlation. However, the *main point* is that $r_{1,2}$ is *not* equal to $r_{2,3}$, suggesting that *load is hour of the day dependent*. The cross-correlation matrix between *all* the partitioned series is calculated and the contour for $r_{i,l}$ =0.99 is shown in Figure 5.3 . An example of the expected contour for the case where the load is hour of the day *independent*, is also shown for comparison (Figure 5.3).

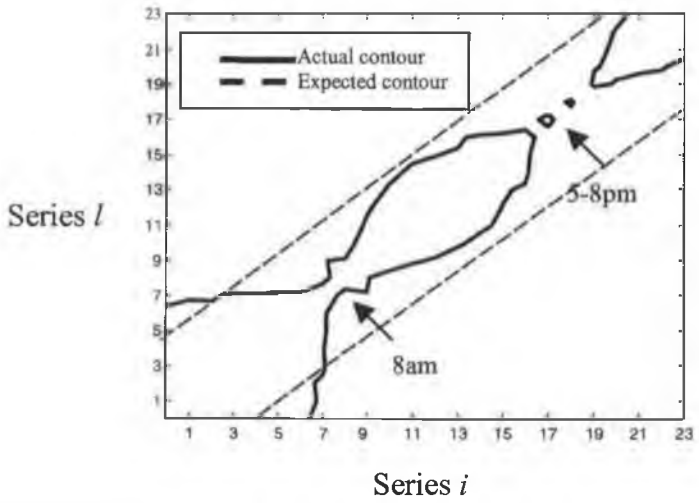

Figure 5.3 Actual and expected contour plot of $r_{i,l}$=0.99 for partitioned series $y_{i,6}$ with $y_{l,6}$.

Inside the contour, the cross-correlation is higher than 0.99 and outside it is lower. The contour changes with each hour and so the assumption that load is hour of the day independent does not appear to hold. The narrowness of the contour at 8 a.m. and 5-8 p.m. show that at these hours especially, the load *may* have an independent component. This suggests that it may be best to model the load at 8 a.m. and 5 p.m. to 8 p.m. separately rather than try to incorporate them into a sequential model.

5.4 Preliminary Modelling of Partitioned Series.

As pointed out in Section 2.3.1, electrical load has a rising trend and a (yearly) seasonal component. The non-stationarity in load is as a result of the rising trend and variability (Section 2.3.1), which changes very slowly from day to day. Thus, for the forecasting horizon required in the current research (i.e. up to seven days ahead), removing this non-stationarity is not a difficult task. The purpose of this section is to present this model, called the *preliminary parallel model* (PPM). This model is used to remove the trend for hour i on day k in day-type j, $d_{i,j}(k)$, and the seasonal component for hour i on day k in day-type j, $\psi_{i,j}(k)$ (Figure 5.4). The PPM for hour i on day-type j is denoted $\text{PPM}_{i,j}$. Modelling the error or *residuals*, $x_{i,j}(k)$, is the subject of later sections (Sections 5.5, 5.6 and 5.7).

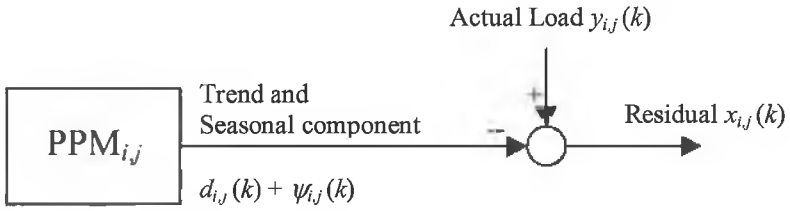

Figure 5.4. Preliminary parallel model overview

5.4.1 Choice of Preliminary Model.

Several techniques are described in Chapter 3 which can model the non-stationarity of a time series, namely:

1. Recurrent neural networks (Section 3.4.3.2),

2. Differencing or another stationarity transform as laid out in Section 3.3.1.1, and

3. State space models as laid out in Sections 3.3.2 and 3.3.4.

As explained in Section 3.4.3.2, the behaviour of recurrent neural networks can be difficult to predict, training can be computationally expensive and large data sets are generally required. In addition, the application of such a complex technique to this problem would seem inappropriate.

As explained in Section 3.3.1.1, differencing can lead to excessive inclusion of high frequency noise in the resulting stationary time series. The gentle differencing transform examined in Section 3.3.1.5 however, does not have this drawback and would seem to be a better choice. One problem with the gentle differencing technique is that the coefficients of the filter are not calculated recursively. This is a disadvantage, as electrical load is a non-stationary time series and allowing the coefficients to vary over the data may be advantageous.

Linear state space models (Section 3.3.2) have the ability to adjust the coefficients of the model as the model proceeds over the data. In fact an optimal algorithm (in the linear least squares sense), the Kalman filter (Section 3.3.2.1)

exists for recursively calculating these coefficients. As mentioned by Harvey (1984, Section 3.3.2.2), the state space equivalent of the gentle differencing algorithm is the Integrated Random Walk (IRW) (Section 3.3.2.2). In addition, all difference equation filters have a state space representation. Thus, a state space model appears to be the best choice.

Although there are an infinite number of state space models, the Basic Structural Model (BSM) (Section 3.3.2.3) is the most popular model for load forecasting (the reasons are laid out in Section 3.3.2.3). The IRW (Section 3.3.2.2) is used to model the trend in a BSM. As Irish load has a trend which rises at an increasing rate (Section 2.3.1) an IRW is particularly useful as the rate of increase of the trend is free to vary *via* the derivative term, $\dot{d}(k)$ (Section 3.3.2.2, Equation (3.55)). The seasonal component in a BSM may be modelled using a Periodic Random Walk (PRW), a Differenced PRW (DPRW) or a Dynamic Harmonic Regression (DHR) model (Section 3.3.2.2). The amplitude of the seasonal component of Irish load data is increasing (Section 2.3.1). Thus, the PRW is not suitable as the seasonal component in this model is assumed to be constant (Section 3.3.2.2). The DPRW and DHR however, allow the seasonal component to increase or decrease over time (Section 3.3.2.2). In contrast to the DPRW, the DHR has twice the number of coefficients to be calculated. Thus, the DPRW is chosen to model the seasonal component of the load data.

5.4.2 Application of the Basic Structural Model.

The BSM chosen consists of an IRW to model the trend and a DPRW to model the seasonal component. From Sections 3.3.2.2 and 3.3.2.3 the form of the model may be specified as:

$$\boldsymbol{\Phi}_{ppm_{i,j}} = \left[\begin{array}{cc|ccccc} 1 & 0 & 0 & & & & \\ 1 & 1 & 0 & & & & \\ \hline 0 & 0 & -1 & . & . & -1 \\ . & . & 1 & . & 0 & . \\ & & & \ddots & & \\ . & . & . & & . & . \\ 0 & 0 & 0 & . & 1 & 0 \end{array}\right] \qquad \boldsymbol{H}_{ppm_{i,j}} = \begin{bmatrix} 1 & 1 & 0 & \cdots & 0 \end{bmatrix} \tag{5.2}$$

where $\boldsymbol{\Phi}_{ppm_{i,j}} \in \mathbf{R}^{(s-1)\times(s-1)}$ and $\boldsymbol{H}_{ppm_{i,j}} \in \mathbf{R}^{1\times(s-1)}$ are the state transition matrix and observation matrix for the PPM for partitioned series i in day-type j and s is the seasonal length (Section 3.3.2.2). Additionally, the states of the PPM and the associated process noise may be expressed as:

$$\boldsymbol{\theta}_{ppm_{i,j}}(k+1) = \begin{bmatrix} d_{i,j}(k+1) \\ \dot{d}_{i,j}(k+1) \\ \psi_{1\,i,j}(k+1) \\ \psi_{2\,i,j}(k+1) \\ \vdots \\ \psi_{s-1\,i,j}(k+1) \end{bmatrix} = \boldsymbol{\Phi}_{ppm_{i,j}} \boldsymbol{\theta}_{ppm_{i,j}}(k) + \begin{bmatrix} \eta_{d\,i,j}(k+1) \\ \eta_{\dot{d}\,i,j}(k+1) \\ \eta_{\psi\,i,j}(k+1) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (5.3)$$

where $\boldsymbol{\theta}_{ppm_{i,j}}$ is the state vector of $PPM_{i,j}$ for partitioned series i in day-type j on day k, $\psi_{1\,i,j}(k),\ldots, \psi_{s-1\,i,j}(k)$ are the seasonal states, $d_{i,j}(k)$ is the trend and $\dot{d}_{i,j}(k)$ is the slope of the trend. $\eta_{d_{i,j}}(k)$, $\eta_{\dot{d}\,i,j}(k)$ and $\eta_{\psi_{i,j}}(k)$ are white noise components with variances $\sigma^2_{d_{i,j}}$, $\sigma^2_{\dot{d}_{i,j}}$ and $\sigma^2_{\psi_{i,j}}$ respectively. In addition, there is a measurement error term (Section 3.3.2.4), $x_{i,j}(k)$, (equivalent to the residual) with a variance of $\sigma^2_{x_{i,j}}$. In order to use this model there are several issues which need to be addressed:

- Treatment of the boundary between years in the partitioned series (Section 5.4.2.1),
- Tuning of the PPM which may be done using Predictive Error Decomposition (PED) or Sequential Spectral Decomposition (SSD) (Section 5.4.2.2), and
- Whether the partition of the data by day-type is better than either of the alternative partitions (Section 5.4.2.3).

5.4.2.1 Treatment of Day-Type Boundary Conditions

Figure 5.2 (Section 5.2) shows the partitioned load series for 00 hrs for the late winter day-type. However, as can be seen from Figure 5.1 (Section 5.2), this series is formed of *slices* of data from several different years. Thus, for example, the last data point taken in 1996 is followed by the first data point taken in 1997. However, between these points there is a gap of approximately 9 months. The growth in the trend component of load data over this period is *significant*. For this reason the boundaries between the years in all day-types must be given special treatment.

In order to estimate the trend states (Equation (5.3)) at the boundaries of the day-types, an IRW model was employed. The entire load data set is first divided into 24 partitioned series, one for each hour of the day. An IRW model is then tuned using PED (Section 3.3.2.4), on each of the 24 partitioned data series. During this process, the states of the IRW models at each point are recorded. The appropriate states at the day-type boundaries are then extracted and used as the estimates of the trend and slope components in the PPMs. The results are found to be quite robust to the parameters of the IRW and so the details are not provided.

As an example consider the PPM for 00 hrs in day-type 6, $PPM_{00,6}$. A Kalman filter is used to predict the states of this model (further details of the PPM are explained in the next section, the boundary conditions are the only concern here). At the last data point in 1996, the Kalman filter gives an estimate of the states of $PPM_{00,6}$ for the first data point in 1997. This estimate is, however, as mentioned above, a 9-month ahead forecast and is thus discarded. Instead, the trend and slope states recorded by the IRW model for 00 hrs are used as the estimates of the trend, $d_{00,6}(k_0)$, and slope, $\dot{d}_{00,6}(k_0)$, states in $\theta_{ppm_{00,6}}(k_0)$, where k_0 is the 1st point in 1997.

5.4.2.2 Tuning the Preliminary Parallel Models.

The PPM are tuned *via* selection of $\sigma^2_{d_{i,j}}, \sigma^2_{\dot{d}_{i,j}}, \sigma^2_{\psi_{i,j}}$ and $\sigma^2_{x_{i,j}}$ as explained in Section 3.3.2.4. Two methods are examined for tuning of the PPMs; PED and SSD (Section 3.3.2.4).

Tuning via PED

The PED algorithm is implemented in the following steps:

1. The state vector is initialised by setting the trend component to the first value of the time series and all the other states to zero (Table 5.2 below),

Table 5.2. Initialisation values for the state vector using PED.

State	Value
$d_{i,j}(0)$	$y_{i,j}(0)$
$\dot{d}_{i,j}(0)$	0
$\psi 1_{i,j}(0),\ldots, \psi s\text{-}1_{i,j}(0)$	0

2. $\sigma^2_{x_{i,j}}$ is set to one (for reasons given in Section 3.3.2.4). The other parameters of the PPM's, $\sigma^2_{d_{i,j}}, \sigma^2_{\dot{d}_{i,j}}$ and $\sigma^2_{\psi_{i,j}}$, are free to vary in order to minimise the log likelihood function, and

3. The log likelihood function to be minimised (Section 3.3.2.4) is:

$$\log L\left[\hat{y}_{i,j}(1), \hat{y}_{i,j}(2), \ldots, \sigma^2_{d_{i,j}}, \sigma^2_{\dot{d}_{i,j}}, \sigma^2_{\psi_{i,j}}\right] =$$

$$\frac{-(N-m_\theta)}{2}\log 2\pi - \frac{1}{2}\sum_{k=m\theta+1}^{N}\log\sigma^2_{\hat{y}_{i,j}}(k) - \frac{1}{2}\sum_{k=m\theta+1}^{N}\log\frac{x_{i,j}{}^2(k)}{\sigma^2_{\hat{y}_{i,j}}(k)} \qquad (5.4)$$

where $\hat{y}_{i,j}(k)$ is the estimated load for hour i in day-type j on day k and $\sigma^2_{\hat{y}_{i,j}}(k)$ is the estimated variance of $\hat{y}_{i,j}(k)$ at day k. m_θ is the number of data points in the transient in the covariance matrix of state vectors (Section 3.3.2.1). m_θ is set to 20 (the choice of 20 is explained below). Equation (5.4)

is minimised by used of the FMINSEARCH algorithm in Matlab which uses the Nelder Mead direct search algorithm (Box *et. al.*,1969).

Table 5.3 below shows the parameters calculated for the PPM for the 13 hrs late winter working day partitioned series.

Table 5.3. Parameters of $PPM_{13,6}$ using PED.

Parameter	Value
$\sigma^2_{x13,6}$	1
$\sigma^2_{d13,6}$	0.0590
$\sigma^2_{\dot{d}13,6}$	3.43×10^{-4}
$\sigma^2_{\psi13,6}$	2.79×10^{-4}

As $\sigma^2_{d13,6}$ is an order of 2 greater than $\sigma^2_{\dot{d}13,6}$ and $\sigma^2_{\psi13,6}$ (Table 5.3) it can be seen that this PPM allows the load estimates to vary mainly through the trend level state.

The forecasts produced by $PPM_{13,6}$ are shown in Figure 5.5 below, by way of example. As can be seen, the forecasts give a MAPE in the training set of 2.25% and in the validation set 1.92%. The forecasts of this partitioned series in the first few days are quite poor. This is due to the transient in the covariance matrix of the state vector estimates, as mentioned in Section 3.3.2.1. Note that the training set MAPE of 2.25% excludes the forecasts made within this transient.

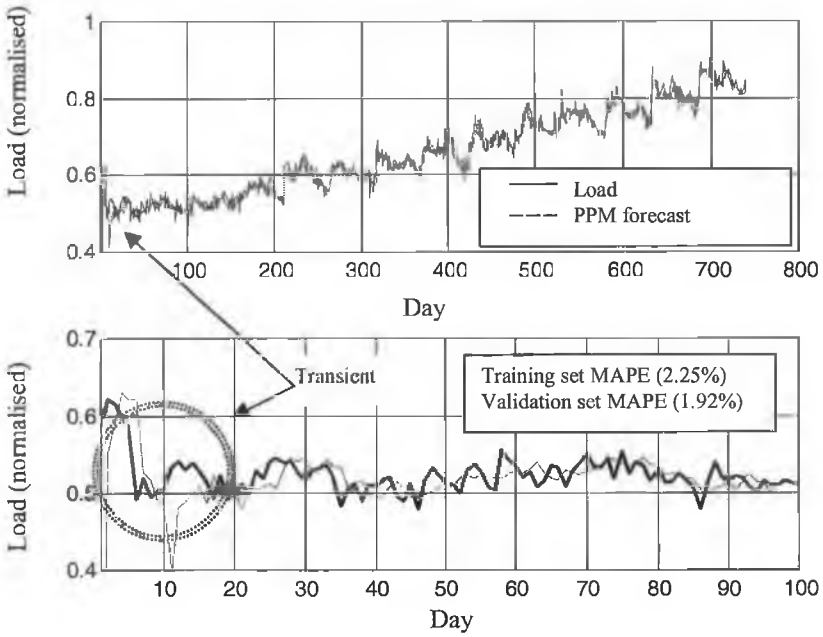

Figure 5.5. A plot of $y_{13,6}(k)$ and $\hat{y}_{13,6}(k)$ using $PPM_{13,6}$ trained with PED.

Figure 5.6 (note the different *y*-axis scales) below, shows how the variance of the estimates, $\sigma^2_{\hat{y}13,6}(k)$, varies as the Kalman filter runs through the data. As can be seen, after approximately 10 data points, the filter has *locked on* to the data. That is, the initial transient has subsided. In order to ensure that the transient has passed m_θ is set to 20 for all the PPMs.

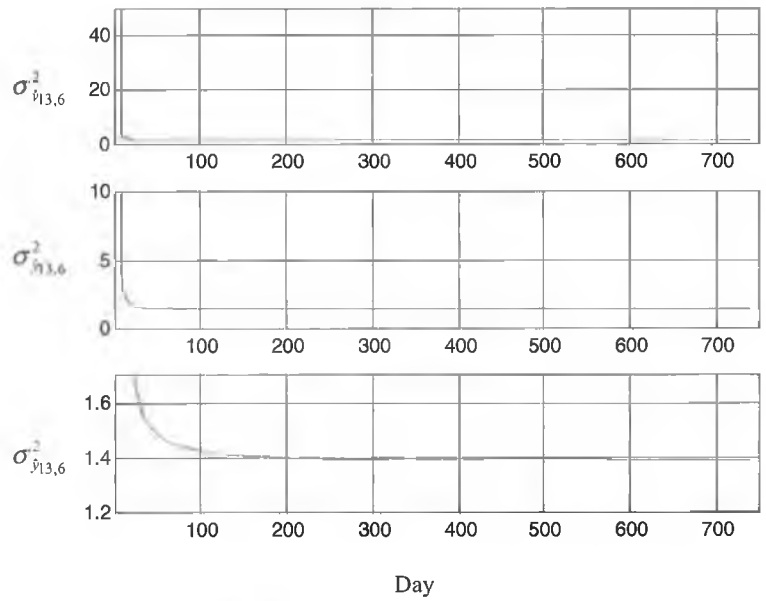

Figure 5.6. A plot of $\sigma^2_{\hat{y}13,6}(k)$ at each point in the data, note the y-axis has 3 different scales (PPM trained using PED).

The Sample Auto Correlation Function (SACF) (Section 3.3.1.4) of the residuals, $x_{13,6}(k)$, are shown in Figure 5.7 below. The confidence bounds represent the level past which a lag is said to be statistically significant 95% of the time (Harvey, 1989). Specifically, if the absolute value of the SACF at a certain lag is greater than the 95% confidence bounds, then the SACF at that lag is statistically different from zero. If the SACF is statistically different from zero, then a correlation exists between the residuals at that lag. As can be seen, the SACF is within the intervals at all lags, except at a lag of 1 and 7. The SACF at a lag of 7 has a value of -0.095 while the lower bound is -0.072. Thus the SACF at this lag is just beyond the lower bound and does not represent a strong correlation. Similarly, the correlation at a lag of 1, although statistically significant, is not very strong with a value of only 1.6. Thus, the residuals are deemed to be sufficiently random to state that no linear autoregressive information remains in the residuals. Finally, the SACFs for other hours and day-types are similar.

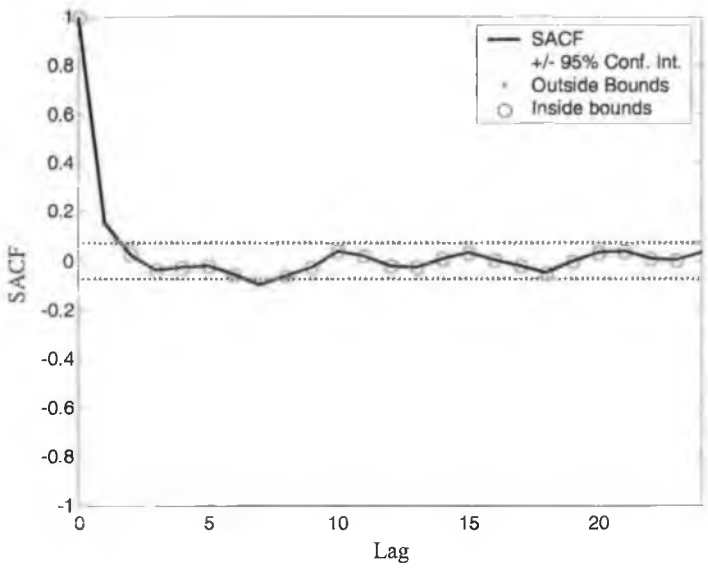

Figure 5.7. A plot of the SACF of $x_{13,6}(k)$ with 95% confidence intervals (PPM trained using PED).

Tuning via SSD.

SSD (Section 3.3.2.4) is implemented in the following steps:

1. The state vector is again initialised by setting the trend component to the first value of the time series and all the other states to zero (Table 5.4 below),

Table 5.4. Initialisation values for the state vector with SSD.

State	Value
$d_{i,j}(0)$	$y_{i,j}(0)$
$\dot{d}_{i,j}(0)$	0
$\psi_{1i,j}(0),\ldots,\psi_{s-1i,j}(0)$	0

2. $\sigma^2_{\eta_{i,j}}$ is set to one (as explained in Section 3.3.2.4). $\sigma^2_{d_{i,j}}$ is set to zero (Section 3.3.2.4). $\sigma^2_{\dot{d}_{i,j}}$ is not calculated *via* the periodogram as suggested in Section 3.3.2.4, as it may easily be specified as occurring at a frequency less than the yearly frequency. There are 53 load points per year in the late winter working day partitioned series for 13 hrs. The cut-off frequency, f_{50}, (Section 3.3.2.4) lies at a point lower than the corresponding yearly period. That is:

$$f_{50} < 1/53 \tag{5.5}$$

Taking f_{50} to be half the yearly frequency and substituting this value into Equation (3.67) gives:

$$\sigma^2_{\dot{d}_{i,j}} = 1605(1/(53\times 2))^4 = 1.25\times 10^{-5} \tag{5.6}$$

The value of $\sigma^2_{\dot{d}_{i,j}}$ may be calculated similarly for the other PPMs, noting that the number of data points per year varies depending on the day-type.

3. $\sigma^2_{\psi_{i,j}}$ is free to vary in order to minimise the log likelihood function, and

4. The log likelihood function to be minimised is (Section 3.3.2.4):

$$\log L\left[\hat{y}_{i,j}(1),\hat{y}_{i,j}(2),\ldots,\sigma^2_{\dot{d}i,j}\right] =$$

$$\frac{-(N-m_\theta)}{2}\log 2\pi - \frac{1}{2}\sum_{k=m\theta+1}^{N}\log\sigma^2_{\dot{x}i,j}(k) - \frac{1}{2}\sum_{k=m\theta+1}^{N}\log\frac{x_{i,j}{}^2(k)}{\sigma^2_{\dot{x}i,j}(k)} \quad (5.7)$$

where m_θ the number of data points in the transient in the covariance matrix of state vectors (Section 3.3.2.1). This is again set to 20 (the choice of 20 is explained below). Equation (5.7) is minimised by used of the FMINSEARCH algorithm in Matlab which uses the Nelder Mead direct search algorithm (Box *et. al.*, 1969).

Table 5.5 below shows the parameters calculated for $PPM_{13,6}$.

Table 5.5. Parameters of $PPM_{13,6}$ (trained using SSD).

Parameter	Value
$\sigma^2_{x13,6}$	1
$\sigma^2_{d13,6}$	0
$\sigma^2_{\dot{d}13,6}$	1.25×10^{-5}
$\sigma^2_{\psi13,6}$	.0044

This PPM allows the load estimates to vary mainly through the trend derivative state, as can be seen in the relative amplitudes of $\sigma^2_{\dot{d}13,6}$ and $\sigma^2_{\psi13,6}$ (Table 5.5). Similar component values were found for the other PPMs.

The forecasts produced by this PPM, for the 13 hrs late winter workings days partitioned series, are shown in Figure 5.8 below. The forecasts of this partitioned series in the first few days are again quite poor, due to the transient in the covariance matrix of the state vector estimates, as mentioned in Section 3.3.2.1.

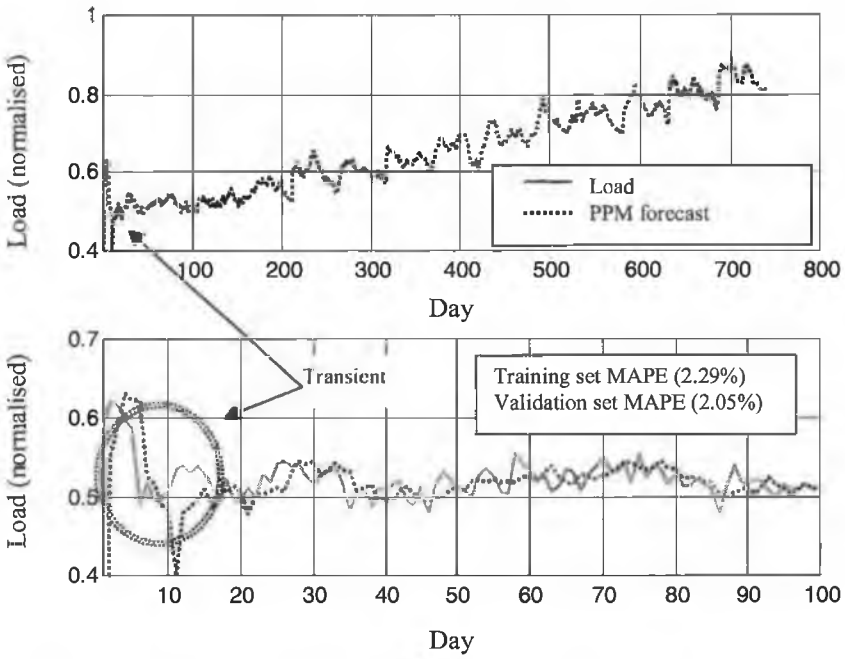

Figure 5.8. A plot of $y_{13,6}(k)$ and $\hat{y}_{13,6}(k)$ using $PPM_{13,6}$ (trained with SSD).

Figure 5.9 (note the different y-axis scales) shows how the variance of the estimates, $\sigma^2_{\hat{y}_{13,6}}(k)$, varies as the Kalman filter runs through the data. As can be seen, after approximately 10 data points, the filter has *locked on* to the data, i.e. the initial transient has subsided. In order to ensure that the transient has passed, m_θ is set to 20 for all the PPMs.

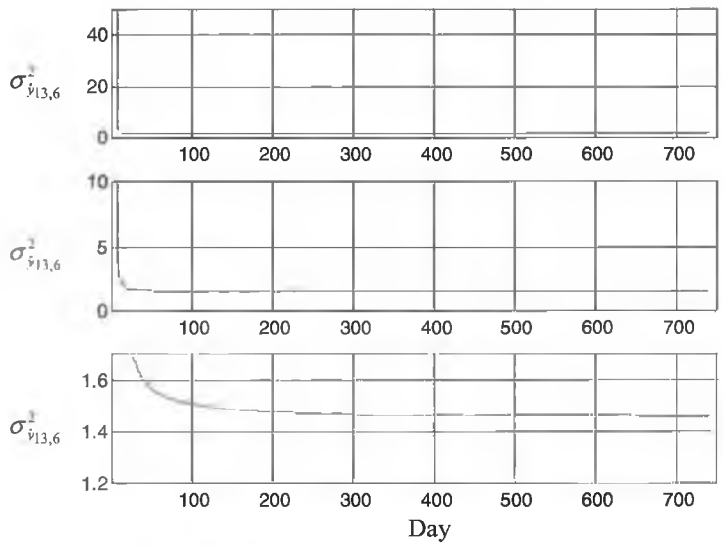

Figure 5.9. A plot of $\sigma^2_{\hat{y}_{13,6}}(k)$ at each point in the data, note the y-axis has 3 different scales (Model trained using SSD).

The SACF (Section 3.3.1.4) of the residuals produced by $PPM_{13,6}$, $x_{13,6}(k)$, using SSD, are shown in Figure 5.10 below. As can be seen, the SACF is very similar to the SACF of the residuals for the PPM trained with PED (Figure 5.7).

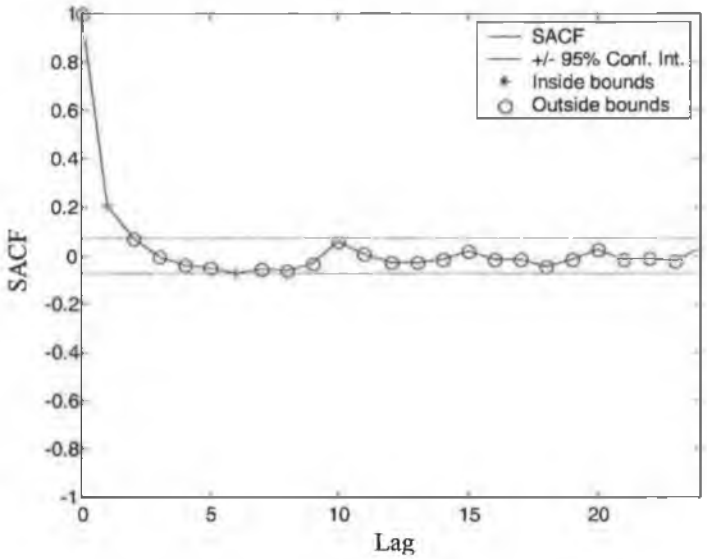

Figure 5.10. A plot of the SACF of $x_{13,6}(k)$ with 95% confidence intervals (trained with SSD).

A Comparison of PED and SSD.

Figure 5.11 shows the MAPE achieved by all the PPMs for late winter working day loads, tuned using PED and SSD. As can be seen, the results are very similar at all hours and the *daily MAPE* (i.e. the average MAPE over all hours of the day) is 2.79% for PED and 2.83% for SSD.

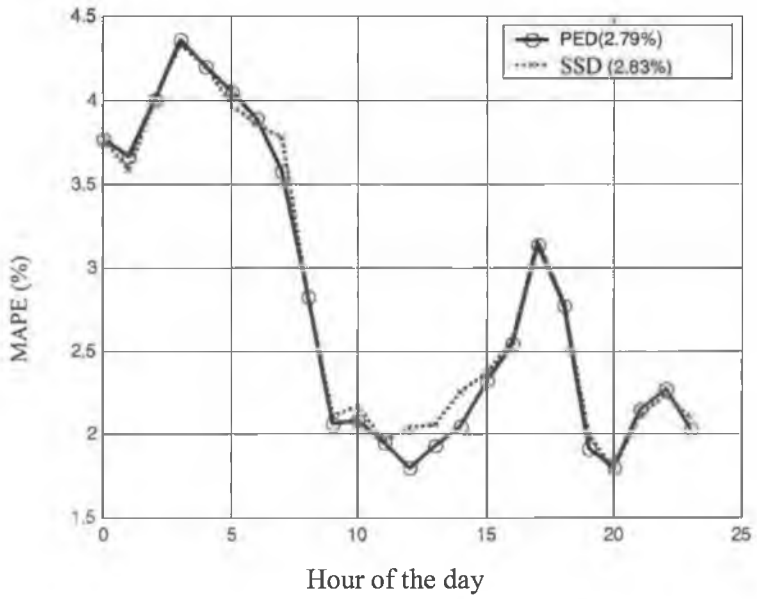

Figure 5.11. A comparison of PED and SSD. (late winter working days, validation set)

Tables 5.6 (validation set) and 5.7 (training set) show the daily MAPEs achieved using the two tuning algorithms for all day-types. As can be seen, PED is the superior tuning method for most day-types. However, the differences between the two tuning algorithms are insufficient to recommend one above the other. PED was chosen as the tuning algorithm.

Table 5.6. *Daily* MAPE for PPMs using different tuning algorithms (Validation set).

Day-type	PED MAPE (%)	SSD MAPE (%)
Early winter Sundays	3.53	**3.51**
Summer Sundays	**3.07**	3.12
Late winter Sundays	2.57	2.57
Early winter working days	2.65	2.65
Summer working days	2.60	**2.51**
Late winter working days	**2.79**	2.83
Early winter Saturdays	**2.76**	2.80
Summer Saturdays	**2.20**	2.27
Late winter Saturdays	2.54	2.54

Table 5.7. *Daily* MAPE for PPMs using different tuning algorithms (Training set).

Day-type	PED MAPE (%)	SSD MAPE (%)
Early winter Sundays	3.58	3.58
Summer Sundays	**3.99**	4.02
Late winter Sundays	3.00	3.00
Early winter working days	**2.98**	2.99
Summer working days	3.05	**2.88**
Late winter working days	2.98	**2.93**
Early winter Saturdays	3.23	3.23
Summer Saturdays	**2.95**	3.01
Late winter Saturdays	2.86	2.85

5.4.2.3 Tuning the Preliminary Parallel Models Using Alternative Data Partitions.

In Section 4.2 ten different day-types were identified. In addition, the reasons for treating each of these day-types with a separate model were explained. In order to test the validity of those assumptions, two alternative partitions of the data are modelled using a BSM:

Alternative partition 1: The entire data set with the exception of Christmas days is partitioned by hour of the day *only*, resulting in 24 partitioned series. As there are seven data points per week, the seasonal length used in the BSMs reflects the weekly seasonal length and is seven, and

Alternative partition 2: The entire data set with the exception of Christmas days is partitioned by hour of the day *and* in the following three categories:

- Sundays (including bank holidays),
- Saturdays, and
- working days.

This results in 24×3 alternative partitioned series. The seasonal lengths used in the BSMs, which model each alternative partitioned series, depend on the series in question. The working day series have a seasonal length of five to reflect the five working days per week (Monday-Friday) that are used. The Sunday and Saturday series have a season of one to reflect that only one day per week is used.

Tables 5.8 and 5.9 compare the daily MAPEs achieved by the PPMs trained with the *partitioned* series and the *alternative partitioned* series. Although the alternative partitions do not use the day-type partitions, the daily MAPE for the forecasts made on the day-types is used to allow comparisons.

The PPMs trained using the first alternative partition have a poor performance compared to the other partitions (Tables 5.8 ,5.9). This implies that trying to model Sundays, Saturdays and working days *together* does not work well with BSMs.

Table 5.8. Daily MAPE for PPMs using alternative data partitions (Validation set).

Day-type	Alternative Partition 1 MAPE (%)	Alternative Partition 2 MAPE (%)	Original Day-type Partition MAPE (%)
Early winter Sundays	14.28	**2.65**	3.53
Summer Sundays	17.95	3.20	**3.07**
Late winter Sundays	15.81	3.21	**2.57**
Early winter working days	4.94	**2.36**	2.65
Summer working days	6.15	3.04	**2.60**
Late winter working days	5.29	2.94	**2.79**
Early winter Saturdays	5.11	**2.19**	2.76
Summer Saturdays	5.60	2.83	**2.20**
Late winter Saturdays	7.92	3.06	**2.54**

Table 5.9. Daily MAPE for PPMs using different data partitions (Training set).

Day-type	Alternative Partition 1 MAPE (%)	Alternative Partition 2 MAPE (%)	Original Day-type Partition MAPE (%)
Early winter Sundays	16.69	**2.95**	3.58
Summer Sundays	19.93	**3.91**	3.99
Late winter Sundays	19.50	3.31	**3.00**
Early winter working days	5.31	**2.87**	2.98
Summer working days	6.75	3.29	**3.05**
Late winter working days	5.70	**2.86**	2.98
Early winter Saturdays	7.07	**2.62**	3.23
Summer Saturdays	7.00	3.37	**2.95**
Late winter Saturdays	9.10	3.07	**2.86**

The PPMs trained using the second alternative partition compare favourably to those trained with the day-type partition. In the validation set, a pattern emerges in which the second alternative partition appears to be superior for early winter day-types (Table 5.8). Table 5.10, below, shows the number of days in each day-type partition. As can be seen the early winter Sunday day-type has the lowest number of days of any Sunday day-type. Similarly, the early winter working day-type has the lowest number of days for any working day day-type. The same situation applies with the early winter Saturday day-types. This would seem to suggest that the early winter day-types can be modelled better by the inclusion of days from the other day-types into the training set. This benefit of including more

data has outweighed the disadvantage that other day-types have a different load curve shape (Section 2.3.2).

Table 5.10. The number of days per day-type (Christmas excluded).

Day-type	Number of days
1. Early winter Sundays	158
2. Summer Sundays	378
3. Late winter Sundays	164
4. Early winter working days	692
5. Summer working days	1382
6. Late winter working days	740
7. Early winter Saturdays	141
8. Summer Saturdays	284
9. Late winter Saturdays	150

In conclusion, the second alternative partition is superior for modelling early winter day-types and so is chosen as the PPM for these day-types (Table 5.11). It should be noted that in order to use this PPM the model must be applied to all the load data in the second alternative partition. However, only the forecasts for the early winter day-types are retained.

Table 5.11. Data partition used by each day-type.

Day-type	Partition
Early winter Sundays	2
Summer Sundays	Day-type
Late winter Sundays	Day-type
Early winter working days	2
Summer working days	Day-type
Late Winter working days	Day-type
Early winter Saturdays	2
Summer Saturdays	Day-type
Late winter Saturdays	Day-type

5.4.3 De-Seasonalisation of Weather Inputs.

There is a high degree of correlation between weather and load (Section 4.2.2). As the residuals, $x_{i,j}(k)$, have had the trend and seasonal components of the load extracted, this distorts the relationship between the weather variables and the residual. The seasonal component of the weather is related to $\psi_{i,j}(k)$, but not to $x_{i,j}(k)$ and so cannot be used to forecast $x_{i,j}(k)$. Thus, the seasonal component

in the weather variables must be removed. As pointed out by Harvey (1994) (in the general case), this may be achieved by filtering the causal variable with the same model used to produce the residual (Section 3.3.1.3). That is, the trend, $d_{w_{i,j}}(k)$, and seasonal components, $\psi_{w_{i,j}}(k)$, of the weather variable, $w_{i,j}(k)$, are removed using a PPM with the same coefficients as the PPM that produced $x_{i,j}(k)$. In this case, the autoregressive variable is $w_{i,j}(k)$, in place of $y_{i,j}(k)$. This results in a weather residual $x_{wi,j}(k)$ (Figure 5.12). The weather residuals $x_{w_{i,j}}(k)$ are also called the *pre-whitened weather variables.*

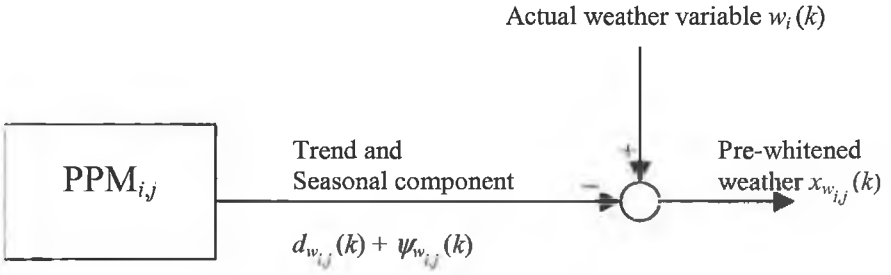

Figure 5.12. Pre-whitening a weather variable.

5.5 Input Selection.

Input selection forms perhaps the most important step in model building (Long *et. al.*, 2000, Abraham and Ledholter, 1983 and Hocking, 1976). Inclusion of non-causal variables leads to poor model generalisation. Hocking (1976) specifically identified two difficulties in linear regression models which incorporate non-causal inputs:

1. The error variance of the estimates is equal to or higher than linear regression models that do not include the non-causal variables, and

2. The non-causal variable may lead to a bias in the forecasts.

In addition, reducing the dimensionality of the inputs aids in model training (Long *et. al.*, 2000).

Another factor in input selection is *multi-colinearity*. Multi-colinearity refers to inputs which share the same or similar information. A good example, given by Abraham and Ledolter (1983), is a model which uses the number of *miles* travelled, $x_1(k)$, and the number of *kilometers* travelled, $x_2(k)$, at time k, as inputs. If the dependant variable is $y(k)$, then a linear regression model relating $y(k)$ to $x_1(k)$ and $x_2(k)$ may be expressed as:

$$y(k) = a_1 x_1(k) + a_2 x_2(k) + \varepsilon(k) \tag{5.8}$$

where a_1 and a_2 are the coefficients relating to $x_1(k)$ and $x_2(k)$, $\varepsilon(k)$ is an error term and k is the time. In matrix notation this may be expressed as:

$$\boldsymbol{y} = \boldsymbol{a}\boldsymbol{x} + \boldsymbol{\varepsilon} \tag{5.9}$$

where $\boldsymbol{y}$ is a vector of dependant variables, $\boldsymbol{a}$ is a column vector of coefficients, $\boldsymbol{x}$ is a matrix of the inputs called the *design matrix* and $\boldsymbol{\varepsilon}$ is vector of error terms. The least squares solution to Equation (5.9) is

$$\boldsymbol{a} = \left[\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}\right]^{-1}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} \tag{5.10}$$

As $x_1(k)$ and $x_2(k)$ contain the same information, $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$ is not of full rank and cannot be inverted. Even if $x_1(k)$ and $x_2(k)$ contain *similar* information, $[\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}]^{-1}$ will have a very small determinant and the variance of $\boldsymbol{a}$ will thus be very high.

However, consider the situation in which both $x_1(k)$ and $x_2(k)$ have added uncorrelated measurement noises with equal variance. In this situation, a new input variable, $x_3(k)$, may be formed by pre-processing the inputs as:

$$x_3(k) = \frac{x_1(k) + x_2(k)}{2} \tag{5.11}$$

The variance of the measurement noise on $x_3(k)$ is half that of $x_1(k)$ (McCabe, 1991) and it is thus a better input than either $x_1(k)$ or $x_2(k)$. Thus, multi-collinearity can be advantageous if dealt with properly.

The type of weather input variables available are listed in Table 2.2 (Section 2.2.1). For the load at any given hour, the previous 72 hours of weather are considered as inputs. Thus, the variables that are considered are:

- $\boldsymbol{t}_{i,j}(k)$, a vector of pre-whitened temperatures from hour i to hour i-72 on day k of day-type j,

- $\boldsymbol{h}_{i,j}(k)$, a vector of pre-whitened humidities from hour i to hour i-72 on day k of day-type j,

- $\boldsymbol{q}_{i,j}(k)$, a vector of pre-whitened wind speeds from hour i to hour i-72 on day k of day-type j,

- $\boldsymbol{o}_{i,j}(k)$, a vector of pre-whitened wind directions from hour i to hour i-72 on day k of day-type j. As the wind direction is a circular measurement, i.e. 0° is equiavalent to 360° the cosine *and* sine of this variable is used, and

- $\boldsymbol{c}_{i,j}(k)$, a vector of pre-whitened cloud covers from hour i to hour i-72 on day k of day-type j.

The set of all possible inputs $[\boldsymbol{t}_{i,j}(k)\ \boldsymbol{h}_{i,j}(k)\ \boldsymbol{q}_{i,j}(k)\ \cos(\boldsymbol{o}_{i,j}(k))\ \sin(\boldsymbol{o}_{i,j}(k))\ \boldsymbol{c}_{i,j}(k)]^{\mathrm{T}}$ contains 6×72 (432) inputs which, in many cases, is larger than the number of days in a partitioned series (Table 5.10). This means that not all of the inputs can be used in a linear regression model, as the coefficients cannot be estimated. For this reason, it is necessary to reduce the number of inputs, prior to any pre-processing.

The input selection procedure is performed in two stages:

1. Input reduction in order to reduce the number of inputs to a number less than the number of days in the partitioned series (Section 5.5.1), and

2. Input pre-processing and selection in order to take advantage of multi-colinearity and choose the optimal number of inputs (Section 5.5.2).

5.5.1 Input Reduction.

There are many methods that can be used to determine which inputs to retain and which to eliminate. A survey of methods is given by Hocking (1976) and with some exceptions (Castellano and Fanelli, 2000), these methods are still in common use today. Ideally, load forecasting models with all possible combinations of the reduced inputs should be constructed and the best selected. However, this is not possible, as the number of possible combinations of 432 inputs choosing 80 at a time is too large to implement, even with the simplest model.

The procedure used in the current research is called *stepwise regression*. Stepwise regression falls into 2 classes; *forward selection* and *backward elimination* (Abraham and Ledolter, 1983).

In forward selection, the *response variable* (output) is first estimated using the causal variables individually. For example, if there are 432 inputs this results in 432 models each with one input variable and 432 sets of response variable estimates. The causal variable that leads to the "best" model (the definition of best is discussed below) is then retained. The next step then uses the retained variable paired with all the un-retained variables individually. Using the example quoted above, this would lead to 431 models with two inputs and 431 sets of response variable estimates (as the retained variable is not paired with itself). The "best" pair of variables are then retained and this process then continues until the required number of input variables has been chosen or some stopping condition has been met. There are several types of forward selection algorithms which differ in their definition of the "best" model. The most popular measure of "best" estimate is the F-statistic of the model errors (Abraham and Ledolter, 1983 and Roecker, 1991). This has several problems, as shown by Grechanovsky and Pinsker (1995). Other measures, such as the mean squared error of the model forecasts on a validation set, have been proposed and found to work well (Roecker, 1991). It should be noted that forward selection is not guaranteed to give an optimal set of inputs. However, as stated by Hocking (1976), many sets

of inputs often exist which are *close* to optimal and forward selection usually finds one of these sets.

Backward elimination is the reverse of forward selection; a model with all the inputs is first trained and then the "worst" input is removed from the model at each iteration. As mentioned above, the number of inputs exceeds the number of data points in many partitioned series, and so calculating the coefficients of a model with all the inputs (i.e. step 1) is not possible. Thus, backward elimination cannot be considered as an option here and forward selection is used.

In the current research, forward selection is required, not to determine the optimal inputs, but rather to reduce the number of inputs to a level were input selection (Section 5.5.2) can be used. The number of inputs to be retained is set at 80. This number is considered sufficiently large to retain all the important inputs, while allowing a large degree of freedom in the data.

For input selection, a linear Regression Model (RM) (Harvey, 1994), which is computationally inexpensive, is used. Though this model is not representative of the full complexity of the system, it is more than sufficient to determine the relative importance of the inputs. The RM model for hour i on day-type j with n inputs, $RM_{n,i,j}$, has the form:

$$x_{i,j}(k) = a_{1,i,j}u_{1,i,j}(k) + a_{2,i,j}u_{2,i,j}(k) + ... + a_{n,i,j}u_{n,i,j}(k) + \varepsilon_{i,j}(k) \qquad (5.12)$$

where $x_{i,j}(k)$ is the residual to be forecast for hour i on day-type j at day k, $u_{l,i,j}$ is the l^{th} input for $RM_{n,i,j}$, $a_{l,i,j}$ is the coefficient applied to that input (calculated by least squares) and n is the number of inputs retained which increases by one at each step in the forward selection procedure (as mentioned above).

For each RM the sum squared error of the forecasts in the validation set is used as the indicator of the "best" model. Table 5.12 shows the delays in each of the input variables which were retained for the late winter working days, 13 hrs partitioned series.

Table 5.12. Delays in input variables retained (late winter working days, 13 hrs partitioned series)[%].

Variable	Retained delays.
Temperature	2, 3, 4, 15, 19, 20, 23, 42, 44, 45, 46, 48, 49
Humidity	0, 1, 6, 7, 8, 9, 10, 11, 12, 24, 25, 26, 27, 29, 32, 33, 34, 37, 40, 42, 43, 44
Cloud Cover	1, 3, 4, 8, 9, 14, 15, 16, 18
Wind speed	1, 5, 6, 7, 8, 11, 12, 13, 14, 15, 19, 21, 23
Cos(wind direction)	2, 3, 7, 12, 14, 15, 16, 17, 19, 20, 21
Sin(wind direction)	3, 4, 6, 7, 8, 11, 15, 16, 17, 19, 21, 23

These results need to be interpreted carefully. The fact that more humidity variables are retained does not necessarily mean that humidity is a more important weather variable than temperature. Rather, the temperature at one hour may be highly correlated to the temperature at other hours and so only a few temperatures are required to include all the temperature information. What is significant is that only one variable from a delay greater than 2 days is chosen (the temperature with a delay of 49 hours). This implies that only the previous 2 days of weather have a significant effect on the load.

5.5.2 Input Pre-Processing and Selection.

At this point there are 80 variables in the set of reduced inputs. The purpose of input pre-processing is to retain the *optimal* number of inputs as opposed to input reduction which seeks to reduce the number of inputs to a manageable size. In addition, multi-colinearity in the reduced input sets is used to advantage. The ideal approach would be to construct load forecasting models with all possible combinations of the reduced inputs and select the best. This is not possible, as the number of possible combinations of the inputs ($80! = 7.1\times10^{118}$) is too large to implement even with the simplest model.

Thus, the same form of linear regression model is utilised as that in Equation 5.12. However, for input pre-processing and selection, the inputs, $u_{l,i,j}$, are selected from the reduced inputs in different ways and may represent the reduced inputs after pre-processing (details are given in Sections 5.5.2.1 to 5.5.2.4).

[%] Note that the current hour has a delay of 0; a delay of 1 implies that the variable at 12 hrs (i.e. 13 hrs -1) on the current day is retained etc.

In order to allow a statistical evaluation of the different input selections a *bootstrapping* technique (van Giersbergen and Kiviet, 2002) is used. Bootstrapping refers to the use of alternate parts of the data set as training and validation sets (as mentioned in Section 2.2). In this case, eight bootstraps are constructed, where the validation set occupies a different range for each set (Figure 5.13, Table 5.13).

Table 5.13. Segmentation of data set for input selection (late winter working days, 13 hrs partitioned series)*.

Set	Training	Validation
Range	Variable	Variable
Size (Days)	624	78

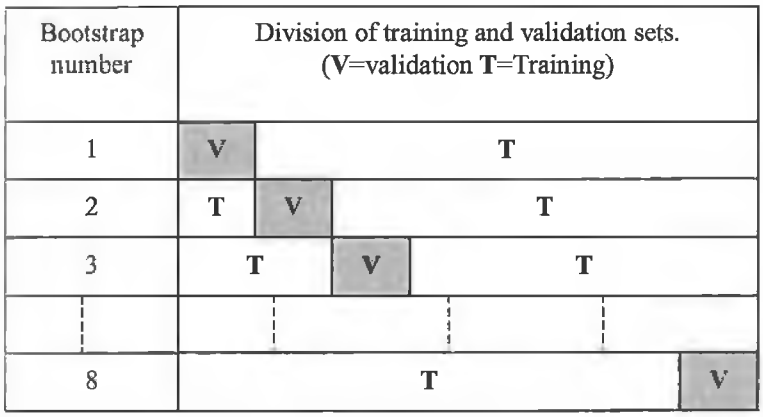

Figure 5.13. Selection of training and validation sets for input selection.

Four methods are now evaluated for input selection.

5.5.2.1 Method 1

Method 1 performs input selection using the following algorithm:

For all inputs:

Train a linear regression model.

Calculate the *T-ratio* (Kazmier and Pohl, 1987) of all the coefficients. This is the ratio of the variance of $a_{l,i,j}$ to the amplitude of $a_{l,i,j}$. A high T-ratio for $a_{l,i,j}$ implies that $u_{l,i,j}$ is of little use in forecasting $x_{i,j}$.

* Note that the days for 1999 and 2000 are excluded as they are part of the test set which is only used after models have been trained and validated.

Order the inputs with increasing value of T-ratio.

For number of inputs NINP = 1 to 80:

Select the first NINP inputs.

Train a linear regression model for these inputs, for each training bootstrap (see Table 5.13 and Figure 5.13).

Calculate each of the Minimum Absolute Errors (MAEs)*, on each of the bootstrap validation sets.

Next NINP

In summary, the technique uses the T-Ratio to order the inputs so that 80 combinations of the inputs can be evaluated (Figure 5.14).

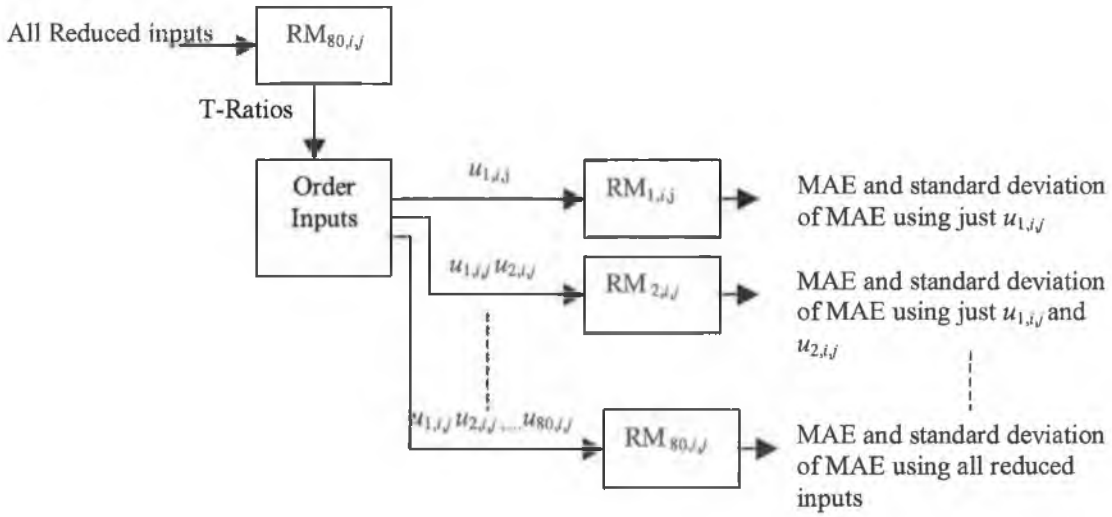

Figure 5.14. A block diagram of Method 1 for input selection for hour *i* day-type *j* model.

* Although the Mean Absolute Percentage Error (MAPE) is the preferred error measure in the field of short-term electrical load forecasting, the data trend changes over time and thus so does the MAPE. This means that the MAPE cannot be used as an error measurement in a *bootstrap*.

5.5.2.2 Method 2

One disadvantage of Method 1 is that it is susceptible to *colinearity* in the inputs (Rencher, 1995). If two inputs are highly correlated, then the likelihood is that neither variable will attain the same significance. This, in turn, can mistakenly push these inputs down the priority list. One means of reducing the colinearity in the input data is to use Principal Component Analysis (PCA) (Rencher, 1995).

PCA is a technique used for input dimension reduction (Moral and Valderamma, 1997). Consider, for example, the case where just two highly correlated input variables are available, $u_1(t)$ and $u_2(t)$ (Figure 5.15). PCA transforms these variables into a set of orthogonal variables $u'_1(t)$ and $u'_2(t)$, such that each variable represents the coefficient along a basis vector in characteristic directions of the original data set (Rencher, 1995) (Figure 5.15). Thus, the transformed variables are not colinear.

Additionally, the transformed variables ($u'_1(t)$ and $u'_2(t)$ in this example), or *components*, are ordered in descending order of variance explained in the original data set ($\sigma^2_{u'_1}$ and $\sigma^2_{u'_2}$ in Figure 5.15), with the first component containing the highest amount of information (Rencher, 1995). As can be seen from Figure 5.15, $u'_2(t)$ accounts for very little information and could be discarded.

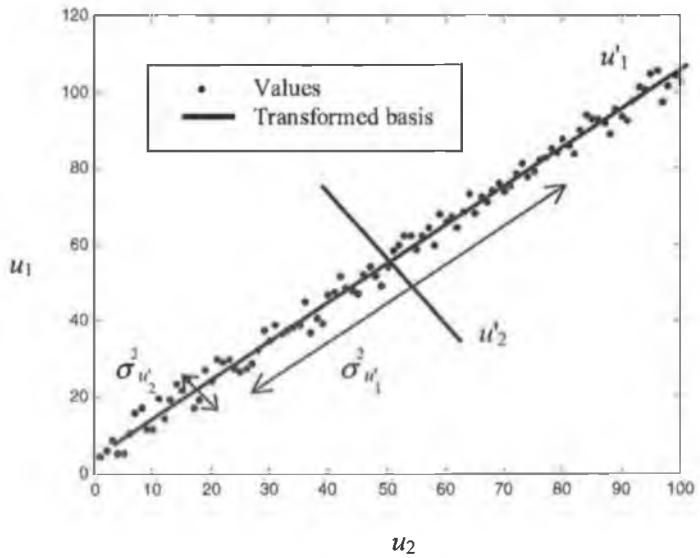

Figure 5.15. An example of two variables transformed using PCA.

Method 2 performs input selection using the following algorithm:

For all inputs:

Transform the inputs using PCA.

Order the transformed components in descending order of variance explained.

For number of components NCOMP = 1 to 80

Select first NCOMP components.

Train a linear regression model for these components, for each training bootstrap (see Table 5.13 and Figure 5.13).

Calculate each of the Minimum Absolute Errors (MAEs), on each of the bootstrap validation sets.

Next NCOMP

Figure 5.16 below gives an overview of Method 2.

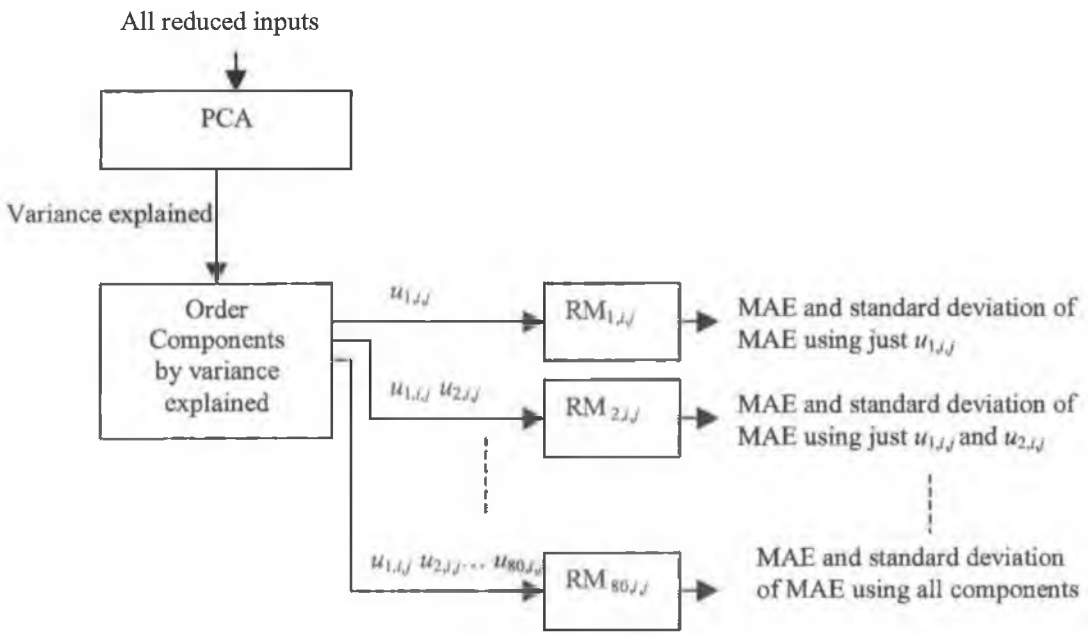

Figure 5.16. A block diagram of Method 2 for input selection for hour *i*, day-type *j*, model.

5.5.2.3 Method 3

One difficulty in Method 2 is that components are ordered by variance explained in the input data, which may not reflect the significance of these inputs with respect to the output data. For example, the first component may have the highest level of variance explained with respect to the input data, whilst still having no correlation with the output data. Method 3 attempts to circumvent this problem by removing the inputs least correlated with the output *prior* to transformation with PCA using the following algorithm:

For all inputs:

Train a linear regression model.

Calculate the *T-ratio* (Kazmier and Pohl, 1987) of all the coefficients.

Choose the inputs with the lowest 50 T-ratio scores.

Transform the inputs using PCA.

Order the transformed components in descending order of variance explained.

For number of inputs NCOMP = 1 to 50:

Select first NCOMP components.

Train a linear regression model for these components, for each training bootstrap (see Table 5.13 and Figure 5.13).

Calculate each of the Minimum Absolute Errors (MAE's), on each of the bootstrap validation sets.

Next NCOMP

Figure 5.17 below gives an overview of Method 3.

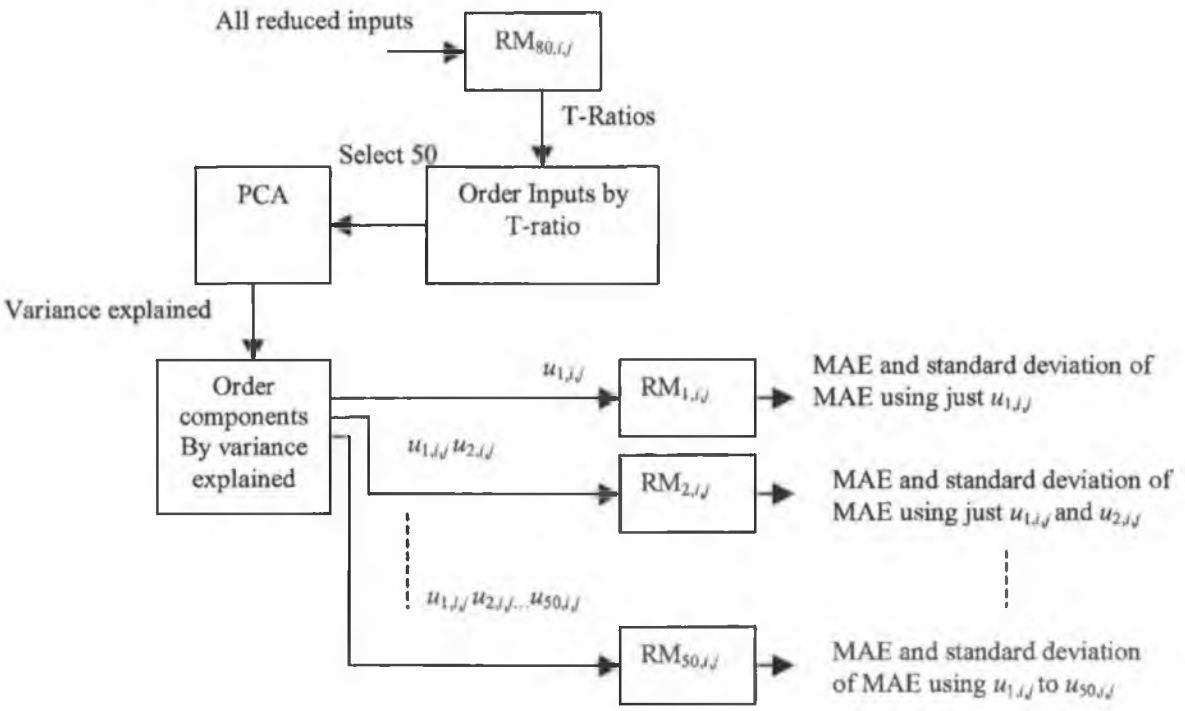

Figure 5.17. A block diagram of Method 3 for input selection for hour *i*, day-tpye *j*, model.

5.5.2.4 Method 4

Method 4 is similar to Method 3 in that a combination of PCA and multiple regression models is used. However, the order of application is reversed and the transformed components are ordered exclusively using the T-ratio scores, so that the correlation between the components and the output is emphasised, rather than the variance explained in the input. Method 4 performs input selection using the following algorithm:

For all inputs:

Transform the inputs using PCA.

Train a linear regression model with the transformed components.

Calculate the T-ratio (Kazmier and Pohl, 1987) of all the coefficients.

Order the transformed components with increasing value of T-ratio.

For number of inputs NCOMP = 1 to 80:

Select first NCOMP components.

Train a linear regression model for these components, for each training bootstrap (see Table 5.13 and Figure 5.13).

Calculate each of the Minimum Absolute Errors (MAEs), on each of the bootstrap validation sets.

Next NCOMP

Figure 5.18 below gives an overview of Method 4.

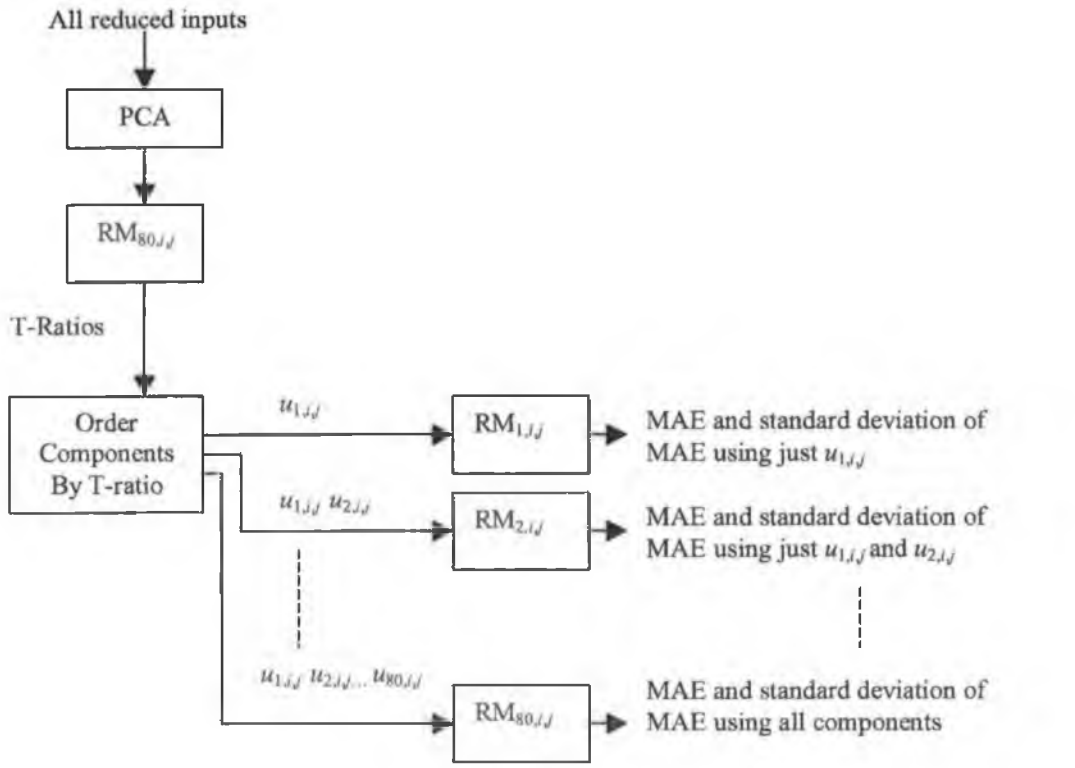

Figure 5.18. A block diagram of method 4 for input selection for hour *i*, day-type *j*, model.

5.5.2.5 A Comparison of Methods 1-4.

For each method and each hour of the day, the optimum selection of inputs (Method 1) or components (Methods 2-4) is that which gives the minimum MAE in the validation set (Figure 5.19). Plots of the MAE as a function of the number of inputs used (or components used in the case of Methods 2-4) are shown in Figure's 5.19 to 5.22 below, for the four methods.

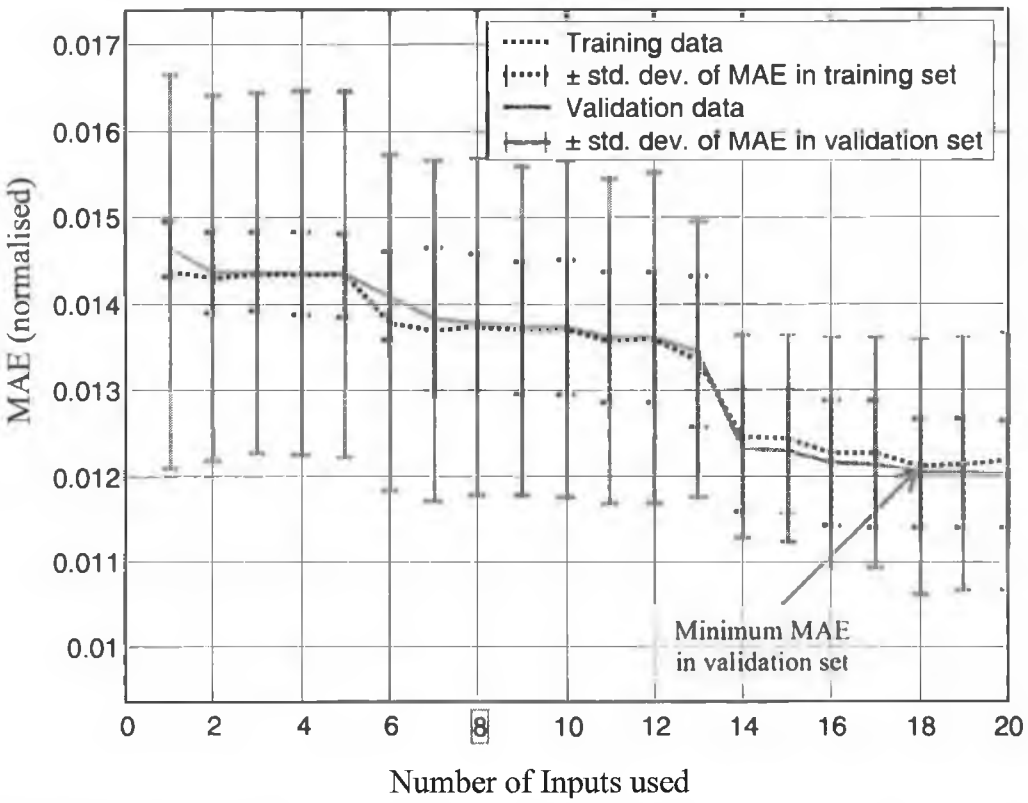

Figure 5.19. Bootstrapped MAE for Method 1 (late winter working days, 13 hrs partitioned series).

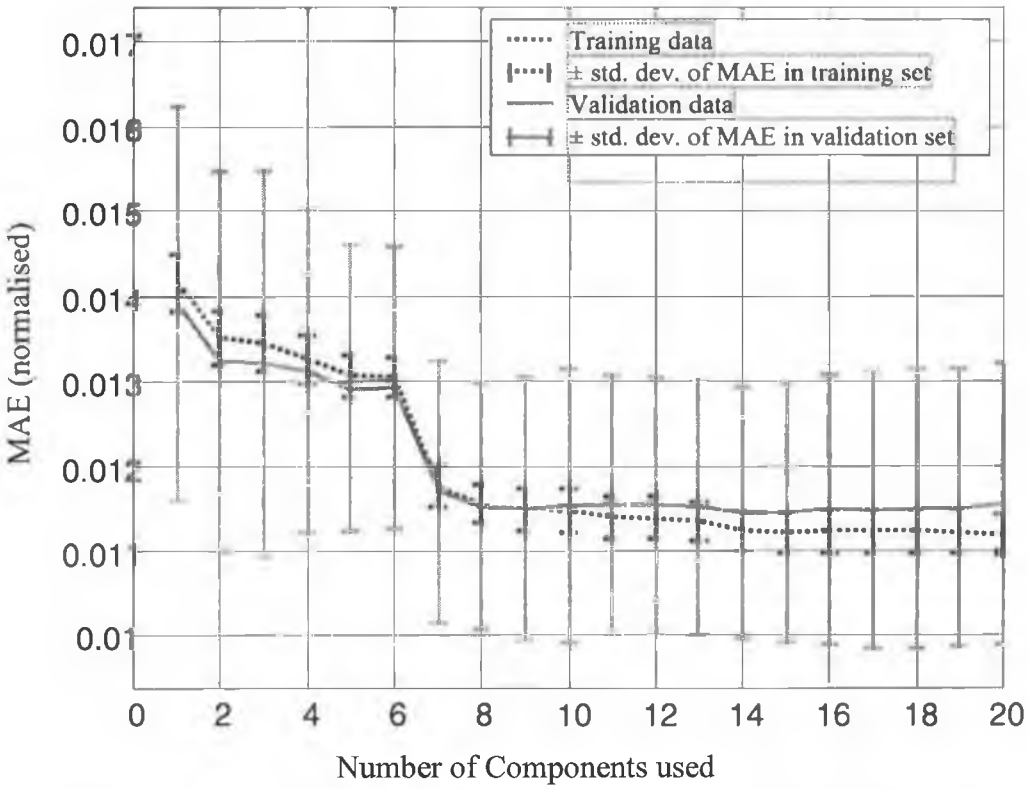

Figure 5.20. Bootstrapped MAE for Method 2 (late winter working days, 13 hrs partitioned series).

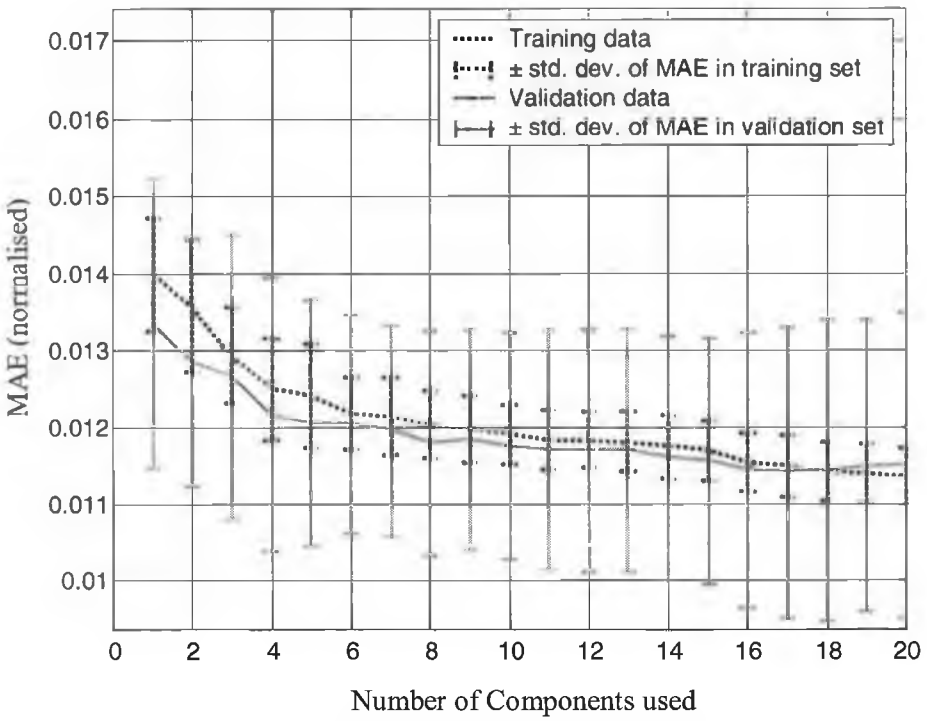

Figure 5.21. Bootstrapped MAE for Method 3 (late winter working days, 13 hrs partitioned series).

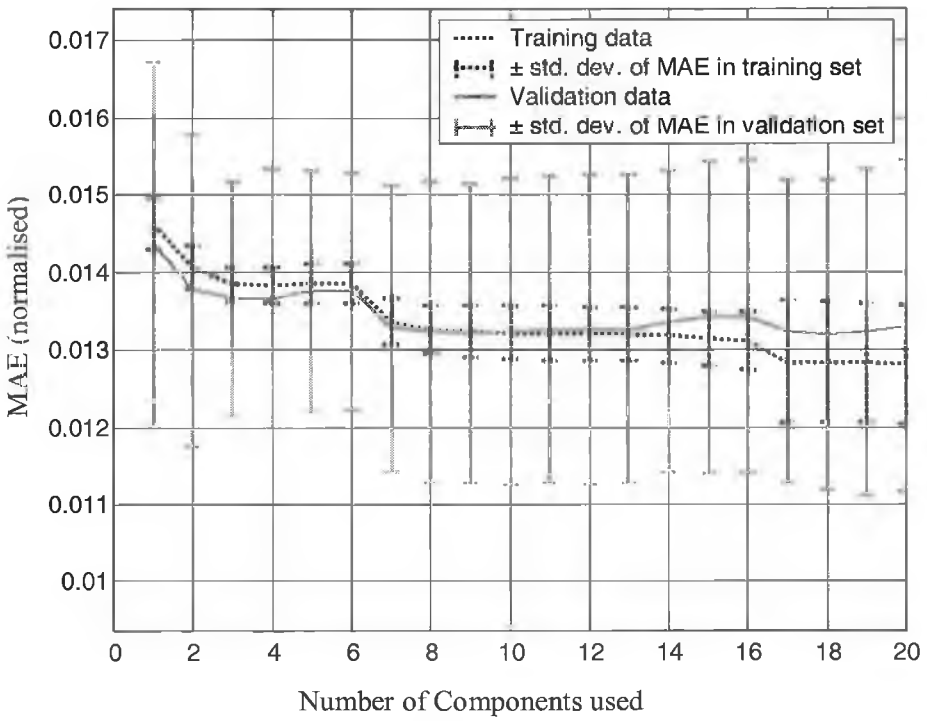

Figure 5.22. Bootstrapped MAE for Method 4 (late winter working days, 13 hrs partitioned series).

As can be seen from Figures 5.19 to 5.22 there is good agreement between the MAEs for the training and validation sets, although the standard deviations of the MAEs in the validation sets is larger. This is expected, as the validation set is not

used in training. However, as the number of inputs or components used increases, the MAE in the validation set starts to deviate from the MAE in the training set. An example of this may be shown by extending the number of components plotted to 50 (Figure 5.23). The deviation occurs as complexity of the model increases past the complexity of the data (this concept is discussed in Section 3.3.1.4).

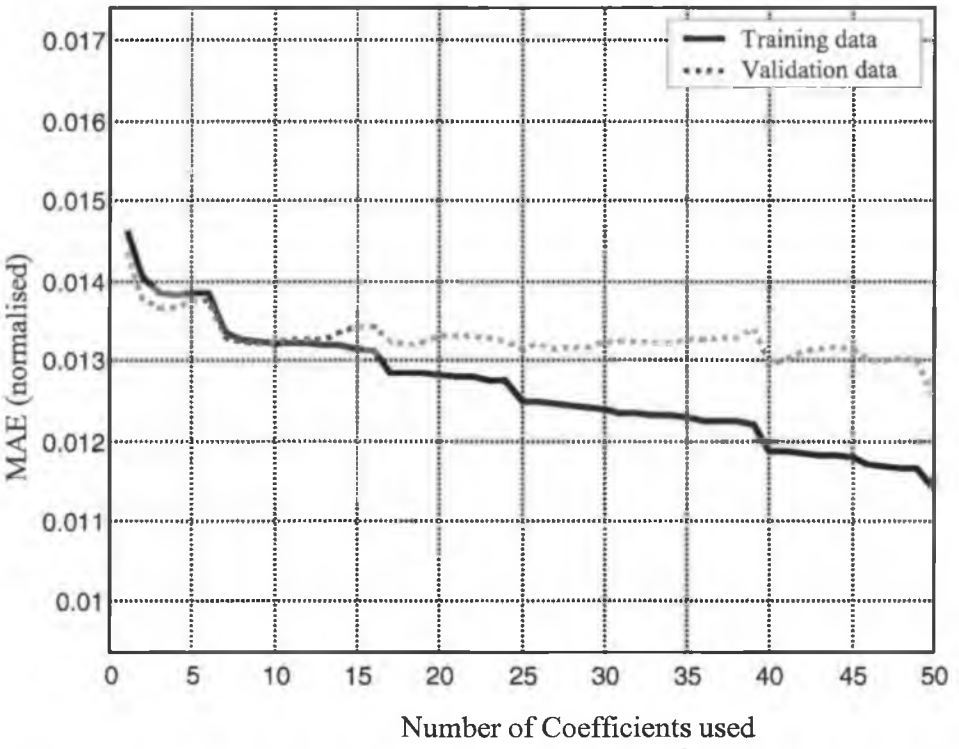

Figure 5.23. Bootstrapped MAE for Method 2 (late winter working days, 13 hrs partitioned series).

Table 5.14 (next page) shows the minimum MAE in the validation set and the corresponding number of components for each method. As can be seen, Method 2 has the lowest MAE at 0.0112 with 50 components. However, Method 3 has an MAE of 0.0114 and uses only 17 components. Closer inspection of Figure 5.20 shows that with 8 components, the MAE for Method 2 is 0.0115. Thus the improvement in Method 2 by the inclusion of the last 42 inputs is not significant. Subjectively, therefore; applying Method 2 and using 8 components would seem to be the best choice for input selection.

Table 5.14. Minimum MAE, corresponding number of components and standard deviations (normalised, late winter working days, 13 hrs partitioned series).

Method	MAE	Number of components	Standard deviation
1	0.0115	46	0.0020
2	0.0112	50	0.0022
3	0.0114	17	0.0019
4	0.0125	48	0.0016

An objective measure of the performance of a model which *penalises the complexity of the model* is the AIC criterion (discussed in Section 3.3.1.4). The AIC for each method, as a function of the number of components, is shown in Figure 5.24 below. As can be seen, the intuitive selection of Method 2 (with 8 components) made above yields the minimum AIC .

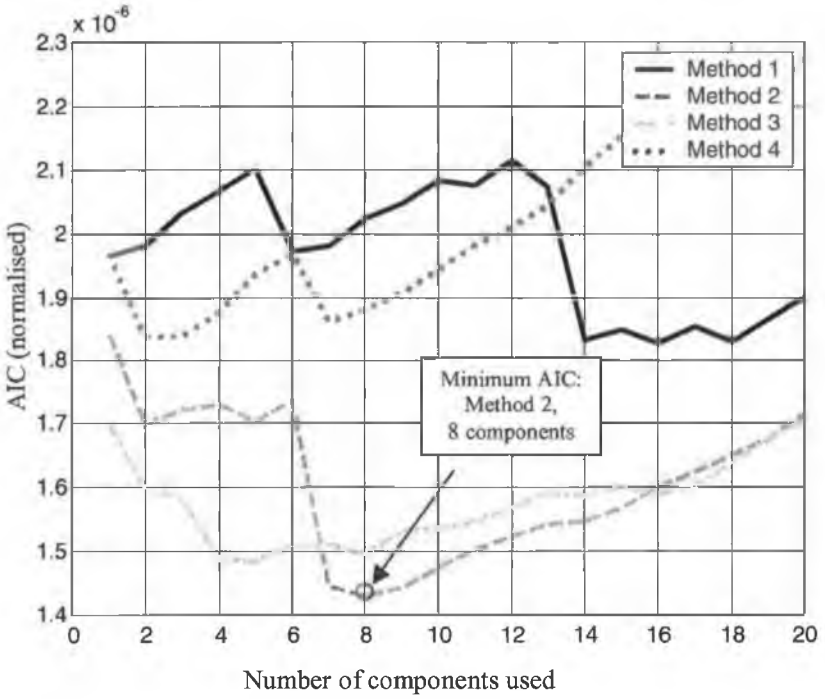

Figure 5.24. AIC for Methods 1 to 4 (late winter working days, 13 hrs partitioned series).

The optimum method and number of components, as selected by the AIC, for each hour of the day in the late winter working day day-type are tabulated in Table 5.15 below.

Table 5.15. Optimum method for input selection and associated statistics (AIC, MAE and standard deviation of MAE have been normalised, late winter working days, validation set).

Hour of the day	Method	AIC ($\times 10^{-5}$)	Number of components	MAE	Standard deviation of MAE
00:00	2	0.3378	3	0.0188	0.0034
01:00	2	0.2328	7	0.0156	0.0029
02:00	2	0.2204	1	0.0152	0.0026
03:00	2	0.2130	9	0.0150	0.0024
04:00	2	0.2035	1	0.0146	0.0024
05:00	2	0.2030	7	0.0146	0.0028
06:00	2	0.1786	2	0.0136	0.0023
07:00	2	0.1902	7	0.0140	0.0020
08:00	2	0.2070	2	0.0147	0.0013
09:00	2	0.1485	7	0.0121	0.0011
10:00	2	0.1421	15	0.0115	0.0016
11:00	2	0.1434	9	0.0115	0.0013
12:00	2	0.1658	7	0.0126	0.0014
13:00	2	0.1428	8	0.0115	0.0014
14:00	2	0.1503	8	0.0119	0.0013
15:00	2	0.1713	17	0.0123	0.0020
16:00	2	0.2931	19	0.0175	0.0026
17:00	2	0.3691	15	0.0197	0.0037
18:00	2	0.2107	7	0.0145	0.0029
19:00	2	0.1262	7	0.0110	0.0023
20:00	2	0.1441	7	0.0123	0.0026
21:00	2	0.1257	6	0.0113	0.0018
22:00	2	0.1605	6	0.0130	0.0021
23:00	2	0.0906	10	0.0092	0.0013

As can be seen, Method 2 performs the best in all cases and the number of components chosen ranges from 1 to 19. The average number of components used is 7.79, while the mode is 10. The histogram of the optimum number of components for all hours in the late winter working day day-type is shown in Figure 5.25 below and shows that 7 to 8 components appears to be the most popular choice. However, the conclusion here is that the optimum number of components is hour of the day dependent and does not follow any discernible pattern.

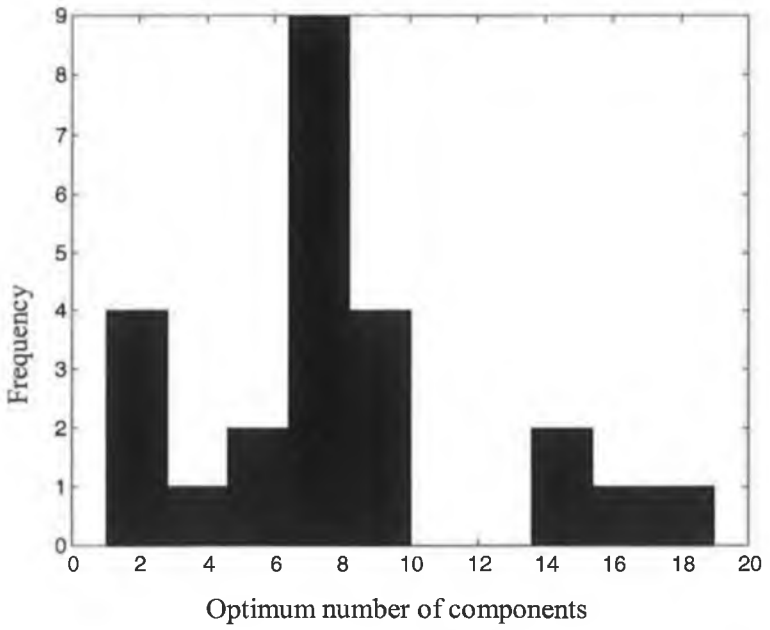

Figure 5.25. A histogram of the number of components chosen by Method 2 over all hours of the day (late winter working days).

5.6 Linear Modelling of Partitioned Series.

This section examines linear modelling of the partitioned series. Section 5.5 used linear models to determine the best inputs and these linear models are used to form the *linear parallel models*. However, while the same linear model *structure* is used in this section, the models are no longer trained using the bootstrapped data sets used in Section 5.5. An overview of the linear parallel models is shown in Figure 5.26 below. Figure 5.26 can also be viewed as a summary of the best preliminary models and input selection techniques selected in Sections 5.4 and 5.5.

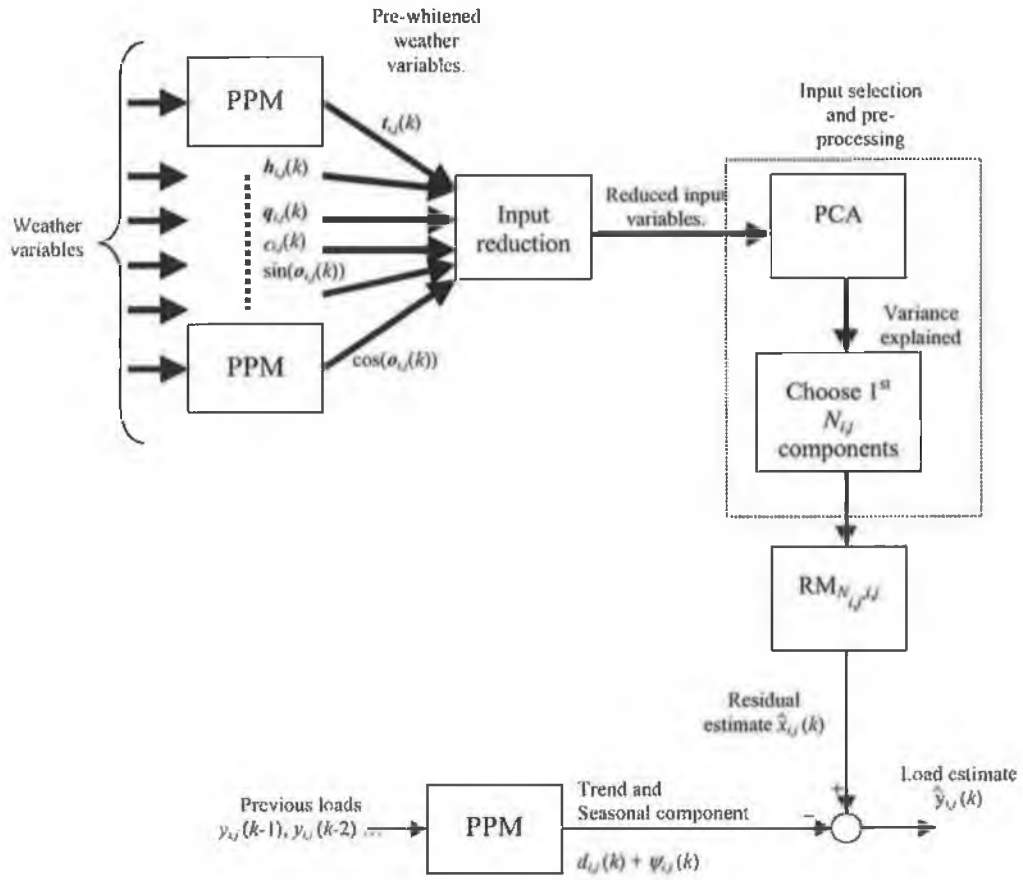

Figure 5.26. Linear parallel model overview

The modelling process may be split into 4 stages:

1. Input Selection and Pre-processing (Section 5.6.1),
2. Structure Determination (Section 5.6.2),
3. Parameter Evaluation (Section 5.6.3), and
4. Model Validation (Section 5.6.4).

Although stages 1-3 have been covered in previous sections they are summarised here for clarity and to allow easy comparison with the non-linear parallel models (Section 5.7).

5.6.1 Input Selection and Pre-Processing.

The inputs and pre-processing are discussed in Section 5.5.

5.6.2 Structure Determination.

The structure of the PPMs has been discussed in Section 5.4 while the structure of the RMs is determined uniquely by the number of inputs as discussed in Section 5.5.

5.6.3 Parameter Evaluation.

The parameters of the PPMs are discussed in Section 5.4. The RMs are trained using least squares. The data set is divided into a training set a validation set and a novelty set as explained in Section 2.2 (Table 2.3, recreated below in Table 5.16)

Table 5.16. Division of data set.

Set	Training	Validation	Novelty
Range	1987-1996	1997-1998	1999-2000

5.6.4 Model Validation.

The coefficients applied to the eight components in the RM applied to the 13:00hrs late winter working day partitioned series are shown in Table 5.17 below. As can be seen each input has a high T-ratio except for the 5^{th} input which may not be required. The sum of the variance explained in the reduced input set by the eight components is 52% and so approximately half the information in the set of reduced inputs has not been used in this model.

Table 5.17. Coefficients applied to the components, the T-ratio of the co-efficients and the % of variance explained in the reduced input set by that component.

Component	1	2	3	4	5	6	7	8
Coefficient	-3.42	4.41	4.64	11.5	-0.74	-10.25	-5.35	-1.35
% Variance Explained	13.5	9.89	8.99	5.70	4.64	4.01	3.17	2.86
T-ratio	4.34	4.84	4.86	9.59	0.54	7.31	3.22	2.45

The Ljung-Box test statistic for the errors in the validation set is 0.017 (Equation 3.36). The number of points in the validation set for the 13:00hrs late winter working day is 104 and so the critical level with a 95% confidence level is $\chi^2_5(\sqrt{104}-8=2) = 5.991$. As the test statistic is less than the critical level this model is deemed to have been validated.

The SACF of the errors in the validation set are shown in Figure 5.27 below. As can be seen, the SACF at a lag of 5 is statistically different from zero. However, the SACF at this lag is -.23 and is not a strong correlation.

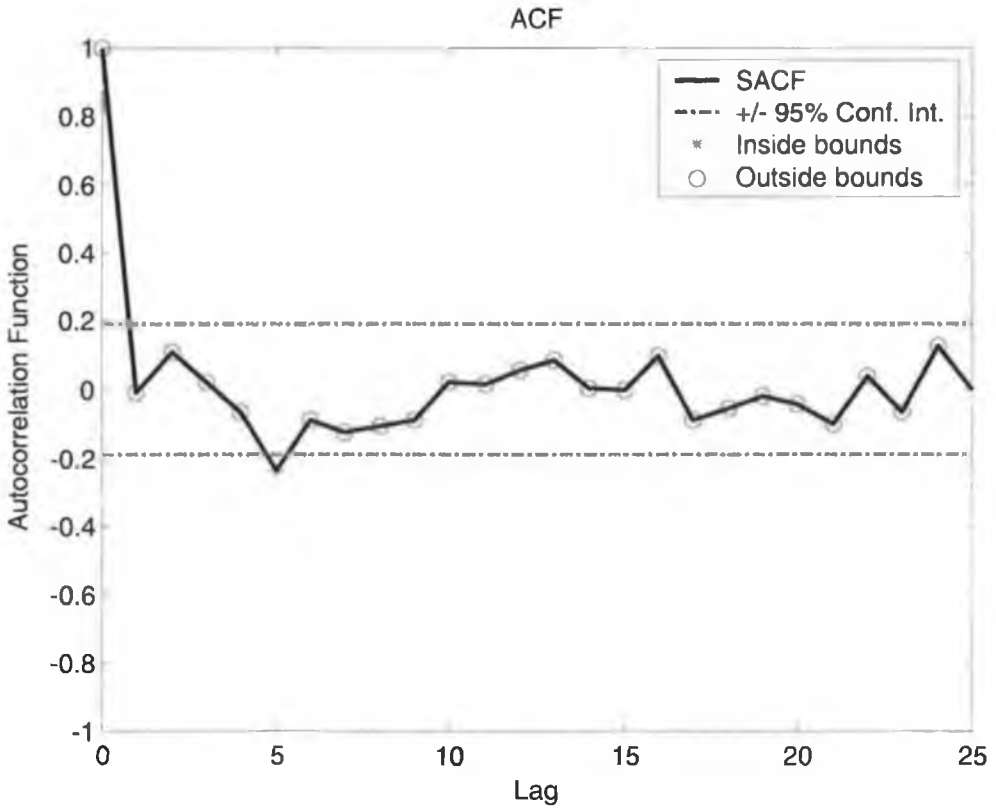

Figure 5.27. The SACF for the errors in the linear parallel model for 13 hrs late winter working day partitioned series.

The MAPE for all hours of the day in the late winter working day-type is shown in Figure 5.28 on the next page (novelty set). As can be seen there is an improvement of 0.4% in the forecast by inclusion of the weather variables.

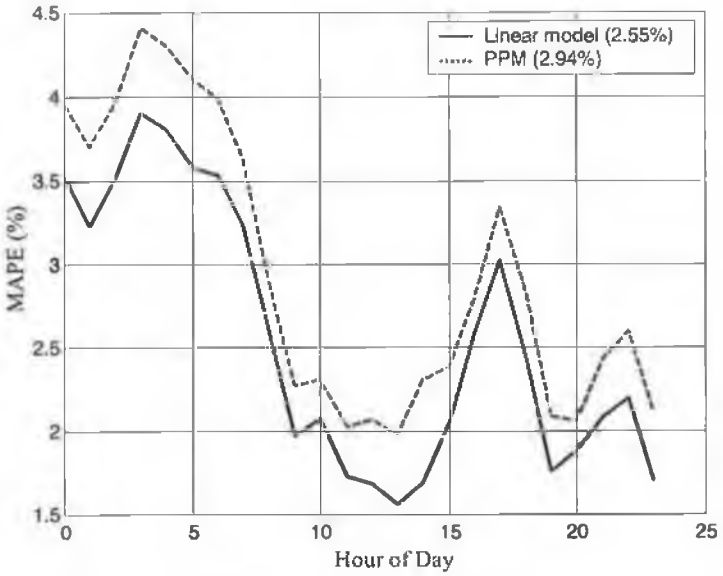

Figure 5.28. The SACF for the errors in the linear parallel model for the 13 hrs late winter working day partitioned series (Novelty Set).

Table 5.18, below, shows the average daily MAPE for all day-types in the Novelty set for both the PPM and Linear Parallel models.

Table 5.18. Daily average MAPE for Linear Parallel models and PPMs (Christmas days excluded, novelty set).

Day-type	MAPE PPM's (%)	MAPE Linear Parallel Models (%)	Improvement (%)
1. Early winter Sundays	2.75	2.70	0.04
2. Summer Sundays	3.22	3.14	0.08
3. Late winter Sundays	2.78	2.75	0.02
4. Early winter Working days	2.36	2.25	0.10
5. Summer working days	3.04	2.99	0.05
6. Late winter working days	2.94	2.55	0.38
7. Early winter Saturdays	2.19	2.20	-0.01
8. Summer Saturdays	2.83	2.81	0.01
9. Late winter Saturdays	3.06	2.86	0.20

5.7 Non-Linear Modelling of Partitioned Series.

This section examines non-linear modelling of the partitioned series. Non-linear models have been found to provide excellent results in Short Term Load Forecasting (STLF), as discussed in Section 3.4. An overview of the *non-linear parallel models* is shown in Figure 5.29, below. In contrast to the linear parallel models (Figure 5.26), the linear Regression Models (RM) are replaced with Neural Networks (NN) and several additional inputs are considered. The motivations for using neural networks and including additional inputs are discussed in Sections 5.7.1 and 5.7.2, respectively. Similar to the linear parallel models, the modelling process is again split into 4 stages:

1. Input Selection and Pre-processing (Section 5.7.2),
2. Structure Determination (Section 5.7.3),
3. Parameter Evaluation (Section 5.7.4), and
4. Model Validation (Section 5.7.5).

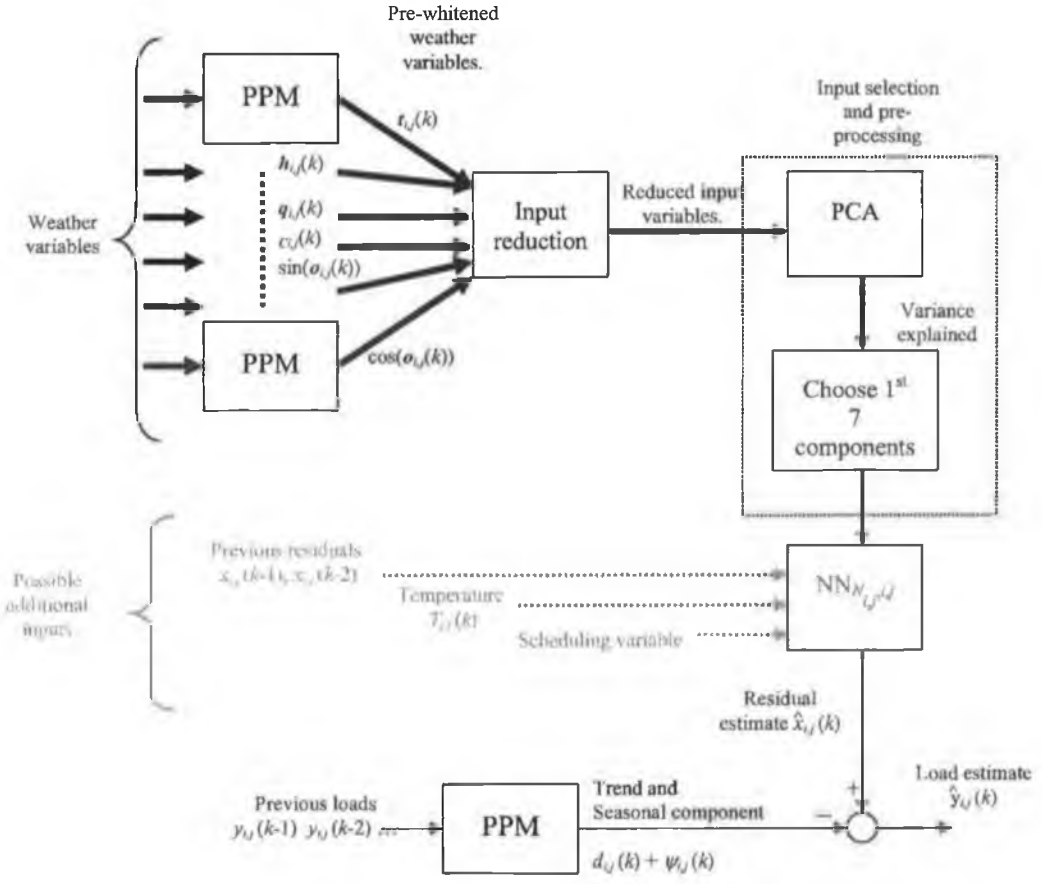

Figure 5.29. Non-linear parallel model overview

5.7.1 Choice of Non-Linear Model.

In Section 3.4 several non-linear modelling techniques are reviewed:

1. Volterra models (Section 3.4.1),
2. Wiener models (Section 3.4.1.1),
3. Hammerstein models (Section 3.4.1.2),
4. Fuzzy logic models (Section 3.4.2.1),
5. Radial Basis Function (RBF) neural networks (Section 3.4.2.2),
6. Feed forward neural networks (Section 3.4.3.1), and
7. Recurrent neural networks (Section 3.4.3.2).

The Wiener model and the Hammerstein model are based on the Volterra model. As pointed out in Section 3.4.1, calculating the Volterra kernels in a Volterra model can be difficult and in addition, they may not converge (Section 3.4.1). Thus a parametric non-linear technique (i.e. techniques one to three above) is not considered.

The fuzzy logic and RBF models suffer from "the curse of dimensionality" as mentioned in Sections 3.4.2, 3.4.2.1 and 3.4.2. Specifically, Mitchell (1992) notes that the use of RBF networks is impractical for more than six inputs. As the optimum number of input variables (for forecasting the load residuals in the current research) determined in Section 5.5 exceeds six in many cases, RBF networks and fuzzy logic techniques are eliminated from the choice of non-linear modelling technique.

In choosing between the two remaining techniques (feed forward neural networks and recurrent neural networks) the following factors must be considered:

1. The partitioning of the data has produced partitioned sets which vary in size from 141 to 1382 (Table 5.10),

2. The residual to be modelled (produced by the PPM) is stationary, and

3. There are 216 partitioned series (24 partitioned series per day-type and nine day-types) and thus 216 non-linear parallel models are required. This is a large number of models and thus the computational expense of the non-linear modelling technique must be considered.

With regard to the first point above, the number of data points in each partitioned series cannot be considered abundant. While both a recurrent neural network and a feed forward neural network would require large data sets for training, recurrent neural networks generally require larger data sets than feed forward neural networks (Section 3.4.3.2).

A recurrent neural network has the ability to model non-stationary time-series. However the second point above means that this ability is not required by the non-linear technique to be chosen. This is a disadvantage for reasons explained in Section 3.4.3.2.

With respect to point three above, recurrent neural networks generally require more computation time than feed forward neural networks (Section 3.4.3.2).

Thus the feed forward neural network appears to be the best choice of non-linear modelling technique in this case, and so is chosen. Although a feed forward neural network is computationally less expensive than a recurrent neural network, it is still quite time consuming to train. Thus there are restrictions on the number of variations in the non-linear parallel models that can be attempted.

5.7.2 Input Selection and Pre-Processing.

Ideally, neural networks with all possible combinations of the weather inputs should be constructed and the best selected. However, this is not possible, as the computational expense of a neural network prohibits more than a few combinations being tested. However, the input selection techniques used in Section 5.5 identified the optimum number of weather components for a *linear*

parallel model. Though this analysis is not representative of the full complexity of the system (specifically non-linear effects), it is more than sufficient to determine the relative importance of the weather inputs. To summarise the selection input procedure used in Section 5.5, the pre-whitened weather inputs are reduced to 80 inputs, transformed using PCA and this produces 80 pre-whitened weather *components*. The components are then ordered by variance explained.

The computational expense of training 216 neural networks will be seen in the next section (Section 5.7.3) to constrict the *structure* of the neural networks in the non-linear parallel models to be the same. This in turn requires that the number of pre-whitened weather component inputs for each neural network is the same. From Figure 5.25 the most popular number of pre-whitened weather components used as inputs is 7-8. Thus the number of pre-whitened weather components used in each non-linear model is thus chosen to be seven.

In addition to the pre-whitened weather components, the following additional inputs are considered:

1. Auto-regressive inputs (i.e. previous residuals). As mentioned in Section 3.4, there is evidence to show that a *non-linear* auto-regressive relationship exists in load data in other utilities. As there is no *linear* auto-regressive information left in the data, there is no way of choosing which auto-regressive residuals to include. The input reduction procedure used with the weather variables suggests however, that only the previous two days of weather variables are significant (Section 5.5.1). By extension, it is assumed that the previous two days of residuals are significant and so these are used as additional inputs,

2. Temperature at the hour to be forecast[%]. In Section 4.2.2 it was shown that the temperature-load *sensitivity* in April and May is significantly different than in the period June-September. This means that the relationship between

[%] This should not be confused with pre-whitened temperature.

load and temperature is dependant on the level of the temperature during summer (and possibly other periods of the year). As the pre-whitened temperatures contain information regarding the change of temperature, as opposed to the level of the temperature, they are of no use in this case. Thus the temperature at the hour to be forecast is included as *operating point* information, and

3. A schedule variable. A plot of the residual for the 00 hrs late winter working day day-type, $x_{00,6}(k)$, versus the 1st pre-whitened weather component is shown in Figure 5.30 below. As can be seen there appear to be two clusters of data. The second cluster of data was identified as occurring on Mondays. Specifically, the effect is more pronounced during the morning and evening hours.

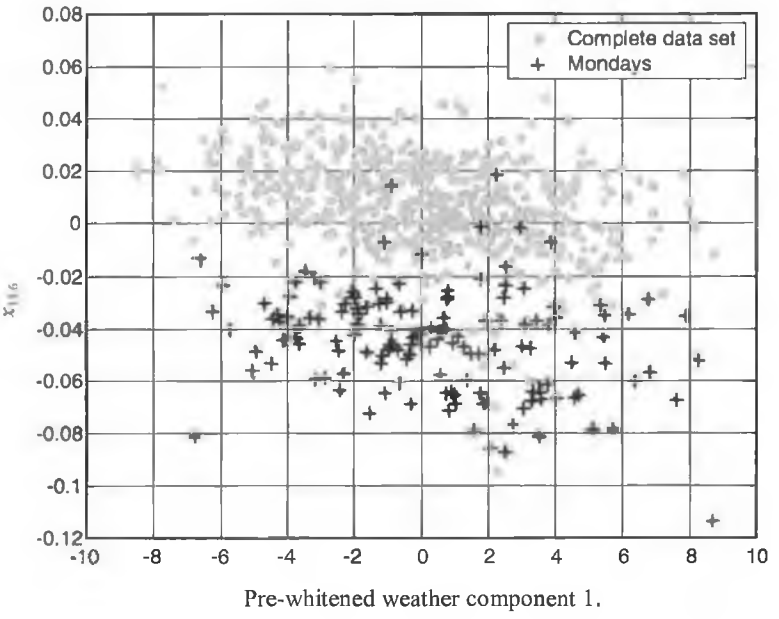

Figure 5.30. A plot of the residual versus the 1st pre-whitened weather variable showing two clusters of data. (00 hrs Late Winter Working Day day-type).

This implies that the relationship between the residual and the pre-whitened weather inputs is different on Mondays. To provide this information to the non-linear parallel models, a scheduling variable is provided for the working day day-types (as this effect is not present in the weekend day-types). This variable has a value of 1 for Mondays and a value of 0 elsewhere.

Several types of non-linear parallel models are trained, each differing by the types of inputs they use. In order to distinguish between these models, they are given the names shown in Table 5.19, below, where *i,j* denotes the model for hour *i* on day-type *j*.

Table 5.19. Non-linear parallel model naming convention.

Model Name	Inputs used.
$NN_{i,j}$	7 Pre-whitened weather components.
$NNAR_{i,j}$	7 Pre-whitened weather components, $x_{i,j}(k\text{-}1)$ $x_{i,j}(k\text{-}2)$
$NNART_{i,j}$	7 Pre-whitened weather components, $x_{i,j}(k\text{-}1)$ $x_{i,j}(k\text{-}2)$, $T_{i,j}(k)$
$NNS_{i,j}$	7 Pre-whitened weather components, Scheduling variable.
$NNARS_{i,j}$	7 Pre-whitened weather components, $x_{i,j}(k\text{-}1)$ $x_{i,j}(k\text{-}2)$, Scheduling variable.
$NNARTS_{i,j}$	7 Pre-whitened weather components, $x_{i,j}(k\text{-}1)$ $x_{i,j}(k\text{-}2)$, $T_{i,j}(k)$, Scheduling variable.

5.7.3 Structure Determination.

Determining the structure of a neural network may be broken down into 3 stages:

1. The number of hidden layers and the activation functions must be chosen (Section 5.7.3.1),
2. The training algorithm must be chosen (Section 5.7.3.2), and
3. The number of nodes in each layer must be determined (Section 5.7.3.3).

5.7.3.1 Choice of network architecture.

The number of layers used in the field of STLF was reviewed in Section 3.4.3.1. As three hidden layers are seldom used (Section 3.4.3.1), the choice lies between a single hidden layer or two hidden layers. The latter is chosen as Lee *et. al.* (1992) found that two hidden layers gave better performance for STLF (Section 3.4.3.1).

The activation functions chosen are *tan sigmoidal* for the hidden layers, and linear for the output layers, as they are the most common activation functions used in feed forward neural networks applied to STLF (Section 3.4.3.1). The input data is normalised between ± 1 so that the *tan sigmoid* activation functions are not driven into saturation (Chihocki and Unbehauen,1993), with a resulting

* This is the neural network for hour *i* on day-type *j*.

speed up in training, since the high gain (gradient) of the neuron characteristic is used.

5.7.3.2 Choice of Training Algorithm.

The training algorithm used is the back propagation algorithm, as a genetic algorithm is too slow to converge (Section 3.4.3.1). Ten neural networks were trained for each topology, with random initial weights to assist in achieving a global (or at least a good local) minimum (Section 3.4.3.1).

Each model is trained using early stopping (Haykin, 1999), in which training ceases when the sum squared error of predictions in the validation set reaches a minimum (an example is shown in Figure 5.31, Section 5.7.3.3). If a minimum is not found, the training stops after ten thousand epochs. Cessation of training at the validation set minimum prevents over-training of the neural network (Haykin, 1999) and assists, in conjunction with topology determination and input selection, in achieving a parsimonious network.

5.7.3.3 Network Topology Determination.

In order to determine the correct topology, 49 neural network architectures were examined, using 1-7 nodes in both the first and second hidden layers, respectively. To perform this for each partitioned series would require training 105,840 (49 architectures ×24 hours/day ×9 day-types ×10) neural networks, which is too computationally expensive. Thus the network topologies are refined for the 13 hrs and 07 hrs late winter working day partitioned series (as representative models) and applied to the others. In addition the NNART and NNARTS models network topologies are refined for the 13 hrs and 07 hrs summer working day partitioned series (as representative models) and applied to the others. This is because the operating point input (i.e. the temperature) in these models (Section 5.7.2) is included, as is thought to be especially relevant in modelling the load during *summer* months (Section 4.2.2). Table 5.20 (next page) shows the results for the model $NN_{07,6}$*.

* (i.e. the 07 hrs Late Winter Working Day partitioned series)

Table 5.20 MAPEs for $NN_{07,6}$ using differing numbers of nodes in hidden layers. (MAPE trn,val and nov are the MAPEs in the training, validation and novelty sets respectively.)

Structure	Layer 2 Nodes:	1				2				3				4				5				6				7			
Layer 1 Nodes	Network Number	Mape Trn	Mape Val	Mape Nov	Index	Mape Trn	Mape Val	Mape Nov	Index	Mape Trn	Mape Val	Mape Nov	Index	Mape Trn	Mape Val	Mape Nov	Index	Mape Trn	Mape Val	Mape Nov	Index	Mape Trn	Mape Val	Mape Nov	Index	Mape Trn	Mape Val	Mape Nov	Index
1	1	3.45	2.94	3.36	9983	3.44	2.94	3.38	5629	3.45	2.95	3.38	9967	3.43	3.00	3.44	4595	3.58	3.15	3.60	9936	3.70	3.23	3.65	2196	3.51	3.00	3.52	9997
	2	3.44	2.94	3.36	9556	3.43	2.94	3.36	4730	3.44	2.95	3.36	7364	3.44	2.96	3.38	4409	3.44	3.00	3.51	2338	3.70	3.23	3.65	6993	3.47	2.96	3.38	9989
	3	3.44	2.94	3.37	1487	3.43	2.94	3.36	9918	3.45	2.94	3.38	5993	3.45	2.95	3.38	9997	3.42	2.97	3.42	981	3.70	3.23	3.65	5549	3.44	2.95	3.37	9939
	4	3.41	2.93	3.36	9696	3.44	2.94	3.35	3712	3.44	2.94	3.38	5968	3.43	2.95	3.38	3067	3.43	2.97	3.40	8826	3.64	3.18	3.64	1486	3.44	2.94	3.38	761
	5	3.43	2.93	3.36	9977	3.41	2.93	3.36	8415	3.44	2.94	3.36	3476	3.44	2.94	3.36	6325	3.42	2.97	3.43	1485	3.47	2.98	3.41	2613	3.43	2.94	3.36	5368
	6	3.44	2.93	3.35	3755	3.43	2.93	3.36	9808	3.44	2.94	3.36	9924	3.45	2.94	3.40	2782	3.44	2.95	3.36	9822	3.49	2.97	3.39	3463	3.44	2.94	3.36	9872
	7	3.43	2.93	3.36	9868	3.43	2.93	3.36	1627	3.44	2.94	3.37	9968	3.43	2.93	3.35	7378	3.46	2.94	3.37	4373	3.46	2.95	3.37	4786	3.43	2.94	3.36	9688
	8	3.43	2.93	3.35	6508	3.44	2.93	3.38	728	3.44	2.93	3.35	2136	3.43	2.93	3.36	9406	3.44	2.94	3.37	9949	3.44	2.94	3.36	4822	3.43	2.93	3.35	3745
	9	3.46	2.93	3.40	966	3.43	2.93	3.35	828	3.44	2.93	3.35	2555	3.43	2.93	3.35	1470	3.43	2.93	3.36	9963	3.43	2.94	3.36	9640	3.44	2.93	3.34	5806
	10	3.44	2.93	3.38	861	3.43	2.93	3.41	1222	3.43	2.92	3.38	1472	3.38	2.91	3.68	48	3.44	2.91	3.38	4698	3.43	2.93	3.36	9867	3.45	2.91	3.40	2492
	Average	3.44	2.93	3.37		3.44	2.93	3.37		3.44	2.93	3.36		3.44	2.93	3.44		3.44	2.93	3.42		3.44	2.94	3.36		3.44	2.93	3.36	
2	1	3.38	2.98	3.42	9955	3.44	2.96	3.45	625	3.69	3.22	3.63	9904	3.40	2.96	3.39	9897	3.39	3.02	3.46	9893	3.50	2.97	3.41	9950	3.66	3.24	3.62	9960
	2	3.38	2.95	3.36	9975	3.43	2.95	3.37	1481	3.39	2.97	3.45	9639	3.43	2.95	3.36	6233	3.43	3.01	3.49	9959	3.40	2.96	3.45	9963	3.70	3.23	3.68	9914
	3	3.41	2.95	3.39	4628	3.43	2.94	3.36	9833	3.42	2.96	3.40	6915	3.39	2.95	3.42	9946	3.42	3.01	3.43	9974	3.43	2.95	3.39	4909	3.43	2.99	3.43	9896
	4	3.43	2.94	3.36	6086	3.43	2.93	3.35	2922	3.43	2.95	3.38	1375	3.43	2.94	3.37	7573	3.36	2.97	3.46	9933	3.43	2.95	3.34	8044	3.42	2.98	3.46	9953
	5	3.44	2.94	3.36	2411	3.43	2.93	3.35	5729	3.43	2.94	3.37	9978	3.40	2.93	3.35	6011	3.46	2.95	3.38	2958	3.39	2.95	3.46	3491	3.38	2.98	3.44	9998
	6	3.41	2.93	3.40	9924	3.40	2.92	3.35	1137	3.43	2.93	3.35	9964	3.41	2.93	3.34	9854	3.42	2.95	3.39	6586	3.38	2.95	3.42	9971	3.46	2.97	3.41	9992
	7	3.44	2.93	3.31	585	3.39	2.92	3.42	9938	3.43	2.93	3.36	7140	3.38	2.93	3.46	9994	3.43	2.94	3.37	9883	3.46	2.94	3.34	2725	3.46	2.95	3.37	10000
	8	3.44	2.93	3.39	5882	3.41	2.92	3.40	4915	3.43	2.92	3.36	9861	3.40	2.92	3.41	9970	3.43	2.91	3.40	1335	3.44	2.94	3.40	510	3.44	2.94	3.33	2547
	9	3.40	2.92	3.40	9728	3.43	2.91	3.40	2350	3.40	2.91	3.40	9607	3.43	2.92	3.38	2547	3.40	2.90	3.38	8229	3.43	2.93	3.36	9950	3.44	2.93	3.45	1085
	10	3.46	2.95	3.35	995	3.40	2.91	3.34	4713	3.45	2.90	3.38	7114	3.40	2.90	3.38	9995	3.43	2.89	3.43	6496	3.41	2.92	3.38	7961	3.38	2.90	3.42	8441
	Average	3.41	2.93	3.38		3.41	2.93	3.38		3.41	2.91	3.37		3.40	2.92	3.41		3.42	2.91	3.40		3.44	2.93	3.37		3.43	2.93	3.40	
3	1	3.33	2.96	3.45	4244	3.38	2.99	3.44	6671	3.39	3.00	3.42	9893	3.41	2.95	3.41	9929	3.38	2.97	3.38	9971	3.82	3.31	3.70	9968	3.45	3.00	3.42	9966
	2	3.41	2.94	3.38	1545	3.37	2.97	3.41	9009	3.43	2.98	3.35	3610	3.40	2.95	3.39	9902	3.44	2.97	3.35	494	3.70	3.24	3.66	9898	3.44	2.96	3.37	3718
	3	3.42	2.94	3.37	3625	3.40	2.95	3.40	9270	3.45	2.95	3.34	3492	3.42	2.95	3.34	1594	3.38	2.95	3.40	4550	3.37	3.01	3.51	9952	3.36	2.96	3.50	9914
	4	3.41	2.94	3.38	5675	3.44	2.93	3.40	1428	3.44	2.94	3.37	9877	3.43	2.94	3.37	4118	3.43	2.95	3.37	5918	3.46	2.99	3.38	621	3.38	2.95	3.41	2292
	5	3.43	2.93	3.33	2259	3.41	2.93	3.38	4377	3.40	2.94	3.42	9960	3.40	2.94	3.37	5470	3.42	2.94	3.40	7697	3.44	2.97	3.39	1192	3.37	2.95	3.41	3485
	6	3.43	2.93	3.36	4845	3.43	2.92	3.39	1268	3.33	2.91	3.40	9946	3.45	2.94	3.36	6288	3.43	2.94	3.37	3539	3.37	2.95	3.41	9961	3.40	2.93	3.39	4527
	7	3.37	2.92	3.39	9088	3.44	2.92	3.31	833	3.42	2.93	3.38	2557	3.43	2.91	3.39	1998	3.39	2.91	3.48	6465	3.36	2.95	3.42	9968	3.46	2.92	3.19	234
	8	3.43	2.91	3.34	2956	3.40	2.91	3.40	6957	3.44	2.92	3.35	1942	3.38	2.90	3.39	9861	3.36	2.90	3.38	4453	3.44	2.95	3.38	4648	3.42	2.91	3.37	9922
	9	3.41	2.91	3.36	3720	3.39	2.90	3.37	4757	3.35	2.92	3.44	10000	3.39	2.88	3.35	3558	3.39	2.90	3.45	9956	3.37	2.94	3.35	9933	3.45	2.90	3.42	541
	10	3.43	2.90	3.37	1264	3.40	2.90	3.39	4344	3.55	2.87	3.28	83	3.36	2.86	3.42	6345	3.51	2.89	3.33	160	3.45	2.90	3.40	2774	3.39	2.88	3.41	9998
	Average	3.41	2.93	3.38		3.41	2.91	3.37		3.44	2.91	3.36		3.39	2.90	3.39		3.40	2.90	3.41		3.41	2.94	3.39		3.43	2.91	3.40	
4	1	3.39	2.96	3.42	9906	3.39	2.97	3.35	9970	3.47	2.98	3.42	739	3.45	3.01	3.35	9986	3.36	2.99	3.45	9902	3.45	3.08	3.41	9969	3.35	2.99	3.44	8015
	2	3.43	2.95	3.37	9999	3.33	2.96	3.50	9917	3.43	2.96	3.38	5078	3.35	2.98	3.47	9911	3.42	2.96	3.38	3183	3.37	3.01	3.41	9942	3.43	2.98	3.37	4938
	3	3.34	2.94	3.47	9968	3.35	2.96	3.45	9976	[illegible]	[illegible]	[illegible]	[illegible]	3.33	2.97	3.45	8355	3.43	2.96	3.36	1385	3.39	3.01	3.47	4139	3.35	2.98	3.43	9963
	4	3.41	2.94	3.36	3441	3.38	2.95	3.38	9928	3.44	2.96	3.38	1786	3.36	2.96	3.35	9968	3.36	2.96	3.46	9967	3.39	3.00	3.39	9252	3.44	2.97	3.42	506
	5	3.43	2.94	3.34	4257	3.36	2.94	3.41	9917	3.41	2.96	3.38	9888	3.35	2.95	3.38	9990	3.38	2.95	3.41	9949	3.45	2.95	3.39	1463	3.37	2.97	3.40	7307
	6	3.41	2.93	3.37	3305	3.46	2.94	3.42	735	3.50	2.95	3.29	117	3.41	2.95	3.39	1319	3.43	2.94	3.35	1085	3.41	2.93	3.45	3671	3.44	2.94	3.45	9961
	7	3.43	2.93	3.38	1681	3.45	2.94	3.35	1631	3.32	2.94	3.43	9963	3.40	2.95	3.40	8429	3.37	2.93	3.43	4760	3.41	2.93	3.16	3690	3.40	2.93	3.39	9982
	8	3.43	2.93	3.34	4818	3.41	2.92	3.33	1767	3.35	2.93	3.38	5184	3.41	2.94	3.36	4784	3.43	2.93	3.43	519	3.45	2.89	3.40	976	3.43	2.93	3.44	523
	9	3.46	2.92	3.40	471	3.46	2.92	3.35	985	3.43	2.93	3.38	2011	3.42	2.94	3.38	2409	3.46	2.92	3.36	450	3.35	2.89	3.40	9610	3.39	2.93	3.37	4346
	10	3.41	2.91	3.35	4583	3.44	2.92	3.41	1826	3.44	2.92	3.39	1498	3.35	2.94	3.44	9941	3.34	2.92	3.49	9987	3.55	2.89	3.14	87	3.41	2.92	3.35	4230
	Average	3.43	2.93	3.38		3.44	2.93	3.38		3.38	2.93	3.38		3.39	2.94	3.39		3.41	2.93	3.41		3.44	2.90	3.38		3.41	2.93	3.39	
5	1	3.40	3.00	3.40	9920	3.70	3.23	3.65	9991	3.71	3.23	3.65	10000	3.44	3.05	3.44	9996	3.52	3.02	3.55	9944	3.57	2.97	3.44	99	3.40	2.98	3.43	2254
	2	3.39	2.99	3.37	9943	3.44	2.97	3.39	9915	3.34	2.99	3.46	9899	3.45	3.00	3.41	9947	3.30	2.99	3.50	9944	3.36	2.95	3.40	9916	3.38	2.97	3.47	3839
	3	3.44	2.97	3.39	703	3.38	2.95	3.44	9884	3.35	2.99	3.41	7232	3.35	2.99	3.49	9999	3.38	2.97	3.38	9975	3.34	2.93	3.47	9956	3.37	2.96	3.41	871
	4	3.42	2.94	3.37	1629	3.41	2.95	3.40	3006	3.37	2.95	3.43	9961	3.38	2.96	3.39	9969	3.31	2.96	3.42	9933	3.44	2.93	3.37	1487	3.40	2.95	3.43	1786
	5	3.36	2.94	3.41	9879	3.34	2.95	3.43	9971	3.35	2.94	3.39	3118	3.38	2.95	3.40	6234	3.34	2.95	3.44	8322	3.49	2.93	3.40	891	3.46	2.94	3.32	728
	6	3.41	2.94	3.33	6865	3.36	2.94	3.42	9989	3.37	2.94	3.47	4075	3.42	2.94	3.36	1491	3.43	2.94	3.38	766	3.41	2.92	3.30	555	3.47	2.93	3.33	9992
	7	3.38	2.93	3.31	8082	3.41	2.92	3.35	1998	3.44	2.93	3.38	2357	3.36	2.93	3.45	3251	3.43	2.94	3.34	331	3.36	2.92	3.45	8765	3.42	2.93	3.42	1184
	8	3.43	2.92	3.37	7890	3.40	2.91	3.35	3459	3.39	2.91	3.37	1187	3.38	2.92	3.38	2381	3.38	2.93	3.40	1472	3.32	2.92	3.54	9978	3.33	2.93	3.41	9071
	9	3.44	2.91	3.42	836	3.45	2.90	3.39	495	3.39	2.90	3.41	7140	3.40	2.92	3.41	1621	3.43	2.92	3.36	9747	3.40	2.91	3.41	9963	3.37	2.92	3.46	3561
	10	3.39	2.91	3.46	2963	3.39	2.89	3.38	9997	3.39	2.87	3.34	5921	3.37	2.90	3.42	5498	3.41	2.88	3.38	1850	3.37	2.87	3.41	3871	3.34	2.91	3.35	316
	Average	3.41	2.95	3.39		3.41	2.94	3.37		3.40	2.94	3.38		3.38	2.97	3.42		3.43	2.92	3.37		3.36	2.91	3.45		3.41	2.93	3.42	
6	1	3.35	2.97	3.44	6413	3.40	2.98	3.41	9821	3.41	3.00	3.40	9943	3.39	2.98	3.39	9986	3.35	3.01	3.42	9963	3.45	3.05	3.43	9950	3.31	2.99	3.41	9968
	2	3.39	2.96	3.38	7443	3.31	2.99	3.52	9543	3.45	2.98	3.37	1481	3.39	2.96	3.42	9991	3.46	2.96	3.35	1347	3.36	2.99	3.40	5021	3.36	2.99	3.52	9906
	3	3.36	2.94	3.47	9923	3.41	2.94	3.39	5364	3.42	2.97	3.38	9999	3.34	2.95	3.45	9988	3.42	2.95	3.40	2745	3.42	2.98	3.37	9907	3.46	2.99	3.43	587
	4	3.36	2.94	3.37	7049	3.43	2.93	3.34	2283	3.37	2.97	3.43	9938	3.36	2.94	3.38	4414	3.38	2.94	3.42	9976	3.39	2.98	3.40	2246	3.35	2.99	3.44	9912
	5	3.45	2.94	3.36	880	3.39	2.92	3.40	5903	3.44	2.96	3.35	3209	3.42	2.93	3.38	962	3.40	2.94	3.38	3226	3.38	2.98	3.40	6706	3.37	2.98	3.51	9958
	6	3.46	2.93	3.37	866	3.44	2.91	3.35	3776	3.42	2.94	3.37	6817	3.34	2.93	3.38	6091	3.43	2.93	3.37	9947	3.41	2.95	3.36	4257	3.37	2.97	3.39	4230
	7	3.45	2.92	3.39	408	3.43	2.90	3.40	74	3.36	2.91	3.36	1630	3.38	2.93	3.41	9990	3.43	2.92	3.46	9997	3.43	2.93	3.41	1619	3.32	2.95	3.48	5389
	8	3.32	2.92	3.49	7356	3.34	2.90	3.42	3531	3.39	2.91	3.39	5626	3.42	2.91	3.37	1221	3.33	2.91	3.44	3330	3.32	2.90	3.46	6290	3.39	2.94	3.37	1382
	9	3.40	2.92	3.39	7464	3.35	2.89	3.35	6889	3.44	2.88	3.41	3070	3.47	2.91	3.32	100	3.41	2.91	3.37	3015	3.33	2.90	3.46	5713	3.39	2.92	3.40	3459
	10	3.43	2.91	3.35	4980	3.41	2.88	3.36	2095	3.43	2.86	3.37	92	3.49	2.87	3.38	107	3.37	2.90	3.41	4641	3.28	2.89	3.45	6152	3.38	2.90	3.44	4444
	Average	3.40	2.93	3.40		3.41	2.89	3.38		3.41	2.89	3.38		3.44	2.91	3.37		3.38	2.91	3.43		3.34	2.93	3.43		3.37	2.93	3.42	
7	1	3.38	2.94	3.37	3475	3.43	2.98	3.47	2118	3.40	2.98	3.40	7221	3.33	3.00	3.48	10000	3.28	3.01	3.51	8012	3.37	2.99	3.45	6186	3.57	3.13	3.51	9976
	2	3.46	2.94	3.44	732	3.41	2.98	3.41	2140	3.27	2.97	3.49	9973	3.32	2.97	3.46	5390	3.39	3.00	3.33	1994	3.32	2.98	3.48	9998	3.33	3.01	3.44	9983
	3	3.31	2.94	3.51	7205	3.34	2.97	3.39	9883	3.35	2.94	3.43	4357	3.37	2.96	3.37	741	3.36	2.98	3.41	7002	3.49	2.98	3.50	108	3.32	2.99	3.49	9926
	4	3.40	2.93	3.32	4422	3.39	2.96	3.36	1966	3.39	2.93	3.43	5483	3.37	2.95	3.45	9893	3.32	2.97	3.45	4793	3.46	2.96	3.50	9970	3.31	2.95	3.48	9967
	5	3.43	2.93	3.35	932	3.45	2.94	3.39	740	3.37	2.92	3.41	3545	3.36	2.95	3.42	3210	3.31	2.96	3.55	9979	3.34	2.95	3.45	8453	3.42	2.95	3.41	3687
	6	3.41	2.93	3.35	3808	3.44	2.94	3.37	387	3.38	2.92	3.45	9902	3.39	2.93	3.37	2643	3.41	2.96	3.42	1231	3.32	2.93	3.39	8490	3.34	2.95	3.35	5370
	7	3.45	2.92	3.38	456	3.37	2.93	3.40	2158	3.38	2.92	3.34	1911	3.43	2.93	3.35	1269	3.39	2.94	3.43	2606	3.37	2.93	3.43	8544	3.37	2.94	3.44	5788
	8	3.46	2.92	3.37	853	3.39	2.89	3.36	4460	3.41	2.91	3.33	9970	3.30	2.92	3.38	7187	3.43	2.92	3.48	505	3.31	2.92	3.46	9969	3.37	2.93	3.42	3477
	9	3.43	2.90	3.37	1447	3.32	2.87	3.32	59	3.31	2.88	3.34	81	3.38	2.91	3.36	2239	3.53	2.91	3.33	422	3.36	2.92	3.37	2218	3.43	2.92	3.38	1112
	10	3.41	2.90	3.33	5779	3.42	2.87	3.33	1846	3.45	2.87	3.39	111	3.35	2.85	3.27	42	3.32	2.89	3.50	4732	3.31	2.87	3.38	5683	3.36	2.84	3.43	4143
	Average	3.40	2.91	3.36		3.47	2.89	3.39		3.49	2.90	3.35		3.46	2.90	3.34		3.42	2.91	3.43		3.34	2.91	3.41		3.38	2.91	3.41	

Table 5.20 may be explained as follows. 'Layer 1 nodes' and 'Layer 2 nodes' refers to the number of nodes in hidden layers 1 and 2 respectively. *Network number* refers to the 10 networks that are trained for each topology. The MAPE in the training, validation and novelty sets is recorded for each network, as is the epoch number at which the training was stopped (called *Index* in Table 5.20). The 10 networks trained for each topology are sorted in descending order, with the network that achieved the lowest MAPE in the validation set last. The four networks with the lowest MAPEs in the validation set are then retained (these are lightly shaded in Table 5.20). The other six networks are discarded as networks that failed to reach a good local minimum, due to their random initial conditions. The MAPEs of the retained networks are averaged to give the *Averaged MAPEs* (AMAPEs). This term is used to distinguish the AMAPE from the *MAPE* which is the MAPE achieved by an individual network. The AMAPE represents the performance of each topology.

To aid the explanation of Table 5.20, the MAPEs for the third network trained with a 4×3×1* topology are shaded diagonally in Table 5.20. The validation MAPE for this network is 2.96% which does not compare well with the best validation MAPE achieved with this topology (2.92%).

Figure 5.31 below shows an example of early stopping using the 4×4×1 topology.

* Note there is only one output and so the all the neural networks have a topology *×*×1.

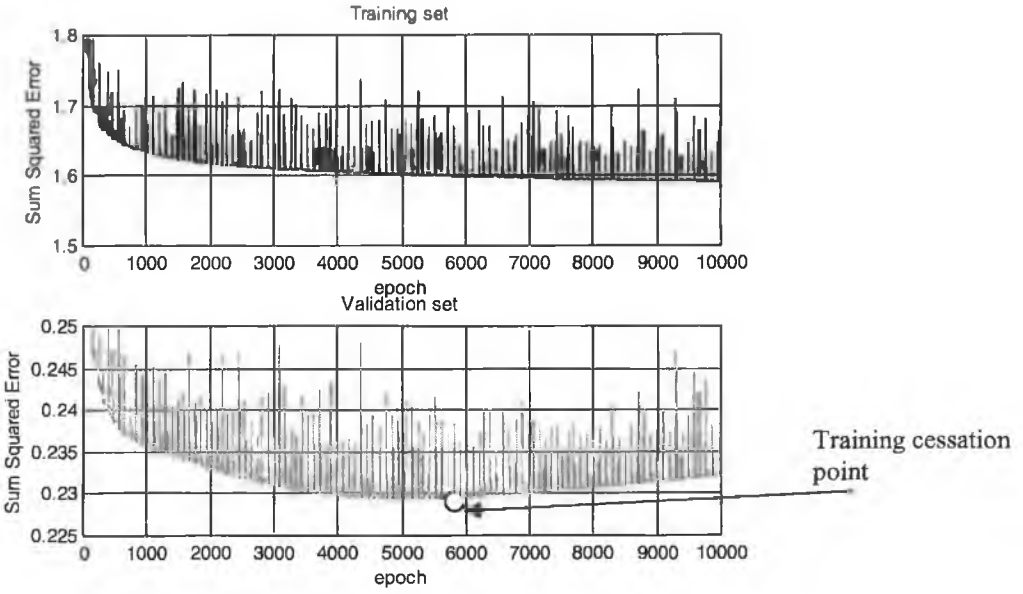

Figure 5.31 An example of early stopping (Topology: 4×4×1, Model: $NN_{07,6}$).

Method for Choosing the Best Topology

Ideally, the optimum topology is that with the lowest AMAPE in the validation set (Table 5.20). However, this approach is only valid if the validation set AMAPEs are exact. However, they are *estimates* of the AMAPE in the validation set. Thus some caution must be exercised in interpreting the results in Table 5.20. The following factors are useful to consider prior to determining the topology:

1. As the complexity (number of nodes) of the networks increase, their ability to over-fit the data increases (this general point is discussed in Section 3.3.1.4) and so the AMAPEs become less reliable estimates. Although there are only four MAPEs used to calculate the AMAPEs this point can still be readily demonstrated by plotting the variance of the AMAPEs against the complexity of the networks (Figure 5.32, below).

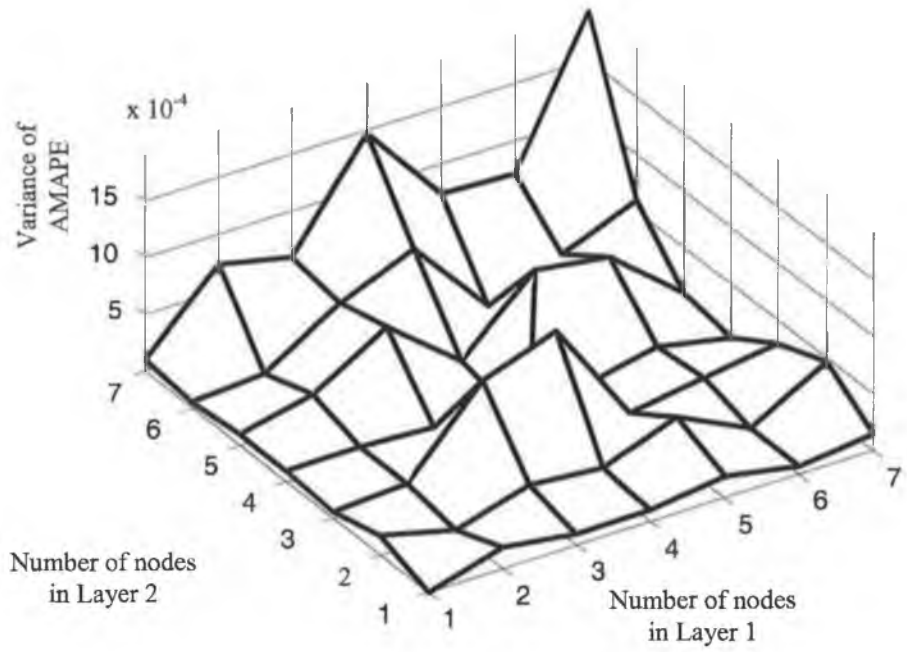

Figure 5.32 The variance of the validation set AMAPEs versus the number of nodes in each topology (Model: $NN_{07,6}$).

As can be seen the *variance* of the validation set AMAPEs rises with complexity and thus their *reliability* decreases*, and

2. Although the differences between two AMAPEs may appear small this is misleading. Table 5.20 for example, shows the AMAPEs achieved by two topologies, 1×1×1 and 3×4×1. The differences between the validation set AMAPEs for each topology in Table 5.20 appear small. However, each neural network is estimating the same *residual*, $x_{07,6}(k)$, which is produced by $PPM_{07,6}$ (Figure 5.29). That is, the forecast of the load produced by $PPM_{07,6}$ has a MAPE of 3.51% (Figure 5.11) and the validation set AMAPEs in Table 5.20 are *improvements* on this figure. As can be seen, the variance of the AMAPEs (Table 5.21) is quite small compared to the difference between the AMAPEs. Thus in this example topology 3×4×1 is considered superior to 1×1×1 although it is more complex and thus it's AMAPE is less reliable.

* This does not mean that topology 1×1×1 is necessarily the best topology; just that the estimated AMAPE for this topology is the most reliable.

Table 5.21 Validation set AMAPEs for $NN_{07,6}$ using 2 different topologies.

Topology	Validation set AMAPE (%)	Variance of validation set AMAPE
1×1×1	2.9320	4.85×10^{-7}
3×4×1	2.8926	1.01×10^{-4}

Considering the above factors, the method for choosing the optimum topology is:

1. Topologies with AMAPEs close to the lowest validation set AMAPE are first identified,

2. Topologies which are more complex than the topology with the minimum validation set AMAPE are eliminated,

3. A preference is shown towards simpler topologies, especially if the difference between topologies is large, and

4. Additionally, the AMAPEs in the training set are used to indicate whether the validation set AMAPEs are a random occurrence or are consistent with a network which has generalised well.

As an example, the AMAPEs for $NN_{07,6}$ in the training and validation sets are shown in Tables 5.22 and 5.23 below (extracted from Table 5.20).

Table 5.22 AMAPEs for $NN_{07,6}$ (Training set, best in bold italic font)

Number of Nodes	Layer 2: 1	2	3	4	5	6	7
Layer 1: 1	3.439	3.436	3.436	3.545	3.442	3.441	3.441
2	3.434	3.407	3.429	3.403	3.424	3.435	3.428
3	3.412	3.409	3.440	***3.389***	3.402	3.405	3.429
4	3.432	3.439	3.379	3.394	3.399	3.441	3.407
5	3.411	3.412	3.401	3.385	3.426	3.361	3.414
6	3.399	**3.408**	**3.405**	3.440	3.383	3.341	3.371
7	3.436	**3.474**	3.493	3.460	3.419	3.340	3.382

Table 5.23 AMAPEs for $NN_{07,6}$ (Validation set, best in bold italic font)

Number of Nodes	Layer 2: 1	2	3	4	5	6	7
Layer 1: 1	2.9320	2.9213	2.9314	2.9267	2.9331	2.9437	2.9266
2	2.9192	2.9162	2.9147	2.9187	2.9095	2.9343	2.9303
3	2.9117	2.9075	2.9092	***2.8926***	2.8953	2.9384	2.9039
4	2.9227	2.9249	2.9306	2.9419	2.9266	2.8979	2.9264
5	2.9199	2.9041	2.9031	2.9181	2.9172	2.9063	2.9231
6	2.9173	**2.8939**	**2.8921**	2.9050	2.9126	2.9043	2.9302
7	2.9083	**2.8879**	2.8952	2.9036	2.9148	2.9112	2.9109

The optimum topology is selected as follows:

1. As can be seen in Table 5.22, the minimum MAPE in the validation set is 2.88% using topology 7×2×1. Topologies 6×2×1, 6×3×1 and 7×2×1 have similar AMAPEs. These are indicated in bold in Tables 5.22 and 5.23,

2. There are other topologies with similar AMAPEs however they are more complex than 7×2×1 and are thus eliminated,

3. It is noted that topology 3×4×1 is simpler than 7×2×1 and that the difference between the validation set AMAPEs is only 0.0005%, and

4. The training set AMAPE for topology 3×4×1 is 0.016% better (a large figure in the current context) than that for topology 7×2×1. Thus topology 3×4×1 is chosen.

Table 5.24 below shows the topologies selected at the representative hours, for each of the models trained for topology determination.

Table 5.24 Selected topologies refined for each model at different hours and day-types.

Model	Hour	Day-type	Topology	MAPE (training set)	MAPE (validation set)
$NN_{07.6}$	07	Late winter	3×4×1	3.408	2.8939
$NN_{13.6}$	13	Late winter	4×7×1	1.842	1.503
$NNS_{07.6}$	07	Late winter	2×3×1	2.369	2.060
$NNS_{13.6}$	13	Late winter	6×3×1	1.808	1.502
$NNAR_{07.6}$	07	Late winter	5×3×1	3.293	2.788
$NNAR_{13.6}$	13	Late winter	3×1×1	1.894	1.552
$NNARS_{07.6}$	07	Late winter	3×4×1	2.216	1.983
$NNARS_{13.6}$	13	Late winter	6×1×1	1.827	1.501
$NNART_{07.6}$	07	Late winter	7×5×1	3.109	2.690
$NNART_{13.6}$	13	Late winter	3×6×1	1.844	1.517
$NNART_{07.5}$	07	Summer	6×3×1	4.104	3.511
$NNART_{13.5}$	13	Summer	6×1×1	1.583	1.417
$NNARTS_{07.6}$	07	Late winter	6×3×1	2.182	1.861
$NNARTS_{13.6}$	13	Late winter	5×7×1	1.800	1.477
$NNARTS_{07.5}$	07	Summer	6×6×1	2.734	2.424
$NNARTS_{13.5}$	13	Summer	5×1×1	1.573	1.401

5.7.4 Parameter Evaluation.

The model parameters are evaluated using the back-propagation algorithm as discussed in Section 5.7.3.2.

5.7.5 Model Validation.

Table 5.24 shows 16 topologies calculated at representative hours for the six types of non-linear parallel model (Table 5.19). The next step is to train neural networks for all hours of all day-types (excluding Christmas as usual) using the 16 topologies listed in Table 5.24 and the appropriate model type.

Tables 5.25 and 5.26 shows the daily AMAPE achieved using the topologies calculated in Section 5.7.3.3 (Table 5.24) for all day-types in the validation and novelty sets respectively.

Table 5.25 Daily AMAPE (%) for each model using selected topologies (validation set, the best model in each day-type is shown in bold italic font. Note: the scheduling variable is only applicable to working days).

Model	Topology	Early winter Sundays	Summer Sundays	Late winter Sundays	Early winter working days	Summer working days	Late winter working days	Early winter Saturdays	Summer Saturdays	Late winter Saturdays
NN	3×4×1	2.54	3.02	2.69	2.69	3.02	2.47	2.62	2.91	2.37
NN	4×7×1	2.55	3.08	2.73	2.69	3.02	2.45	2.62	2.92	2.42
NNS	2×3×1	-	-	-	2.00	2.31	2.00	-	-	-
NNS	6×3×1	-	-	-	1.96	2.28	1.95	-	-	-
NNAR	5×3×1	2.53	3.01	2.50	2.60	2.98	2.44	2.58	2.91	2.32
NNAR	3×1×1	2.51	***2.98***	2.49	2.61	2.99	2.44	2.60	2.90	2.35
NNARS	3×4×1	-	-	-	1.97	2.37	1.92	-	-	-
NNARS	6×1×1	-	-	-	1.93	2.24	1.91	-	-	-
NNART	7×5×1	2.47	3.05	2.35	2.62	2.99	2.42	2.62	2.95	2.29
NNART	3×6×1	2.51	3.07	2.41	2.62	2.99	2.45	2.61	2.98	2.44
NNART	6×3×1	***2.41***	3.03	***2.32***	2.59	2.97	2.43	***2.52***	***2.90***	2.33
NNART	6×1×1	***2.41***	2.99	2.33	2.61	2.99	2.41	2.53	2.91	***2.28***
NNARTS	6×3×1	-	-	-	***1.92***	2.29	***1.89***	-	-	-
NNARTS	5×7×1	-	-	-	1.96	2.28	***1.89***	-	-	-
NNARTS	6×6×1	-	-	-	1.95	***2.23***	1.90	-	-	-
NNARTS	5×1×1	-	-	-	1.94	***2.23***	1.90	-	-	-

Table 5.26 Daily AMAPE (%) for each model using selected topologies (Novelty set, the best model in each day-type is shown in bold italic font. Note: the scheduling variable is only applicable to working days).

Model	Topology	Early winter Sundays	Summer Sundays	Late winter Sundays	Early winter working days	Summer working days	Late winter working days	Early winter Saturdays	Summer Saturdays	Late winter Saturdays
NN	3×4×1	2.73	3.19	2.75	2.31	2.99	2.69	2.19	2.71	2.76
NN	4×7×1	2.69	3.20	2.73	2.31	3.00	2.68	2.28	2.77	2.73
NNS	2×3×1	-	-	-	1.74	2.20	2.16	-	-	-
NNS	6×3×1	-	-	-	1.72	2.18	2.10	-	-	-
NNAR	5×3×1	2.71	3.15	2.70	2.30	2.93	2.66	2.12	2.60	2.69
NNAR	3×1×1	2.57	***3.13***	2.66	2.25	2.95	2.63	2.17	2.62	2.60
NNARS	3×4×1	-	-	-	1.72	2.25	2.02	-	-	-
NNARS	6×1×1	-	-	-	1.71	2.09	2.01	-	-	-
NNART	7×5×1	2.63	3.19	2.57	2.29	2.93	2.61	2.26	2.63	2.67
NNART	3×6×1	2.62	3.15	2.60	2.30	2.96	2.64	2.21	2.70	2.64
NNART	6×3×1	***2.56***	3.14	***2.44***	2.31	2.95	2.60	***2.08***	***2.61***	2.65
NNART	6×1×1	***2.55***	3.13	2.44	2.26	2.94	2.59	2.07	2.64	***2.55***
NNARTS	6×3×1	-	-	-	***1.70***	2.15	***1.98***	-	-	-
NNARTS	5×7×1	-	-	-	1.73	2.16	***2.01***	-	-	-
NNARTS	6×6×1	-	-	-	1.72	***2.09***	1.99	-	-	-
NNARTS	5×1×1	-	-	-	1.72	***2.08***	1.98	-	-	-

The first thing to note is that the scheduling variable is applicable only to working day-types. Thus there are no entries in the weekend day-types for the NNS, NNARS and NNARTS models.

For the working day day-types, the scheduling variable is beneficial in all cases. For example, the NNS models are superior to the NN models for late and early winter working days (Table 5.25), the NNARS models are superior to the NNAR models etc.

Figure 5.33 indicates the benefit of the scheduling variable at each hour of the day (as opposed to daily averages in Table 5.25). There is little difference between 10 hrs and 16 hrs. This implies that the scheduling variable is also of *no benefit* during normal working hours of the day.

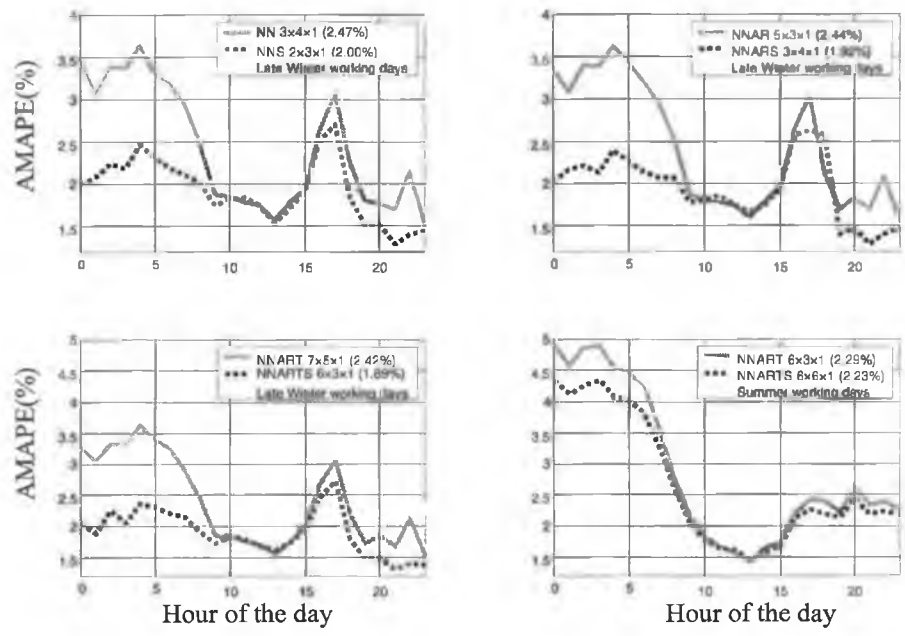

Figure 5.33 The AMAPE for non-linear models with and without the scheduling variable (Validation set, Models: differing types shown in legends).

For a given model type (NN, NNS etc.) the simpler topologies tend to give superior results (Table 5.25). Figure 5.34 shows the AMAPE achieved by using $NN_{07,6}$ with topologies 4×7×1 and 3×4×1 when applied to all hours of the day. Topology 4×7×1 is superior at 13 hrs and 3×4×1 is superior at 07 hrs, as these are the topologies optimised at these hours, respectively (Section 5.7.3.3, Table 5.24). However, there is not a consistent difference between their performances at other hours of the day. The conclusion here is that the best topology for one hour of the day is *not an indicator* of the best topology for *other* hours of the day. A similar situation applies to the topologies calculated for the other models listed in Table 5.25.

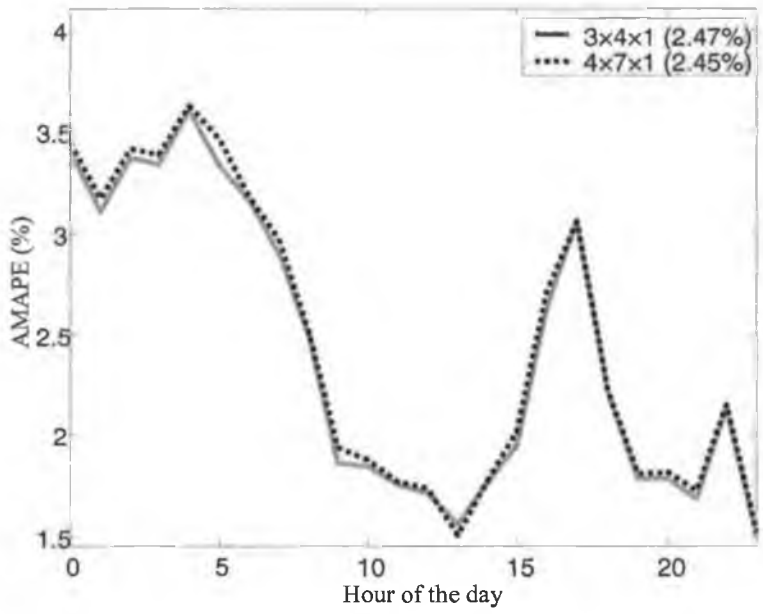

Figure 5.34 The AMAPE for topologies 3×4×1 and 4×7×1 versus the hour of the day (Validation set, Models: $NN_{00,6}$ to $NN_{23,6}$).

Comparing models with and without the AR inputs (e.g. NN and NNAR or NNS and NNARS), shows that the AR variables are beneficial in most cases (Table 5.25). Similarly, the operating point inputs are beneficial in most cases except summer Saturdays and Sundays (Table 5.25). This is a surprising result as the operating point input was thought to be especially useful for predicting summer loads (Section 4.2.2). The temperature-load relationship in summer is operating point dependant but the correlation between temperature and load is low (Section 4.2.2). This low correlation may override the advantage of using an operating point input for summer Saturdays and Sundays.

In summary, the analysis above proposes:

1. Simpler topologies are selected within each model type,
2. The scheduling variable is used (where applicable), but not at working hours (i.e. 10 hrs to 16 hrs) or during summer, and
3. The operating point input is beneficial except for summer Saturdays and summer Sundays.

5.8 Comparison of Linear and Non-linear parallel models

Table 5.27, below, shows the daily MAPEs achieved by the PPMs and linear parallel models in the novelty set. In addition the AMAPEs of the best non-linear parallel models chosen in Section 5.7.5 (Table 5.26) are shown. As can be seen, the parallel models are superior to the PPMs in all cases. The non-linear parallel models also give better forecasts in all cases, although the gap is small in the case of summer Sundays. It can therefore be concluded that there is a non-linear relationship between the load and the inputs chosen, justifying the use of a non-linear forecasting technique.

Table 5.27. Daily MAPEs for PPMs, Linear and non-linear parallel models (Christmas days excluded, novelty set).

Day-type	MAPE PPM's (%)	MAPE Linear Parallel Models (%)	MAPE Non-linear Parallel Models (%)
1. Early winter Sundays	2.75	2.70	***2.55***
2. Summer Sundays	3.22	3.14	***3.13***
3. Late winter Sundays	2.78	2.75	***2.44***
4. Early winter working days	2.36	2.25	***1.70***
5. Summer working days	3.04	2.99	***2.08***
6. Late winter working days	2.94	2.55	***1.98***
7. Early winter Saturdays	2.19	2.20	***2.08***
8. Summer Saturdays	2.83	2.81	***2.61***
9. Late winter Saturdays	3.06	2.86	***2.55***

5.9 Conclusion.

The linear and non-linear parallel models were presented in this chapter. In Section 5.3, a technique was presented to determine whether the parallel approach was justified based on the data alone. From this analysis, there is some evidence to support the view that some hours of the day have independent components, thus justifying the parallel approach. However, this evidence is not conclusive and requires comparison with a sequential approach for confirmation (Chapter 6).

The preliminary parallel models were constructed in Section 5.4. These models are based on the day-types identified in Chapter 4. However, in order to check that the day-types determined in Chapter 4 are the optimum means of partitioning the data, two other alternative partitions achieved through amalgamation were

examined (Section 5.4.2.3). It was found that the day-type partitions are superior in most cases, and significantly improved on partitioning the data by hour of the day alone (alternative partition 1, Section 5.4.2.3). However, alternative partition 2 (partitioning the data as Saturdays, Sundays and working days and by hour of the day) was found to give superior forecasts over the early winter periods. Thus, the optimum partition of the data is a mix between the day-type partition and alternative partition 2.

Section 5.4.2.2 examined two techniques for tuning the parameters in the PPMs, PED and SSD. It was found that there were negligible differences between the PPMs trained with both techniques. However, the main difference between SSD and PED is that the parameters may be chosen explicitly with SSD, making this technique more appealing in a practical sense.

Input selection was examined in Section 5.5. The first finding was that only up to two days of past (pre-whitened) weather was significant in forecasting load (Section 5.5.1). Four techniques were then examined for pre-processing, and selecting which of these weather inputs should be used. It was found that the techniques which used PCA (Methods 2, 3 and 4, Section 5.5.2) provided the best results. PCA uses the colinearity between the weather inputs to produce transformed inputs or components (Section 5.5.2.2). It can therefore be concluded that reducing the colinearity in model inputs aids in the modelling process. In addition, the techniques which order the components by variance explained (Methods 2 and 3) were superior to Method 4 which ordered the components via the T-ratio. The reason that variance explained is a good indicator of which components are important in load forecasting is explained below.

High frequency information in the original weather variables will tend to have a low cross-correlation. For example, the high frequency information in the temperature at 5 p.m. will be uncorrelated with the temperature at 4 p.m. (as this information changes at a high frequency). The variance explained orders the components by the amount of information in the original weather inputs that is present in the components. Thus, high frequency information will appear in

components ordered lower down the list. From the coherence plot in Section 2.3.3.2 (Figure 2.11) it can be seen that high frequency information in temperature has a *low* correlation with high frequency information in load. Thus there is a correspondence between the variance explained in each component and the correlation of each component with the load.

Section 5.6 presented the linear parallel models. It was seen that these models were superior to the PPMs and so it can be concluded that inclusion of external variables can improve load forecasts.

Section 5.7 presented the non-linear parallel models. The inclusion of three types of inputs, not present in the linear parallel models, was investigated: auto-regressive inputs, temperature (note: not the pre-whitened temperature) and a scheduling variable. The AR input was found to be beneficial in almost all cases and was present in all the best models selected in Table 5.25. Thus, the presence of a non-linear autoregressive component in Irish load (as suggested in Section 5.1) is confirmed. The inclusion of the other two inputs is dependent on the hour of the day and day-type.

Finally Section 5.8 compares the results for the linear and non-linear parallel models (Sections 5.6 and 5.7, respectively). It is found that the non-linear parallel models are superior to the linear parallel models in all cases, justifying the use of a non-linear technique in load forecasting.

Chapter 6

Multi-Timescale Modelling for Short-Term Load Forecasting

6.1 Introduction

This chapter outlines *and extends* the application of the Multi-TimeScale (MTS) technique used by Murray (1996, Section 3.3.4.2 and henceforth referred to as the MTS technique for clarity) to the problem of short term load forecasting.

The MTS approach is based on combining parallel models with sequential models, as discussed in Sections 3.2.2 and 3.3.4.2. The advantages of this approach are discussed in Section 3.3.4.2. As discussed in Section 3.2.2, there is no consensus in the literature regarding a choice between the parallel and sequential approaches. Thus, another motivation for using this approach is to draw a comparison with the parallel approach used in Chapter 5.

As pointed out by Murray (1996), the MTS technique may be applied in three steps:

1. Develop a lower-timescale model or Sequential load forecasting Model (SM) (Section 6.2),

2. Generate forecasts of the cardinal points and end-sum points (Section 6.3) and

3. Combine the SM with the cardinal point and end-sum forecasts using the MTS technique. The MTS technique requires choosing the freed states and the appropriate weight matrices (see Section 3.3.4.2). The freed states are chosen using a method proposed by Murray. Two new methods are proposed here for determining the weight matrices; a numerical approach (Section 6.4.1) and a deterministic approach (Section 6.4.2).

As in Chapter 5, the late winter working day-type is again used as an indicator of other day-types. Differences between the results for day-types will be mentioned as appropriate.

Note: It is not practical to index the models in this chapter in the same manner as the parallel models in Chapter 5. Specifically, the variables in the models used in this chapter are indexed with the number of hours from the start of the data, k, as opposed to indexing by hour i on day k as in the case of the parallel models (Chapter 5). For example, the notation for the load is now $y_j(k)$, for the load k hours form the start of the data for day-type j. In Chapter 5 the notation for the load was $y_{i,j}(k)$, where i was the hour of the day, j was the day-type and k was the k^{th} day in partitioned series j. Note: that this is because the MTS approach only disaggregates the load by day-type.

Finally; note that the load data used in this chapter is disaggregated by day-type, but not by hour of the day.

6.2 The Sequential Model.

As a linear state space model structure is required for the SM (Section 3.3.4.2), a BSM using a DPRW (Section 3.3.4.2) is selected. As explained in Section 3.3.2.3 a BSM is ideal for modelling short-term load. The BSM may be expressed as:

$$y_j(k) = d_j(k) + \psi_j(k) + \varepsilon_j(k) \tag{6.1}$$

where $d_j(k)$ is the trend component, $\psi_j(k)$ is the seasonal component and $\varepsilon_j(k)$ is the SM error (or residual, although *error* is used here to distinguish that the residual will not be forecast as in the parallel models) k hours from the start of the sequential time-series, for day-type j. Also, the SM for day-type j is referred to as SM_j.

As with the preliminary linear parallel model (also a BSM) in Section 4.1, the trend component is modelled using an Integrated Random Walk (IRW) (Section 3.3.2.2), with the seasonal component modelled using a Differenced Periodic Random Walk (DPRW) (Section 3.3.2.2) as:

$$\boldsymbol{\theta}_j(k) = \begin{bmatrix} d_j(k) \\ \dot{d}_j(k) \\ \hline \psi_j(k) \\ \psi_j(k-1) \\ \vdots \\ \psi_j(k-(24-2)) \end{bmatrix} = \left[\begin{array}{cc|cccccc} 1 & 1 & 0 & 0 & . & . & 0 \\ 0 & 1 & 0 & 0 & . & . & 0 \\ \hline 0 & 0 & -1 & -1 & . & . & -1 \\ 0 & 0 & 1 & 0 & . & . & 0 \\ . & . & . & . & . & . & 0 \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & . & 1 & 0 \end{array}\right] \begin{bmatrix} d_j(k-1) \\ \dot{d}_j(k-1) \\ \hline \psi_j(k-1) \\ \psi_j(k-2) \\ \vdots \\ \psi_j(k-(24-1)) \end{bmatrix} + \begin{bmatrix} 0 \\ \eta_{d_j}(k-1) \\ \hline \eta_{\psi_j}(k-1) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{6.2}$$

where $\boldsymbol{\theta}_j(k)$ is the state vector, $\dot{d}_j(k)$ is the rate of change of the trend, k hours from the start of the sequential time series for day-type j, and the other notation is explained below. The seasonal length in this case is 24, since there are 24 hours in each day. $\eta_{d_j}(k)$ and $\eta_{\psi_j}(k)$ are white noise components with

variances $\sigma^2_{d_{i,j}}$ and $\sigma^2_{\psi_{i,j}}$, respectively. In addition there is a measurement error term $\varepsilon_j(k)$ (Section 3.3.2.4) with a variance of $\sigma^2_{\varepsilon_j}$.

Figure 6.1 below summarises the application of the BSM to the sequential model.

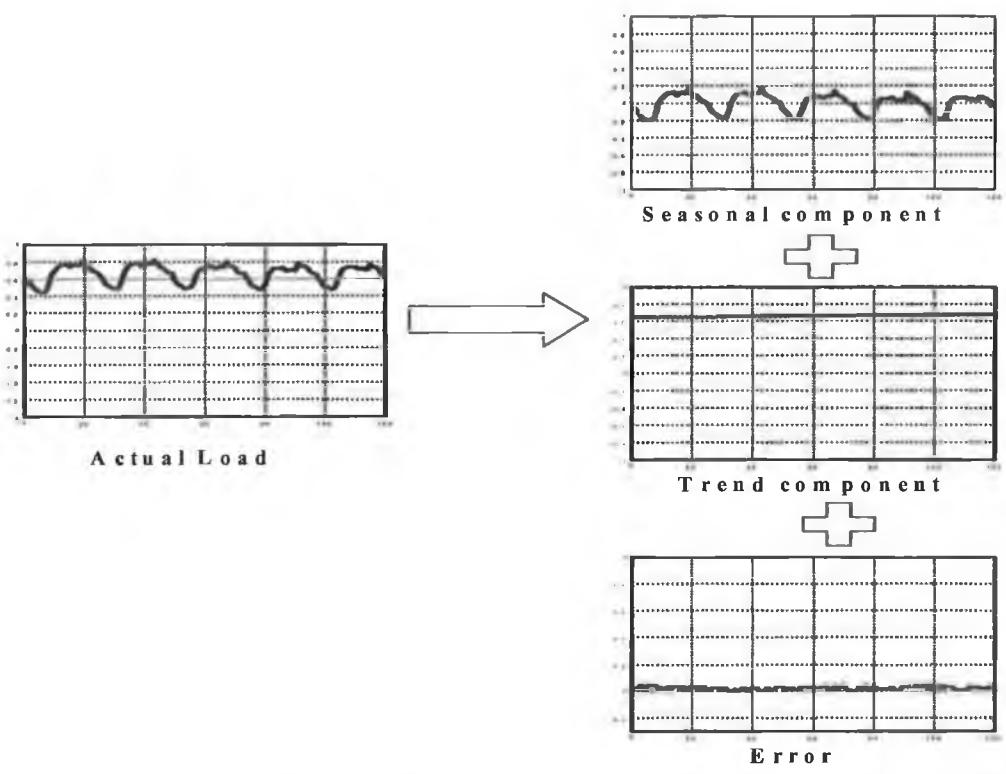

Figure 6.1 Overview of how the SM decomposes the load into seasonal and trend components.

As with the other models presented in this thesis, the modelling process for the SM may be split into four stages:

1. Input Selection and Pre-processing (Section 6.2.1),
2. Structure Determination (Section 6.2.2),
3. Parameter Evaluation (Section 6.2.3), and
4. Model Validation (Section 6.2.4).

6.2.1 Input Selection and Pre-processing.

The SM is an Auto-Regressive (AR) model and has no external inputs.

6.2.2 Structure Determination.

The structure of the SM is given by the state transition matrix in Equation (6.2), above. This is a fixed structure and so no structure determination is required.

6.2.3 Parameter Evaluation.

As with the PPM (Section 5.4.2), the SM also uses a BSM. Again, the variances of the noise components in the BSM, $\sigma^2_{\dot{d}_j}$ and $\sigma^2_{\psi_j}$ and $\sigma^2_{\varepsilon_j}$ are the parameters that must be determined. The SSD (Section 5.4.2.2) method is used and is summarised below.

The initial values of the state vector are shown in Table 6.1, below.

Table 6.1. Initialisation values for the state vector with SSD.

State	Value
$d_j(0)$	$y_j(0)$
$\dot{d}_j(0)$	0
$\psi_{1j}(0),\ldots,\psi_{s-1j}(0)$	0

As in Section 5.4.2.2, $\sigma^2_{r_j}$ is set to one and $\sigma^2_{a_j}$ is set to zero. $\sigma^2_{\dot{d}_j}$ is again not calculated *via* the periodogram as suggested in Section 3.3.2.4, as it may easily be specified as occurring at a frequency less than the yearly frequency. For example, there are 1272 (53×24) load points per year in the late winter working day day-type (i.e. for $j = 6$). The cut-off frequency, f_{50}, (Section 3.3.2.4) lies at a point lower than the corresponding yearly period. That is:

$$f_{50} < 1/1272 \tag{6.3}$$

Taking f_{50} to be half the yearly frequency and substituting this value into Equation (3.54) gives:

$$\sigma^2_{\dot{d}_6} = 1605(1/(1272\times 2))^4 = 3.83\times 10^{-11} \tag{6.4}$$

The value of $\sigma^2_{d_j}$ may be calculated similarly for the other SMs (i.e. for j = 1,…,9), noting that the number of data points per year varies depending on the day-type.

$\sigma^2_{d_j}$ is free to vary in order to minimise the log likelihood function as in Section 5.4.2.2. Table 6.2 below shows the parameters calculated for SM_6.

Table 6.2. Parameters of SM_6 (trained using SSD).

Parameter	Value
$\sigma^2_{x_6}$	1
$\sigma^2_{d_6}$	0
$\sigma^2_{\dot{d}_6}$	3.83×10^{-11}
$\sigma^2_{\psi_6}$	6.61×10^{-6}

6.2.4 Model Validation.

Figure 6.2 below shows the MAPE for each hour of the day made using a 1-hour ahead forecast, i.e. the 4 a.m. forecasts were made at 3 a.m., the 10 p.m. forecasts were made at 9 p.m. etc. The daily MAPE (i.e. the mean value of the graph in Figure 6.2) is 2.92%.

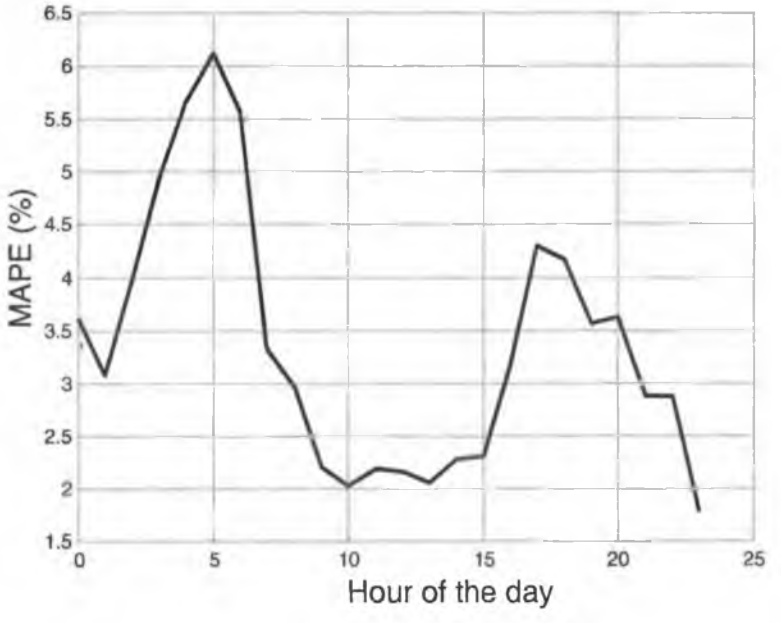

Figure 6.2. The MAPE as a function of hour of the day for SM_6. (1-hour ahead forecasts, training set)

Figure 6.3, below, shows the MAPE as a function of the hour of the day, for forecasts again made 1-step ahead and also 10-steps ahead. As can be seen, the forecast MAPEs are similar, but the 10-step ahead forecasts are slightly less accurate due to propagation error (Section 3.2.2). The daily MAPEs for the 1 and 10-step ahead forecasts are 2.92% and 3.04%, respectively.

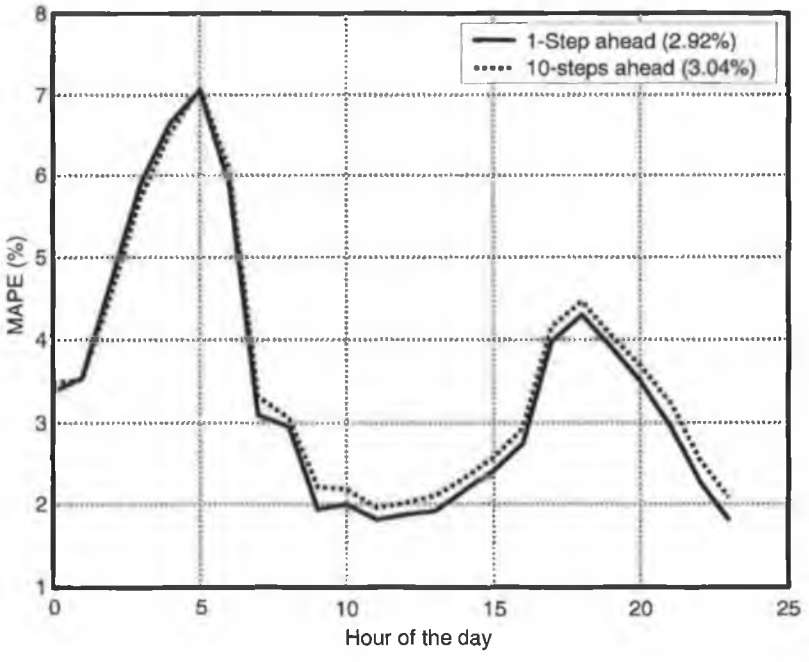

Figure 6.3. The MAPE as a function of hour of the day for SM_6. (1-hour and 10-hour ahead forecasts, training set)

Figure 6.4 below shows how the *daily* MAPE changes, with the forecast horizon from 1 to 24-steps ahead. As can be seen, the propagation error in this model accumulates. The minimum occurs at 1-step ahead (2.92%) and the maximum occurs at 24-steps ahead (3.19%).

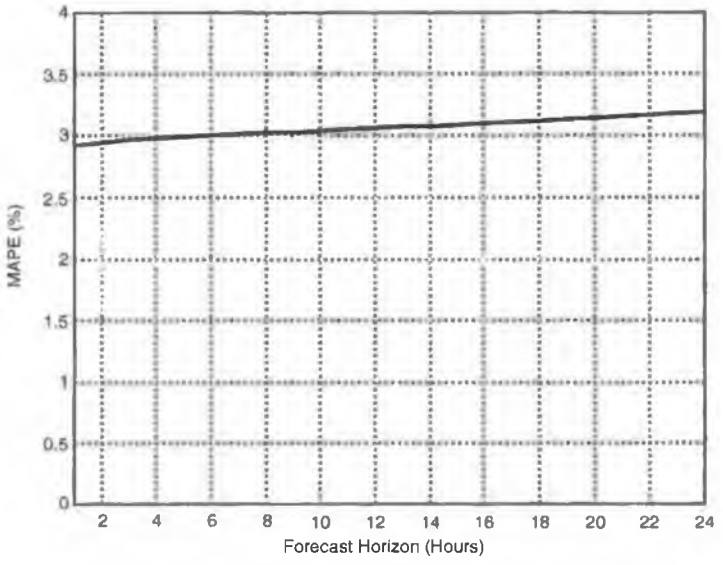

Figure 6.4. The MAPE as a function of forecast horizon for SM_6. (training set)

Tables 6.3 (training set) and 6.4 (validation set) show the daily MAPEs achieved for all day-types at four different forecast horizons. The daily MAPEs for each SM are high compared to the performance of the PPMs (see Tables 5.8 and 5.9, Section 5.4.2.3). For example, the daily MAPE achieved by the (best) PPMs for the summer working days day-type is 3.05% (Table 5.8), compared with 4.65% using the SM on the same day-type. The cause of this relatively bad performance is discussed below.

Table 6.3. Mean absolute percentage *daily* errors of SMs at differing forecast horizons (Training set).

Day-type	Daily MAPE (%)			
	1-step ahead	8-steps ahead	16-steps ahead	24-steps ahead
Early winter Sundays	4.75	5.54	5.07	5.33
Summer Sundays	7.30	7.29	7.29	7.27
Late winter Sundays	5.89	6.72	5.54	5.25
Early winter working days	3.66	3.91	3.99	4.11
Summer working days	4.40	4.40	4.40	4.46
Late winter working days	2.92	3.02	3.10	3.19
Early winter Saturdays	3.63	4.22	4.19	4.63
Summer Saturdays	3.82	3.91	3.96	4.12
Late winter Saturdays	3.48	3.86	3.81	4.25

Table 6.4. Mean absolute percentage *daily* errors of SMs at differing forecast horizons (Validation set).

Day-type	Daily MAPE (%)			
	1-step ahead	8-steps ahead	16-steps ahead	24-steps ahead
Early winter Sundays	4.73	5.18	4.31	5.17
Summer Sundays	5.35	5.38	5.39	5.74
Late winter Sundays	5.24	6.00	4.66	4.82
Early winter working days	3.99	4.33	4.30	4.36
Summer working days	4.58	4.66	4.59	4.65
Late winter working days	3.37	3.56	3.64	3.75
Early winter Saturdays	4.02	5.12	5.00	5.03
Summer Saturdays	3.73	4.37	4.01	3.97
Late winter Saturdays	3.38	3.21	3.28	3.88

Figure 6.5, below, shows a plot of six days of load taken from the late winter working days day-type. The six days shown are Wednesday, Thursday, Friday, (Saturday and Sunday are not working days) Monday, Tuesday and Wednesday. As can be seen, there is a *shift* in the level of the load in the centre of the plot caused by the removal of Saturday and Sunday during construction of the late winter working days series (Section 5.2).

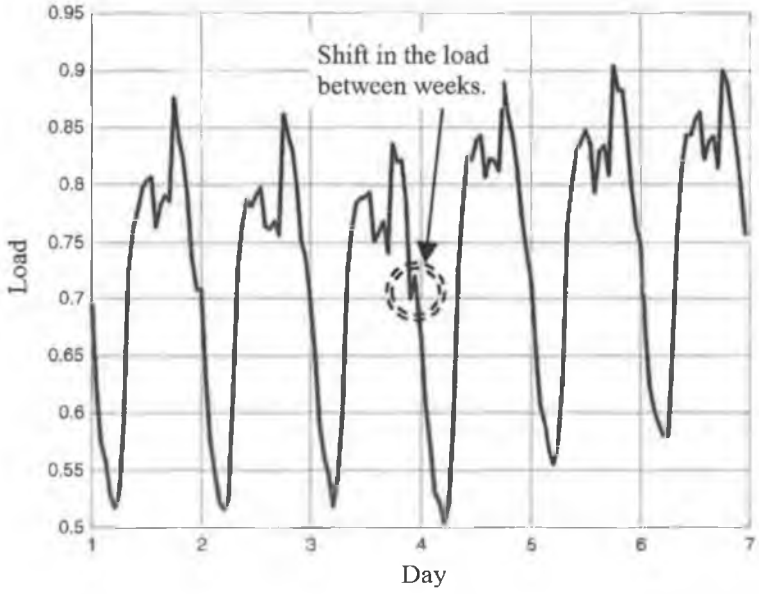

Figure 6.5. Loads in the late winter working day day-type from 03/02/1999 to 10/02/1999 (Note: Day 3 = Friday, Day 4 = Monday)

The SM attempts to model the load as a trend and seasonal component (Equation 6.1). As already stated (Section 3.3.2.4), the trend component acts as a low pass-filter and so is unable to adjust to the shift (which occurs at a single point and is

thus at a high frequency). Therefore, the shift must be accounted for by the seasonal component. However, the seasonal component requires the expected sum of the load (minus the trend) over the season (24-hours) to be zero (Equation 3.58, Section 3.3.2.2). As the shift violates this condition, it introduces a large error into the seasonal component forecast. It should be noted that this effect is more pronounced with weekend day-types, as they have a shift after every day. It was also found that the MTS model using these SMs performed badly compared to the parallel models. Thus a different approach is required, in which the trend component accounts for the shift.

The seasonal component, as stated above, requires the expected sum of the load (minus the trend) over the season to be zero. The solution proposed here is to remove the (forecasted) average value of the load for each day from each day. That is, the trend component is now modelled as fixed over the duration of a day i.e. as a *daily trend*. As the change in the trend component over the course of a day is negligible, this approximation may be justified. The remaining load is then zero-mean as required and may be modelled as a seasonal process. Figure 6.6 below, contrasts this approach with the previous one. This approach is called the Adjusted Sequential Model (ASM).

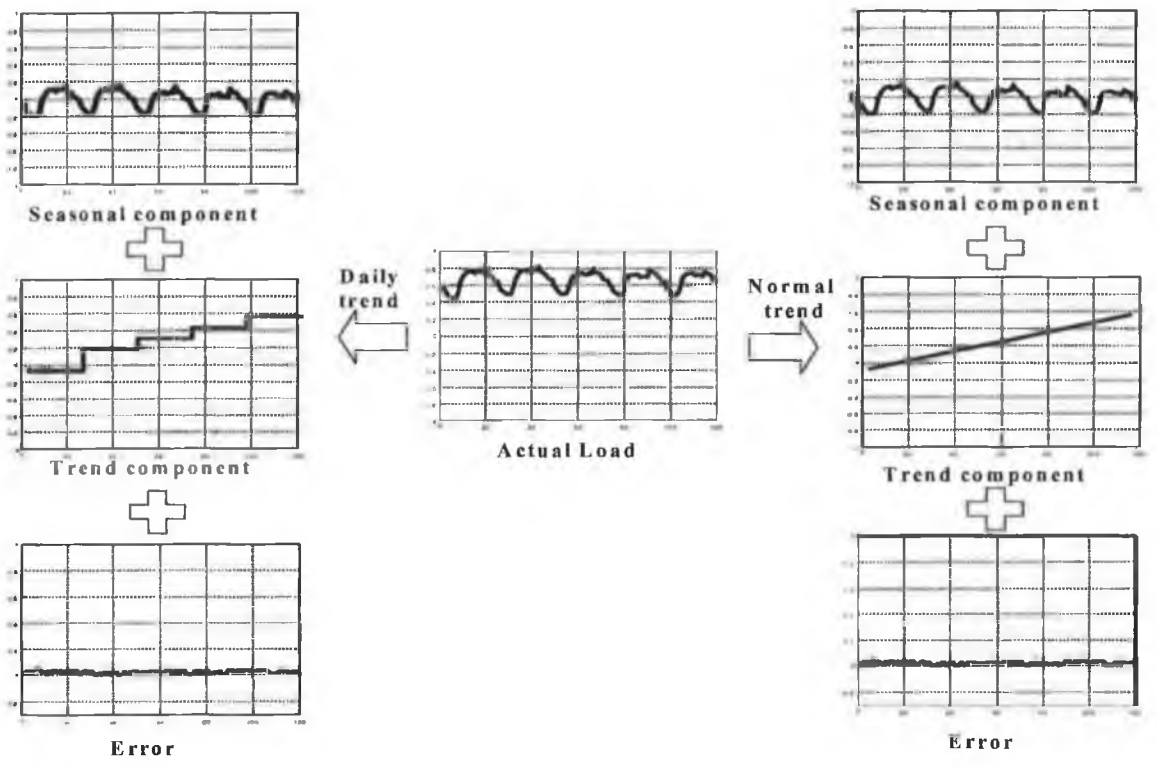

Figure 6.6 Comparison of the SM and Adjusted SM approach (trend component exaggerated for clarity).

In order to model the average load per day, a series is constructed by taking the average value of each day. This series has one point per day and thus has similar characteristics to a partitioned series (which is composed of loads from one hour per day). Thus the average load is modelled using a model similar to a PPM (Section 5.4). This model will be referred to as a Trend Model (TM), and TM_j will be used to denote the trend model for day-type j.

The trend model forecasts are then removed from the actual load leaving a seasonal component (see Figure 6.6 above). This seasonal component is then forecast using a DPRW (Section 3.3.2.2) as before. Figure 6.7, below, again contrasts the SM and ASM approach, but from a model building perspective. It should be noted that the SM models the trend and seasonal component jointly, while the ASM models the trend and seasonal component separately.

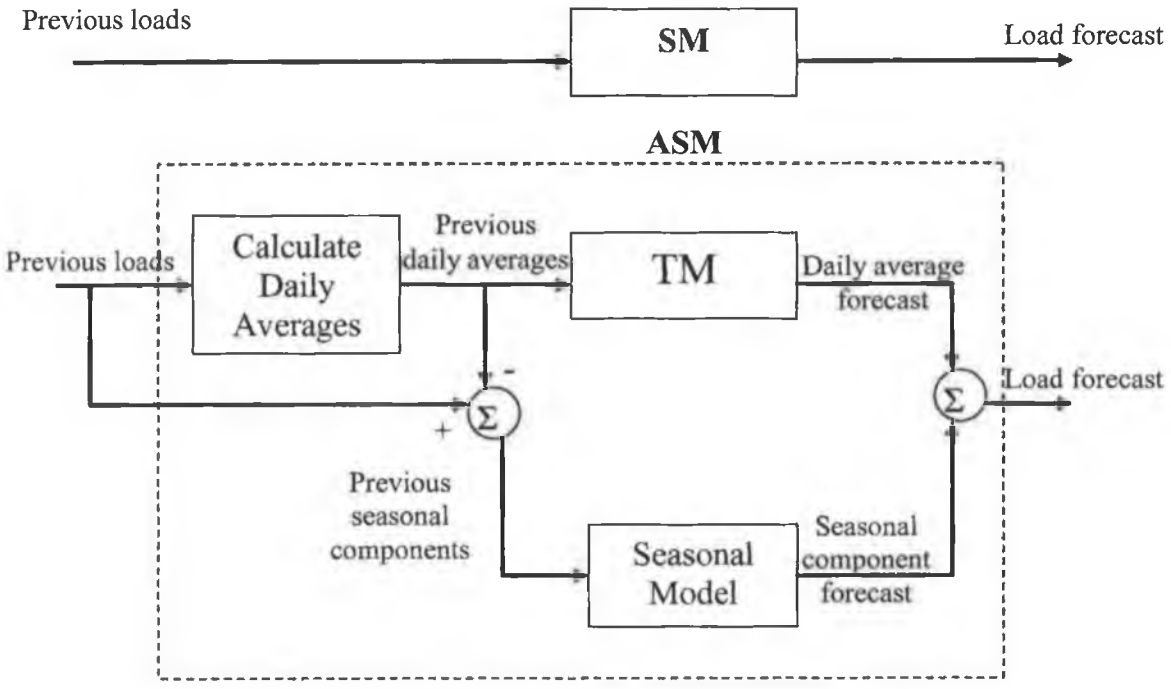

Figure 6.7 Comparison of the SM and Adjusted SM approach from a model building perspective (trend component exaggerated for clarity).

Table 6.5 shows the parameters calculated for ASM_6 (i.e. the late-winter working day day-types model). Comparing Tables 6.2, above, and 6.5 below it can be seen that the parameters calculated for the ASM and SM models differ greatly. $\sigma^2_{\dot{d}_6}$ for the SM has a far smaller value than that of the ASM. However, as the trends in both models are calculated differently (Figure 6.7), they cannot be directly compared. In contrast, the seasonal component in both models is trying to forecast the same component of the load (Figure 6.6). Comparing $\sigma^2_{\psi_6}$ for both models shows that the ASM has a seasonal component with a greater variance than the SM; 1.33×10^{-2} and 6.61×10^{-6}, respectively. Thus the SM reacts to a change in the seasonal component far slower than the ASM. This may be caused by the effect of the shift in the load between weeks (Figure 6.5) for the SM.

Table 6.5. Parameters of ASM_6 using SSD.

Parameter	Value
$\sigma^2_{x_6}$	1
$\sigma^2_{d_6}$	0
$\sigma^2_{\dot{d}_6}$	1.25×10^{-5}
$\sigma^2_{\psi_6}$	0.0133

Figure 6.8, below shows the daily MAPE as a function of forecast horizon for ASM_6. In contrast to SM_6 (Figure 6.3) the MAPE remains relatively constant despite the increase in forecast horizon.

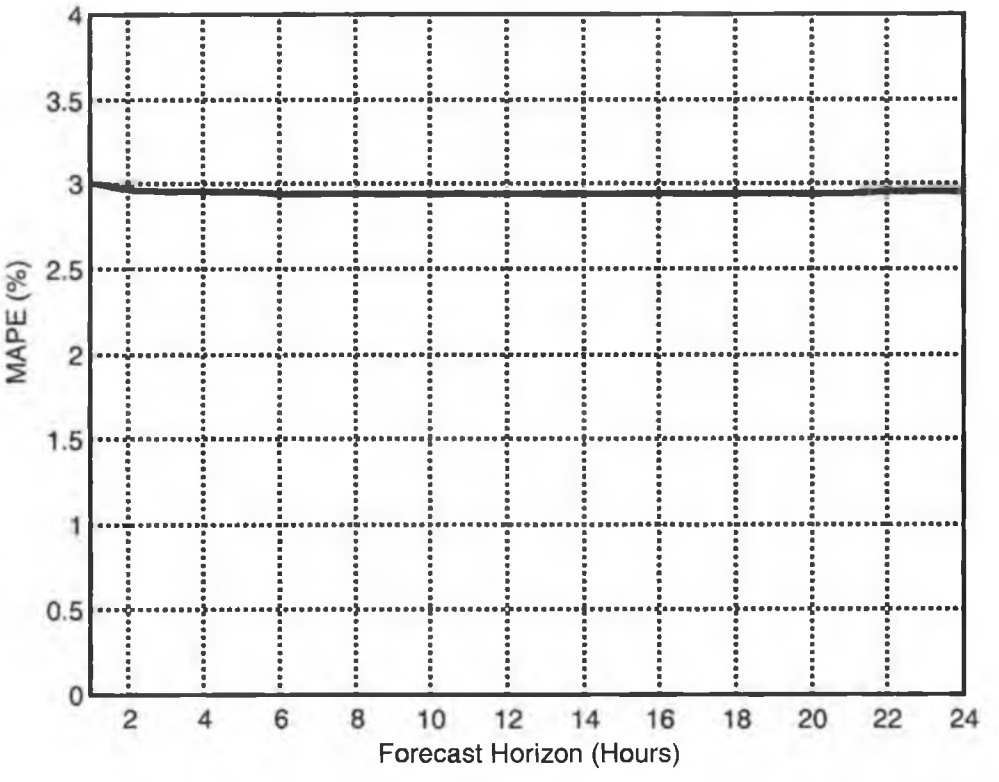

Figure 6.8. The MAPE as a function of forecast horizon for ASM_6. (training set)

Tables 6.6 and 6.7, below, show the daily MAPE achieved at four different forecast horizons for all the day-types in the training and validation sets. Comparing these results with those for the SMs (Tables 6.3 and 6.4) it can be seen that the daily MAPEs for the ASMs are lower. They are also broadly in line with the results for the PPMs used in the parallel models (Tables 5.6 and 5.7, Section 5.4.2.3).

Table 6.6. Mean absolute percentage *daily* errors of ASMs at differing forecast horizons (Training set).

Day-type	Daily MAPE (%)			
	1-step ahead	8-steps ahead	16-steps ahead	24-steps ahead
Early winter Sundays	4.15	4.15	4.15	4.15
Summer Sundays	4.70	4.63	4.63	4.61
Late winter Sundays	3.65	3.65	3.65	3.65
Early winter working days	3.18	3.04	3.04	3.03
Summer working days	3.14	2.93	2.93	2.94
Late winter working days	3.04	2.97	2.97	2.97
Early winter Saturdays	3.67	3.67	3.67	3.67
Summer Saturdays	3.07	3.02	3.02	2.99
Late winter Saturdays	3.14	3.05	3.05	3.05

Table 6.7. Mean absolute percentage *daily* errors of ASMs at differing forecast horizons (Validation set).

Day-type	Daily MAPE (%)			
	1-step ahead	8-steps ahead	16-steps ahead	24-steps ahead
Early winter Sundays	4.20	4.20	4.20	4.20
Summer Sundays	3.57	3.48	3.49	3.45
Late winter Sundays	3.70	3.70	3.71	3.70
Early winter working days	3.18	3.03	3.03	3.02
Summer working days	3.07	2.85	2.85	2.88
Late winter working days	2.95	2.89	2.89	2.90
Early winter Saturdays	4.06	4.06	4.07	4.06
Summer Saturdays	3.04	2.98	2.98	2.94
Late winter Saturdays	2.87	2.75	2.75	2.75

6.3 The Cardinal Point and End-Sum Models.

The cardinal point models generate forecasts for the load at specific hours of the day. The end-point model generates forecasts for the load at the *end-point* which is the last hour of the day. The parallel models generate forecasts at specific hours of the day and so these are used as the cardinal and end-point models. The specific parallel models used are the non-linear models selected in Section 5.7.5 and presented in Table 5.25 which are summarised below (Table 6.8).

Table 6.8 Daily AMAPE (%) for non-linear parallel models using selected topologies (validation set).

	Early winter Sundays	Summer Sundays	Late winter Sundays	Early winter working days	Summer working days	Late winter working days	Early winter Saturdays	Summer Saturdays	Late winter Saturdays
Model	NNART	NNAR	NNART	NNARTS	NNARTS	NNARTS	NNART	NNART	NNART
Topology	6×1×1	3×1×1	6×3×1	6×3×1	5×1×1	6×3×1	6×3×1	6×3×1	6×1×1
AMAPE	2.41	2.98	2.32	1.92	2.23	1.89	2.52	2.90	2.28

The end-sum model forecasts the sum of the load over the period of a day. The first part of constructing this model was to form a series made up of the daily sums. This series was then modelled in a manner similar to the partitioned series. That is; a preliminary linear model is first used to remove the trend and seasonal component, and then the residual is modelled using a neural network. The preliminary linear model has been described in Section 6.2 and is in fact the trend model (TM). This is because the sum of the load over the period of a day is simply the average load for that day multiplied by twenty-four.

Table 6.9 below summarises the end-sum model performance, type and topology used for each of the day-types. It should be noted that the AMAPEs for the end-sum models are lower than for the parallel models as it is easier to forecast the sum of load over a day rather than at particular hours.

Table 6.9 AMAPE (%) for non-linear end-sum models using selected topologies (validation set).

	Early winter Sundays	Summer Sundays	Late winter Sundays	Early winter working days	Summer working days	Late winter working days	Early winter Saturdays	Summer Saturdays	Late winter Saturdays
Model	NNART	NNAR	NNART	NNARTS	NNARTS	NNARTS	NNART	NNART	NNART
Topology	3×1×1	2×1×1	2×2×1	2×1×1	2×2×1	2×3×1	2×3×1	3×3×1	2×2×1
AMAPE	1.95	2.17	1.67	1.25	1.43	1.45	2.61	2.01	1.81

6.4 The Multi-Timescale Model.

The sequential model (Section 6.2), cardinal point, end-point and end-sum forecasts are now combined by using the MTS technique (Section 3.3.4). Specifically, the following are combined:

- Sequential model forecasts from a forecast horizon of 1 to N (see Section 3.3.4.2), where $N = 72$, i.e. a three day ahead SM forecast. Note: the adjusted SM forecasts (Section 6.2.4) are used in this section,
- Cardinal point forecasts at the hours of 5 a.m., 1 p.m., 2 p.m., 6 p.m. and 11 p.m. for one day ahead only. These hours are chosen as they have been identified by Eirgrid as the most important hours of the day (Murray, 2000),
- End-sum forecasts for the first, second and third day ahead.

The first state in the SM is fixed and the other states are allowed to vary in the MTS technique in order to combine the forecasts above. It was found by Murray, (1996) that fixing the first state (the trend component, see Equation 6.2) and freeing the other states in a DPRW (as is used in the SM here) gave the best results.

In addition, all of the forecast origins are chosen to be at 12 a.m. (00:00 hrs) as this is the first hour of each day. The parameters of the multi-time scale model are summarised in Table 6.10 below.

Table 6.10 Parameters for the MTS model (see Section 3.3.4.2 for definitions).

Variable	Values/Dimension	Description
n	25	Dimension of the state vector $\theta(k)$
r	24	Number of freed states of $\theta(k)$
N	72	Number of points used in the smoothing constraint
M	5	Number of cardinal point forecasts used.
P	3	Number of end-sum forecasts used.
c_i	[5 13 14 18 23]	The distance from the forecasting origin to the i^{th} cardinal point.
S	24	Length of the summations for end-sum model.
w	1×80	Weight vector.
W	80×80	Diagonal weight matrix.

The weight matrix, $\boldsymbol{W}$, must now be determined. This is examined in Sections 6.4.1 and 6.4.2.

6.4.1 Weight determination: A numerical approach

The diagonal elements of the weight matrix, w_i, may be expressed in terms of the weights applied to the five cardinal point deviations, $w_1,\ldots, w_5$, three end-sum deviations, $w_6,\ldots, w_8$, and 72 sequential model deviations, $w_9,\ldots, w_{80}$, as:

$$diag(\boldsymbol{W}) = [\underbrace{w_1 \quad \cdots \quad w_5}_{\text{Cardinal points}} \quad \underbrace{w_6 \quad w_7 \quad w_8}_{\text{End - Sum}} \quad \underbrace{w_9 \quad \cdots \quad w_{32}}_{\text{SM day 1}} \quad \underbrace{w_{33} \quad \cdots \quad w_{56}}_{\text{SM day 2}} \quad \underbrace{w_{57} \quad \cdots \quad w_{80}}_{\text{SM day 3}}] \tag{6.5}$$

where 'SM day 1' refers to the SM deviations from the MTS forecasts for day 1, etc. In order to reduce the dimension of the weight matrix above, it is assumed that the weights applied to the sequential model deviations are the same for each day:

$$[w_9 \quad \cdots \quad w_{32}] = [w_{33} \quad \cdots \quad w_{56}] = [w_{57} \quad \cdots \quad w_{80}] \tag{6.6}$$

Three types of technique are now examined for determining the weight matrix:

1. **A random approach (Section 6.4.1.1)**, where $w_1,\ldots,w_{32}$ are assigned randomly, with no regard to the differing accuracies of the models. This is included to demonstrate the importance of calculating the weights properly,

2. **Weight profiles (Section 6.4.1.2)**. Murray (2000) suggests the use of profiles based on experience, and

3. **An optimised weight matrix (Section 6.4.1.3)**, calculated by use of a non-linear search routine.

6.4.1.1 Random Weight Selection.

Three random weight matrices are constructed using a uniform pseudo-random number generator. These weights are shown below in Figure 6.9.

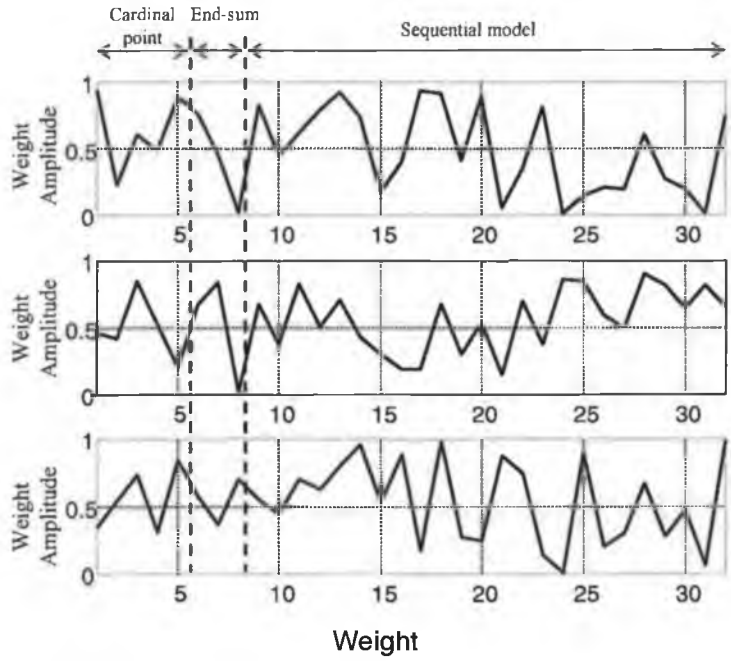

Figure 6.9. A plot of the diagonal of the three random weight matrices (Note: elements 33 to 56 and 57 to 80 are the same as elements 9 to 32 and so are not shown (see Equation 6.6)).

6.4.1.2 Weight Profile Selection.

The weight profiles constructed by Murray (1996) were developed from empirical evidence. They are constructed using the following procedure:

- The weights applied to the cardinal point and end-sum deviations are given values of 100,

- The weights applied to the first sequential model deviation, w_9, are given values approximately $1/10^{th}$ of the weight applied to the cardinal and end-sum deviations. This recognises that the end-sum and cardinal point forecasts have a greater accuracy than the sequential model forecasts,

- Weights w_{10} to w_{32} may then be constructed using three different approaches, resulting in three different profiles:

1. SM deviations closer to the forecasting origin are penalised more, allowing the MTS forecast more freedom to meet the cardinal and end-sum forecasts away from the forecasting origin. This is achieved by decrementing the weights by a constant amount:

$$w_i = w_9 - (i-9)\frac{w_9}{24} \qquad i = 10,...,32 \qquad (6.7)$$

2. Alternatively, the SM deviations closer to the end of the day are given a greater weight (for example, w_{32} is the weight applied to the SM deviation for 23:00 hrs and is given a greater weight than w_9, which is at the forecasting origin). This is achieved by incrementing the weights by a constant amount:

$$w_i = w_9 + (i-9)\frac{w_9}{24} \qquad i = 10,...,32 \qquad (6.8)$$

3. SM deviations close to the forecasting origin and the end of the day are given equal weights, so as to obtain more agreement (between the SM and MTS forecasts) at both the forecasting origin and at the boundaries between the days. The intervening weights are then assigned lower values using a quadratic curve as:

$$w_i = 0.38i^2 - 1.55i + 20.89 \qquad i = 9,...,32 \qquad (6.9)$$

- Weights w_{33} to w_{80} are again constructed by repeating weights w_9 to w_{32} (see Equation 6.6).

These three profiles are shown below in Figure 6.10.

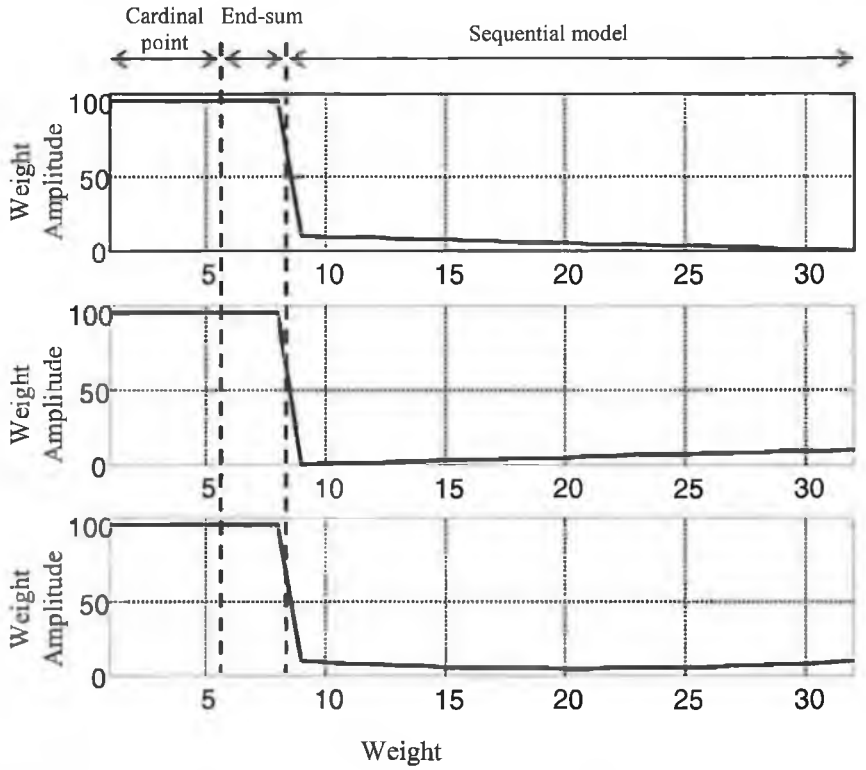

Figure 6.10. A plot of the diagonal of the three profile weight matrices (Note: elements 33 to 56 and 57 to 80 are the same as elements 9 to 32 and so are not shown).

6.4.1.3 Optimised Weight Selection.

Given a particular weight matrix (and the various components required for the MTS technique, which are described in Section 6.4), the MTS technique may be used to produce load forecasts. The load forecasts that are produced over the validation set may then be compared to the actual load, to give a MAPE over the validation set. By allowing elements $w_1,\ldots,w_{32}$ of the weight matrix to vary (elements $w_{33},\ldots,w_{80}$ are again calculated using equation (6.6)), a cost function, J, may be constructed as:

$$J(w_1,\ldots,w_{32}) = \sum_{i=1}^{K} \frac{|y(i) - y_{mts}(i)|}{y(i)} \frac{100}{K} \tag{6.10}$$

where $y(k)$ is the load at time k, $y_{mts}(k)$ is the MTS load forecast at time k and K is the number of points in the validation set. The optimised weight selection technique consists of optimising Equation (6.10) via a non-linear optimisation routine. The routine used is the FMINSEARCH algorithm in Matlab which uses the Nelder Mead direct search algorithm (Box *et. al.*, 1969).

To start the optimisation, $w_1,\ldots,w_{32}$ are initialised randomly using the same technique used in Section 6.4.1.2. The optimisation routine is stopped after five thousand iterations. To help avoid local minima or at least find a good local minimum, the optimisation is carried out using three different initial conditions. Finally, unlike in Sections 6.4.1.2 and 6.4.1.3, different weight matrices are optimised for each day-type (i.e. 9×3 optimisations).

Figure 6.11, below, shows three optimised weight matrices for the late winter day-type. The optimised values are quite similar, though there are some important differences.

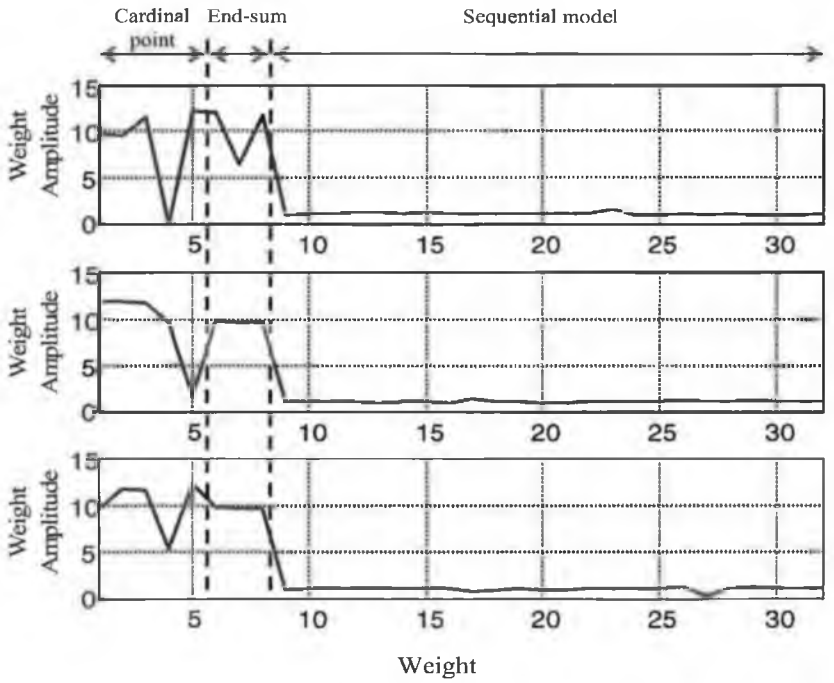

Figure 6.11. A plot of the diagonal of the three optimised weight matrices (Note: elements 33 to 56 and 57 to 80 are the same as elements 9 to 32 and so are not shown).

The amplitudes of the weights applied to the cardinal and end-point deviations are approximately 10 times larger than those applied to the SM deviations (Figure 6.11, above). This is in agreement with the weight profiles (Section 6.4.1.2). However, some of the weights have much lower amplitudes. For example, the first optimised weight matrix has a value of 0.5 for w_4; the weight applied to the 6 p.m. cardinal point deviation. The second optimised weight matrix has a value of 1.75 for w_5; the weight applied to the 11 p.m. cardinal point deviation. From experience it has been found that applying a large weight to w_4

requires a low weight to be applied to w_5, in order to reach a good minimum in Equation (6.10) and *vice versa*.

In the third optimised weight matrix, the value for w_{28} (the weight applied to the 5 p.m. SM deviation) has a lower value than the other SM deviation weights, implying that the SM forecast for 5 p.m. may have a lower accuracy than the other SM forecasts. The cause of these low weights will be further examined in Section 6.5.

Although the optimised weight matrices are similar, the differences between them imply that local minimums have been found to Equation (6.10), and that the weights are dependent on the initial conditions.

6.4.1.4 Results and Analysis.

Figure 6.12 below, shows the MAPE as a function of the hour of the day using the SM and the MTS technique, with the three *random* weight matrices (Section 6.4.1.1) over the novelty set (late winter day-type). As can be seen, the MTS forecast MAPEs are better than the SM model forecast MAPEs at some hours but the results are inconsistent.

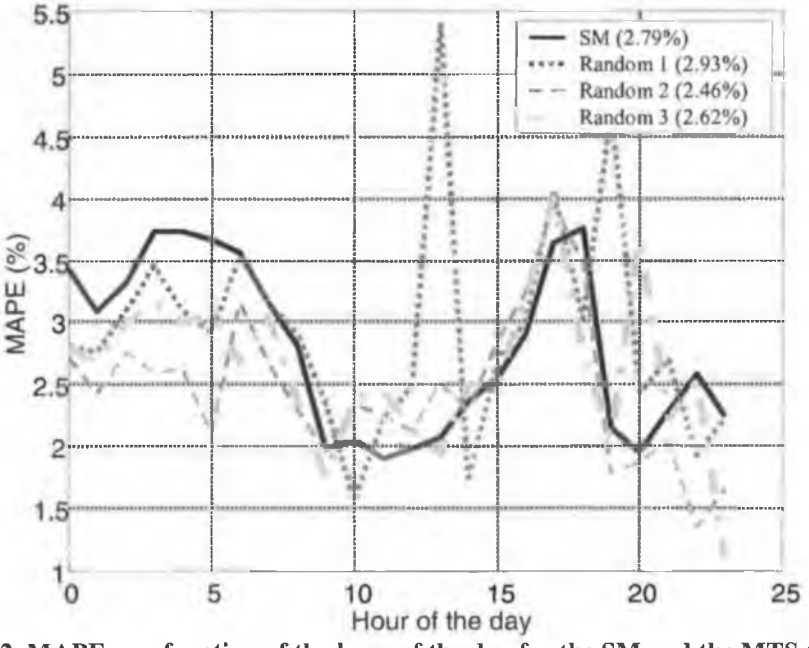

Figure 6.12. MAPE as a function of the hour of the day for the SM and the MTS technique with three random weight matrices (late winter working days, novelty set).

In addition, the accuracy of the forecasts made using the random weight matrices at some hours are very poor (for example, at 13 hrs for random weight matrix 1, and at 20 hrs for random matrix 2, in Figure 6.12).

The cardinal point forecasts and end-sum forecasts are superior to the SM forecasts. Yet, the forecasts produced by combining the SM, cardinal point and end-sum forecasts (via the MTS technique) using the first random weight matrix (2.93%) are inferior to the SM forecasts (2.79%). This shows that if the weights in the MTS technique are not estimated correctly, the technique may result in very poor forecasts.

Figure 6.13, next page, shows the MAPE as a function of the hour of the day, using the SM and the MTS technique, with the three *profile* weight matrices (Section 6.4.1.2) over the novelty set (late winter day-type). The MTS MAPEs are superior to the SM MAPEs at most hours of the day and the daily MAPEs for the profiles (2.30%, 2.52% and 2.36%) are significantly better than the SM daily MAPE. Thus the use of weight profiles in the MTS technique can generally be said to provide increased accuracy.

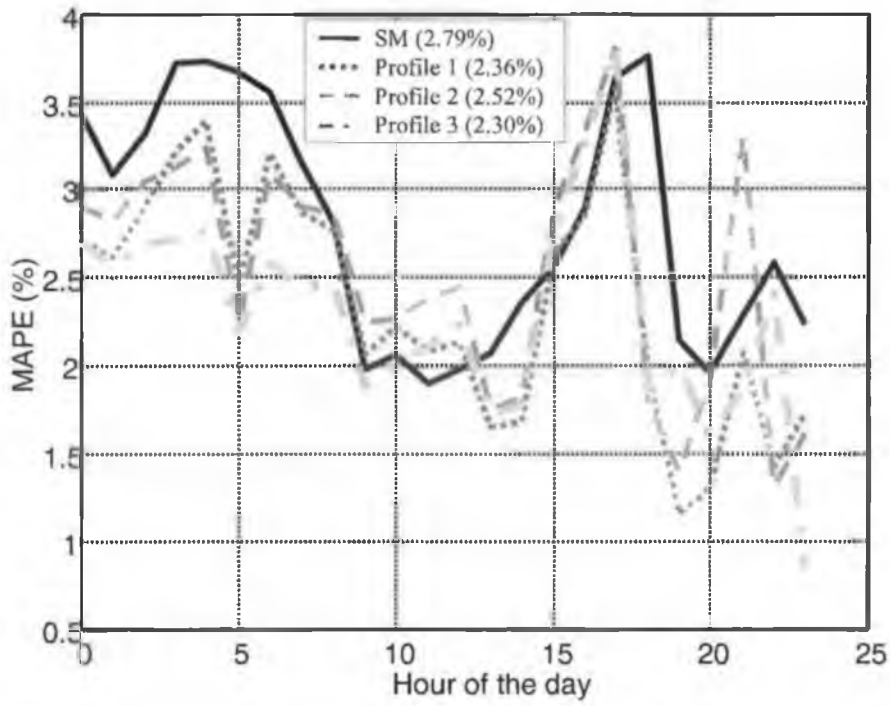

Figure 6.13. MAPE as a function of the hour of the day using three weight profiles (late winter working days, novelty set).

As can be seen, Profile 1 and Profile 3 result in superior forecasts to Profile 2 at 00:00 hrs (Figure 6.13). 00:00 hrs is the forecasting origin and the weights given to the SM deviations at the forecasting origin are given greater weights in Profile 1 and Profile 3, than in Profile 2. Conversely, Profiles 2 and 3 result in superior forecasts to Profile 1 at 23:00 hrs (Figure 6.13). This is because the weight given to the SM forecast at this point has a greater weight in Profiles 2 and 3.

At the cardinal points (05:00 hrs, 13:00 hrs, 14:00 hrs, 18:00 hrs and 23:00 hrs) the MTS forecasts are all significantly better than the SM forecasts. However at the other hours, the results are mixed. This implies that the weight profiles may be weighting the cardinal point deviations too heavily, at the expense of the forecasts at other hours of the day (Figure 6.13).

Figure 6.14, below shows the MAPE as a function of the hour of the day, using the SM and the MTS technique with the three *optimised* weight matrices (Section 6.4.1.3) over the novelty set (late winter day-type). As can be seen, the optimised weight matrices lead to similar MAPEs for each hour of the day and are significantly better at all hours than the SM forecasts. The first and third optimised weight matrices have very similar performance. The second optimised weight matrix leads to poorer results and the optimisation procedure in this case may have found a poor local minimum.

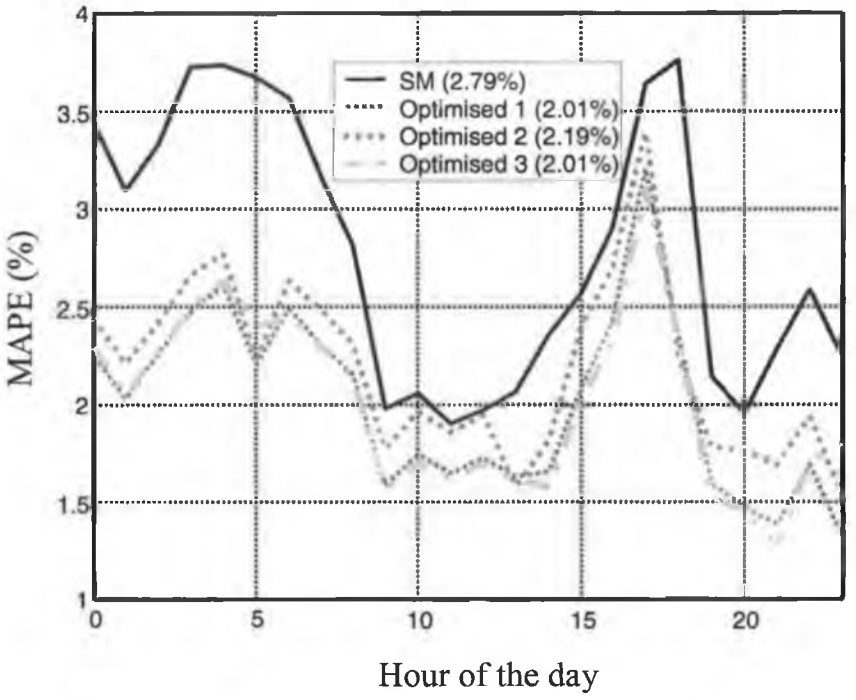

Figure 6.14. MAPE as a function of the hour of the day using three optimised weight matrices (Late winter working days, Novelty set).

Tables 6.11 and 6.12, below, summarise the MTS forecast daily MAPEs produced over all day-types using the weight matrices defined in Sections 6.4.1.1 to 6.4.1.3. The first thing of note is that the optimised weight matrices give superior results, in almost all cases, to both the weight profiles and random weight matrices. However, there are some exceptions; for example, weight profile 3 gives the best results in the validation set during late winter Saturdays (Table 6.11). This result may just be a random occurrence as it does not carry over to the novelty set (Table 6.12).

The random weight matrices in many cases give inferior results when compared to the SM. Thus, the MTS technique requires appropriate estimation of weight matrices in order to improve on SM forecasts (as noted in earlier in this section).

Table 6.11 Daily MAPEs for the SM and the MTS model using different types of weight matrix (Validation set).

Weight type	Early winter Sundays	Summer Sundays	Late winter Sundays	Early winter working days	Summer working days	Late winter working days	Early winter Saturdays	Summer Saturdays	Late winter Saturdays
Random 1.	5.15	4.27	4.41	3.12	2.76	2.75	5.14	4.43	3.97
Random 2.	5.25	4.02	4.31	2.86	2.82	2.29	4.98	4.31	3.46
Random 3.	5.02	4.15	4.12	2.95	2.69	2.39	4.64	4.07	3.35
Profile 1.	4.81	3.69	3.96	2.53	2.46	2.39	4.47	4.04	2.97
Profile 2.	4.77	3.77	3.79	2.34	2.35	2.08	4.56	3.85	2.76
Profile 3.	4.86	3.74	3.82	2.42	2.53	2.24	4.62	3.95	2.65
Optimised 1	3.86	3.32	3.22	2.23	2.13	2.08	3.78	2.79	2.72
Optimised 2	3.84	3.34	3.21	2.23	2.24	2.15	3.69	2.85	2.83
Optimised 3	3.85	3.32	3.24	2.25	2.14	2.09	3.80	2.81	2.83
SM	4.20	3.49	3.70	3.04	2.88	2.90	4.06	2.98	2.75

Table 6.12 Daily MAPEs for the SM and the MTS model using different types of weight matrix (Novelty set).

Weight type	Early winter Sundays	Summer Sundays	Late winter Sundays	Early winter working days	Summer working days	Late winter working days	Early winter Saturdays	Summer Saturdays	Late winter Saturdays
Random 1.	5.12	4.23	4.36	2.64	2.55	2.93	5.01	3.74	3.45
Random 2.	5.23	3.98	4.23	2.56	2.58	2.46	4.87	3.89	3.01
Random 3.	5.04	4.12	4.10	2.76	2.46	2.62	4.55	3.67	2.98
Profile 1.	4.83	3.67	3.97	2.14	2.23	2.36	4.36	3.42	2.54
Profile 2.	4.75	3.76	3.78	2.12	2.13	2.52	4.45	3.24	2.39
Profile 3.	4.87	3.73	3.85	2.22	2.19	2.30	4.62	3.95	2.65
Optimised 1	4.03	3.10	3.27	1.85	1.93	2.01	3.64	2.15	2.31
Optimised 2	4.01	3.13	3.27	1.85	1.98	2.19	3.60	2.16	2.36
Optimised 3	4.01	3.10	3.29	1.87	1.95	2.01	3.63	2.15	2.35
SM	4.23	3.46	3.59	2.67	2.63	2.79	4.19	2.36	2.30

6.4.2 Weight determination: A deterministic approach

The following derivation seeks to find an analytic solution for the weight vector in the MTS technique. The aim is to minimise (in some sense) the *error* between the MTS forecasts of load and the actual load. However, this aim is in conflict with the MTS technique which minimises the *deviations* between the MTS forecast and the SM, PM and end-sum forecasts. The reason for this conflict is shown diagrammatically in Figure 6.15 below, which shows a plot of a MTS forecast, a SM forecast and the actual load (note: the end-sum, and PM forecasts are excluded for clarity). As can be seen from Figure 6.15; if the deviation of the

MTS forecast from the SM forecast at k+40, δ_{40} (see Section 3.3.4.2), is forced to zero (i.e. the MTS forecast at this point is forced to be equal to the SM forecast), then the forecast error at this point, say ε_{40} (note: ε_{45} is shown in Figure 6.15 for clarity), will be non-zero. Alternatively if ε_{40} is forced to zero (i.e. the MTS load forecast at this point is forced to be equal to the actual load), then the deviation will be non-zero and hence the conflict.

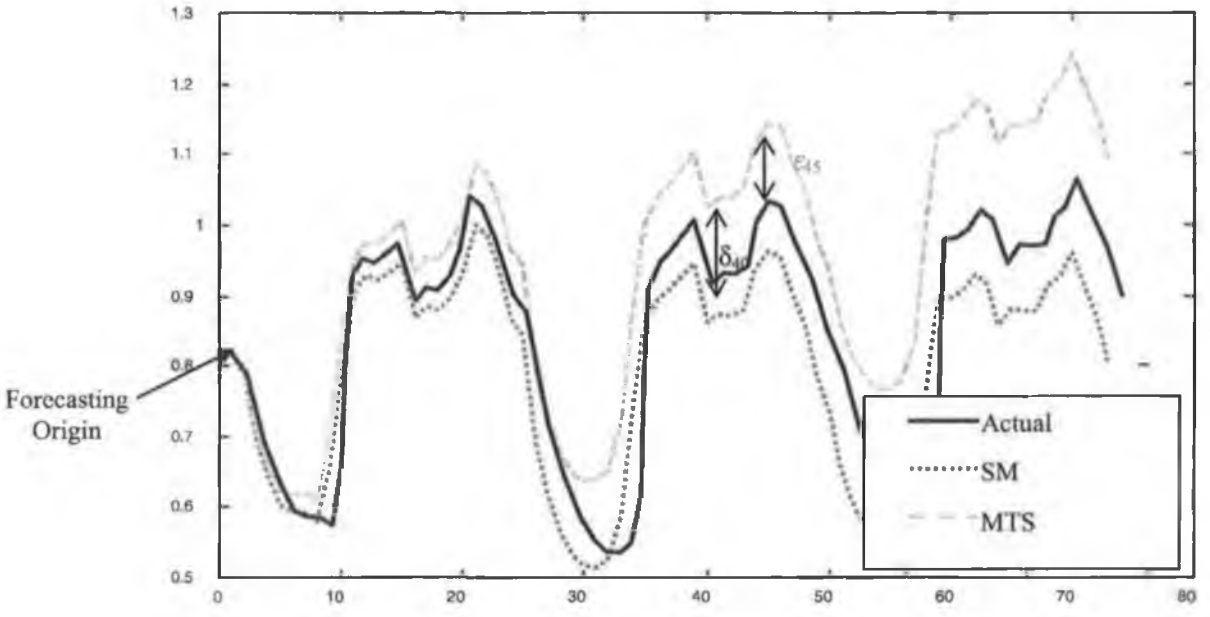

Figure 6.15 Diagram showing the conflict between minimising δ_i and ε_i together.

The final aim of all forecasting models is to minimise the forecasting error in some sense. The MTS technique is based on the assumption that by maintaining the *shape* of the SM forecast with adjustments from the PM and end-sum models, the forecasting error will be reduced. In this sense the MTS technique is similar to a *regularising term* (see Haykin, 1999); i.e. a term added to the cost function which places a penalty on some geometric quality of the model, so as to aid its ability to generalise. The approach presented below constructs an error cost function based on the standard sum of squared errors plus a regularisation term based on the MTS technique.

Define a vector $\boldsymbol{w}$ as the diagonal of the weight matrix, $\boldsymbol{W}$, as:

$$\boldsymbol{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{N+M+P} \end{bmatrix} = diag(\boldsymbol{W}) \qquad (6.11)$$

where N is the number of points used in the smoothing constraint, M is number of cardinal point forecasts used, P is the number of end-sum forecasts used and w_i is the i^{th} element of $\boldsymbol{w}$. The aim of this derivation is to find the optimal value of $\boldsymbol{w}$.

Define a vector, $\underline{\boldsymbol{y}}$, of actual loads in the training set as:

$$\underline{\boldsymbol{y}} = [y(1) \quad y(2) \quad \cdots \quad y(K)] \tag{6.12}$$

where K is the number of points in the training set and $y(k)$ is the load at time k. In addition, define a vector of forecasts made by the eventual model, $\underline{\boldsymbol{y}}_{mts}$, as:

$$\underline{\boldsymbol{y}}_{mts} = [\hat{y}_{mts}(1) \quad \hat{y}_{mts}(2) \quad \cdots \quad \hat{y}_{mts}(K)] \tag{6.13}$$

where $\hat{y}_{mts}(k)$ is a one-step ahead forecast for time k. A standard sum of squared errors cost function may now be defined as:

$$J(\boldsymbol{w}) = \left(\underline{\boldsymbol{y}} - \underline{\boldsymbol{y}}_{mts}\right)^T \left(\underline{\boldsymbol{y}} - \underline{\boldsymbol{y}}_{mts}\right) \tag{6.14}$$

where $J(\boldsymbol{w})$ is the sum of squared forecast errors (Note: this is not the only cost function that could be used). Note that J is a function of $\boldsymbol{w}$, which is the unknown parameter. However, it is desired to maintain the shape of the forecast by use of the MTS technique, i.e. Equation (6.14) is constrained by Equation (3.79) (Section 3.3.4.2) evaluated at $k = 1,\ldots,K$:

$$C(\boldsymbol{w}) = \begin{bmatrix} \boldsymbol{\theta}_2^*(1,\boldsymbol{w}) - \left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{A}(1) \\ \boldsymbol{\theta}_2^*(2,\boldsymbol{w}) - \left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{A}(2) \\ \vdots \\ \boldsymbol{\theta}_2^*(K,\boldsymbol{w}) - \left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{A}(K) \end{bmatrix} = \underline{0} \tag{6.15}$$

where $C(\boldsymbol{w})$ is the constraint, $\boldsymbol{\theta}_2^*(k,\boldsymbol{w})$ is the vector of freed states which is now indexed by k to show that the altered state vector changes as a function of time and is also a function of the weight vector, $\boldsymbol{w}$. $\boldsymbol{W}$ is a diagonal weight matrix, $\boldsymbol{B}$ and $\boldsymbol{A}(k)$ are matrices as defined in Section 3.3.4.2 and $\boldsymbol{A}(k)$ is also now indexed by k as it is different at each k. Thus Equation (6.14) is to be solved subject to the constraint in Equation (6.15). This problem may be classified as *constrained*

optimisation and may be solved with the use of Lagrange multipliers (Apostol, 1974).

Given a cost function $J(\boldsymbol{w})$ and an equality constraint $C(\boldsymbol{w})$ the objective is to find a value $\boldsymbol{w}_0$ of $\boldsymbol{w}$ (Apostol, 1974) such that:

$$\boldsymbol{w}_0 = \arg\min_{\{\boldsymbol{w}:C(\boldsymbol{w})=0\}} J(\boldsymbol{w}) \tag{6.16}$$

The Langrangian may be defined as:

$$G(\boldsymbol{w},\boldsymbol{\Lambda}) = J(\boldsymbol{w}) - C(\boldsymbol{w})\boldsymbol{\Lambda} \tag{6.17}$$

where $G(\boldsymbol{w},\boldsymbol{\Lambda})$ is the Langrangian and $\boldsymbol{\Lambda}$ is a column vector of Lagrange multipliers, $\boldsymbol{\Lambda}= [\lambda_1\ \lambda_2\ \ldots\ \lambda_{rK}]^{\mathrm{T}}$. It is well known that under regularity conditions, $\boldsymbol{w}_0$ in Equation (6.16) can be equivalently solved as the solution (Apostol, 1974) to the following extremal problem:

$$(\boldsymbol{w}_0, \boldsymbol{\Lambda}_0) = \arg\min_{\boldsymbol{w}} \arg\max_{\boldsymbol{\Lambda}} G(\boldsymbol{w},\boldsymbol{\Lambda}) \tag{6.18}$$

Taking the partial derivatives of Equation (6.18) with respect to $\boldsymbol{w}$ and $\boldsymbol{\Lambda}$ and setting equal to zero gives:

$$\frac{\partial G(\boldsymbol{w},\boldsymbol{\Lambda})}{\partial \boldsymbol{w}} = \frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} - \frac{\partial C(\boldsymbol{w})}{\partial \boldsymbol{w}}\boldsymbol{\Lambda} = 0 \tag{6.19}$$

and

$$\frac{\partial G(\boldsymbol{w},\boldsymbol{\Lambda})}{\partial \boldsymbol{\Lambda}} = C(\boldsymbol{w}) = 0 \tag{6.20}$$

Defining:

$$\Omega = \frac{\partial C(\boldsymbol{w})}{\partial \boldsymbol{w}} \tag{6.21}$$

and substituting into Equation (6.19) gives:

$$\frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} - \Omega\boldsymbol{\Lambda} = 0 \tag{6.22}$$

In solving Equations (6.20) and (6.22) one could eliminate $\boldsymbol{w}$ and then solve for $\boldsymbol{\Lambda}$ or eliminate $\boldsymbol{\Lambda}$ and then solve for $\boldsymbol{w}$. If Ω is of full rank then the easiest approach is to eliminate $\boldsymbol{\Lambda}$ and then solve for $\boldsymbol{w}$.

Pre-multiplying Equation (6.22) by Ω^T gives (Apostol, 1974):

$$\Omega^T \frac{\partial J(w)}{\partial w} - \Omega^T \Omega \Lambda = 0 \tag{6.23}$$

As the determinant of $\Omega\Omega^T \neq 0$ Equation (6.23) can be solved for Λ as:

$$\Lambda = \left(\Omega^T \Omega\right)^{-1} \Omega^T \frac{\partial J(w)}{\partial w} \tag{6.24}$$

Substituting back into Equation (6.23) gives (Apostol, 1974):

$$\left(I - \Omega\left(\Omega^T \Omega\right)^{-1} \Omega^T\right) \frac{\partial J(w)}{\partial w} = 0 \tag{6.25}$$

The matrix $\Omega\left(\Omega^T\Omega\right)^{-1}\Omega^T$ has well known properties and is idempotent (Apostol, 1974) i.e.:

$$\left(\Omega\left(\Omega^T \Omega\right)^{-1} \Omega^T\right)\left(\Omega\left(\Omega^T \Omega\right)^{-1} \Omega^T\right) = \left(\Omega\left(\Omega^T \Omega\right)^{-1} \Omega^T\right) \tag{6.26}$$

In addition, the rank of $\Omega\left(\Omega^T\Omega\right)^{-1}\Omega^T$ is the same as the rank of Ω. If Ω is of full rank then there is a unique solution for to $w_0 = w$ using Equations (6.26) and (6.20) (Apostol, 1974).

Returning to the MTS problem considered here, the first thing to consider is whether Ω is of full rank. Examining Equation (6.21) it can be seen that Ω is of full rank *if* each of the partial differentials of $C(w)$ with respect to w are independent. The only term in $C(w)$ (Equation 6.15) that differs between the rows of $C(w)$ is $A(k)$, $k = 1,\ldots,K$. Thus we require that:

$$A(k) \neq A(j) \quad k,j = 1,\ldots,K \text{ and } k \neq j \tag{6.27}$$

otherwise one row will be equal to another and $C(w)$ will not be full rank. $A(k)$ is a function of the SM states, cardinal point forecasts and end-sum forecasts (see Equations 3.77 and 3.78). Thus it is likely that Equation (6.27) *holds*; as the SM states, the cardinal point forecasts and the end-sum forecasts vary over time (due among other things to a rising trend in load). Thus a unique minimum of w probably exists.

The gradient of Equation (6.15) with respect to w now needs to be calculated. However, the mathematics soon became intractable and a closed form solution could not be found (details are given in Appendix B). Another solution would be

to solve Equations (6.25) and Equation (6.20) numerically. Details of this approach can be found in Apostol (1974).

6.5 A Comparison of Parallel and Multi-Timescale Models.

Figure 6.16 below, shows the MAPE as a function of the hour of the day for the SM, PMs and MTS model during the late winter working day day-type (novelty set). As can be seen, the MTS model and PMs perform significantly better than the SM at all hours. In addition, the MTS' and PM's performance is similar at most hours of the day. The MTS model gives better results over the hours of 09 hrs to 14 hrs. It is interesting to note that within these hours, there are two cardinal points (1 p.m. and 2 p.m.). Conversely, the MTS forecasts at 17 hrs (5 p.m.) are significantly inferior to the PM forecasts, despite the presence of a cardinal point at 18 hrs (6 p.m.).

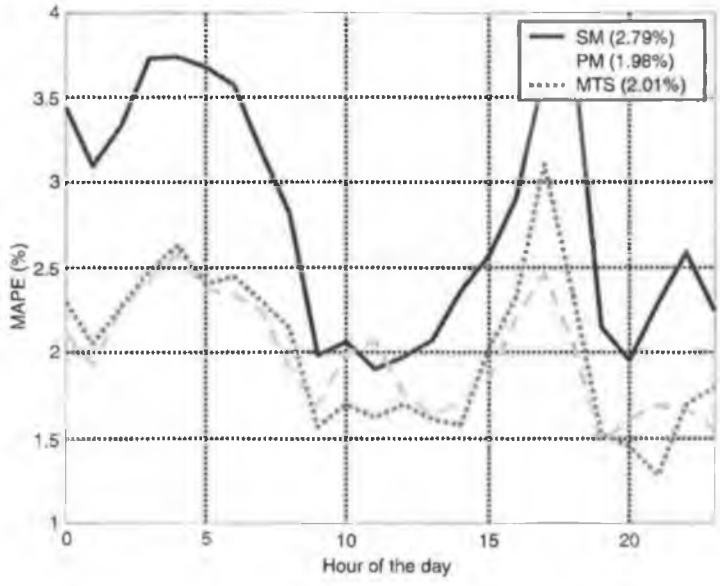

Figure 6.16. MAPE as a function of the hour of the day using for the SM, PMs and MTS model (optimised weight matrix 3, late winter working days, novelty set).

Figures 6.17, below, features the same graphs as Figure 6.16, above, except that the data used is the summer working day day-type (novelty set). As can be seen, the MTS and PMs forecasts again have similar accuracy at most hours of the day. However, it can be seen that at 5 p.m. and 6 p.m., the MTS forecasts are again inferior to the PM forecasts. In addition, the forecasts of load at 8 p.m. for the

summer working days, using the MTS model, are significantly inferior to those of the PMs.

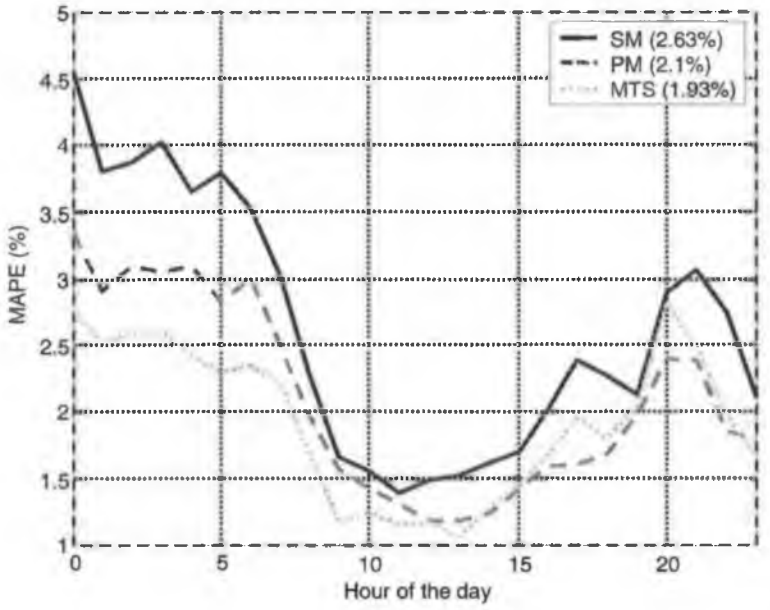

Figure 6.17. MAPE as a function of the hour of the day using for the SM, PMs and MTS model (optimised weight matrix 1, Summer working days, novelty set).

Figures 6.18, below, features the same graphs as Figure 6.17 above, except that the data used is the early winter working day day-type (novelty set). Again it can be seen that at 5 p.m. and 6 p.m. the MTS forecasts are inferior to the PM forecasts. In addition, the MTS forecasts of load at 7 p.m. are significantly inferior to those of the PMs.

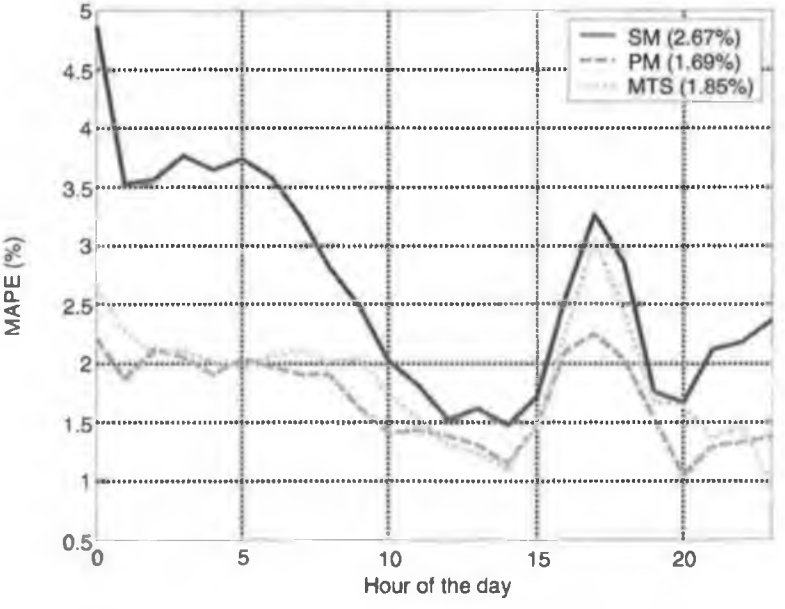

Figure 6.18. MAPE as a function of the hour of the day using for the SM, PMs and MTS model (optimised weight matrix 1, early winter working days, novelty set).

In Section 5.3 the partitioned series were examined for independence and it was found that at the hours of 8 a.m. and 5 to 8 p.m., the load may have independent components. This implied that a parallel approach would be superior to a sequential approach at these hours (see Section 3.2.2). The MTS model, which is a mix of parallel and SMs, performs well compared to the PMs at all hours except between 5 p.m. and 8 p.m. as seen above. Thus there is more evidence to support the view that the load at these hours may have independent components. However, there appeared to be no significant differences between the MTS and PM performance at 8 a.m. and so the presence of an independent component at this hour of the day is questionable.

Tables 6.13 and 6.14 below, summarise the performance of the SMs, PMs and MTS models in all day-types. The MTS models used the three optimised weight matrices and the average of the MAPE is taken.

As can be seen, the MTS technique does not perform well for some weekend day-types compared to the PM forecasts (specifically in the early winter Sundays, late winter Sundays and early winter Saturday day-types). This is because the MTS technique attempts to maintain the shape of the SM forecast up to three days into the future (Section 6.4). However, as discussed in Section 6.2.4, there is a 'shift' in the load after *every* day in the weekend day-types. The MTS forecasts do not have this shift and so have the same problem encountered with the sequential models in Section 6.2.4. In the case of the SM, the state vector, $\boldsymbol{\theta}(k)$, could be changed to account for the 'shift' in the load by estimating the trend and seasonal states separately (Section 6.2.4). However, the MTS technique estimates an altered state vector, $\boldsymbol{\theta}^*(k)$ (see Section 3.3.4.2), via a weighted least squares solution and so estimating the trend and seasonal component separately may not be possible. The working day day-types also have a 'shift' in the load (Section 6.2.4). However, in this case it occurs after every five days and does not affect performance dramatically.

The MTS forecasts perform well compared to the PM forecasts during summer Saturdays and summer Sundays. A possible reason for this, despite the effect of the 'shift' described above, is that load is relatively uncorrelated to weather during the summer months (Section 4.2.2). Thus the 'shifts' are not as pronounced in summer as they are in winter.

Table 6.13 Daily MAPEs for PMs, SM and MTS model (validation set).

Model Type	Early winter Sundays	Summer Sundays	Late winter Sundays	Early winter working days	Summer working days	Late winter working days	Early winter Saturdays	Summer Saturdays	Late winter Saturdays
SM	4.20	3.49	3.70	3.04	2.88	2.90	4.06	2.98	2.75
PM	2.41	2.98	2.32	1.92	2.23	1.89	2.52	2.90	2.28
MTS*	3.85	3.33	3.22	2.24	2.17	2.11	3.76	2.82	2.79

Table 6.14 Daily MAPEs for PMs, SM and MTS model (novelty set).

Model Type	Early winter Sundays	Summer Sundays	Late winter Sundays	Early winter working days	Summer working days	Late winter working days	Early winter Saturdays	Summer Saturdays	Late winter Saturdays
SM	4.23	3.46	3.59	2.67	2.63	2.79	4.19	2.36	2.30
PM	***2.67***	3.21	***2.39***	***1.69***	2.10	***1.98***	***2.23***	2.68	***2.23***
MTS	4.02	***3.11***	3.28	1.86	***1.95***	2.07	3.62	***2.15***	2.34

6.6 Conclusion.

The multi-timescale technique consists of combining the forecasts of several different models. The models combined (the sequential models, parallel models and end-sum models) were presented, and methods for determining the weight matrix used in the technique were examined in depth. Finally, a comparison of the MTS and parallel models was performed.

The sequential models were examined in Section 6.2, but it was found that they performed inadequately. Partitioning the data into day-types results in several intervening days being removed (Section 5.2). This in turn resulted in a *shift* in the load at the point were the days were removed (Section 6.2.4, Figure 6.5). The underlying requirement of a sequential approach is that a data set is hour of the day independent. However, the shift mentioned above occurs only at 23 hrs and 0

hrs, i.e. between days. Thus, partitioning the data by day-type produces data sets that are not hour of the day independent.

For a working-day day-type, a shift occurs once every 5 days, which may appear negligible. However, as was seen in Section 6.2.4, this has an adverse effect on estimation of the seasonal component in the SM. The solution involved modelling the trend component as constant at all hours of a day (Section 6.2.4). This is equivalent to modelling the trend using a parallel approach, as there is only one value that needs to be estimated per day.

The multi-timescale technique was examined in Section 6.4. An essential part of this technique is determining appropriate values for the weight matrix. A novel technique was examined in which the weights are determined by means of numerical optimisation. It was found that the weights determined improved the performance of the MTS technique above previous procedures (i.e. weight profiles, Section 6.4.1.2).

Three optimised weight matrices were trained for each day-type (Section 6.4.1.3). It was found that each optimised weight matrix led to good results when used in the MTS technique. This is important as the optimisation procedure is computationally expensive, and is impractical to perform numerous times.

Section 6.4.2 examined the theory behind the MTS technique. It was proposed that the MTS technique should be viewed as a regularisation term, as the estimation of the altered state vector, $\boldsymbol{\theta}_2^*(k, \boldsymbol{w})$, is not related to forecasting error, but rather to the shape of the multi-step ahead forecasts. A technique based on constrained optimisation was then proposed for altering the MTS technique to include a measure of the forecast error. This is achieved by optimising the weight matrix as a function of forecast error. However, due to the complexity of the equations involved, a closed form solution cannot be found and a numerical solution is instead proposed.

---

* Three optimised weight matrices are used and an average MAPE taken.

In Section 6.5 the MTS technique was compared to the parallel models. The results showed that overall the MTS and parallel models provide similar forecast accuracy. At some hours of the day, the MTS forecasts are marginally superior to the parallel model forecasts. However, at other hours of the day (5 p.m. to 8 p.m.) the parallel model forecasts are significantly better than the MTS forecasts.

Finally, due to the problems with a shift in the load between day-types, the MTS technique was found not to perform well during weekend days in winter (Section 6.5). In summer the shift is less significant and the MTS technique was found to compare well to the PMs during weekend days.

Chapter 7

Fusion of Models for Load Forecasting.

7.1 Introduction.

Model fusion consists of taking the forecasts from several models (which forecast the same variable) and combining them into one forecast known as a *fused forecast*. The concept of model fusion is well known in the general field of forecasting and was pioneered mainly by Bates and Granger (1969) and Reid (1968). Fused forecasts are often more accurate than any of the individual model forecasts (Palit and Popovic, 2000, Xiong et. al., 2001, Shamseldin *et. al.*, 1997 and Perrone and Cooper, 1993, among others). This is because different models are often better at modelling different aspects of an underlying process and thus combining the models appropriately gives a better forecast. In addition, a single model incorporating all aspects of an underlying process may be more complex and difficult to train than the individual models (Palit and Popovic, 2000). The concept of model fusion therefore agrees with the 'divide and conquer' strategy discussed in Section 4.2.1 (in that case segmentation of the data set was discussed).

Model fusion has been applied to many fields. Examples are rainfall run-off models (Shamseldin *et. al.*, 1997, Xiong *et. al.*, 2001), wine analysis (Rong et. al., 2000) and chemical processes (Sridhar *et. al.*, 1999). However, at the time of this author's publication in the area, model fusion was new to the field of short - term load forecasting (Fay et. al., 2000). Model fusion is particularly suited to short term load forecasting, due to the problems encountered when using weather forecasts in STLF models (Section 3.5). In Section 3.5 it was pointed out that STLF models should be trained with actual weather inputs, even though they are used with weather forecasts. However, if information is available on weather forecast errors, and this can be incorporated into the STLF model, then models using forecasted inputs can be substantially improved.

Model fusion offers the following solution procedure:

1. Several models (called *sub-models*) are first trained. The models differ in the inputs presented, e.g. the weather and non-weather inputs are presented to different models, and

2. The sub-model forecasts are fused together. At this point the errors in the weather forecasts may be taken into account.

This solution methodology allows the problem of weather forecast errors to be dealt separately from the parameter evaluation stage of the model.

In general terms, model fusion may be expressed as a function approximation problem of the form:

$$f_{\text{fuse}}(f_1(\boldsymbol{x}),f_2(\boldsymbol{x}),\ldots,f_M(\boldsymbol{x})) = f(\boldsymbol{x}) + \varepsilon \tag{7.1}$$

where $f(\boldsymbol{x})$ is the function to be approximated, $\boldsymbol{x}$ is a vector of inputs, f_{fuse} is the model fusion algorithm, $f_i(\mathbf{x})$ is sub-model i, N is the number of sub-models used and ε is an associated error term. Note that each sub-model also attempts to approximate the function $f(\boldsymbol{x})$ i.e.:

$$f_i(\boldsymbol{x}) = f(\boldsymbol{x}) + \nu_i \tag{7.2}$$

where $f_i(\boldsymbol{x})$ is the function implemented by sub-model i and ν_i is the error associated with that sub-model.

As with STLF techniques, model fusion techniques may be categorised as either linear or non-linear. In the linear case, Equation (7.1) may be expressed as a weighted sum of the sub-model forecasts (McCabe, 1991), as:

$$f(\boldsymbol{x}) = w_1 f_1(\boldsymbol{x}) + w_2 f_2(\boldsymbol{x}) + \ldots + w_N f_N(\boldsymbol{x}) \ldots + \varepsilon \tag{7.3}$$

where w_i is the weight applied to sub-model forecast i. There are several linear fusion algorithms available but they all share the advantage that the weights may be determined *uniquely* and *optimally* (in some sense, see below) from the data set. For example, McCabe's fusion algorithm minimises the trace of the covariance matrix of fused forecast errors (see Section 7.4.2) while the weighted average method (Shamseldin et. al., 1997) minimises the sum of squared errors.

Thus linear combination methods will give an optimal result (in the *linear least variance* sense in the case of McCabe's algorithm (1991) and the best *linear fit* sense in the case of the weighted average method).

As pointed out by Fiordoliso, (1998), the function approximation performed by Equation (7.3) is not a universal approximator. That is; linear fusion models do not have the ability to approximate an arbitrary function with an arbitrarily small error. This is because linear fusion models can only express $f(\boldsymbol{x})$ as a (weighted) sum of several other functions (i.e. Equation 7.3). In the opinion of Palit and Popovic (2000) it is unlikely that an underlying process $f(\boldsymbol{x})$ can be expressed as such. Thus they conclude that the use of linear fusion models is questionable.

Non-linear fusion models use a non-linear function to implement $f_{\text{fuse}}(\bullet)$ in Equation (7.1). Several techniques may be used, such as fuzzy logic (Section 3.4.2) or neural networks (Section 3.4.3). Although fuzzy logic and neural network techniques are themselves universal approximators, the function approximation implemented by the overall *non-linear model fusion* is not (contrary to the view expressed by Fiordoliso, 1998). This is because information may be lost by the sub-models which cannot be recovered by $f_{\text{fuse}}(\bullet)$ (Palit and Popovic, 2000).

In contrast to the linear fusion model, non-linear fusion modelling techniques do not have a unique structure or solution for the parameters. As discussed in Sections (3.4.2) and (3.4.3) it may be difficult to obtain an optimal or near optimal model.

In conclusion, linear and non-linear fusion models both have disadvantages which are dependent on the underlying process and so the choice between them is application specific.

In practical terms, the first factor to be noted is the availability of the weather forecast data. From Table 2.2 (Section 2.2) weather forecasts are available for the period of 1st February 2000 to 1st March 2000. These forecasts are available only for temperature, cloud cover, wind speed and wind direction; thus humidity

forecasts are not available. In addition, the weather forecasts that are available are for days in the late winter day-types and so these are the only day-types analysed. Also note that statistics on the weather forecasts such as confidence intervals etc. are *not* available.

7.2 Modelling Weather Forecast Errors.

Due to the sparseness of weather forecasts and the fact that weather forecast errors change over time (see Section 3.5), it is necessary to model the error on the weather forecasts to produce what are termed *pseudo-weather forecasts*. The approach here is to first model the temperature forecast errors as temperature is the dominant input variable. The other weather forecast errors are then analysed.

7.2.1 Modelling Temperature Forecast Errors.

Figure 7.1 below shows a plot of the actual and forecast temperatures for 1st February 2000 to 1st March 2000. As can be seen, the temperature forecasts are reasonably accurate.

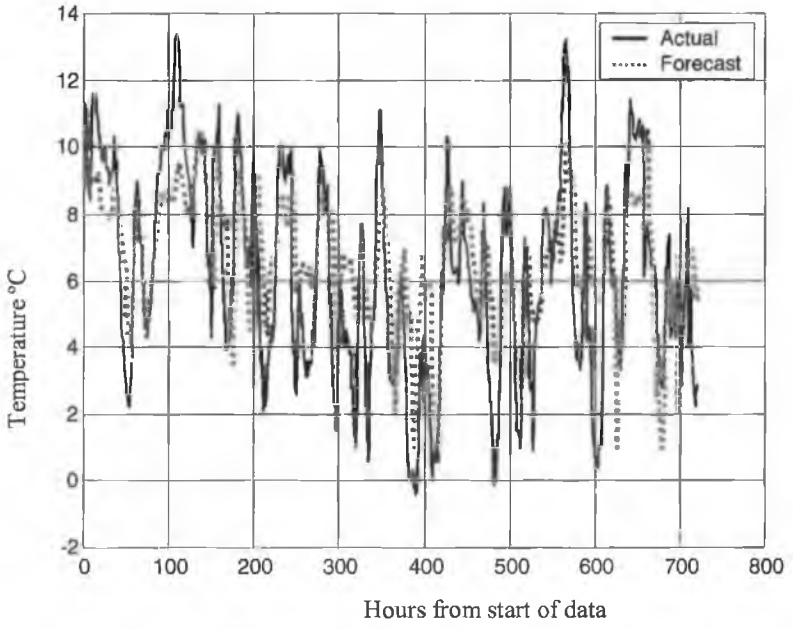

Figure 7.1. Actual and forecast temperature (1st February 2000 to 1st March 2000).

However, the errors in the forecasts do not seem to follow a Gaussian distribution but rather are either above or below the actual for prolonged periods (Figure 7.2). The weather in Ireland is dominated by Atlantic weather systems.

When a weather system or *front* reaches Ireland, there is a shift in the level of the temperature and other weather variables. This shift is also a factor that the Irish Meteorological Office must forecast. In Figure 7.2 it can be seen that the sign of the weather forecast error typically changes when there is a large change in the temperature.

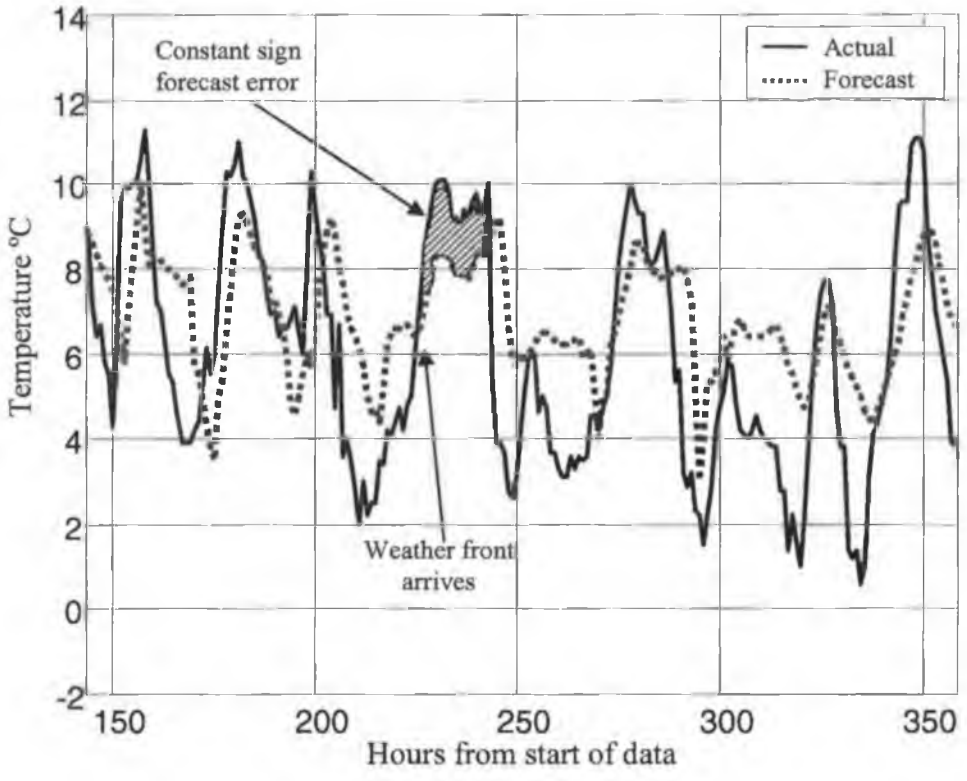

Figure 7.2. Actual and forecast temperature (6th to 15th February 2000).

The points at which there is a shift in the temperature are called *turning points*. Several different procedures were attempted to find the turning points. However, the following procedure was found to give excellent results:

1. The temperature is first smoothed to provide an easier means of detecting the turning points. To this end the temperature is filtered using an IRW (Section 3.2.2.2). The states of the filter are estimated using a Kalman filter (Section 3.3.2.1). In contrast to previous state space approaches in the current research, the *a posteriori* state estimate, $\hat{\boldsymbol{\theta}}^{+}(k)$, (as opposed to the *a priori* state estimate) is used to estimate the temperature. The smoothed temperature at time k, $\widehat{T}(k)$ is then:

$$\bar{T}(k) = \boldsymbol{H}_{irw}\hat{\boldsymbol{\theta}}^{+}(k) \tag{7.4}$$

Where $\boldsymbol{H}_{irw}$ is the observation matrix of an IRW (Section 3.3.2.2). The *a posteriori* state estimate is used, because the aim is to smooth the temperature and not to forecast it. The noise-variance ratio, ξ, (Section 3.3.2.4) of the filter was set to .03. This algorithm was found to be robust to a wide range of values for ξ., and

2. A window, with a width of eleven hours, is then passed over the smoothed temperature. If the smoothed temperature at the centre of the window is the maximum within that window:

$$\hat{T}(k) = \max\left[\hat{T}(k-5), \hat{T}(k+5)\right] \tag{7.5}$$

and it is greater than half the average value of the other smoothed temperatures in the window:

$$\hat{T}(k) \geq .5\sum_{i=0}^{11} \frac{\hat{T}(k-5+i)}{10} \qquad i \neq 5 \tag{7.6}$$

Then point k is designated a high turning point. The procedure for finding a low turning point is equivalent except that the minimum is used in Equation (7.5). Figure 7.3 below shows the actual weather, the weather forecasts and the high and low turning points.

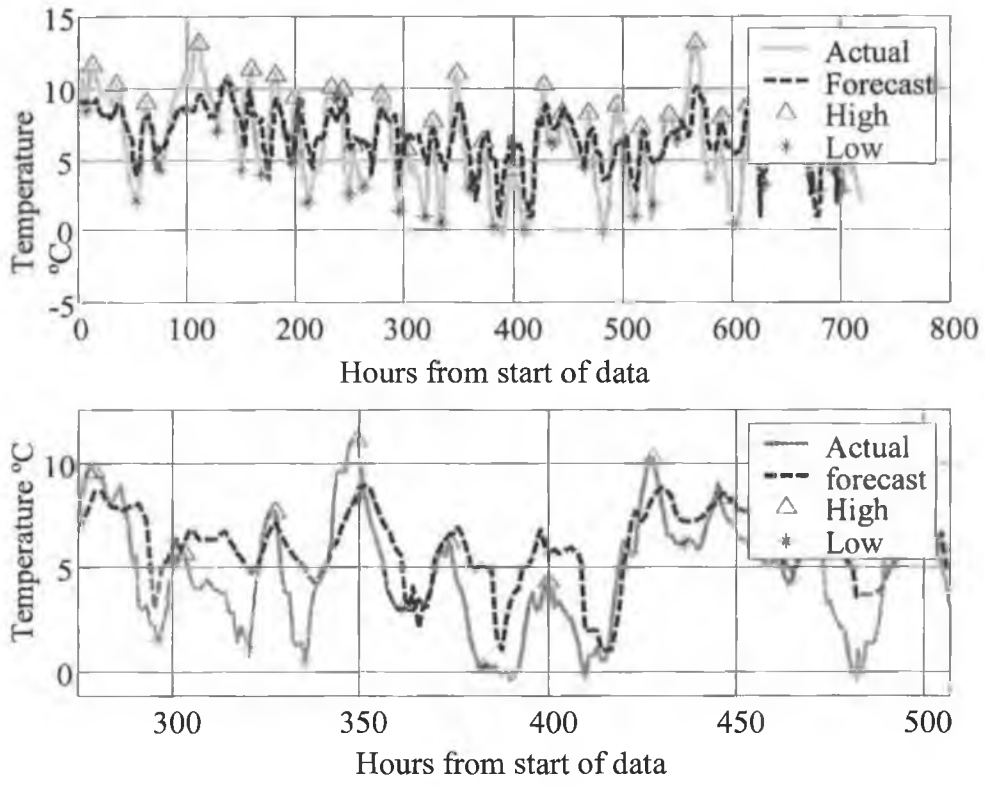

Figure 7.3. Detected turning points in temperature (1st February 2000 to 1st March 2000, second graph is a close-up).

Had the Irish Meteorological Office known the level of the temperature in advance then it is the assumption of this analysis that they would have no forecast errors. To test this assumption the mean (called the *level* here) and standard deviation (called the *shape* here) of the actual and forecast temperatures *between the turning points* is calculated. An adjusted forecast is then produced by removing the level and shape of the temperature forecast and introducing the actual level and shape as:

$$\tilde{T}(k) = \left(\frac{\hat{T}(k) - \overline{\hat{T}}_{k_{tp_i}, k_{tp_{i+1}}}}{\sigma_{\hat{T}_{k_{tp_i}, k_{tp_{i+1}}}}} \right) \sigma_{T_{k_{tp_i}, k_{tp_{i+1}}}} + \overline{T}_{k_{tp_i}, k_{tp_{i+1}}} \qquad k \in \left[k_{tp_i}, k_{tp_{i+1}} \right] \qquad (7.7)$$

Where $\tilde{T}(k)$ is the adjusted temperature forecast k hours from the start of the forecast data (1st February 2000), $\hat{T}(k)$ is the forecast temperature and k_{tp_i} is the i^{th} turning points. The following statistics are calculated between the i^{th} and i+1th turning points k_{tp_i} and $k_{tp_{i+1}}$; $\overline{\hat{T}}_{k_{tp_i}, k_{tp_{i+1}}}$ is the average forecast temperature or

level, $\sigma_{\hat{T}_{k_{tp_i},k_{tp_{i+1}}}}$ is the forecast temperature standard deviation or shape, $\overline{T}_{k_{tp_i},k_{tp_{i+1}}}$ is the average (or level) of the actual temperature and $\sigma_{T_{k_{tp_i},k_{tp_{i+1}}}}$ is the standard deviation (or shape) of the actual temperature. Figure 7.4 below shows the adjusted temperature forecast and the actual temperature.

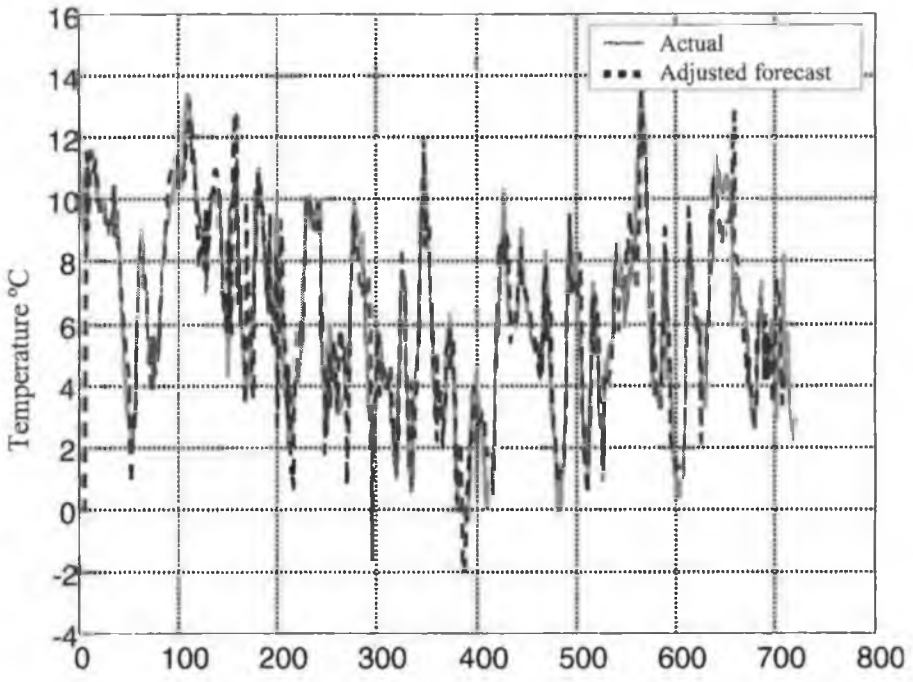

Figure 7.4. Adjusted temperature forecast and actual temperature (1st February 2000 to 1st March 2000).

As can be seen, there is good agreement between the adjusted and actual temperatures. The temperature forecast errors can now be divided into the following components:

- The difference between the adjusted and actual temperature, $\tilde{T}(k)-T(k)$, called the *random error*,
- The difference between the forecast temperature level and the actual level (between the turning points), $\overline{T}_{k_{tp_i},k_{tp_{i+1}}} - \overline{\hat{T}}_{k_{tp_i},k_{tp_{i+1}}}$, called the *level error*, and
- The difference between the forecast temperature shape and the actual shape, $\sigma_{\hat{T}_{k_{tp_i},k_{tp_{i+1}}}} - \sigma_{T_{k_{tp_i},k_{tp_{i+1}}}}$, called the *shape error*.

Figure 7.5 below, shows the histograms of the three errors listed above with Gaussian distributions which have been fitted to the data. In addition the SACF of the random temperature forecast errors is shown. Although the SACF of the *random* forecast errors is statistically significant for some lags the auto-correlation is still quite small and thus it can be assumed that the *random forecast error* are indeed taken from a random population. Similarly, the level and shape errors are found to be taken from a random population.

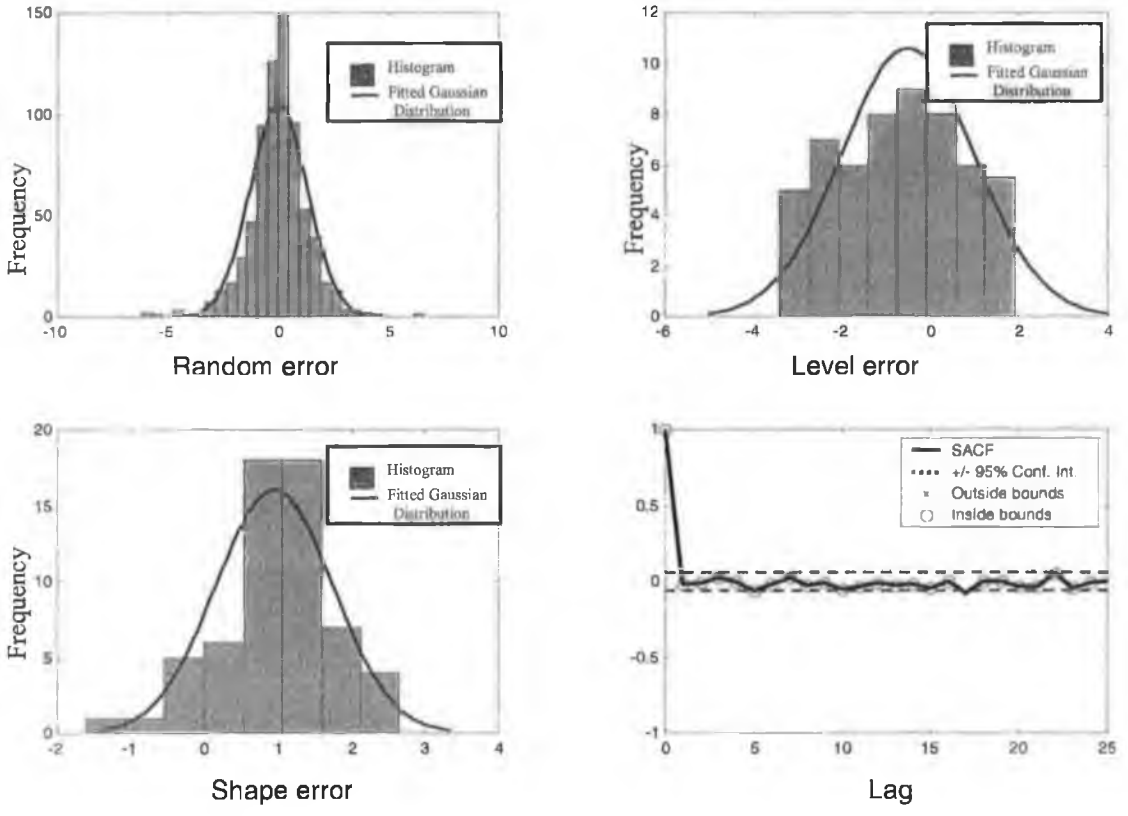

Figure 7.5. Various statistics of temperature forecast errors.

As can be seen, the Gaussian distributions fit the histograms well. Thus, it is assumed that the temperature forecast shape, level and random errors are normally distributed.

The values of the fitted normal distributions in Figure 7.5 are shown in Table 7.1 below.

Table 7.1 Fitted normal distribution values for the difference between weather forecast statistics and actual weather statistics.

Distribution	Mean	Standard Deviation
$\tilde{T}(k)$ - $T(k)$ (Random)	-.5312	1.4902
$\bar{\tilde{T}}_{k_{tp_i},k_{tp_{i+1}}}$ - $\bar{\hat{T}}_{k_{tp_i},k_{tp_{i+1}}}$ (Level)	0.0292	1.3131
$\sigma_{\hat{T}_{k_{tp_i},k_{tp_{i+1}}}}$ - $\sigma_{\tilde{T}_{k_{tp_i},k_{tp_{i+1}}}}$ (Shape)	0.9407	0.8008

Reversing the previous analysis gives a mechanism by which pseudo-temperature forecast errors can be produced. This process involves the following algorithm called the *pseudo-weather forecast generation algorithm*:

1. The turning points in the actual temperature are identified,

2. A Gaussian random number generator is used to generate pseudo values for the random temperature forecast errors (using the values in Table 7.1). These are then subtracted from the actual temperature to produce a *pseudo adjusted weather forecast*, $\tilde{T}'(k)$,

3. A Gaussian random number generator (using the distribution values in Table 7.1) is used to generate *pseudo* values for $\bar{\hat{T}}_{k_{tp_i},k_{tp_{i+1}}}$ and $\sigma_{\hat{T}_{k_{tp_i},k_{tp_{i+1}}}}$ denoted $\bar{\hat{T}}'_{k_{tp_i},k_{tp_{i+1}}}$ and $\sigma'_{\hat{T}_{k_{tp_i},k_{tp_{i+1}}}}$ respectively, and

4. Equation (7.7) is inverted as:

$$\hat{T}'(k) = \left(\frac{\tilde{T}'(k) - \bar{T}_{k_{tp_i},k_{tp_{i+1}}}}{\sigma_{T_{k_{tp_i},k_{tp_{i+1}}}}} \right) \sigma'_{\hat{T}_{k_{tp_i},k_{tp_{i+1}}}} + \bar{\hat{T}}'_{k_{tp_i},k_{tp_{i+1}}} \qquad k \in [k_{tp_i}, k_{tp_{i+1}}] \qquad (7.8)$$

Where $\hat{T}'(k)$ is the pseudo weather forecast required.

Figure 7.6 below shows the pseudo-temperature forecasts produced for the period 1st February 2000 to 1st March 2000.

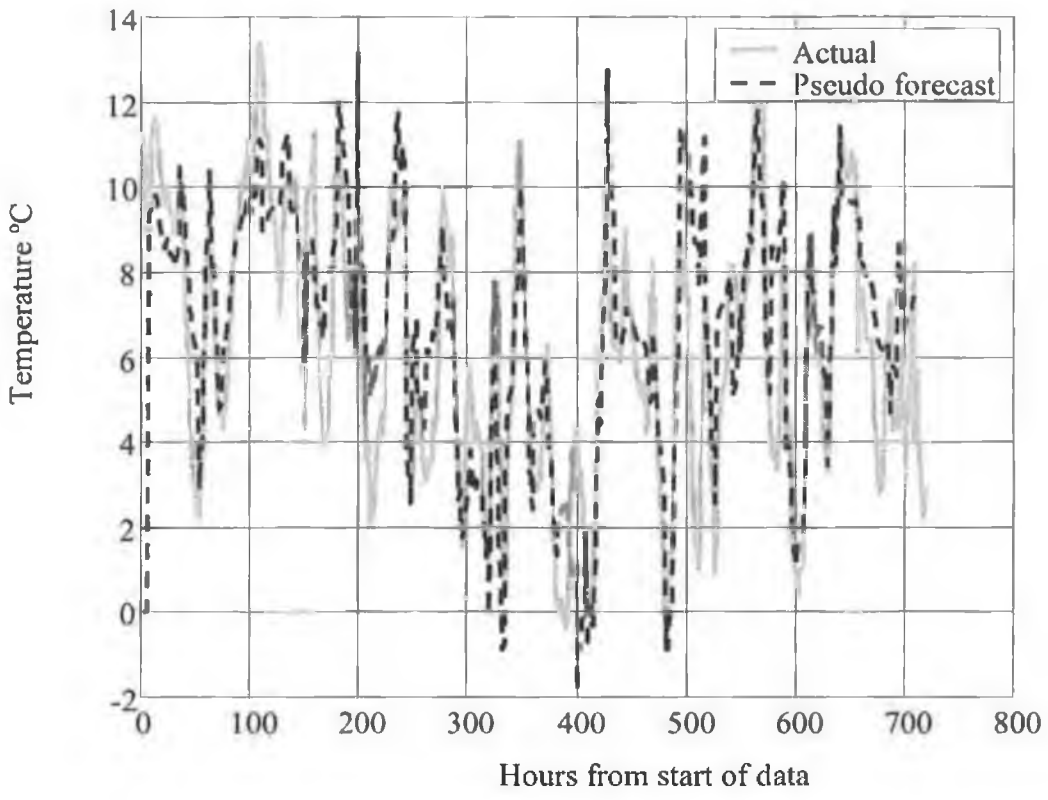

Figure 7.6. Actual and pseudo-forecast temperature (1st February 2000 to 1st March 2000).

Figure 7.7 shows the SACFs of the temperature forecast errors and the pseudo-temperature forecast errors. As can be seen, they are in good agreement, showing that the pseudo-temperature forecast errors have similar statistics to the temperature forecast errors. Finally, the sum squared error of the weather forecast errors (3.32×10^3) compares well with that of the pseudo-weather forecast errors (3.02×10^3).

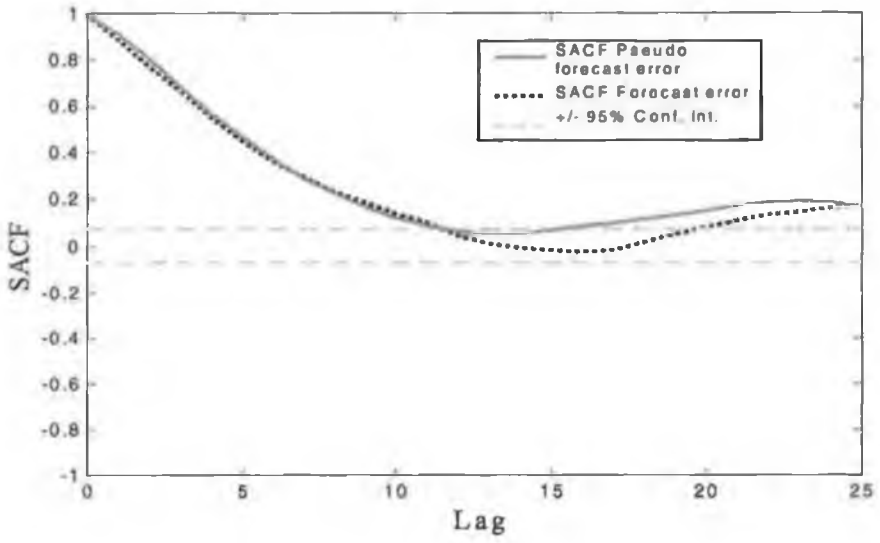

Figure 7.7. SACF of forecast and pseudo-forecast temperature errors.

7.2.2 Modelling Cloud Cover, Wind Speed and Wind Direction Forecast Errors.

The cloud cover, wind speed, and wind direction weather forecast errors are modelled in a similar way to the temperature forecast errors. Note that the turning points are assumed to be the same for all weather variables. Figures 7.8 to 7.10 show the distributions and SACFs of these weather variables.

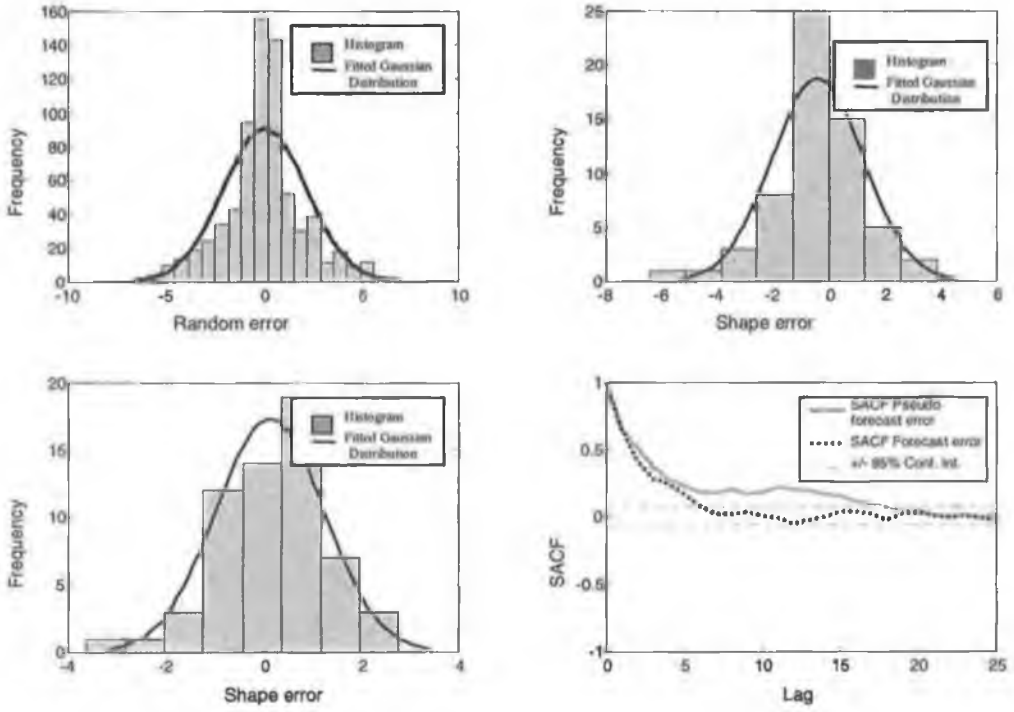

Figure 7.8. Histograms for cloud cover forecast errors and the SACF of the cloud cover forecast errors and the pseudo forecast errors.

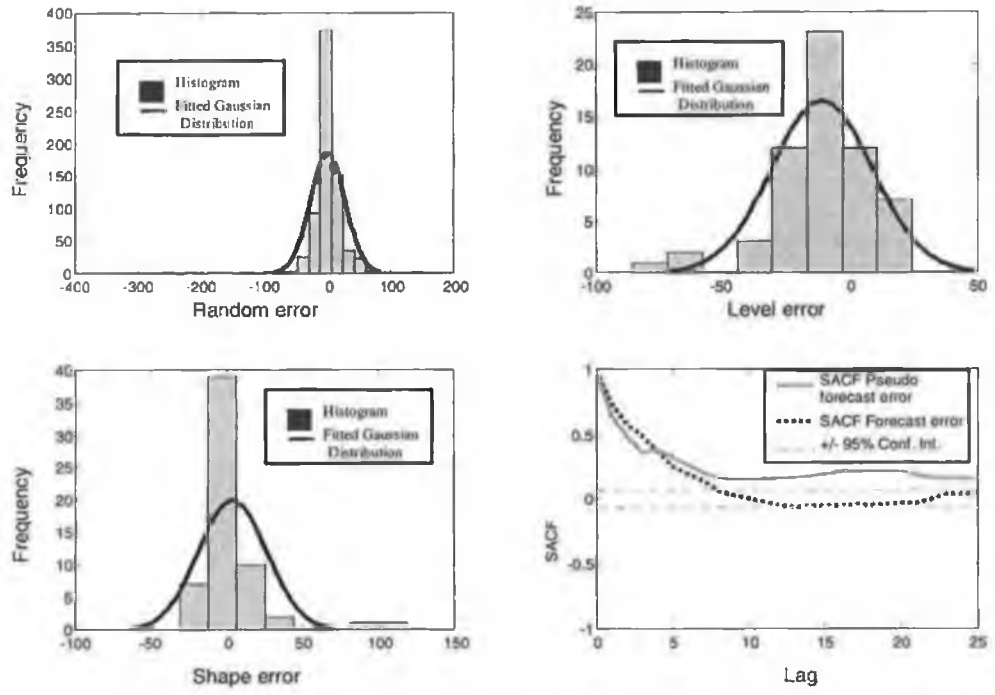

Figure 7.9. Histograms for wind direction forecast errors and the SACF of the wind direction forecast errors and the pseudo forecast errors.

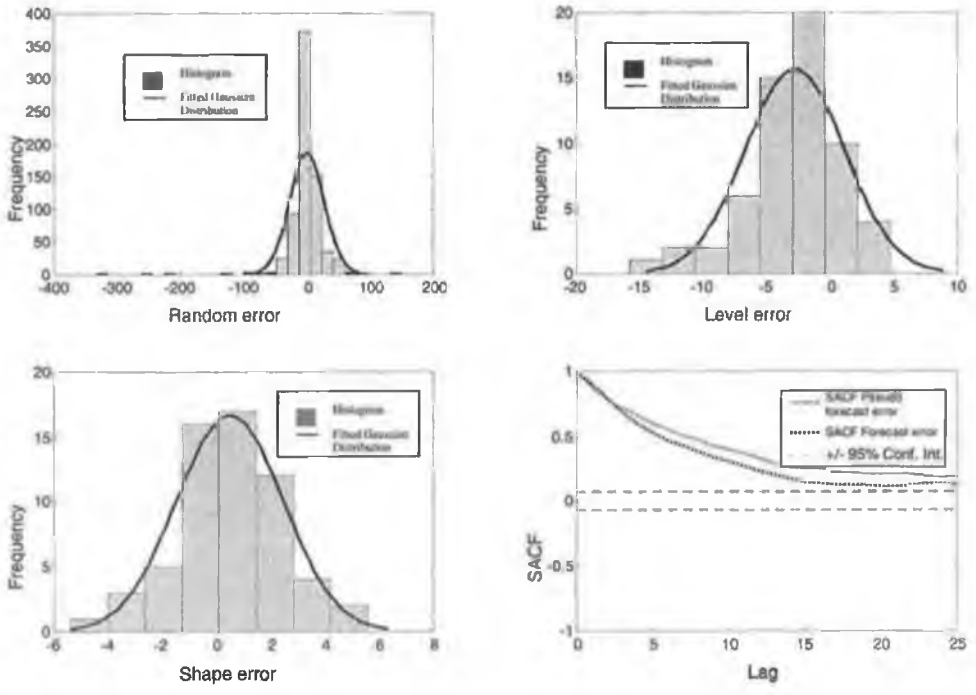

Figure 7.10. Histograms for wind speed forecast errors and the SACF of the wind speed forecast errors and the pseudo forecast errors.

7.2.3 Joint Modelling of Weather Forecast Errors.

The cross-correlation (co-efficients) of the three types of weather forecast errors in temperature, cloud cover, wind speed, and wind direction weather forecast errors shown in Tables 7.2 and 7.3 below*.

Table 7.2 The cross-correlation matrix of random forecast errors of temperature, cloud cover, wind speed and wind direction.

Random error type	Temperature	Cloud Cover	Wind Direction	Wind Speed
Temperature	**1.00**	0.06	0.02	*0.21*
Cloud Cover	0.06	**1.00**	0.02	0.00
Wind Direction	0.02	0.02	**1.00**	0.06
Wind Speed	*0.21*	0.00	0.06	**1.00**

Table 7.3 The cross-correlation matrix of shape and level forecast errors of temperature, cloud cover, wind speed and wind direction.

Error type	Temperature Level	Temperature Shape	Cloud Cover Level	Cloud Cover Shape	Wind Direction Level	Wind Direction Shape	Wind Speed Level	Wind Speed Shape
Temperature Level	**1.00**	*-0.26*	0.17	0.16	0.26	-0.02	0.32	0.16
Temperature Shape	*-0.26*	**1.00**	*-0.37*	0.16	0.00	-0.02	-0.01	0.04
Cloud Cover Level	0.17	*-0.37*	**1.00**	-0.09	0.08	0.08	0.02	0.14
Cloud Cover Shape	0.16	0.16	-0.09	**1.00**	0.08	0.19	0.04	0.17
Wind Direction Level	*0.26*	0.00	0.08	0.08	**1.00**	*-0.37*	0.07	0.19
Wind Direction Shape	-0.02	-0.02	0.08	0.19	-0.37	**1.00**	-0.22	0.26
Wind Speed Level	***0.32***	-0.01	0.02	0.04	0.07	-0.22	**1.00**	-0.19
Wind Speed Shape	0.16	0.04	0.14	0.17	0.19	*0.26*	-0.19	**1.00**

As some of the cross-correlations are quite large this implies that some of the errors are jointly distributed. For example, as the cross-correlation coefficient between the wind speed level error and temperature level error has a correlation coefficient of 0.32 (highlighted in Table 7.3), these errors are most likely correlated.

It is assumed that as the distributions of the all the weather forecast errors (i.e. the random, level and shape errors) are normally distributed, they are also *jointly* normally distributed.

* As the random weather forecast errors are not serially correlated (Figure 7.5) they are assumed to be uncorrelated to the level and shape forecasts and so are shown separately.

In order to take account of the fact that the weather forecast errors are jointly normally distributed the pseudo-weather forecast generation algorithm laid out in Section 7.2.1 must be adapted. This is achieved by changing step 3 of the algorithm to:

3. A multivariate Gaussian random number generator (using the distribution values in Table 7.2) is used to generate *pseudo* values for the random errors in temperature, cloud cover, wind speed and wind direction *simultaneously*. A multivariate Gaussian random number generator (using the distribution values in Table 7.3) is then used is used to generate *pseudo* values for the level and shape errors in temperature, cloud cover, wind speed and wind direction *simultaneously*.

The multivariate Gaussian random number generator used is based on the Cholesky decomposition and details may be found in Dagpunar (1988). The jointly generated pseudo-weather forecasts are those used in the remainder of this chapter.

7.3 Choice of Load Forecasting Model.

Four models were identified in Section 3.5.2. for minimising the effect of weather forecast error on load forecasts:

1. The fuzzy logic model of Bitzer and Rößer (1998),

2. The fuzzy logic model of Rahman and Hazim (1993),

3. The ensemble models of Taylor and Buizza (2003), and

4. The Hammerstein model of Miyake *et. al.*, (1995).

The first approach seeks to reduce the effect of weather forecast error by fuzzification of the inputs. This is equivalent to using a qualitative rather than quantitative measure of the inputs (e.g. temperature is *high*). It is presumed

however that the magnitude of the weather forecast error is not sufficient to change the classification of the inputs (e.g. temperature may be classed as *high* when in fact it is *low*). The approach here however is to deal with the weather forecast errors in a quantitative way.

The second and third approaches uses statistics of the weather forecasts as inputs These are provided for each weather forecast by the weather forecast service used by Rahman and Hazim (1993) and by Taylor and Buizza (2003). As confidence intervals or any other statistics regarding individual weather forecasts are not available in the current research this approach cannot be used.

The third approach (Miyake *et. al.*, 1995) constructs a set of N candidate models (or *sub-models*) each with differing inputs and selects the best model (Section 3.5.2). Each sub-model is evaluated using the FPEEV (Section 3.5.2) which is a function of the error covariance matrix of the weather forecast errors. This approach has the advantage that the effect of weather forecast error can be separated by sub-model. This circumvents the problems created when weather forecast errors are not present in the training set (Section 3.5) as the sub-model which performs best with weather forecast errors is chosen.

However, the FPEEV is specific to the Hammerstein sub-models used. As the load forecasting models used in the current research (Chapters 5 and 6) are not Hammerstein models (for reasons given in Section 5.7.2), the Miyake *et. al.*, (1995) approach is not used in the current research. Nevertheless, the principles used in this approach may be generalised.

The approach of Miyake *et. al.*, (1995) may be viewed as *a gating network* (Haykin, 1999). That is, a weighted sum approach in which the model with the best FPEEV is given a weight of 1 and all others a weight of 0 (Figure 7.11).

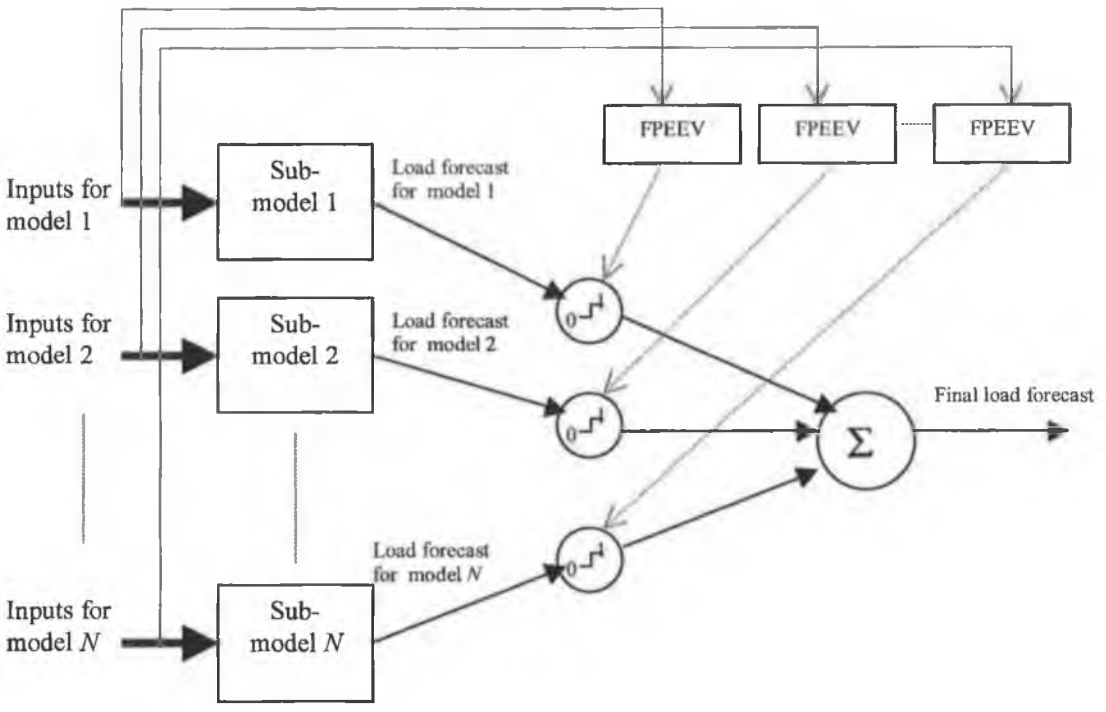

Figure 7.11. A weighted sum view of the approach of Miyake *et. al.*, (1995).

The binary weights used by Miyake *et. al.*, (1995) may be generalised by combining the output of the models i.e. *model fusion.*

As the FPEEV is model specific, it can only be used to weight the output of Hammerstein models. In the approach proposed here however, the data fusion algorithm of McCabe (1991) (explained in Section 7.4.2) was chosen as it is not model specific. This algorithm depends on the *covariance matrix of sub-model forecast errors* (explained in more detail in Section 7.4.2). In addition, the distribution of the sub-model forecast errors is required only to be un-biased and symmetric i.e. normality is not required. This is an advantage as the weather forecast inputs to the sub-models are not Gaussian as discussed in Section 7.2. Also, the weights can be easily changed to cater for perceived shifts in the relative significance of the various forecasts used to form the composite output. The entire model is called the *data fusion model* and is shown in Figure 7.12 below.

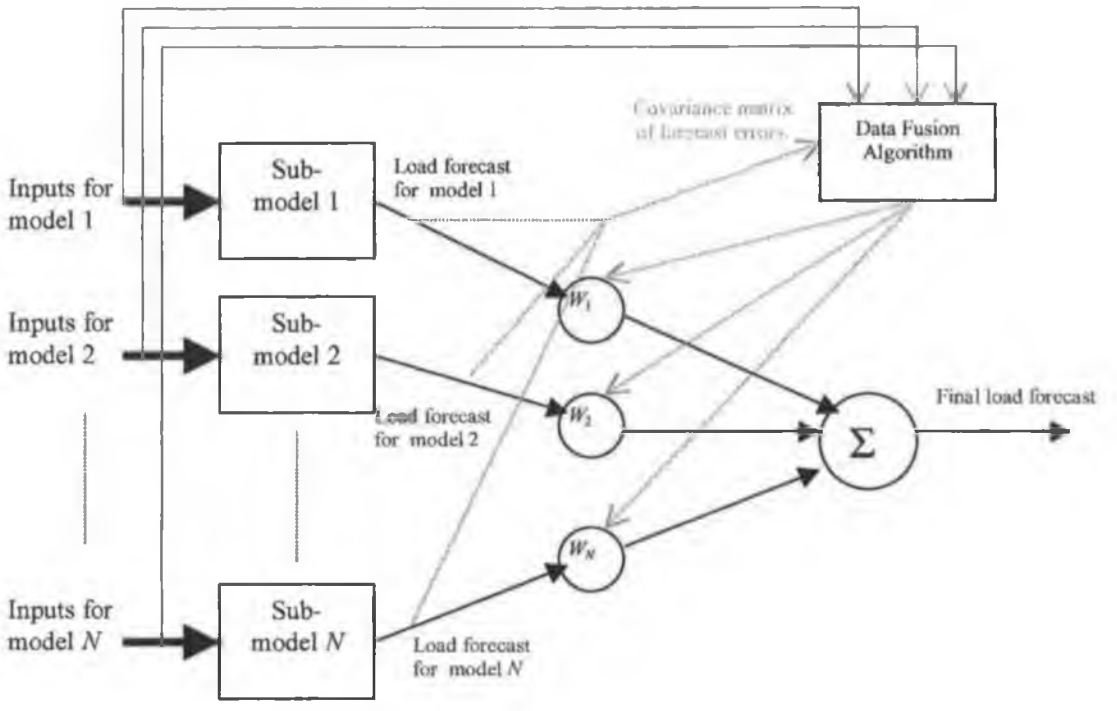

Figure 7.12. The data fusion model overview.

7.4 Sub-Modelling of Partitioned Series.

As shown in Figure 7.12 the Data Fusion Model consists of several sub-models which have their outputs combined to form a final forecast. The sub-models used are similar to the best non-linear parallel models, which are described in Section 5.7.5 (Figure 7.13, notation explained below).

The following variables are re-defined (from Chapter 5) for clarity*:

- $d_{i,j}(k)$ and $\psi_{i,j}(k)$ are the trend and seasonal components, respectively, for hour i on day k in day-type j in the Preliminary Parallel Model (PPM),
- $NN_{i,j}$ denotes a neural network for hour i on day k in day-type j,
- $x_{i,j}(k)$ is the load residual from the PPM and $\hat{x}_{i,j}(k)$ is the estimated load residual given by $NN_{i,j}$
- $y_{i,j}(k)$ and $\hat{y}_{i,j}(k)$ are the actual and estimated load respectively,

* Note that humidity is not included as forecasts of humidity are not available for this study.

- $T_{i,j}(k)$ the temperature,
- $\boldsymbol{t}_{i,j}(k)$, is a vector of pre-whitened temperatures from hour i to hour i-72 on day k of day-type j *with pseudo-weather forecast errors,*
- $\boldsymbol{q}_{i,j}(k)$, is a vector of pre-whitened wind speeds from hour i to hour i-72 on day k of day-type j *with pseudo-weather forecast errors,*
- $\boldsymbol{o}_{i,j}(k)$, is a vector of pre-whitened wind directions from hour i to hour i-72 on day k of day-type j *with pseudo-weather forecast errors*. As the wind direction is a circular measurement, i.e. 0° is equivalent to 360° the cosine *and* sine of this variable is used, and
- $\boldsymbol{c}_{i,j}(k)$, is a vector of pre-whitened cloud covers from hour i to hour i-72 on day k of day-type j *with pseudo-weather forecast errors*.

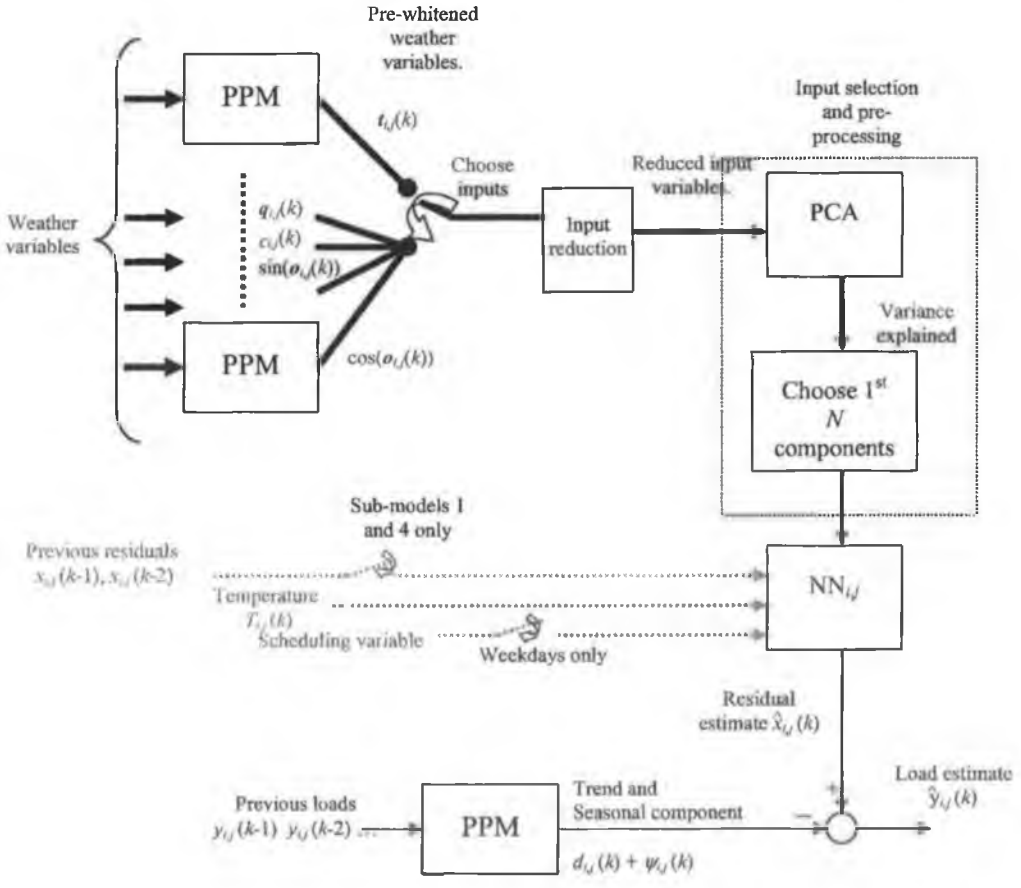

Figure 7.13. Sub-model overview (note: the inputs to $NN_{i,j}$ vary depending on the sub-model)

Three types of sub-models were chosen which have different types of inputs. These are chosen so that forecast errors can be attributed to particular inputs. A fourth sub-model type is included using all the available inputs to capture any non-linear relationships between the inputs and the residual. These sub-models are:

1. A model with AR inputs only; *Sub Model 1*,
2. A model with temperature inputs only; *Sub Model 2*,
3. A model with cloud cover, wind speed and wind direction inputs; *Sub Model 3*, and
4. A model with AR, temperature, cloud cover, wind speed and wind direction inputs; *Sub Model 4*.

As the sub-models are parallel models, there are 24×4 (one for each hour of the day and four sub-model types) sub-models per day-type*.

7.4.1 Input Selection and Pre-processing.

All of the sub-models use the operating point input, as they all forecast the load for the late winter day-types (see Section 5.7.5). The scheduling variable is used with all late winter weekday sub-models (see Section 5.7.5). Sub-model 1 uses the AR inputs selected in Section 5.2.

The external input selection and processing for sub-models 2 to 4 is similar to that used for the non-linear parallel models (Section 5.7.2).

7.4.2 Structure Determination.

The structure determination for the neural networks uses the same approach as for the non-linear parallel models (Section 5.7.3).

* Note that only the load forecasts of the sub-models for hour i on day-type j are combined together and not with the forecasts at any other hour or day-type.

7.4.3 Parameter Evaluation.

The parameter evaluation for the neural networks uses the same approach as for the non-linear parallel models (Section 5.7.4).

7.4.4 Model Validation.

Tables 7.4 and 7.5 below, show the AMAPEs achieved by each of the four models in the novelty, validation and training sets. Table 7.4 is for the case where actual weather inputs are used. Table 7.5 is for the case where pseudo-weather forecasts are used. As can be seen the performance of sub-models 2 to 4 deteriorates with the inclusion of pseudo-weather forecasts (Table 7.6). Sub-model 1 has no external inputs and so it's performance is unaffected by pseudo-weather forecast error (Table 7.6).

Sub-model 4 performs best in most day-types when using actual weather inputs (Table 7.4). However, with the inclusion of pseudo-weather forecast errors sub-model 4's performance deteriorates and the best model is now dependent on the day-type (Table 7.5).

Table 7.4 The AMAPEs of the sub-models using actual weather inputs.

		Training Set			Validation Set			Novelty Set		
Sub-model	Topology	Late Winter Sundays	Late Winter Working days	Late Winter Saturdays	Late Winter Sundays	Late Winter Working days	Late Winter Saturdays	Late Winter Sundays	Late Winter Working days	Late Winter Saturdays
1	3×4×1	3.13	2.3	2.96	2.6	2.17	2.65	2.69	2.31	2.83
2	4×7×1	2.85	2.31	2.86	2.58	2.15	**2.48**	**2.51**	2.32	**2.69**
3	2×3×1	3.06	2.39	2.98	2.65	2.22	2.68	2.79	2.44	2.88
4	6×3×1	**2.74**	**2.21**	**2.83**	**2.46**	**2.08**	2.52	2.55	**2.2**	2.81

Table 7.5 The AMAPEs of the sub-models using pseudo weather forecast inputs.

		Training Set			Validation Set			Novelty Set		
Sub-model	Topology	Late Winter Sundays	Late Winter Working days	Late Winter Saturdays	Late Winter Sundays	Late Winter Working days	Late Winter Saturdays	Late Winter Sundays	Late Winter Working days	Late Winter Saturdays
1	3×4×1	3.13	2.3	2.96	**2.6**	2.17	2.65	2.69	2.31	2.83
2	4×7×1	**2.99**	2.36	3.07	2.69	2.2	2.56	**2.49**	2.35	2.81
3	2×3×1	3.52	2.4	3.11	2.92	2.24	2.74	2.86	2.44	2.92
4	6×3×1	3.04	**2.25**	**2.96**	2.68	**2.12**	**2.52**	2.69	**2.21**	**2.75**

Table 7.6 The difference between the AMAPEs of the sub-models with and without pseudo weather forecast inputs (i.e. the values in Table 7.5 minus those in Table 7.4).

		Training Set			Validation Set			Novelty Set		
Sub-model	Topology	Late Winter Sundays	Late Winter Working days	Late Winter Saturdays	Late Winter Sundays	Late Winter Working days	Late Winter Saturdays	Late Winter Sundays	Late Winter Working days	Late Winter Saturdays
1	3×4×1	0	0	0	0	0	0	0	0	0
2	4×7×1	0.14	0.05	0.21	0.11	0.05	0.08	-0.02	0.03	0.12
3	2×3×1	0.46	0.01	0.13	0.27	0.02	0.06	0.07	0	0.04
4	6×3×1	0.3	0.04	0.13	0.22	0.04	0	0.14	0.01	-0.06

7.5 The Linear Model Fusion Algorithm.

The model fusion algorithm described by McCabe, (1991) seeks to minimize the variance of a fused forecast based on the covariance error matrix of sub-model forecasts. The cross-covariance error matrix of the sub-model forecasts is considered and the distribution of the forecast error noise is not restricted to Gaussian but merely required to be unbiased and symmetric.

In this case a combined forecast of the load on day k at hour i of day-type j, $\hat{y}_{i,j}(k)$, is created using a weighted average of the four sub-model forecasts $\hat{y}_{1i,j}(k) \ldots \hat{y}_{4i,j}(k)$ (McCabe, 1991), as:

$$\hat{y}_{i,j}(k) = \sum_{m=1}^{4} w_{m_{i,j}} \hat{y}_{m_{i,j}}(k) \tag{7.9}$$

where $w_{m_{i,j}}$ is the weight applied to the m^{th} sub-model forecast, $\hat{y}_{m_{i,j}}(k)$, and is derived from the error covariance matrices of $\hat{y}_{1i,j}(k) \ldots \hat{y}_{4i,j}(k)$. As this analysis is similar for each parallel model, the hour and day-type indices, i and j, are at this point suppressed for clarity. The weights may be derived (McCabe, 1991), as:

$$\begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} = \begin{bmatrix} C'_{4,1} & C'_{4,2} & C'_{4,3} \end{bmatrix} \begin{bmatrix} C'_{1,1} & C'_{1,2} & C'_{1,3} \\ C'_{2,1} & C'_{2,2} & C'_{2,3} \\ C'_{3,1} & C'_{3,2} & C'_{3,3} \end{bmatrix}^{-1} \tag{7.10}$$

where $C'_{4,m}$ (for $m \neq 4$) is an auxiliary variable defined as:

$$C'_{4,m} = C_{4,4} - C_{4,m} \qquad m \neq 4 \tag{7.11}$$

and $C'_{m,n}$ (for $m,n \neq 4$) is another auxillary variable defined as:

$$C'_{m,n} = C_{m,n} - C_{4,n} - C_{m,4} + C_{4,4} \qquad m \neq 4, n \neq 4 \tag{7.12}$$

where $C_{m,n}$ (or $C_{m,n_{i,j}}$) is the (sample) cross covariance of the forecast errors from sub-model m with sub-model n, defined (Papoulis, 1991), as:

$$C_{m,n} = \mathrm{E}[(y(k)-\hat{y}_m(k))(y(k)-\hat{y}_n(k))] = \frac{1}{M}\sum_{k=1}^{M}(y(k)-\hat{y}_m(k))(y(k)-\hat{y}_n(k)) \tag{7.13}$$

where E denotes the expectation operator and M is the number of samples used. The final weight w_4 is determined using the constraint that $\hat{y}_{i,j}(k)$ is unbiased (McCabe, 1991), as:

$$w_4 = 1 - \sum_{m=1}^{3} w_m \tag{7.14}$$

In addition an estimate of the error variance of $\hat{y}_{i,j}(k)$, $\hat{\sigma}_{\hat{y}_{i,j}}$, may be calculated (McCabe, 1991), as:

$$\hat{\sigma}_{\hat{y}_{i,j}} = C_{4,4} - \begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} C_{4,1} \\ C_{4,2} \\ C_{4,3} \end{bmatrix} \tag{7.15}$$

7.5.1 Fusing Sub-Model Outputs.

Fusing sub-model outputs consists of calculating the weights in Equations (7.10) and (7.14) using covariance matrices calculated with all the sub-model output errors in the training set.

The following algorithm, describes the approach in more detail:

For partitioned series day-type j =3,6 and 9*
    For partitioned series hour i = 0 to 23
        For Sub-Model = 1 to 4
            Train Sub-Model
            Record the model errors in the training set
        Next Sub-Model
        Estimate the cross-covariance matrix of sub-model load forecast errors (Equation 7.13)
        Calculate the weights using Equations (7.10) and (7.14)
        Calculate a fused forecast using a weighted average of the sub-model forecasts (Equation 7.9)
    Next hour
Next day-type.

The results for $y_{6,13.00}$ are now presented as a sample of the wider results. Aggregated results are presented in Sections 7.4.3.1 and 7.4.3.2.

Table 7.7 below shows the cross-covariance matrix of sub-model load forecast errors (using the partitioned series for hour 13:00hrs in the late winter day-type and without weather forecast error). Note that the matrix is symmetric, as are all covariance matrices. The diagonal elements of the matrix are the variance's of the individual sub-model forecast errors. As can be seen sub-models 2 and 3 have the lowest and highest variances respectively. Also note the large degree of cross-correlation between the load forecast errors indicated by the high off-diagonal elements. This shows that the forecasts made by each model are highly correlated. In this situation the improvement gained by use of data fusion may not be substantial (see Section 7.1).

* These are the Late Winter Day-types, see Table 5.10

Table 7.7 The cross-covariance matrix of sub-model load forecast errors (normalised, day-type 6, 13:00hrs, actual weather inputs).

Sub-model	1	2	3	4
1	1.18	1.12	1.18	1.16
2	1.12	1.16	1.13	1.12
3	1.18	1.13	1.20	1.18
4	1.16	1.12	1.18	1.18

The weights calculated for this fusion model are shown in Table 7.8 below. As can be seen, the models have roughly equal (absolute) weights reflecting that their performances are similar.

Table 7.8 The weights applied to each sub-model forecast (day-type 6, 13:00hrs, actual weather inputs).

Sub-model	1	2	3	4
Weight	0.54	0.59	-0.60	0.47

The estimated error variance of the fused estimate (calculated using Equation (7.15)) is 1.14*. This figure is lower than any of the individual sub-model error variances (i.e. the diagonal elements of Table 7.7), demonstrating that the fused load forecast is superior (in the least error variance sense).

The MAPEs (or AMAPEs in the case of the sub-models) of the load forecasts from the fusion and sub-models for $y_{6,13:00}$ are presented in Table 7.9 below. As can be seen, the fusion model has the lowest MAPE and variance in the training and novelty sets. However, in the validation set, the MAPE for the fusion model is not the lowest. This is because the fusion model is trained to minimise the error variance and not the MAPE.

Table 7.9 The MAPEs and sample error variances of the load forecasts from the fusion and sub models (Notes: day-type 6, 13:00hrs, actual weather inputs, sample error variances have been normalised).

	Training Set		Validation Set		Novelty Set	
Model	MAPE	Sample Error Variance	MAPE	Sample Error Variance	MAPE	Sample Error Variance
Sub-model 1	2.28	1.18	1.97	1.31	1.99	1.74
Sub-model 2	2.30	1.16	2.02	1.34	2.03	1.68
Sub-model 3	2.30	1.20	1.97	1.34	2.05	1.83
Sub-model 4	2.27	1.18	**1.96**	1.33	2.01	1.76
Fusion	**2.25**	**1.14**	1.97	**1.30**	**1.97**	**1.64**

* This figure has been normalised by the same normalisation factor used throughout this thesis.

Note that the previous example is for the situation in which no weather forecast errors are included in the sub-model inputs. When pseudo-weather forecast errors are added to the weather inputs, the covariance matrices, $C_{m,n_{i,j}}$ (for m,n =1,…,4, i=1,…,9 and j = 0,…,23), will be different. As a consequence the weights, w_1,…, w_4, will be different.

7.5.1.1 Results without Pseudo-Weather Forecast Errors.

Figure 7.14 below, shows the MAPE of the load forecasts produced by the fusion and sub models for *all hours* of the late winter working day-type over the period of the novelty set. As can be seen, the fusion model achieves the best MAPE at most hours, or is close to the best. The daily MAPE is shown in the key of Figure 7.14 and again the Fusion model has the lowest MAPE with a value of 2.17%.

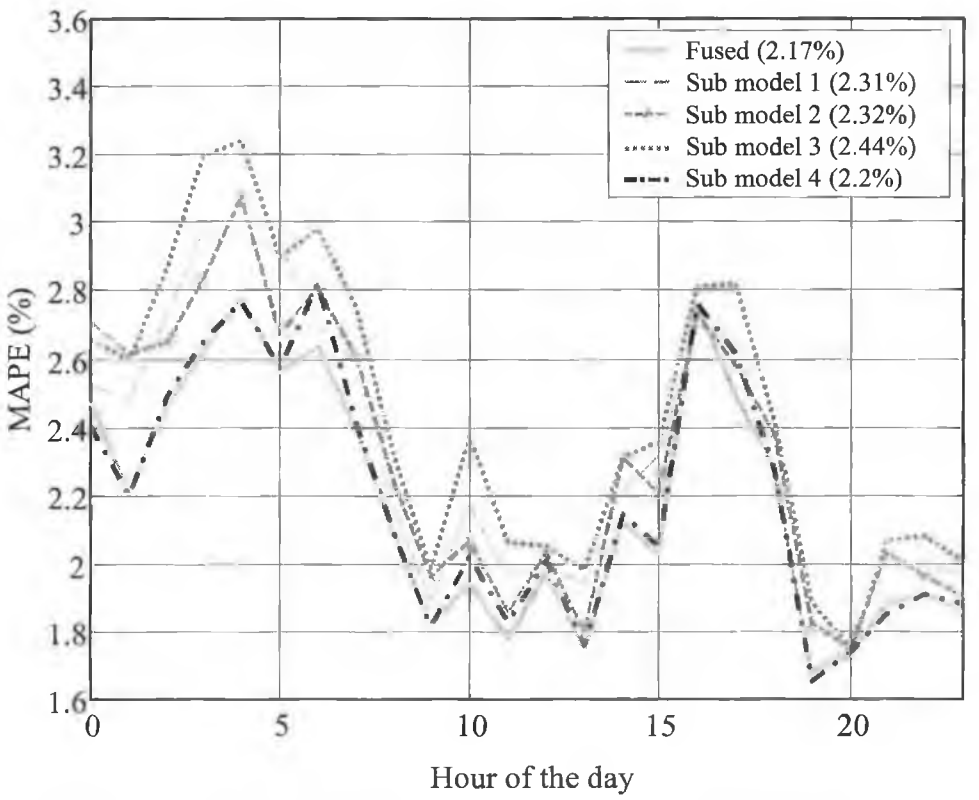

Figure 7.14. MAPE as a function of hour of the day for fusion and sub-models (notes: late linter working day-type, novelty set, actual weather inputs used)

Tables 7.10, 7.11 and 7.12, below, summarise the daily MAPEs achieved by all of the models in all the day-types used in this chapter (i.e. the late winter day-types). Similar to results presented in Section 7.4.3, the fusion model load forecasts have the best MAPEs and lowest variances of most of the models in the

training and validation sets (Tables 7.10 and 7.11). In addition, the fusion model is also the best in all data sets for the working days (Tables 7.10, 7.11 and 7.12). This is important, as the working day series have more data than the Sunday or Saturday day-type series and thus are more reliable estimates of the performance of the fusion model relative to the sub-models.

Table 7.10 The daily MAPEs of the load forecasts from the fusion and sub-models (Notes: Training set)*.

Model	Late winter Sundays		Late winter working days		Late winter Saturdays	
	MAPE	Sample Error Variance	MAPE	Sample Error Variance	MAPE	Sample Error Variance
Sub-model 1	3.13	1.10	2.30	0.93	2.96	1.24
Sub-model 2	2.85	0.96	2.31	0.91	2.86	1.15
Sub-model 3	3.06	1.07	2.39	0.98	2.98	1.23
Sub-model 4	2.74	0.87	2.21	0.86	2.83	1.11
Fusion	**2.54**	**0.72**	**2.19**	**0.83**	**2.60**	**0.91**

Table 7.11 The daily MAPEs of the load forecasts from the fusion and sub-models (Notes: Validation set)*.

Model	Late winter Sundays		Late winter working days		Late winter Saturdays	
	MAPE	Sample Error Variance	MAPE	Sample Error Variance	MAPE	Sample Error Variance
Sub-model 1	2.60	1.19	2.17	1.31	2.65	1.52
Sub-model 2	2.58	1.23	2.15	1.28	**2.48**	1.36
Sub-model 3	2.65	1.33	2.22	1.37	2.68	1.60
Sub-model 4	2.46	1.07	2.08	1.22	2.52	**1.35**
Fusion	**2.38**	**1.01**	**2.07**	**1.20**	**2.48**	1.36

Table 7.12 The daily MAPEs of the load forecasts from the fusion and sub-models (Notes: Novelty set)*.

Model	Late winter Sundays		Late winter working days		Late winter Saturdays	
	MAPE	Sample Error Variance	MAPE	Sample Error Variance	MAPE	Sample Error Variance
Sub-model 1	2.69	1.61	2.31	1.73	2.83	2.13
Sub-model 2	**2.51**	**1.37**	2.32	1.74	**2.69**	**1.91**
Sub-model 3	2.79	1.75	2.44	1.91	2.88	2.20
Sub-model 4	2.55	1.46	2.20	1.59	2.81	2.10
Fusion	2.54	1.51	**2.17**	**1.56**	**2.69**	1.96

* Variances have been normalised.

7.5.1.2 Results *with* Pseudo-Weather Forecast inputs.

Figure 7.15, below, shows the MAPE of the load forecasts produced by the fusion and sub-models for all hours of the late winter working day-type over the period of the novelty set (with pseudo-weather forecast errors on the weather inputs). As can be seen the fusion model again achieves the best MAPE at most hours or is close to the best. The daily MAPE is shown in the key of Figure 7.14 and again the fusion model has the lowest MAPE with a value of 2.22%. Note that the performance of sub-models 2 to 4 deteriorates relative to the performance without pseudo weather forecast error (Figures 7.14 and 7.15).

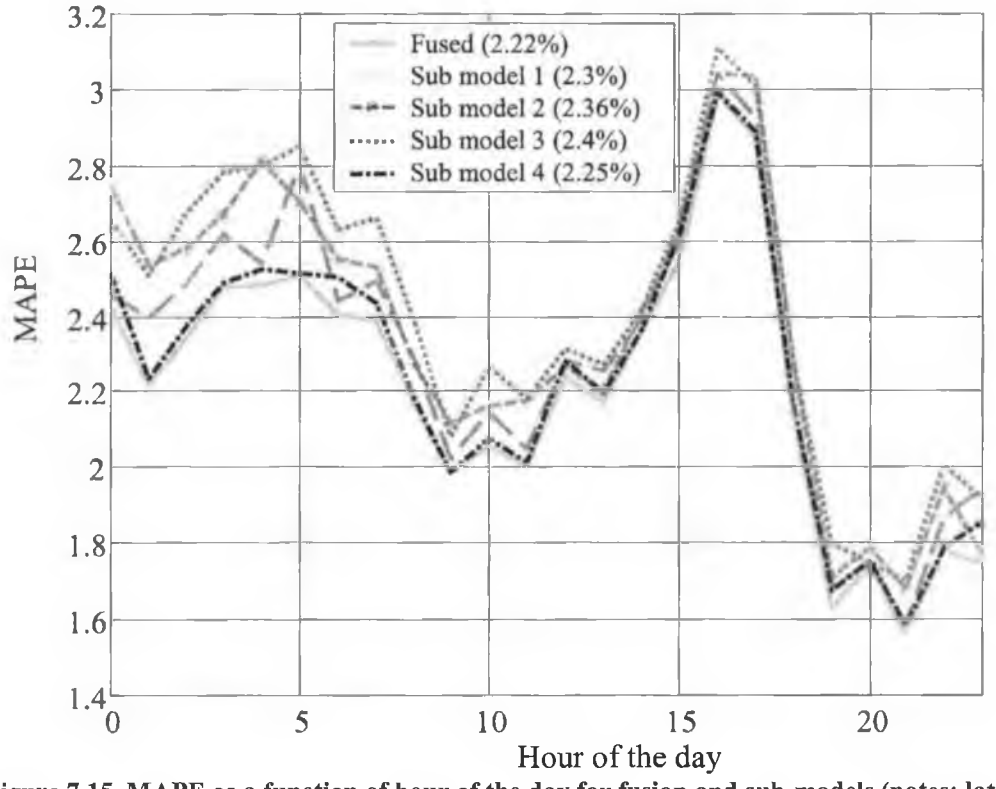

Figure 7.15. MAPE as a function of hour of the day for fusion and sub-models (notes: late winter working day-type, novelty set, pseudo weather forecasts used)

The results in Tables 7.13, 7.14 and 7.15 below are similar to those in the last section (Section 7.4.3.1). The fusion model is the best model, except in the novelty set for late winter Sundays (Table 7.15) and the validation set for late winter Saturdays (Table 7.14). However, the fusion model has again the best performance in all sets for the late winter working days which, as mentioned in Section 7.4.3.1, is the most reliable estimate of the relative performance of the models.

Table 7.13 The daily MAPEs of the load forecasts from the fusion and sub-models (Notes: Training set, pseudo weather forecasts used)*.

Model	Late winter Sundays		Late winter working days		Late winter Saturdays	
	MAPE	Sample Error Variance	MAPE	Sample Error Variance	MAPE	Sample Error Variance
Sub-model 1	3.13	1.10	2.30	0.93	2.96	1.24
Sub-model 2	2.99	1.03	2.36	0.95	3.07	1.32
Sub-model 3	3.52	1.99	2.40	0.99	3.11	1.40
Sub-model 4	3.04	1.04	2.25	0.89	2.96	1.25
Fusion	**2.79**	**0.88**	**2.22**	**0.87**	**2.82**	**1.12**

Table 7.14 The daily MAPEs of the load forecasts from the fusion and sub-models (Notes: Validation set, pseudo weather forecasts used)*.

Model	Late winter Sundays		Late winter working days		Late winter Saturdays	
	MAPE	Sample Error Variance	MAPE	Sample Error Variance	MAPE	Sample Error Variance
Sub-model 1	2.60	1.19	2.17	1.31	2.65	1.52
Sub-model 2	2.69	1.28	2.20	1.35	2.56	1.45
Sub-model 3	2.92	1.64	2.24	1.39	2.74	1.64
Sub-model 4	2.68	1.28	2.12	1.27	**2.52**	**1.37**
Fusion	**2.52**	**1.10**	**2.10**	**1.25**	2.56	1.40

Table 7.15 The daily MAPEs of the load forecasts from the fusion and sub-models (Notes: Novelty set, pseudo weather forecasts used)*.

Model	Late winter Sundays		Late winter working days		Late winter Saturdays	
	MAPE	Sample Error Variance	MAPE	Sample Error Variance	MAPE	Sample Error Variance
Sub-model 1	2.69	1.61	2.31	1.73	2.83	2.13
Sub-model 2	**2.49**	**1.34**	2.35	1.78	2.81	2.05
Sub-model 3	2.86	1.88	2.44	1.91	2.92	2.26
Sub-model 4	2.69	1.61	2.21	1.62	2.75	2.00
Fusion	2.50	1.38	**2.20**	**1.59**	**2.73**	**1.98**

Next, the situation in which the fusion model weights are *trained* using actual weather inputs, but the model is *operated* using pseudo-weather forecasts is considered. This situation is used as an example of how a load model's performance (in this case the fusion model's) can degenerate if weather forecast errors are not taken into account (this point is discussed in Section 3.5).

* variances have been normalised

Table 7.16 below shows the results achieved in the novelty set for this situation. It can be seen that the fusion model in Table 7.16 has higher MAPEs than any other model. However, the performance of the fusion model has not deteriorated significantly.

Table 7.16 The daily MAPEs of the load forecasts from the fusion and sub-models (Notes: Novelty set)*.

Type of input used to calculate weights	Type of input used to generate forecasts	Late winter Sundays		Late winter working days		Late winter Saturdays	
		MAPE	Sample Error Variance	MAPE	Sample Error Variance	MAPE	Sample Error Variance
Pseudo-weather forecasts.	Pseudo-weather forecasts.	2.50	1.38	2.20	1.59	2.73	1.98
Actual weather	Pseudo-weather forecasts.	2.65	1.61	2.21	1.61	2.75	2.01

7.6 The Non-Linear Model Fusion Algorithm.

The non-linear model fusion algorithm uses an MLP (Section 3.4.3) to implement the model fusion. The overall forecast produced by the non-linear model fusion algorithm may be expressed as:

$$\hat{y}_{i,j}(k) = f_{i,j}\left(\hat{y}_{1_{i,j}}(k), \hat{y}_{2_{i,j}}(k), \hat{y}_{3_{i,j}}(k), \hat{y}_{4_{i,j}}(k)\right) \tag{7.16}$$

where $f_{i,j}$ represents the function implemented by the MLP for hour i on day-type j, $\hat{y}_{i,j}(k)$ is the non-linear fusion model estimate and $\hat{y}_{1_{i,j}}(k),\ldots,\hat{y}_{4_{i,j}}(k)$ are the forecasts from the four sub-models. Note that a different MLP is trained for each hour of the day and for each day-type.

The details of the MLPs are similar to those used in Section 5.7 and are now summarised. There are four inputs given by the four sub-model forecasts (which have been normalised). As there are only four inputs, no input pre-processing is performed. The MLPs have two hidden layers with sigmoidal activation functions and a linear output layer. The structure of the MLPs is determined by training 64 different structures (from 1 to 8 nodes in hidden layer 1, and 1 to 8 nodes in hidden layer 2) and choosing the structure which performed best (in terms of the AMAPE, see below) in the validation set. Each network is initialised

using random starting conditions and 10 networks are trained for each structure, the top four being retained. These four networks are then used to give an AMAPE (as in Section 5.7.3.3). The back-propagation training algorithm is used with early stopping and cross-validation.

The results of the non-linear model fusion algorithm are now compared to those using the linear model fusion algorithm.

7.6.1 Results without Pseudo-Weather Forecast Errors.

Figure 7.16 below, shows the MAPE of the load forecasts produced by the linear and non-linear fusion models for *all hours* of the late winter working day-type over the period of the novelty set. As can be seen the non-linear fusion model achieves the best MAPE at most hours or is close to the best. The daily MAPE is shown in the key of Figure 7.16, as can be seen there is a minor improvement by use of a non-linear fusion algorithm.

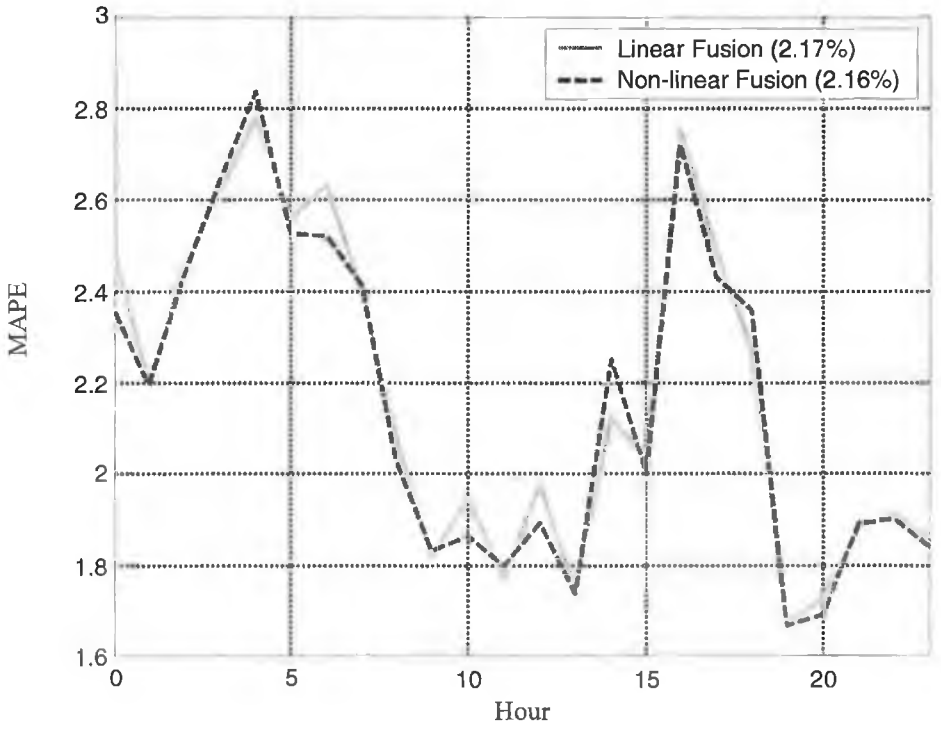

Figure 7.16. MAPE as a function of hour of the day for linear and non-linear fusion models (notes: late winter working day-type, novelty set, actual weather inputs used)

---

* variances have been normalised

Tables 7.17, 7.18 and 7.19 compare the performance of the linear and non-linear fusion models in the training, validation and novelty data sets for all day-types. As can be seen, the non-linear fusion model has the best performance in the training and validation sets. However, in the novelty set the non-linear fusion model has the lowest daily MAPE only for the late winter working days day-type (Table 7.19). Even in this case the non-linear fusion model has a higher variance than the linear fusion model (1.58 compared to 1.56 respectively). In conclusion, there seems to be no advantage in using this type of non-linear fusion model in this case.

Table 7.17 The daily MAPEs of the load forecasts from the fusion and sub-models (Notes: training set, actual weather inputs used)*.

Model	Late winter Sundays		Late winter working days		Late winter Saturdays	
	MAPE	Sample Error Variance	MAPE	Sample Error Variance	MAPE	Sample Error Variance
Non-linear	**2.21**	**0.57**	**2.15**	**0.81**	**2.24**	**0.70**
Linear	2.54	0.72	2.19	0.83	2.60	0.91

Table 7.18 The daily MAPEs of the load forecasts from the fusion and sub-models (Notes: validation set actual weather inputs used)*.

Model	Late winter Sundays		Late winter working days		Late winter Saturdays	
	MAPE	Sample Error Variance	MAPE	Sample Error Variance	MAPE	Sample Error Variance
Non-linear	**2.27**	**0.936**	**2.03**	**1.16**	**2.34**	**1.20**
Linear	2.38	1.01	2.07	1.20	2.48	1.36

Table 7.19 The daily MAPEs of the load forecasts from the fusion and sub-models (Notes: novelty set actual weather inputs used)*.

Model	Late winter Sundays		Late winter working days		Late winter Saturdays	
	MAPE	Sample Error Variance	MAPE	Sample Error Variance	MAPE	Sample Error Variance
Non-linear	2.74	1.77	**2.16**	1.58	2.79	2.09
Linear	**2.54**	**1.51**	2.17	**1.56**	**2.69**	**1.96**

7.6.2 Results with Pseudo-Weather Forecast inputs.

The MLPs in the non-linear fusion algorithm are again trained for this simulation. However, in this case the forecasts from the sub-models produced with *pseudo weather forecasts* are used (Section 7.5.1.2). Figure 7.17 below,

* Variances have been normalised.

compares the linear and non-linear fusion model MAPEs for all hours of the day for the late winter working day day-type. As can be seen, the non-linear fusion model has a lower daily MAPE (2.16% compared with 2.20%). However, the non-linear fusion model is not the best at all hours of the day. Thus no clear difference between these models can be discerned.

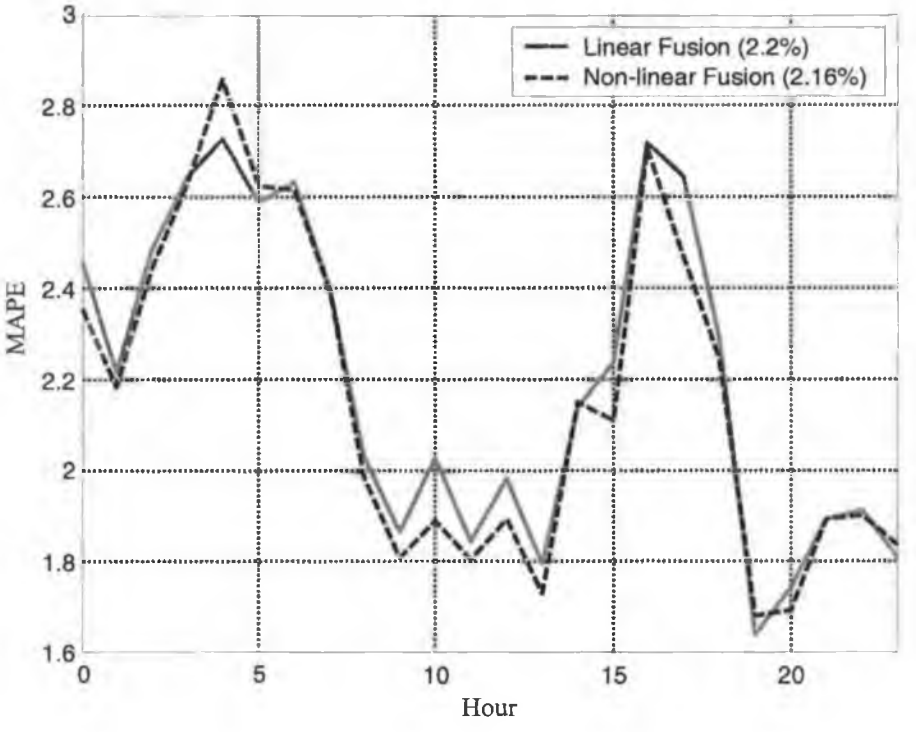

Figure 7.17. MAPE as a function of hour of the day for linear and non-linear fusion models (notes: late winter working day-type, novelty set, pseudo weather forecasts used)

Tables 7.20, 7.21 and 7.22 compare the performance of the linear and non-linear fusion models in the training, validation and novelty data sets for all day-types. As can be seen, the non-linear fusion model again has the best performance in the training and validation sets. However, in the novelty set the non-linear fusion model has the lowest daily MAPE only for the late winter working days day-type (Table 7.19 and Figure 7.17 above). In conclusion; the non-linear fusion models do not appear to be generalising well and a non-linear combiner may be too complex for this situation.

Table 7.20 The daily MAPEs of the load forecasts from the Fusion and sub models (Notes: Training set, pseudo weather forecasts used)*.

Model	Late winter Sundays		Late winter working days		Late winter Saturdays	
	MAPE	Sample Error Variance	MAPE	Sample Error Variance	MAPE	Sample Error Variance
Non-linear	**2.21**	**0.57**	**2.16**	**0.82**	**2.25**	**0.71**
Linear	2.79	0.88	2.22	0.87	2.82	1.12

Table 7.21 The daily MAPEs of the load forecasts from the Fusion and sub models (Notes: Validation set, pseudo weather forecasts used)*.

Model	Late winter Sundays		Late winter working days		Late winter Saturdays	
	MAPE	Sample Error Variance	MAPE	Sample Error Variance	MAPE	Sample Error Variance
Non-linear	**2.28**	**0.98**	**2.04**	**1.17**	**2.40**	**1.25**
Linear	2.52	1.10	2.10	1.25	2.56	1.40

Table 7.22 The daily MAPEs of the load forecasts from the Fusion and sub models (Notes: Novelty set, pseudo weather forecasts used)*.

Model	Late winter Sundays		Late winter working days		Late winter Saturdays	
	MAPE	Sample Error Variance	MAPE	Sample Error Variance	MAPE	Sample Error Variance
Non-linear	2.73	1.78	**2.16**	**1.55**	2.84	2.19
Linear	**2.50**	**1.38**	2.20	1.59	**2.73**	**1.98**

7.7 Conclusion.

This chapter examined the use of fusion models with a view to minimising the effect of weather forecast errors on load forecasts.

Section 7.2 modelled and examined the statistics of weather forecast errors. Previous approaches in STLF have modelled the weather forecast error as an independent Gaussian random variable (Park et. al., 1993a and Chen and Yu, 1992). However, it was found that this assumption does not apply to Irish weather forecasts in two respects; the weather forecast errors have serial correlation (Figure 7.2) and they are cross-correlated (Tables 7.2 and 7.3). Typically, some form of aggregate weather variables are used in STLF models (e.g. average daily temperature). In the sub-models used here, the transformation of the weather inputs by PCA results in a similarly aggregated variable. The error

* variances have been normalised

in an aggregate weather variable will have a non-zero mean (Figure 7.2) and a Gaussian approximation would underestimate this. For example, independent Gaussian random noise added to hourly temperature would tend to cancel itself out when the average is taken. Thus the effect of weather forecast errors on load forecasting models would be underestimated.

The structure of the weather forecast errors was then used to produce pseudo-weather forecast errors from 1986 to 2000 which have the accuracy of current weather forecasts. This is important as, for example, weather forecasts from 1986 are less accurate than current weather forecasts and would thus be of no relevance in predicting future loads.

A model fusion technique was then proposed for minimising the effect of weather forecast errors. In general, weather forecast error causes approximately 1% deterioration in load forecasts of all models used here. This figure, though important, is not as high as suggested by Douglas *et. al.* (1998b) and the IEEE Committee report (1985), for their systems. However, the fusion model was capable of adjusting the weighting of the sub-models to reflect that the weather based sub-models deteriorated relative to the AR model. The fusion model was shown to successfully separate the tasks of model training and rejecting weather forecast errors.

Finally, a comparison of linear and non-linear fusion models found little difference between them. Considering that the novelty set results for the non-linear fusion model were generally worse than for the linear fusion model, it would appear that the non-linear fusion models used are over-complex for this problem.

Chapter 8

Conclusions

8.1 Broad Conclusions.

This thesis examined a strategy for short-term load forecasting in Ireland. As seen in Chapter 3, short-term load forecasting is a popular area of research and there are a multitude of modelling approaches which have been proposed. In addition, no single benchmark exists that would allow a comparison of these approaches. Indeed, differences between electrical grid systems mean that the choice of modelling approach is application specific. Thus, the overall tactic taken in this thesis has been to identify the major choices facing a short-term load forecaster, and identify the correct choice for Irish load data.

The first choice to be made was the level of disaggregation of the data (Section 3.2). Overall, it was found that the amount of data contained in each day-type was sufficient to allow modelling of that day-type (Chapters 5, 6 and 7). With some exceptions, the day-type partition was found to give the best results, while alternative partition 1 (which partitioned the data by hour of the day only, i.e. no day-types at all) led to significantly inferior model performance (Section 5.4.2.3). Alternative partition 2 (which partitioned the data by hour of the day and Saturday, Sunday and working day) removed the winter and summer part of the day-type partition and led to good model performance in all day-types (Section 5.4.2.3). In addition, this partition was found to be superior when forecasting the load in early winter.

Disaggregation of the data by hour of the day depends on whether there are independent components in load at some hours of the day (Section 5.3). Empirical evidence suggests that there is a case for modelling the hours of 5 p.m. to 8 p.m. separately from other hours of the day as they have independent components (Section 6.5). As these hours are the hours at which many people return from work and use electrical appliances at home, it would seem that electricity consumption at home is influenced by different factors than those for other hours of the day. In general though, it was found that neither the parallel

nor the sequential approach was best for all day-types or even for all hours of the day. Rather, each day-type and each hour of the day have separate characteristics which need to be taken into account when constructing models (more detail is given in Section 8.3 below).

Linear and non-linear models were examined in Chapter 5. Empirical evidence suggests that there is a non-linear relationship between Irish load and weather variables and also a non-linear auto-regressive component (Section 5.8). This suggests the use of non-linear models in short-term load forecasting can provide considerable improvement on linear models.

Chapter 6 examined and extended the multi-timescale technique of Murray (2000). This technique was found to work well during most hours of the day and methods for estimating the weight matrix, used in the technique, were examined (Section 6.4.1). A novel approach was proposed in which the weights could be determined numerically and it was found that this approach gave excellent results (Section 6.4.1). However, the MTS technique was susceptible to the way in which the data was partitioned (Section 6.5) and perhaps a different means of partitioning the data *or* adjustments to the technique could be examined. This is an area of future research.

A novel view of the MTS technique was then presented in which the technique was viewed as a regularisation term as opposed to a model combination technique. It was seen that attempting to optimise the weight matrix estimates to minimise the forecast error, while simultaneously attempting to minimise the deviations (of the SM, cardinal point and end-sum forecasts from the MTS forecasts), were conflicting objectives. A numerical approach solving this conflicting objective was suggested and is left for future research. In addition, a route is given for a derivation of a closed form solution.

Chapter 7 presented a fusion model which used a novel approach to minimise the effect of weather forecast errors. This model is a generalisation of the model proposed by Miyake *et. al.*, (1995) but it not restricted in terms of the type of model or the distributions of the weather forecast errors. The fusion model was

found to give excellent load forecasts even in the presence of weather forecast errors. The statistics of weather forecast errors in Ireland were also examined. It was found, contrary to previous research in this area (Section 3.5), that weather forecast error is not independently normally distributed, but has a structure. A novel algorithm was presented which allows pseudo-weather forecasts to be produced based on this structure.

8.2 Analysis of Models.

The parallel models are based on disaggregation of the load by hour of the day and are found to give very good results in all day-types. This is partly due to the nature of the partitioned series, which are constructed by taking a single hour from each day-type. By taking a single hour from each day, the 'shift' in the load, which gave rise to problems in the sequential and MTS models was avoided (see Section 6.2.4). This is because the load at 3 p.m., for example, on a Friday is similar to the load at 3 p.m. on the following Monday (the data point that would come after the Friday load in a working data partitioned series). Also, the rise in the trend component from Friday to Monday is negligible.

The input selection procedures for the parallel models were examined in Section 5.5.2. It was concluded that Method 2 was the best method. As explained in Section 5.8, this is because there is correspondence between the frequency information in the pre-whitened weather variables, the variance explained by each transformed pre-whitened weather component and the correlation of the weather inputs to the load. However, this correspondence is application specific and so this technique may not work well in other situations.

The feed forward neural networks used in Section 5.7 were found to work well. The following suggestions were noted during their construction:

- The topology determination technique used was very computationally expensive and did not allow for non-fully connected architectures. This is an area for future research,

- Cross validation and early stopping improved network performance considerably and should always be used,

- The multi-step ahead performance of the networks was found in most cases to be stable, but this should be checked,

- Training several networks with different initial conditions is recommended, as it allowed those that did not generalise well to be identified and discarded,

- Weighting the network errors during training to discount older data did not bring about much improvement. This is also an area for future research,

- Using a single hidden layer resulted in relatively bad model performance and

- No discernable improvement resulted from the use of different activation functions, contrary to the findings of Choueiki *et. al.* (1997).

The MTS models performed well in the working day day-types, but did not compare well to the parallel models in the winter weekend day-types. The reason for this poor performance lay with the way in which the data was partitioned, which resulted in a 'shift' in the load (Section 6.5). It was found that the sequential model had to be adjusted to account for this shift and that even then, the MTS models performed badly in some day-types (Section 6.5).

The sequential model was found not to be very robust to 'bad data', or outliers in the data set. During operation, an outlier tended to persist in the sequential model forecasts for several days. Additionally, a 'few' outliers in the training set result in underestimates of the seasonal component parameters. For these reasons, it is proposed that research into a different model for the seasonal component be pursued or that any application of this model should verify the data prior to using it. In addition, the SSD algorithm for BSM parameter estimation (Section 3.3.2.4) should be used, as it allows the trend component parameters to be estimated independent of any outliers in the data.

The fusion models were presented in Chapter 7. These were designed to minimise the effect of weather forecast errors. The construction of each sub-model is similar to that of a parallel model. However, the fusion approach may be applied to other model types and requires only that the weather inputs be separated from the auto-regressive inputs, at the sub-model stage. The non-linear fusion algorithm was found to give no improvement over the linear fusion algorithm. However, feed forward neural networks may not be the ideal modelling technique for non-linear fusion. Fuzzy logic techniques combine the output of several models and may thus be superior in this case. These are suggested as an avenue of future research.

8.3 Recommendations and Caveats.

Figure 8.1 below, summarises the best models found for each day-type and hour of the day.

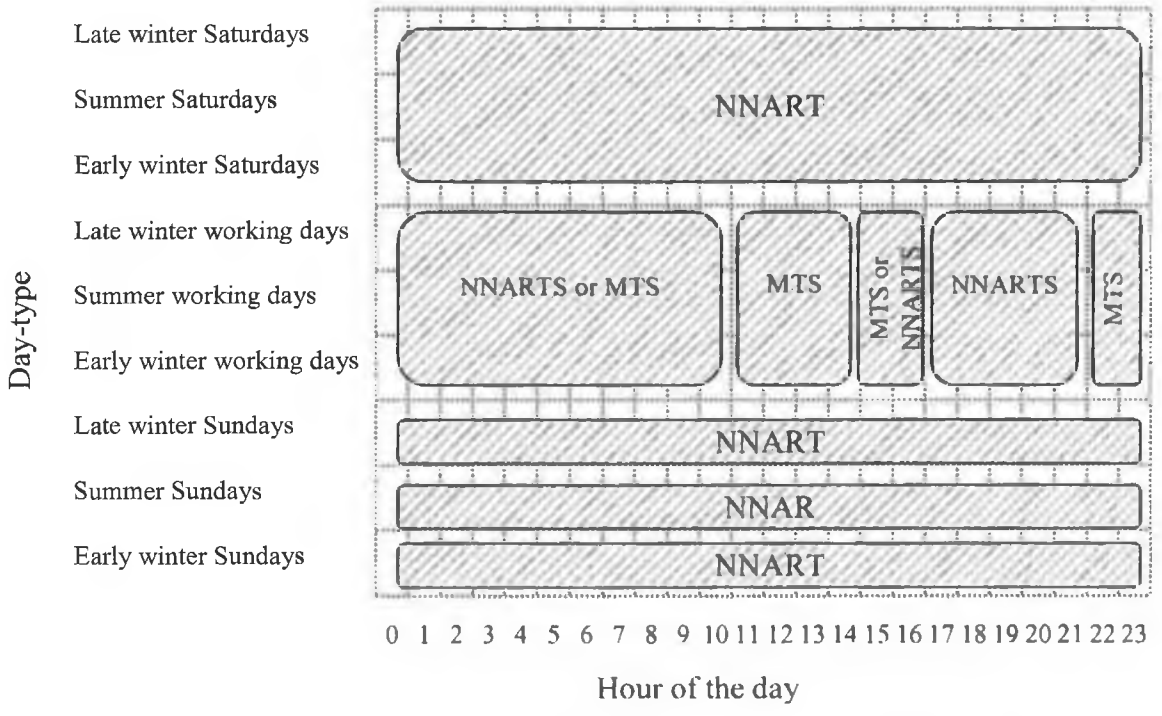

Figure 8.1 An overview of the best model to use for each hour of the day in each day-type.

The following are recommendations for Eirgrid (project sponsors) regarding construction of a load forecasting computer application and more generally characteristics of Irish electrical load data:

- The MTS and parallel models should both be used but at different times, as indicated in Figure 8.1,
- The MTS technique combines the output of several models and thus requires more development and maintenance costs. The use of the parallel models alone will reduce load forecast accuracy slightly, but will be cheaper, less time consuming and more robust to outliers/missing data,
- The loads between 5 p.m. and 8 p.m. have independent components, but are still highly correlated with the load at other hours of the day. This means that approaches which depend on forecasting the load using load curves are not obsolete but need to have their forecasts at these hours adjusted,
- The non-linear parallel models are quite robust, but it is recommended that they are retrained every year,
- The fusion model shows promise, but requires more research and weather forecast data,
- There is no discernable cooling effect in Irish load although this situation may change in the future, and
- Bank holidays and Sundays have similar load curves and should be modelled together.

8.4 Future Research.

The following areas require further attention and are suggested as topics for future research:

1. **The level of day-type disaggregation**. Disaggregation by day-type was found, in general, to be advantageous. However, alternative partition 2 was found to be the best partition for early winter working days, but not for summer working days or late winter working days (Section 5.4.2.3). The day-type partitions and alternative partition 2 are *overlapping* sets (Figure 8.2), yet empirical evidence suggests the use of both for different

situations. The day-types were identified by dividing the output grid of a Kohonen map (Section 4.2) into several *non-overlapping* sets (i.e. the day-types) in Section 4.2. It may however, be fruitful to divide the output grid into overlapping sets; this is an area for future research.

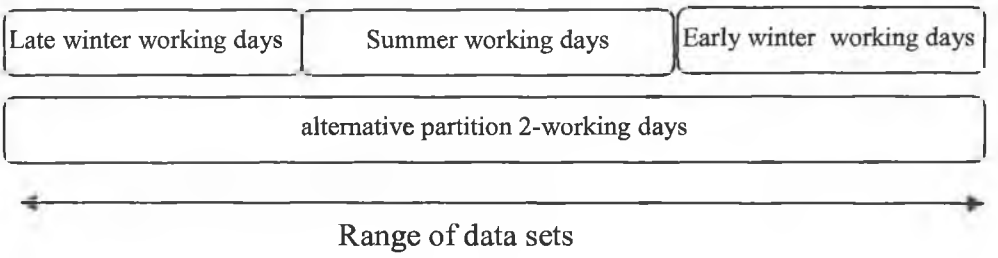

Figure 8.2 The overlap of alternative partition 2 and the working day day-types.

2. **The non-linear parallel modelling technique**. The feed forward neural networks used to model the residual component in the parallel models gave results superior to a linear model. In addition, these models provided an appropriate tool to investigate a strategy for load forecasting in Ireland. However, feed forward neural networks are just one type of non-linear model, and there exist many others (as seen in Chapter 3),

3. **A non-linear input selection and pre-processing procedure**. The input selection procedure relied on identifying those elements of the data that were *linearly* correlated with the load. This procedure may have excluded inputs which had a non-linear relationship with the load. Specifically, non-linear PCA and Independent Component Analysis (ICA) are two techniques which may provide better input selection. However, the computational expense of any non-linear technique would have to be taken into account,

4. **Calculation of the weight matrix in the MTS technique**. In Section 6.4.1 the importance of calculating W was demonstrated. A numerical approach is given, but this approach is computationally expensive and may not give an optimum result. In Section 6.4.2 a deterministic approach was attempted and the route for a solution was given. This approach

appears promising, but the equations became intractable quite early in the derivation. It is suggested that a statistical approach based on the expected value of the cost function would result in an easier formulation. In addition, any solution from a statistical approach could lend itself easily to a recursive solution.

5. **Development of the fusion models**. The fusion models attempt to minimise the effect of weather forecast errors on load forecasts and are also valid load forecasting models in their own right. Only two fusion algorithms were examined for combining the sub-model forecasts and more could be examined in future research. It was found during development of the fusion models that when the weather based sub-model forecasts deviated significantly from the auto-regressive sub-model forecasts, the former were more accurate. However, this effect was not consistent and requires more research,

6. **Christmas and exceptional days**. The Christmas period was not examined in this thesis and a model for this period is required. Christmas is distinct from all other day-types as each day in the period is significantly different (i.e. the load on Christmas day is different from the load on 2nd January, etc.). In addition the amount of data in the Christmas day-type is limited and may benefit from inclusion of data outside the day-type in a similar manner to the early winter day-types (Section 5.4.2.3), and

7. **Multi-step ahead performance.** The performance of models for a forecast horizon of several days ahead was not examined in this thesis. It is possible that different models may be superior for different forecast horizons and also that weather forecasts for several days ahead will be less accurate than twenty-four hour ahead weather forecasts.

References

Abdelaziz, A.R., Gouda, S.A.M., 1998, A fuzzy based electrical load forecasting. *In*: Jurgen, H., Zimmermann, H.J., eds., *Proceedings*, *European Congress on Intelligent Techniques and Soft Computing*, Aachen, Germany, Sept 1998. Aachen: Elite foundation, 1985-1989.

Abraham, B., Ledolter, J., 1983, *Statisitcal Methods for Forecasting*, New York: J. Wiley.

Apostol, T.M., 1974, *Mathematical Analysis*, 2, New York: Addison-Wesley.

Arahal, M.R., Camacho, E.F., 1999, Application of the RAN algorithm to the problem of short term load forecasting, *In*: *Proceedings*, *the* 5th *European Control Conference*, Karlsruhe, Germany, August 1999, (not paginated on CD-ROM).

Arinze, B., Kim, S.L., Anandarajan, M., 1997, Combining and selecting forecasting models using rule based induction, *Computer Operations Research*, 24 (5), 423-433.

Aussem, A., 1999, Dynamical recurrent neural networks towards prediction and modelling of dynamical systems, *Neurocomputing*, 28, 207-232.

Bai, E.W., 2002, A blind approach to the Hammerstein-Wiener model identification, *Automatica*, 38, 967-979.

Barakat, E.H., Eissa, M.A.M., 1989, Forecasting monthly peak demand in fast growing electric utility using a composite multiregression-decomposition model, *IEE Proceedings C*, 136 (1) , 1989, 35-41.

Barakat, E.H., Qayyum, M.A., Hamed, M.N., Al Rashed, S.A., 1990, Short-term peak demand forecasting in fast developing utility with inherit dynamic load characteristics. I. Application of classical time-series methods. II. Improved modelling of system dynamic load characteristics, *IEEE Transactions on Power Systems*, 5 (3), 813-824.

Bates, J.M., Granger, C.W.J., 1969, The combination of forecasts, *Operational Research Quarterly*, 20, 451-468.

Bhattacharyya, K., Crow, M.L., 1996, A fuzzy logic based approach to direct load control, *IEEE Transactions on Power Systems*, 11 (2), 708-714.

Billor, N., Hadi, A.S., Velleman, P.F., 2000, BACON: blocked adaptive computationally efficient outlier nominators, *Computational Statistics and Data Analysis*, 34, 279-298.

Bitzer, I.B., Rößer, I.F., 1998, A hybrid fuzzy neural network system for short term electrical load forecasting. *In*: Jurgen, H., Zimmermann, H.J., eds., *Proceedings*, *European Congress on Intelligent Techniques and Soft Computing, Aachen, Germany, September*, Aachen: Elite foundation, 1920-1924.

Bowerman, B., O'Connell, R.T., *Time series forecasting unified concepts and computer implementations*, Duxbury press, 1987.

Box, G.E.P., Cox, D.R., 1964, An analysis of transformations, *Journal of the Royal Statistical Society*, 26 (B), 211-243.

Box. G.E.P., Jenkins, G.M., 1970, *Time Series Analysis: Forecasting and Control*, San Francisco: Holden Day.

Box. G.E.P., Pierce, D.A., 1970, Distribution of residual autocorrelations in autoregressive integrated moving average time series models, *Journal of the American Statistical Association*, 65, 1509-1526.

Box, M.J., Davies, D., Swann, W.H., 1969, *Non-Linear Optimization Techniques*, Edinburgh: Oliver and Boyd ltd.

Bretschneider, P., Rauschenbach, T., Wernstedt, J., 1998, Hybrid forecast strategy using an adaptive fuzzy classification algorithm. *In*: Jurgen, H., Zimmermann, H.J., eds., *Proceedings, European Congress on Intelligent Techniques and Soft Computing*, Aachen, Germany, Sept, Aachen: Elite foundation, 1916-1919.

Bretschneider, P., Rauschenbach, T., Wernstedt, J., 1999, Forecast using an adaptive fuzzy classification algorithm for load, *In*: *Proceedings, the* 5th *European Control Conference*, Karlsruhe, Germany, August, (not paginated on CD-ROM)

Brockwell, P.J., Davis, R.A., 1987, *Time Series: Theory and Methods*, New York USA: Springer Verlag Inc.

Broemeling, L., Shaarawy, S., 1985, Time series: A Bayesian analysis in the time domain, *In:* Bunn, D.W, Farmer, E.D., eds. *Comparative Models for Electrical Load Forecasting*, NY: John Wiley and sons, 109-119.

Capasso, A., Grattieri, W., Lamedica, R., Prudenzi, A., 1994, A bottom-up approach to residential load modelling, *IEEE Transactions on Power Systems,* 14 (4), 957-964.

Castellano, G., Fanelli, A.M., 2000, Variable selection using neural-network models, *Neurocomputing*, 31, 1-13.

CER, 2000, *A helicopter guide to trading and settlement*, CER report 00/18, Dublin: CER.

Charytoniuk, W., Chen, M.S., Kotas, P., Van Olinda, P., 1999, Demand forecasting in power distribution systems using non-parametric probability density estimation, *IEEE Transactions on Power Systems,* 14 (4),
1200-1206.

Chen, S.L., Kao, F.C., 1996, An efficient algorithm to model and forecast hourly weather sensitive load, *Journal of the Chinese Institute of Electrical Engineering*, 3(3), 231-243.

Chen, S.T., Yu, D.C., Moghaddamjo, A.R., 1992, Weather sensitive short-term load forecasting using non-fully connected artificial neural network, *IEEE Transactions on Power Systems,* 7 (3), 1098-1104.

Chihocki, A., Unbehauen, R., 1993, *Neural networks for optimisation and signal processing*, John Wiley and sons.

Chiu, C.C., Kao, L.J., Cook, D.F., 1997, Combining a neural network with a rule based expert system approach for short term power load forecasting in Taiwan, *Expert Systems with Applications*, 13 (4), 299-305.

Choueiki, M.H., Mount-Campbell, C.A., Ahalt, S.C., 1997, Building a 'quasi optimal' neural network to solve the short-term load forecasting problem, *IEEE Transactions on Power Systems,* 12 (4), 1432-1439.

Chow, T.W.S, Leung, C.T., 1996, Neural networks based short-term load forecasting using weather compensation, *IEEE Transactions on Power Systems*, 11 (4), 1736-1742.

Cloarec, G.M., Ringwood, J.V., Kelly, M.V., 1998, Neuro-fuzzy forecasting of weekly electricity consumption. *In*: Jurgen, H., Zimmermann, H.J., eds., *Proceedings, European Congress on Intelligent Techniques and Soft Computing, Aachen, Germany, September 1998*. Aachen: Elite foundation, 1939-1939.

Connor, J., Atlas, L.E., Martin, D.R., 1992, Recurrent networks and NARMA modelling, *Advances in Neural Information Processing Systems*, 4, 301-308.

Commarmond, P., Ringwood, J.V., 1997, Quantitative fuzzy modelling of short-time-scale electricity consumption, *In: Proceedings Irish Signals and Systems Conf., Derry, Ireland, June* 1997, Derry: Magee College, (not paginated on CD-ROM).

Cryer, J.D., 1986, *Time Series Analysis*, USA, Boston: Duxbury Press..

Dagpunar, J., 1988, *Principles of Random Variate Generation*, UK, Oxford: Clarendon Press.

Darbellay, G.A., 1999, An estimator of the mutual information based on a criterion for independence, *Journal of Computational Statistics and Data Analysis*, 32, 1-17.

Darbellay, G.A., Slama, M., 2000, Forecasting the short-term demand for electricity, do neural networks stand a better chance?, *International Journal of Forecasting*, 16, 71-83.

Dash, P.K., Liew, A.C., Rahman, S., 1995a, Peak load forecasting using a fuzzy neural network, *Electric Power Systems Research*, 32, 19-23.

Dash, P.K., Liew, A.C., Rahman, S., Dash, S., 1995b, Fuzzy and neuro-fuzzy computing models for electric load forecasting, *Engineering Applications of Artificial Intelligence*, 8 (4), 423-433.

Dash, P.K., Liew, A.C., Rahman, S., Satpathy, J.K., Ramakrishna, G., 1994, *Engineering Intelligent Systems*, 2 (3), 185-199.

Dash, P.K., Satpathy, H.P., Liew, A.C., 1998, A real-time short-term load forecasting system using a self-organising fuzzy neural network, *Engineering Applications of Artificial Intelligence*, 11, 307-316.

Dash, P.K., Satpathy, H.P., Liew, A.C. , Rahman, S., 1997, A real-time short-term load forecasting system using functional link network, *IEEE Transactions on Power Systems,* 12 (2), 675-681.

Dehdashti, A.S., Tudor, J.R., Smith, M.C., 1982, Forecasting of hourly load by pattern recognition, *IEEE Transactions on Power Apparatus and Systems,* 101 (9), 3290-3295.

Di Caprio, U., Genesio, R., Pozzi, S., Vinino, A., 1985, Comparison of ARMA and extended Wiener filtering for load prediction at ENEL. *In*: Bunn, E.D., Farmer, E.D., eds. *Comparative Models for Electrical Load Forecasting*, NY: John Wiley and sons, 109-119.

Douglas, A.P., Breipohl, A.M., Lee, F.N., Adapa, R., 1998a, Risk due to load forecast uncertainty in short term power system planning, *IEEE Transactions on Power Systems,* 13 (4), 1493-1499.

Douglas, A.P., Breipohl, A.M., Lee, F.N., Adapa, R., 1998b, The impacts of temperature forecast uncertainty on Bayesian load forecasting, *IEEE Transactions on Power Systems,* 13 (4), 1507-1513.

Drezga, S, Rahman, S., 1998, Input variable selection for ANN-based short-term load forecasting, *IEEE Transactions on Power Systems*, 13 (4), 1238-1244.

Dunn, J.C., 1974, A fuzzy relative of the ISODATA process and its use in detecting compact well separated clusters, *Journal of Cybernetics*, 3 (3), 35-57.

Elkateb, M.M., Solaiman, K., Al-Turki, Y., 1998, A comparative study of medium-weather-dependant load forecasting using enhanced artificial/fuzzy neural network and statistical techniques, *Neurocomputing*, 23, 3-13.

Fan, J.Y., McDonald, J.D., 1994, A real-time implementation of short-term load forecasting for distribution power systems, *IEEE Transactions on Power Systems,* 9 (2), 988-994.

Fay, D, Ringwood, J.V., Condon, M., Kelly, M., 2000, A data fusion model for Irish electricity load forecasting , *in: Proceedings, Irish Signals and Systems Conference,* Maynooth, Ireland, 25th - 27th June, Techman, Maynooth, 135-140.

Feng, W., Keng, Y.E., Qi, L.Y., Jun., L., Shan, Y.C., 1998, Short term load forecasting based on weather information, *in: Proceedings, 1998 International Conference on Power System Technology,* Beijing, China, *August*, IEEE, 572-575.

Fiordaliso, A., 1998, A non-linear forecasts combination method based on Takagi-Sugeno fuzzy systems, *International Journal of Forecasting*, 14, 367-379.

Franses, P.H., Koehler, A.B., 1998, A model selection strategy for time series with increasing seasonal variation, *International Journal of Forecasting*, 14, 405-414

Fuller, S., Harris, G., 1999, Verification: analysis of forecasting accuracy, *In: Numerical Weather Prediction Gazettte*, Bracknell, UK: U.K. Meteorological Office, 14-15.

Funahashi, K.I., 1989, On the approximate realization of continuous mapping by neural networks, *Neural Networks*, 2 (3), 183-192.

Gelb, A., 1984, *Applied Optimal Estimation*, 8th ed., Massachusetts USA: MIT Press.

Geweke, J., 2001, Bayesian econometrics and forecasting, *Journal of Econometrics*, 100, 11-15.

Giersbergen, N.P.A., Kiviet, J.F., 2002, How to implement the bootstrap in static or stable dynamic regression models: test statistic versus confidence region approach, *Journal of Econometrics*, 108 (1), 133-156

Gontar, Z., Sideratos, G., Hatziargyriou, N.D., 2004, Short-term load forecasting using radial basis function networks, *In: proceedings, 3rd Hellenic Conference on AI, SETN 2004*, Samos, Greece, May 5-8, 432-438

Grechanovsky, E., Pinsker, I., 1995, Conditional p-values for the F-statistic in a forward selection procedure, *Computational Statistics and Data Analysis*, 20, 239-263.

Gupta, P.C., 1985, Adaptive short-term forecasting of hourly loads using weather information. *In*: Bunn, E.D., Farmer, E.D., eds. *Comparative Models for Electrical Load Forecasting*, NY: John Wiley and sons, 43-56.

Haida, T., Muto, S., 1994, Regression based peak load forecasting using a transformation technique, *IEEE Transactions on Power Systems,* 9 (4), 1788-1794.

Handschin, E., Dörnemann, C., 1988, Bus load modelling and forecasting, *IEEE Transactions on Power Systems,* 3 (2), 627-633.

Hara, K., Fukui, S., Iba, K., 1997, A new environment for improving accuracy of load forecasting using load and weather data. *In: proceedings, 4th International Conference on Advances in Power System Control, Operation and Management, Hong Kong, November*, London, UK: IEE, 135-8.

Harris, J.L., Liu, M., 1993, Dynamic structural analysis and forecasting of residential electricity consumption, *International Journal of Forecasting*, 9, 437-455.

Harris, C.J., Moore, C.G., Brown, M., 1993, *Intelligent control, aspects of fuzzy logic and neural nets*, NJ : World Scientific.

Harvey, A.C., 1981, *Time Series Analysis*, Hertfordshire UK: Philip Allan.

Harvey, A.C., 1984, A unified view of statistical forecasting procedures, *Journal of Forecasting*, 3, 245-275.

Harvey, A.C., 1994, *Forecasting, Structural Time Series Models and the Kalman Filter*, 4, Cambridge UK: Cambridge University Press.

Hassibi, B., Stork, D.G., Wolff, G.J., 1992, Optimal brain surgeon and general network pruning, *in: Proceedings, IEEE International Conference on Neural Networks*, 1, San Francisco: CA, IEEE, 293-299.

Haykin, S., 1999, *Neural Networks, A comprehensive foundation*, 2, London UK: Prentice Hall Int.

Hippert, S.H., Pedriera, C.E., Souza, R.C., 2001, Neural networks for short-term load forecasting: a review and evaluation, *IEEE Transactions on Power Systems*, 16 (1), 44-55.

Ho, K.L., Hsu, Y.Y., Liang, C.C., Lai, T.S., 1990, Short term load forecasting of Taiwan power system using a knowledge-based expert system, *IEEE Transactions on Power Systems*, 5 (4), 1214-1221.

Hocking, R.R., 1976, The analysis and selection of variables in linear regression, *Biometrics*, 32, 1-49.

Hsu, Y.Y., Yang, C.C., 1991a, Design of artificial neural networks for short-term load forecasting Part I: Self-organizing feature maps for day type identification, *IEE Proceedings-C*, 138(5), page 407-413.

Hsu, Y.Y., Yang, C.C., 1991b, Design of artificial neural networks for short-term load forecasting Part II: Multilayer feedforward networks for peak load and valley load forecasting, *IEE Proceedings-C*, 138(5), page 414-418.

Hubele, N.F., Cheng, C.S., 1990, Identification of seasonal short-term forecasting models using statistical decision functions, *IEEE Transactions on Power Systems*, 5 (1), 40-45.

Hyde, O., Hodnett, P.F., 1997a, An adaptable automated procedure for short-term electricity load forecasting, *IEEE Transactions on Power Systems*, 12 (1), 84-94.

Hyde, O., Hodnett, P.F., 1997b, Modelling the effect of weather in short-term electricity load forecasting, *Math. Engng. Ind.*, 6 (No.2), 155-169.

IEEE committee report, 1985, Problems associated with unit commitment in uncertainty, *IEEE Transactions on Power Apparatus and Systems*, 104 (8), 2072-2078.

Ihara, S., Tani, M., Tomiyama, K., 1994, Residential load characteristics observed at KEPCO power system, *IEEE Transactions on Power Systems*, 9 (2), 1092-1101.

Imai, H., Tanaka, A., Miyakoshi, M., 1998, A method of identifying influential data in fuzzy clustering, *IEEE Transactions on Fuzzy Systems*, 6 (1), 90-101.

Infield, D.G., Hill, D.C., 1998, Optimal smoothing for trend removal in short term load forecasting, *IEEE Transactions on Power Systems*, 13 (3), 1115-1120.

Jabbour, K., Riveros, J.F.V., Landsbergen, D., Meyer, W., 1988, ALFA: Automated load forecasting assistant, *IEEE Transactions on Power Systems*, 3 (3), 908-914.

Jain, A.K, Dubes, R.C., 1998, *Algorithms for Clustering Data*, N.J.: Prentice Hall.

Kalaitzakis, K. Stavrakakis, G.S. Anagnostakis, E.M., 2002, Short-term load forecasting based on artificial neural networks parallel implementation, *Electric Power Systems Research*, 63, 185-196.

Kassaei, H.R., Keyhani, A., Woung, T., Rahman, M., 1999, A hybrid neural network bus load modelling and prediction, IEEE *Transactions on Power Systems*, 14 (2), 718-724.

Kazmier, L.J., Pohl, N.F., 1987, *Basic Statistics for Business and Economics*, McGraw Hill, Singapore.

Khotanzad, A., Davis, M.H., Abaye, A., Maratululam, D.J., 1996, An artificial neural network hourly temperature forecaster with applications in load forecasting, *IEEE Transactions on Power Systems*, 11 (2), 870-876.

Kiartzis, S., Kehagias, A., Bakirtzis, A., Petridis, V., 1997, Short term load forecasting using a Bayesian combination method, *Electrical Power and Energy Systems*, 19 (3), 171-177.

Kiartzis, S.J., Bakirtzis, A.G., Petridis, V., 1995, Short-term load forecasting using neural networks, *Electric Power Systems Research*, 33, 1-6.

Kim, K.H., Youn, H.S., Kang, Y.C., 2000, Short-term load forecasting for special days in anomalous load conditions using neural networks and fuzzy inference models, *IEEE Transactions on Power Systems*, 15 (2), 559-565.

Kodogiannis, V.S., Anagnostakis, E.M., 1999, A study of advanced learning algorithms for short-term load forecasting, *Engineering Applications of Artificial Intelligence*, 12, 159-173.

Kohonen, T., 1990, The self-organising map, *Proceedings IEEE*, 78 (9), 1464-1480.

Kosko, B., 1997, *Fuzzy Engineering*, N.J.: Prentice Hall.

Le Cun, Y., Denker, J.S., Solla, S.A., 1990, Optimal brain damage, *In: proceedings, Advances in Neural Information Processing Systems, 2, San Mateo: CA:* Morgan-Kaufmann, 598-605.

Lee, K.Y., Cha, Y.T., Park, J.H., 1992, Short-term load forecasting using an artificial neural network, *IEEE Transactions on Power Systems*, 7 (1), 124 - 132.

Lertpalangsunti, N., Chan, C.W., 1998, An architectural framework for the construction of hybrid intelligent forecasting systems: application for electricity demand prediction, *Engineering Applications of Artificial Intelligence*, 11, 549-565.

Liang, R.H., Cheng, C.C, 2000, Combined regression-fuzzy approach for short-term load forecasting, *IEE Proceedings-Generation, Transmission and Distribution*, 147 (4), 261-266.

Liu, K., Subbarayan, S., Shoults, R.R., Manry, M.T., Kwan, C., Lewis, F.L., Naccarino, J., 1996, Comparison of very short term load forecasting techniques, *IEEE Transactions on Power Systems*, 11 (2) , 877-882.

Ljung, L., 1987, *Systems Identification – Theory for the user*, Prentice Hall.

Lonergan, T., 1994, Linguistic modelling of electricity consumption using fuzzy logic, *Masters Thesis*, Dublin City University, 1994.

Lonergan, T., Ringwood, J.V., 1995, Linguistic modelling of electricity consumption using fuzzy modelling techniques, *In: proceedings, Irish DSP and Control Colloquium, Belfast, UK, June* 1995.

Long, D.W., Brown, M., Harris, C., 2000, Principal component analysis in time series modelling, *In: proceedings, 35th Universities Power Engineering Conference, Belfast, 6th-8th September*. Belfast, UK: Queens University, (Not paginated on CD-ROM).

Lu, Q.C., Grady, W.M., Crawford, M.M., Anderson, G.M., 1989, An adaptive nonlinear predictor with orthogonal escalator structure for short-term load forecasting, *IEEE Transactions on Power Systems,* 4 (1), 158-164.

Lu, C.N., Wu, H.T., Vemuri, S., 1993, Neural network based short term load forecasting, *IEEE Transactions on Power Systems,* 8 (1), 336-342.

Mars, P, Chen, J.R., Nambiar, R., 1996, *Learning Algorithms: Theory and Applications in Signal Processing, Control and Communications*, N.Y.: CRC Press.

Mastorocotas P.A., Theocharis, J.B., Bakirtzis, A.G., 1999, Fuzzy modelling for short term load forecasting using the orthogonal least squares method, *IEEE Transactions on Power Systems,* 14 (1), 29-35.

Mbamalu, G.A.N., El-Hawary, M.E., 1993, Load forecasting via sub-optimal seasonal auto-regressive models and iteratively reweighted least squares method, *IEEE Transactions on Power Systems,* 8 (1), 343-348.

McCabe, H.,1991, Minimum trace fusion of N sensors with arbitrary correlated sensor to sensor errors, *Proceedings of IFAC Conference on Distributed Intelligent Systems*, Virginia, USA, August 1991, 229-234.

Mitchell, J., 1992, Comparing feedforward neural network models for time series prediction, *In: proceedings, Neural Research and Applications, Belfast, 25th-26th June*. Belfast, UK: Queens University, 175-182.

Miyake, T., Murata, J., Hirasawa, K., 1995, One-day through seven-day-ahead electrical load forecasting in consideration of uncertainties of weather information, *Electrical Engineering in Japan*, 115 (8), 135-142.

Mizukami, Y., Nishimori, T., 1993, Maximum electric power demand prediction by neural networks, *In: proceedings, Second Joint Forum on Applications of Neural Networks to Power Systems, Yokohama, Japan, 19th-22nd April.*

Moghram, I., Rahman, S., 1989, Analysis and evaluation of five short-term load forecasting techniques, *IEEE Transactions on Power Systems,* 4 (4), 1484-1491.

Mohamad, E.A., Mansour, M.M., El-Debeiky, Mohamad, K.G., Rao, N.D., Ramakrishna, G., 1996, Results of Egyptian unified grid hourly load forecasting using an artificial neural network with expert system interface, *Electric Power Systems Research*, 39, 171-177.

Mohammed, O., Park, D., Merchant, R., Dinh, T., Tong, C., Azeem, A., 1995, Practical experiences with an adaptive short-term load forecasting system, *IEEE Transactions on Power Systems,* 10 (1), 254-265.

Moral, M.H., Valderrama, M.J., 1997, A principal component approach to dynamic regression models, *International Journal of Forecasting*, 13, 237-244.

Mori, H., Kobayashi, 1996, Optimal fuzzy inference for short-term load forecasting, *IEEE Transactions on Power Systems,* 11 (1), 390-396.

Morrissey, A.P., Van Toai, N.N., 1988, Forecasting the total impact of major new industrial plant on company electrical energy requirements, *IEEE Transactions on Power Systems,* 3 (3), 1084-1089.

Moutter S.P., Bodger, B.E., Gough, P.T. 1986, Spectral decomposition and extrapolation of variations in electricity loading, *IEE Proceedings*, 113 (C), 247-255.

Muller, H., Petrisch, G., 1998, Energy and load forecasting by fuzzy-neural networks. *In*: Jurgen, H., Zimmermann, H.J., eds., *Proceedings*, *European Congress on Intelligent Techniques and Soft Computing, Aachen, Germany, September*. Aachen: Elite foundation, 1925-1929.

Muller, H., Schatzel, F., 1999, Daily load curve clustering and prediction by neural model tool box for power systems with non-stochastic load components, *In*: *Proceedings*, *the* 5th *European Control Conference*, Karlsruhe, Germany, August, (not paginated on CD-ROM)

Murray, F.T., 1996, Forecasting methodologies for electricity supply systems, Dublin: *Ph.D. thesis*, School of Electronic Engineering, Dublin City University, Ireland.

Murray, F.T., Ringwood, J.V., Austin, P.C., 2000, Integration of multi-time-scale models in time series forecasting [electricity consumption], *International Journal of Systems Science*, 31 (10), 1249-1260.

Ng, C.N., Young, P.C., 1990, Recursive estimation and forecasting of non-stationary time series, *Journal of Forecasting*, 9, 173-204.

Osula, D.O.A., Adebisi, O, 2001, Testing the stability of travel expenditures in Nigeria, *Transportation Research*, 35 (A), 269-287.

Otto, P., Schunk, P.O., 1999, Fuzzy based time series forecasting of electric load. *In*: *Proceedings*, *the* 5th *European Control Conference*, Karlsruhe, Germany, August 1999, (not paginated on CD-ROM).

Palit, A.K., Popovic, D., 2000, Nonlinear combination of forecasts using artificial neural network, fuzzy logic and neuro-fuzzy approaches, *in: Proceedings, IEEE international conference on fuzzy systems*, 2, 566-571.

Papadakis, S.E., Theocharis, J.B., Bakirtzis, A.G., 1999, Fuzzy short-term load forecasting models based on load-curve prototype fuzzy clustering, *In*: *Proceedings*, *the* 5th *European Control Conference*, Karlsruhe, Germany, August, (not paginated on CD-ROM)

Papalexopoulos, A.D., Hesterburg, C.T., 1990, A regression based approach to short term system load forecasting, *IEEE Transactions on Power Systems,* 5 (4), 1535-1547.

Papoulis, A., 1991, *Probability, Random Variables, and Stochastic Processes*, 3rd ed., Singapore: McGraw Hill International.

Park, D.C., El-Sharkawi, M., Marks, R., 1991a, An adaptively trained neural networks, *IEEE Transactions on Neural Networks,* 23(3), 334-345.

Park, D.C., El-Sharkawi, M., Marks, R., 1991b, Electric load forecasting using an artificial neural network, *IEEE Transactions on Power Systems,* 6(2), 442-449.

Park, D., Mohammed, O., Azeem, A., Merchant, R., Dinh, T., Tong, H., 1993a, Load curve shaping using neural networks, *in: Proceedings, Second International Forum on Applications of Neural Networks to Power Systems, Yokohama, Japan, April.* IEEE, 290-295.

Park, J.H., Park, Y.M., Lee, K.Y., 1991c, Composite modelling for adaptive short-term load forecasting, *IEEE Transactions on Power Systems,* 6 (2), 450-457.

Parzen, E., 1982, ARARMA models for time series analysis and forecasting, *Journal of Forecasting*, 1, 67-82.

Pelikan, E., Matejka, P., Slama, M., Vinkler, K., 1996, Interactive forecasting of the electric load using Kohonen self-organizing feature maps. *In*: *Proceedings, International Neural Network Society 1996 Annual Meeting*, Mahwah, NJ, USA. NJ: Lawrence Erlbaum Assoc, 443-446.

Peng, T.M., Hubele, N.F., Karadg, G.G., 1992, Advancement in the application of neural networks for short-term load forecasting, *IEEE Transactions in Power Systems*, 7 (1), 250–257.

Pere, A., 2000, Comparison of two methods for transforming height and weight to normality, *Annals of Human Biology*, 27 (1), 35-45.

Perrone, M.P, Cooper, L.N., 1993, When networks disagree: Ensemble methods for hybrid neural networks, in *Artificial Neural Networks for Speech and Vision*, R.J. Mammone (eds), NY: Chapman & Hall.

Priestly, M.B., 1981, *Spectral Analysis and Time Series. Volume I: Univariate Series*. Academic Press.

Priestly, M.B., 1981, *Spectral Analysis and Time Series. Volume II: Multivariate Series, Prediction and Control*. Academic Press.

Rahman, S., Bhatnagar, R., 1988, An expert system based algorithm for short-term load forecasting, *IEEE Transactions on Power Systems,* 3 (2), 392-399.

Rahman, S., Hazim, O., 1993, A generalised knowledge-based short-term load forecasting technique, *IEEE Transactions on Power Systems,* 8 (2), 508-514.

Rajurkar, K.P.; Newill, R.E., 1985, Multiple series modelling and forecasting of short-term load demand by data dependent systems. *In*: *Proceedings of the International Conference on Cybernetics and Society, New York, USA*. NY: IEEE, 448-52.

Ramanathan, R., Engle, R., Granger, C.W.J., Araghi, F.V., Brace, C., 1997, Short-run forecasts of electricity loads and peaks, *International Journal of Forecasting*, 13, 161-174.

Ranaweera, D.K., Karady, G.G., Farmer, R.G., 1995, A probabilistic approach to handle uncertainties in load forecasting, *In*: *Proceedings of the American Power Conference, Chicago, IL, USA*. NY: IEEE, 1701-1706.

Ranaweera, D.K., Hubele, N.F., Karady, G.G., 1996, Fuzzy logic for short term load forecasting, *Electrical Power and Energy Systems*, 18 (4), 215-222.

Reid. D.J., 1968, Combining three estimates of gross domestic products, *Economica*, 35, 431-444.

Reinschmidt, K.F., 1995, Artificial neural networks for short term load forecasting, *In: Proceedings Annual Meeting of the American Power Conference, Chicago, Illinois*, 1144-1149.

Rencher, A.C., 1995, *Methods of multivariate analysis*, John Wiley and Sons, New York.

Roecker, E.B., 1991, Prediction error and it's estimation for subset selected models, *Technometrics*, 33, 459-468.

Rong, L., Ping, W., Wenlei, H., 2000, A novel method for wine analysis based on sensor fusion technique, *sensors and actuators*, B 66, 246-250.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986, Learning representations by back propagating errors error, *Nature*, 323, 533-536.

Schalkoff, R.J., 1997, *Artificial Neural Networks*, Singapore: McGraw Hill.

Schetzen, M., 1980, *The Volterra and Weiner Theories of Non-Linear Systems*, USA: John Wiley and sons.

Seber, G.A.F., Wild, C.J., 1989, *Nonlinear Regression*, USA, NY: John Wiley and sons.

Sforna, M., 1995, Searching for the electric-weather temperature function by using the group method of data handling, *Electric Power Research*, 32, 1-9.

Shamseldin, A.Y., O'Connor, K.M., Liang, G.C., 1997, Methods for combining the outputs of different rainfall-runoff models, *Journal of Hydrology*, 197, 203-229.

Sharaf, A.M., Lie, T.T., 1995, A novel neuro-fuzzy based self-correcting online electric load forecasting model, *Electric Power and System Research*, 34, 121-125.

Sinha, N.K., 1991, *Linear Systems*, NY, USA: John Wiley and sons.

Song, X.H., Hopke, P.K., 1996, Kohonen neural network as a pattern recognition method based on weight interpretation, *Analytica Chimica Acta*, 334, 57-64.

Sridhar, D.V., Bartlett, E.B., Seagrave, R.C., 1999, An information theoretic approach for combining neural network process models, *Neural Networks*, 12, 915-926.

Srinivasan, D., 1998, Evolving artificial neural networks for short term load forecasting, *Neurocomputing*, 23, 265-276.

Srinivasan, D., Chang, C.S., Liew, A.C., 1995, Demand forecasting using fuzzy neural computation, with special emphasis on weekend and public holiday forecasting, *IEEE Transactions on Power Systems*, 10 (4), 1897-1903.

Srinivasan, D., Tan, S. S., Chang, C. S., Chan, E. K., 1999, Parallel neural network-fuzzy expert system for short-term load forecasting: system implementation and performance evaluation, *IEEE Transactions on Power Systems,* 14 (3), 1100-1106.

Tamimi, M., Egbert, R., 2000, Short term electric load forecasting via fuzzy neural collaboration, *Electric Power Systems Research*, 56, 243-248.

Taylor, J.W., Buizza, R., 2003, Using weather ensemble predictions in electricity demand forecasting, *International Journal of Forecasting*, 19 (1), 57-70.

Train, T., Ignelz, P., Engle, R., Granger, C., Ramanathan, R., 1984, The billing cycle and weather variables in models of electricity sales, *Energy*, 9, 1041-1047.

Vemuri, S., Huang, W.L., Nelson, D.J., 1981, On-line algorithm for forecasting hourly loads of an electric utility, *IEEE Transactions on Power Apparatus and Systems,* PAS-100, 3775-3784.

Vermaak, J., Botha, E.C., 1998, Recurrent neural networks for short-term load forecasting, *IEEE Transactions on Power Systems,* 13 (1), 126-132.

Weigend, A.S., Rumelhart, D.E., Huberman, B.A., 1991, Generalisation by weight elimination with application to forecasting, *Advances in Neural Information Processing Systems*, 3, 875-882.
Wisnowski, J.W., Montgomery, D.C., Simpson, J.R., 2001, A comparative analysis of multiple outlier detection procedures in the linear regression model, *Computational Statistics and Data Analysis*, 36, 351-382.

Xiong, L., Shamseldin, A.Y., O'Connor, K.M., 2001, A non-linear combination of the forecasts of rainfall-runoff models by the first order Takagi-Sugeno fuzzy system, *Journal of Hydrology*, 245, 196-217.

Yang, H.T., Huang, C.M., 1998, A new short-term load forecasting approach using self-organising fuzzy ARMAX models, *IEEE Transactions on Power Systems,* 13 (1), 217-225.

Yang, H.T., Huang, C.M., Huang, C.L., 1996, Identification of ARMAX model for short-term load forecasting: an evolutionary programming approach, *IEEE Transactions on Power Systems,* 11 (1), 403-408.

Yoo, H., Pimmel, R.L., 1999, Short-term load forecasting using a self-supervised adaptive neural network, *IEEE Transactions on Power Systems,* 14 (2), 779-784.

Young, P.C., 1988, Recursive extrapolation, interpolation and smoothing of non-stationary time-series, *In: proceedings, IFAC Symposium on Identification and System Parameter Estimation, Beijing, China.* PRC, 35-46.

Yousef, M.T., El-Alayly, A.A., 2000, Long term planning using artificial neural networks technique, *In: proceedings, 35th Universities Power Engineering Conference, Belfast, UK, September* 2000. London, UK: IEEE, (Not paginated on CD-Rom).

Yousef, M.T., 2000, Neural networks as a tool for load forecasting, *In: proceedings, International Conference on Engineering Applications of Neural Networks, Cagliari, Italy, July* 2000. Italy: University of Cagilari, 144-148.

Yu, Z., 1996, A temperature match based optimisation method for daily load prediction considering DLC effect, *IEEE Transactions on Power Systems,* 11 (2), 728-733.

Appendix A

Table of variables used in the multi-time scale technique.

Variable	Dimension	Description
n	1×1	Length of the state vector $\theta(k)$
r	1×1	Number of freed states of $\theta(k)$
$x(k)$	1×1	Load at time k
$\theta(k)$	$n\times 1$	State vector at time k
$\boldsymbol{H}$	$1\times n$	Observation matrix
$\boldsymbol{\Phi}$	$n\times n$	Transition matrix
$\theta_1(k)$	$(n-r)\times1$	Fixed states at time k
$\theta_2(k)$	$r\times1$	Freed states at time k
$\boldsymbol{\Phi}_1$	$(n-r\times(n-r)$	Partition of $\boldsymbol{\Phi}$ for fixed states
$\boldsymbol{\Phi}_2$	$r\times r$	Partition of $\boldsymbol{\Phi}$ for freed states
$\boldsymbol{H}_1$	$1\times(n-r)$	Partition of $\boldsymbol{H}$ for fixed states
$\boldsymbol{H}_2$	$1\times r$	Partition of $\boldsymbol{H}$ for freed states
N	1×1	Number of points used in the smoothing constraint
M	1×1	Number of cardinal point forecasts used.
P	1×1	Number of end-sum forecasts used.
$\hat{y}_{c_i}$	1×1	i^{th} cardinal point forecast
c_i	1×1	The distance from the forecasting origin to the i^{th}
δ_{c_i}	1×1	The deviation of the i^{th} cardinal point from the MTS
$\hat{y}_{s_i}$	1×1	i^{th} end-sum forecast.
S	1×1	Length of the summations for end-sum model.
δ_{s_i}	1×1	The deviation of $\hat{y}_{s_i}$ from the summation of the MTS
δ_i	1×1	The deviation of the state space model forecast at $k+i$
$\boldsymbol{A}(k)$	$(N+M+P)\times1$	Left hand side of Equation 3.74.
$\boldsymbol{B}$	$(N+M+P)\times r$	1^{st} term on the right hand side of Equation 3.74.
$\boldsymbol{w}$	$(N+M+P)\times1$	Diagonal of weight vector.
$\boldsymbol{W}$	$(N+M+P)\times(N+M+P)$	Diagonal weight matrix.
$\Lambda(k)$	$(N+M+P)\times1$	Vector of deviations.
$\underline{\boldsymbol{A}}$	$(N+M+P)\times K$	$[\boldsymbol{A}(1)\quad \boldsymbol{A}(2)\quad \cdots\quad \boldsymbol{A}(K)]$
$\underline{\boldsymbol{\theta}}_1$	$n-(r\times K)$	$[\boldsymbol{\theta}_1(1)\quad \boldsymbol{\theta}_1(2)\quad \cdots\quad \boldsymbol{\theta}_1(K)]$
$\underline{\boldsymbol{\theta}}_2$	$r\times K$	$[\boldsymbol{\theta}_2^*(1)\quad \boldsymbol{\theta}_2^*(2)\quad \cdots\quad \boldsymbol{\theta}_2^*(K)]$
y	$1\times K$	Vector of actual loads
y_{mts}	$1\times K$	Vector of 1-step MTS load forecasts

Appendix B

Suggested derivation route for gradients used in the multi-time scale technique.

From Section 6.4.2 the derivative of the MTS constraint with respect to the weights may be expressed as:

$$\frac{\partial C(\boldsymbol{w})}{\partial \boldsymbol{w}} = \Omega = \begin{pmatrix} \dfrac{\partial C(\boldsymbol{w})}{\partial w_1} \\ \dfrac{\partial C(\boldsymbol{w})}{\partial w_2} \\ \vdots \\ \dfrac{\partial C(\boldsymbol{w})}{\partial w_{N+M+P}} \end{pmatrix} \tag{B1}$$

The following gradient is now considered which is required in evaluating Equation (B1):

$$\frac{\partial\left(\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{A}\right)}{\partial w_i} = \frac{\partial\left(\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\right)}{\partial w_i}\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{A}(k) + \left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\frac{\partial\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)}{\partial w_i} \tag{B2}$$

Using the following matrix gradient:

$$\frac{\partial\left(X^{-1}\right)}{\partial z} = -X^{-1}\frac{\partial X}{\partial z}X^{-1} \tag{B3}$$

where X is a matrix and z is a scalar, Equation (B2) becomes:

$$\frac{\partial\left(\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{A}(k)\right)}{\partial w_i} =$$

$$\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\frac{\partial\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)}{\partial w_i}\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{A}(k) + \left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\frac{\partial\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)}{\partial w_i} \tag{B4}$$

Considering the partial derivative term in Equation (B4):

$$\frac{\partial\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)}{\partial w_i} = \boldsymbol{B}^T\frac{\partial(\boldsymbol{W})}{\partial w_i}\boldsymbol{B} = \boldsymbol{B}_i\boldsymbol{B}_i^T \tag{B5}$$

where $\boldsymbol{B}_i$ is the i^{th} row of $\boldsymbol{B}$. The result above is obtained by noting that the result of taking the partial derivative is a sparse matrix with an entry equal to one on the i^{th} diagonal element:

$$\frac{\partial(\boldsymbol{W})}{\partial w_i}=\begin{bmatrix}\frac{\partial w_1}{\partial w_i} & 0 & 0 & 0 & 0\\ 0 & \ddots & 0 & 0 & 0\\ 0 & 0 & \frac{\partial w_i}{\partial w_i} & 0 & 0\\ 0 & 0 & 0 & \ddots & 0\\ 0 & 0 & 0 & 0 & \frac{\partial w_{N+M+P}}{\partial w_i}\end{bmatrix}=\begin{bmatrix}0 & 0 & 0 & 0 & 0\\ 0 & \ddots & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & \ddots & 0\\ 0 & 0 & 0 & 0 & 0\end{bmatrix} \quad \text{(B6)}$$

Substituting Equation (B5) into Equation (B4) gives:

$$\frac{\partial\left(\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{A}(k)\right)}{\partial w_i}=$$

$$\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}_i\boldsymbol{B}_i{}^T\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{A}(k)+\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}_i\boldsymbol{B}_i{}^T \quad \text{(B7)}$$

This may be rearranged to give:

$$\frac{\partial\left(\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{A}(k)\right)}{\partial w_i}=\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}_i\boldsymbol{B}_i{}^T\left(\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{A}(k)+I\right) \quad \text{(B8)}$$

Equation (B1) is now reconsidered and expanding $C(\boldsymbol{w})$ using Equation (6.13) gives:

$$\Omega=\begin{pmatrix}\frac{\partial C(\boldsymbol{w})}{\partial w_1}\\ \frac{\partial C(\boldsymbol{w})}{\partial w_2}\\ \vdots\\ \frac{\partial C(\boldsymbol{w})}{\partial w_{N+M+P}}\end{pmatrix}=\begin{bmatrix}\frac{\partial}{\partial w_1}\left(\boldsymbol{\theta}_2^*(1,\boldsymbol{w})-\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{A}(1)\right)\\ \vdots\\ \frac{\partial}{\partial w_{N+M+P}}\left(\boldsymbol{\theta}_2^*(1,\boldsymbol{w})-\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{A}(1)\right)\\ \frac{\partial}{\partial w_1}\left(\boldsymbol{\theta}_2^*(1,\boldsymbol{w})-\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{A}(2)\right)\\ \vdots\\ \frac{\partial}{\partial w_{N+M+P}}\left(\boldsymbol{\theta}_2^*(1,\boldsymbol{w})-\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{A}(2)\right)\\ \vdots\\ \frac{\partial}{\partial w_1}\left(\boldsymbol{\theta}_2^*(1,\boldsymbol{w})-\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{A}(K)\right)\\ \vdots\\ \frac{\partial}{\partial w_{N+M+P}}\left(\boldsymbol{\theta}_2^*(1,\boldsymbol{w})-\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{A}(K)\right)\end{bmatrix} \quad \text{(B9)}$$

The partial differential of the constraint for time k with respect to weight i is now considered:

$$\frac{\partial}{\partial w_i}\left(\boldsymbol{\theta}_2^*(k,\boldsymbol{w})-\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{A}(k)\right)=\frac{\partial\boldsymbol{\theta}_2^*(k,\boldsymbol{w})}{\partial w_i}-\frac{\partial}{\partial w_i}\left(\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{A}(k)\right) \quad \text{(B10)}$$

Equation (B.8) is then substituted into Equation (B10) to give:

$$\frac{\partial}{\partial w_i}\left(\boldsymbol{\theta}_2^*(k,\boldsymbol{w})-\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{A}(k)\right)=$$

$$\frac{\partial\boldsymbol{\theta}_2^*(k,\boldsymbol{w})}{\partial w_i}-\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}_i\boldsymbol{B}_i^T\left(\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{A}(k)+I\right) \quad \text{(B11)}$$

Finally the differential of the weight matrix with respect to the weights may be expressed as:

$$\Omega=\begin{bmatrix}
\frac{\partial\boldsymbol{\theta}_2^*(1,\boldsymbol{w})}{\partial w_1}-\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}_i\boldsymbol{B}_i^T\left(\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{A}(1)+I\right)\\
\vdots\\
\frac{\partial\boldsymbol{\theta}_2^*(1,\boldsymbol{w})}{\partial w_{N+M+P}}-\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}_i\boldsymbol{B}_i^T\left(\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{A}(1)+I\right)\\
\frac{\partial\boldsymbol{\theta}_2^*(2,\boldsymbol{w})}{\partial w_1}-\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}_i\boldsymbol{B}_i^T\left(\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{A}(2)+I\right)\\
\vdots\\
\frac{\partial\boldsymbol{\theta}_2^*(2,\boldsymbol{w})}{\partial w_{N+M+P}}-\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}_i\boldsymbol{B}_i^T\left(\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{A}(2)+I\right)\\
\vdots\\
\frac{\partial\boldsymbol{\theta}_2^*(K,\boldsymbol{w})}{\partial w_i}-\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}_i\boldsymbol{B}_i^T\left(\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{A}(K)+I\right)\\
\vdots\\
\frac{\partial\boldsymbol{\theta}_2^*(K,\boldsymbol{w})}{\partial w_{N+M+P}}-\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}_i\boldsymbol{B}_i^T\left(\left(\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^T\boldsymbol{W}\boldsymbol{A}(K)+I\right)
\end{bmatrix} \quad \text{(B12)}$$

The gradient of $J(\boldsymbol{w})$ with respect to $\boldsymbol{w}$ is now considered. From Equation (6.12), $J(\boldsymbol{w})$ may be expressed as:

$$J(\boldsymbol{w})=\sum_{k=1}^{K}\left(y(k)-y_{mts}(k)\right)^2 \quad \text{(B13)}$$

where the individual elements of $\underline{y}$ and $\underline{y}_{mts}$ have been expanded in the summation above. Differentiating $J(\boldsymbol{w})$ with respect to w_i and noting that $\underline{y}$ is fixed gives:

$$\frac{\partial J(\boldsymbol{w})}{\partial w_i}=2\sum_{k=1}^{K}\frac{\partial y_{mts}(k)}{\partial w_i}\left(y(k)-y_{mts}(k)\right) \quad \text{(B14)}$$

From Equation (3.76) the MTS forecast may be expressed in terms of the altered state vector as:

$$\hat{y}_{mls}(k) = \boldsymbol{H}\boldsymbol{\theta}^{*}(k) = \begin{bmatrix} \boldsymbol{H}_1 & \boldsymbol{H}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}_1(k) \\ \boldsymbol{\theta}_2^{*}(k) \end{bmatrix} \tag{B15}$$

Taking the partial derivative of Equation (B15) with respect to w_i gives:

$$\frac{\partial \hat{y}_{mls}(k)}{\partial w_i} = \frac{\partial}{\partial w_i}\left(\begin{bmatrix} \boldsymbol{H}_1 & \boldsymbol{H}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}_1(k) \\ \boldsymbol{\theta}_2^{*}(k) \end{bmatrix} \right) = \begin{bmatrix} \boldsymbol{H}_1 & \boldsymbol{H}_2 \end{bmatrix} \begin{bmatrix} \dfrac{\partial \boldsymbol{\theta}_1(k)}{\partial w_i} \\ \dfrac{\partial \boldsymbol{\theta}_2^{*}(k)}{\partial w_i} \end{bmatrix} \tag{B16}$$

Noting that $\boldsymbol{\theta}_1(k)$ is not a function of w_i gives:

$$\frac{\partial \hat{y}_{mls}(k)}{\partial w_i} = \begin{bmatrix} \boldsymbol{H}_1 & \boldsymbol{H}_2 \end{bmatrix} \begin{bmatrix} 0 \\ \dfrac{\partial \boldsymbol{\theta}_2^{*}(k)}{\partial w_i} \end{bmatrix} = \boldsymbol{H}_2 \frac{\partial \boldsymbol{\theta}_2^{*}(k)}{\partial w_i} \tag{B17}$$

Substituting Equation (B17) and into Equation (B14) gives:

$$\frac{\partial J(\boldsymbol{w})}{\partial w_i} = \begin{pmatrix} 2\sum_{k=1}^{K} \boldsymbol{H}_2 \dfrac{\partial \boldsymbol{\theta}_2^{*}(k)}{\partial w_i} (y(k) - \boldsymbol{H}_1\boldsymbol{\theta}_1(k) - \boldsymbol{H}_2\boldsymbol{\theta}_2^{*}(k)) \\ 2\sum_{k=1}^{K} \boldsymbol{H}_2 \dfrac{\partial \boldsymbol{\theta}_2^{*}(k)}{\partial w_i} (y(k) - \boldsymbol{H}_1\boldsymbol{\theta}_1(k) - \boldsymbol{H}_2\boldsymbol{\theta}_2^{*}(k)) \\ \vdots \\ 2\sum_{k=1}^{K} \boldsymbol{H}_2 \dfrac{\partial \boldsymbol{\theta}_2^{*}(k)}{\partial w_i} (y(k) - \boldsymbol{H}_1\boldsymbol{\theta}_1(k) - \boldsymbol{H}_2\boldsymbol{\theta}_2^{*}(k)) \end{pmatrix} \tag{B18}$$

At this point the mathematics was found to be intractable and a numerical solution is proposed as mentioned in Section 6.4.2.