Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®
Neurocomputing 55 (2003) 469–498

NEUROCOMPUTING

www.elsevier.com/locate/neucom

24-h electrical load data—a sequential or partitioned time series?

Damien Fay[a,*], John V. Ringwood[b], Marissa Condon[a], Michael Kelly[c]

[a] *Dublin City University, Glasnevin, Dublin 9, Ireland*
[b] *NUI Maynooth, Maynooth, Co. Kildare, Ireland*
[c] *Electricity Supply Board, Dublin 2, Ireland*

Accepted 11 March 2003

Abstract

Variations in electrical load are, among other things, hour of the day dependent, introducing a dilemma for the forecaster: whether to partition the data and use a separate model for each hour of the day (the *parallel* approach), or use a single model (the *sequential* approach). This paper examines which approach is appropriate for forecasting hourly electrical load in Ireland. It is found that, with the exception of some hours of the day, the sequential approach is superior. The final solution however, uses a combination of linear sequential and parallel neural models in a multi-time scale formulation.

Keywords: Load forecasting; Time series analysis; Multi-layer perceptrons; Principal component analysis

1. Introduction

Short term load forecasting refers to forecasts of electricity, on an hourly basis, from one to several days ahead. The amount of excess electricity production (or spinning reserve) required to guarantee supply, in the event of an underestimation, is determined by the accuracy of these forecasts. Conversely, overestimation of the load leads to sub-optimal scheduling (in terms of production costs) of power plants (unit commitment). In accordance with the Electricity Regulation Act of 1999, a deregulated market

* Corresponding author.
E-mail address: fayd@eeng.dcu.ie (D. Fay).

0925-2312/03/$ - see front matter
doi:10.1016/S0925-2312(03)00390-4

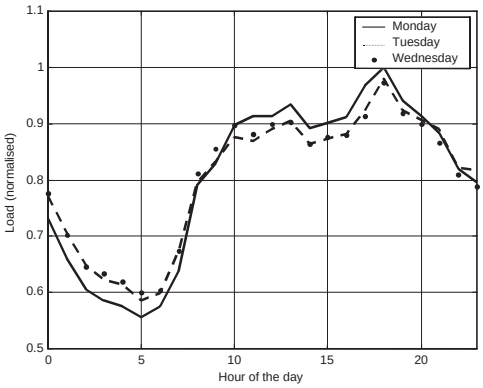

Fig. 1. Typical working day loads.

structure was set up, which should lead to increased impetus to reducing forecast error and the associated costs.

Electrical demand is driven by economic and human activity, which has obvious daily, weekly and yearly cycles, as well as a long-term trend and special periods such as bank holidays, Christmas etc., all of which are reflected in load data. The *load curve* (i.e. the load over a day) for three typical *working days* (Mondays to Fridays) is shown in Fig. 1, reflecting daily human activity.

Day to day variations in the load curves are dependent on weather, hour of the day and previous loads. This hour of day dependence is the focus of this paper. In the general case, the load on hour i of day k, $y_i(k)$ may be expressed as a function, f, of previous loads, current and previous weather inputs, the hour of the day, the day, and an error term as

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

where $\boldsymbol{U}_i(k)$ is a vector of causal variables (weather inputs) 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 Eq. (1) 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 Eq. (1) becomes:

$$y_i(k) = f_s(y_{i-1}(k),\ldots,y_{i-N}(k), y_{i-1}(k-1),\ldots,y_{i-N}(k-P), \boldsymbol{U}_{i-1}(k),\ldots,\boldsymbol{U}_{i-M}(k), \boldsymbol{U}_{i-1}(k-1),\ldots,\boldsymbol{U}_{i-M}(k-Q)) + \varepsilon_i(k). \quad (2)$$

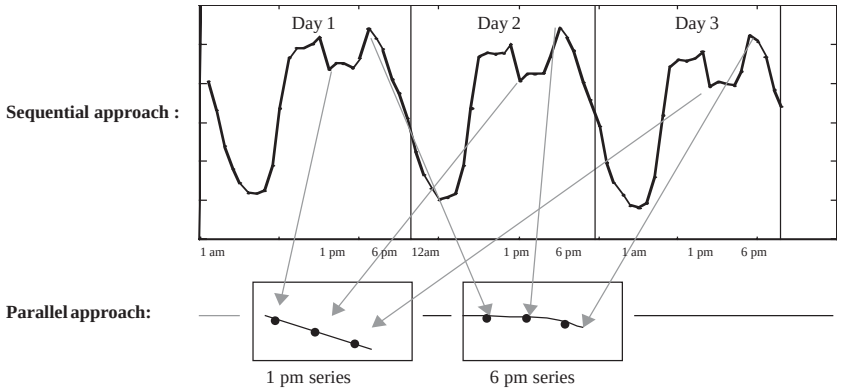

Fig. 2. Constructing partitioned series from electrical load data.

However, in the *parallel* approach, the data is partitioned so that each hour of the day is modelled by a separate function and so Eq. (1) becomes:

$$\begin{aligned}
y_1(k) &= f_1(y_1(k-1),\ldots,y_1(k-P_1),\boldsymbol{U}_1^*(k-1),\ldots,\boldsymbol{U}_1^*(k-Q_1))+\varepsilon_1(k)\\
y_2(k) &= f_2(y_2(k-1),\ldots,y_2(k-P_2),\boldsymbol{U}_2^*(k-1),\ldots,\boldsymbol{U}_2^*(k-Q_2))+\varepsilon_2(k)\\
&\vdots\\
y_{24}(k) &= f_{24}(y_{24}(k-1),\ldots,y_{24}(k-P_{24}),\boldsymbol{U}_{24}^*(k-1),\ldots,\boldsymbol{U}_{24}^*(k-Q_{24}))\\
&\quad+\varepsilon_{24}(k),
\end{aligned} \tag{3}$$

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 01 : 00 h 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.

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

1. although $f_{1,\ldots,24}$ is no longer hour of the day dependent (i is excluded from Eq. (3)), this may not result in $f_{1,\ldots,24}$ being any less complex than f_s. The partitioned series are created by daily sampling of load (Fig. 2). As shown by Harvey [13], a sub-sampled (i.e. taking every pth sample) auto-regressive moving average (ARMA) process is itself an ARMA process of equal or *higher* order. Although it is questionable that load is generated by an ARMA process it has been modelled as such with varying degrees of success by several authors [1,8,27],
2. the number of parameters that need to be calculated in the parallel approach (24 sets of parameters) exceeds that of the sequential approach, where only one set of parameters needs to be calculated [20],
3. the data set is partitioned into 24 separate time series (Fig. 2), reducing the number of input–output pairs for training of the model,

4. training 24 separate models can be overly computationally expensive and
5. calculating the topologies of 24 separate neural networks is prohibitive, as was found in [10].

1.1. Sequential and parallel approaches to load forecasting

For many statistical techniques a sequential approach is taken, in which hour of the day dependence is ignored, such as [1,6–8,27]. As observed by Connor et al. [6], 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. [6] 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 and a multi-layer perceptron (MLP) neural network). This was reported as being due to the increased complexity of the modelling task. Darbellay and Slama [7] similarly observed that an ARMA model, which ignores hour of the day dependence, was superior to a feedforward neural network, 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 [15,25]. In the study by Connor et al. [6], it was found that the parallel approach vastly improved the forecasting performance, where recurrent neural networks were used as the modelling tool. In fact, this was found to be the optimal technique. In contrast, Lee et al. [20] found the performance of parallel and sequential models which used a MLP neural network, 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. [22] 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 [18,22].

A number of other studies [12,17,24] have examined combining the sequential and parallel approaches. This combined approach is known as the multi-time scale approach and adjusts the forecasts of a sequential model with those of parallel models. For example, [12] 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.

1.2. Neural networks for load forecasting

Neural networks have been found by many authors to give excellent results for short term load forecasting [5,6,14], due to the presence of non-linear auto-regressive components in load [7,10] and the non-stationary nature of the series [28]. For implementing dynamic time series models, basic choices of network type include feedforward

networks (MLP [18] or radial basis function (RBF) [19] networks) with delayed inputs (external recurrence), or locally recurrent networks [6].

In terms of feedforward networks, the use of RBF neural networks as time series models (as is the case here) has been questioned by Mitchell [23] for several reasons, mainly due to dimensionality difficulties. Kodogannias and Anagnostakis [19] similarly noted that, for the specific case of short-term load forecasting, the number of RBF inputs is severely restricted.

As electrical load data is, in general, a non-stationary time series, recurrent neural networks would appear to be a suitable choice of model [6,7,28]. However, as noted by Connor et al. [6] the difference between load forecasting models which use recurrent neural networks and those using feedforward neural networks, is the assumption in the latter case that the non-stationarity of the load can be removed by a stationarity transform. The non-stationarity in load arises from the long-term trend, which changes very slowly from day to day. Thus, for the forecasting horizon required in short term load forecasting (i.e. up to several days ahead), removing the non-stationarity is not a difficult task. Also, recurrent neural networks are difficult to train and require large data sets [4].

Like recurrent neural networks, MLPs can model stationary time series and do not suffer from the restrictions on the dimension of the input as RBF networks do. However, feedforward neural networks are less complex and less computationally expensive to train than locally recurrent neural networks [4]. Therefore, MLPs are used in this study.

1.3. Paper layout

The paper is laid out as follows: Section 2 describes the data set under examination and Section 3 examines the hour-of-day dependency in that data. Sections 4 and 5 develop parallel and sequential models, respectively, with Section 5 also containing a multi-time-scale formulation which allows extra inputs from the parallel models to be included in a sequential formulation. Section 6 presents comparative results for both parallel and sequential philosophies with conclusions drawn in Section 7.

2. Data set details

A database containing electricity demand, actual temperature, wind speed and humidity from 1987 to 1998 on an hourly basis is available. Data between Tuesday and Thursday in the months January to March has been selected so as to avoid the exceptions associated with weekend, Christmas and changes due to the daylight saving hour. Thus in total there are 30 days of data selected from each full year of data. Though likely that the summer period would require a separate model set, it is felt that focusing on the winter period presents the greatest forecasting challenge, since the winter load data has considerably more variability than summer. Also note that while actual weather data has been used in this analysis, future forecasts will utilise forecasted weather variables, as documented in [9].

Table 1
Segmentation of data set

Set	Training	Validation	Novelty
Range	20th Jan 1987 20th Mar 1996	21st Mar 1996 19th Mar 1997	20th Mar 1997 26th Mar 1998
Size (Days)	300	30	30

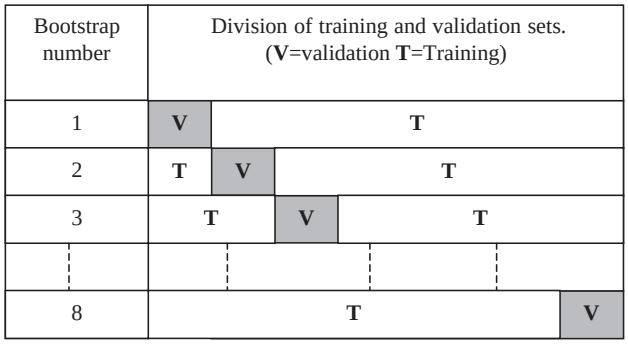

Fig. 3. Selection of training and validation sets for input determination.

Three sets of data are used to train and test the models (Table 1):

- *the training set* is used to calculate model parameters,
- *the validation set* is used to aid in model structure determination. In the case of neural network models, the validation set is used for early stopping and topology 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.

The overall training, validation and novelty sets, used for performance determination, topology and training cessation point in the parallel and sequential models, is shown in Table 1.

The techniques used for input selection (Section 4.2) utilise different training and validation sets where the dates vary due to the use of a bootstrapping technique [11], 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 (Fig. 3 and Table 2).

3. Determining hour of the day dependence

Consider the partitioned series $y_{1,\ldots,24}$ which represent the data partitioned by hour of the day. If electricity demand is hour of the day *independent* (which is the underlying

Table 2
Segmentation of data set for input selection

Set	Training	Validation
Range	Variable	Variable
Size (Days)	287	41

Table 3
Cross-correlation matrix of y_{13}, y_{14} and y_{15}

Hour	1 p.m.(13:00)	2 p.m.(14:00)	3 p.m.(15:00)
1 p.m.(13:00)	1	0.9958	0.9924
2 p.m.(14:00)	0.9958	1	0.9934
3 p.m.(15:00)	0.9924	0.9934	1

assumption of the sequential approach), then the cross-correlation between any two adjacent parallel series should be independent of which two hours are chosen. For example, the correlation between y_1 and y_2 should equal the correlation between y_4 and y_5. If $f_{1,\ldots,24}()$ is a linear function then this hypothesis can be tested using the linear cross-correlation coefficient $r_{i,j}$ between parallel series i and j. Even if $f_{1,\ldots,24}()$ are non-linear, the linearising assumption of using linear cross-correlation analysis is sufficient to either confirm or reject the hypothesis. The cross-correlation coefficient is defined as [2]:

$$r_{i,j} = \frac{\mathrm{E}[(\bar{y}_i - y_i)(\bar{y}_j - y_j)]}{\sqrt{\mathrm{E}[(\bar{y}_i - y_i)^2]}\sqrt{\mathrm{E}[(\bar{y}_j - y_j)^2]}} \tag{4}$$

where E[] denotes the expectation operator and $\bar{y}_i$ is the average of y_i. An example of the cross-correlation matrix between the 1 p.m., 2 p.m. and 3 p.m. series is shown in Table 3.

As can be seen the cross-correlations are very high (Table 3). This is not surprising; the load profiles for 3 typical days shown in Fig. 1 are very similar, showing how a large component of the data is highly correlated. 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 $r_{i,j} = 0.99$ is shown in Fig. 4. An example of the expected contour for the case where the load is hour of the day *independent*, is also shown for clarity (Fig. 4).

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 is not true. The narrowness of the contour at 9 a.m. and 6–8 p.m. show that at these hours especially, the load may have an independent component.

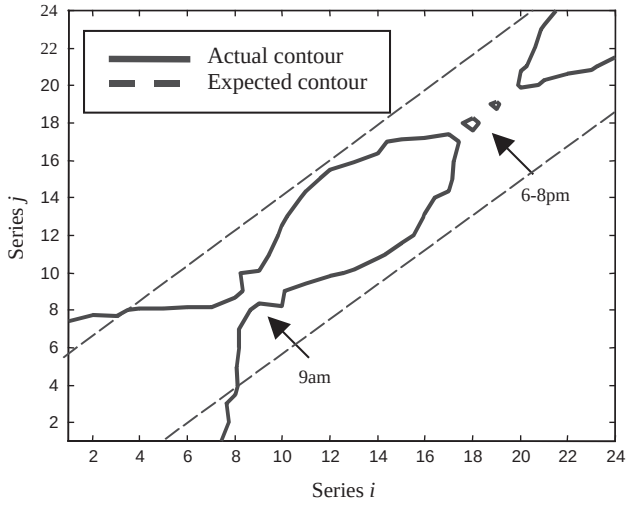

Fig. 4. Actual and expected contour plot of $r_{i,j} = 0.99$ for partitioned series i with j.

This suggests that it may be best to model the load at 9 a.m., 6 p.m., 7 p.m. and 8 p.m. separately rather than try to incorporate them into a sequential model. The next step is to model the data using both the parallel (Section 4) and sequential (Section 5) approaches and examine the results.

4. Parallel models

4.1. Preliminary auto-regressive linear model

As explained earlier, the non-linearities in electrical load data are to be modelled using a feed forward neural network, which requires that the data be stationary. This can be achieved in two ways:

- stationarity transformations can be used, such as differencing [2] and
- a preliminary model can be used to remove the non-stationary elements of the data, leaving a stationary residual. This is known as a *preliminary linear auto-regressive* (AR) model.

The latter approach is taken in this study as stationarity transforms can introduce noise to the training data [13].

The partitioned series for hour i on day k, $y_i(k)$ has a low frequency trend $d_i(k)$ due to year on year changes in usage of electricity and a seasonal component $s_i(k)$ due to more electricity being used in winter for heating than in summer. The preliminary linear AR model is composed of a basic structural model (BSM) [13] which removes

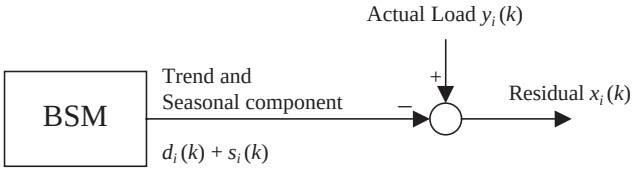

Fig. 5. Preliminary AR linear model overview.

$d_i(k)$ and $s_i(k)$ from $y_i(k)$ leaving a residual $x_i(k)$ (Fig. 5). The residual is composed of weather, non-linear AR and white noise components.

The BSM is a state space model which represents a time series as a sum of a trend, $d_i(k)$, a seasonal, $s_i(k)$, and a residual, $x_i(k)$, [13] as

$$y_i(k) = d_i(k) + s_i(k) + x_i(k). \tag{5}$$

If the trend component is modelled using an integrated random walk [13] and the seasonal component is modelled using a differenced periodic random walk [13], then the complete state-space model is defined by the following [13]:

$$\begin{bmatrix} d_i(k) \\ \dot{d}_i(k) \\ - - - - - - - \\ s_i(k) \\ s_i(k-1) \\ . \\ . \\ s_i(k-(T-2)) \end{bmatrix} = \theta_i(k) = \left[\begin{array}{cc|ccccc} 1 & 1 & 0 & 0 & . & . & 0 \\ 0 & 1 & 0 & 0 & . & . & 0 \\ - & - & - & - & - & - & - \\ 0 & 0 & -1 & -1 & . & . & -1 \\ 0 & 0 & 1 & 0 & . & . & 0 \\ . & . & . & . & . & . & 0 \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & . & 1 & 0 \end{array}\right]$$

$$\times \begin{bmatrix} d_i(k-1) \\ \dot{d}_i(k-1) \\ - - - - - - \\ s_i(k-1) \\ s_i(k-2) \\ . \\ . \\ s_i(k-(T-1)) \end{bmatrix} + \begin{bmatrix} 0 \\ \eta_d(k-1) \\ - - - \\ \eta_s(k-1) \\ 0 \\ . \\ . \\ 0 \end{bmatrix}, \tag{6}$$

where $\dot{d}_i(k)$ is the rate of change of the trend, $\theta_i(k)$ is the state vector, $\eta_d(k)$ and $\eta_s(k)$ are error terms. T is the seasonal length in this case 30 as there are 30 days per

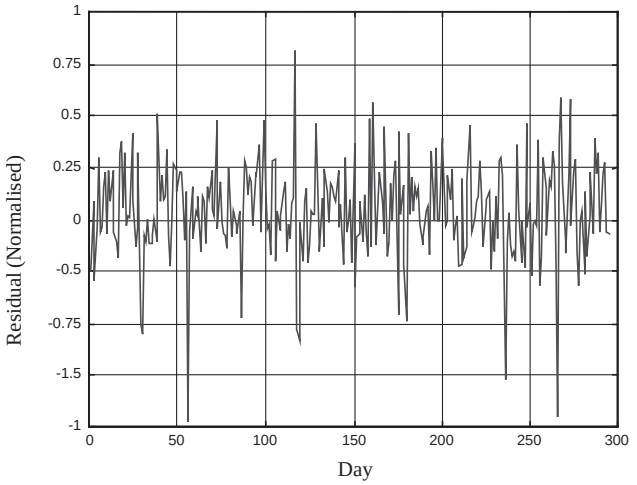

Fig. 6. A plot of the residuals for 1 a.m., i.e. $x_1(k)$.

year in data set (see Section 2). The load may be extracted from the state vector by using the *observation matrix* [13] defined as

$$y_i(k) = [1 \quad 0 \quad 1 \quad 0 \quad \cdots \quad 0 \quad 0 \quad 0]\theta_i(k) + x_i(k) \tag{7}$$

In order to perform a prediction, a Kalman filter [13] is used over the identification data set to provide initial state estimates for the model. Covariances for the (process) noise sources $\eta_s(k)$ and $\eta_d(k)$ the measurement noise, $\varepsilon(k)$, are determined using maximum likelihood optimisation [13].

The residuals are then tested to ensure that they are stationary using the sample auto correlation function (SACF). The SACF of a time series $v(k)$ represents the linear correlation between observations separated by a lag, and may be expressed [2] as

$$\hat{r}_v(m) = \frac{\sum_{t=1}^{n-m}(v(t) - \bar{v})(v(t+m) - \bar{v})}{\sum_{t=1}^{n}(v(t) - \bar{v})^2}, \tag{8}$$

where $\hat{r}_v(m)$ is the SACF value for a lag of m, n is the number of observations used and $\bar{v}$ is the average value of $v(k)$. Note that the SACF is an estimate of the auto-correlation function, as sample data is used. Box and Jenkins [2] suggest that a process may be considered non-stationary if the SACF *dies away slowly*, which is determined subjectively with experience. A plot of the residuals for the 1 a.m. series is shown in Fig. 6 with the associated SACF in Fig. 7.

As can be seen the SACF dies away quickly and is close to zero for all lags greater than zero. Thus this residual is deemed to be stationary (using the approach suggested by Box and Jenkins [2]). The SACF for the other partitioned series are similar and this confirms the stationarity of the residuals.

There is a high degree of correlation between weather and load. As the residual has the trend and seasonal components of the load extracted, this distorts the relationship between the weather and the residual. The seasonal component of the weather is related

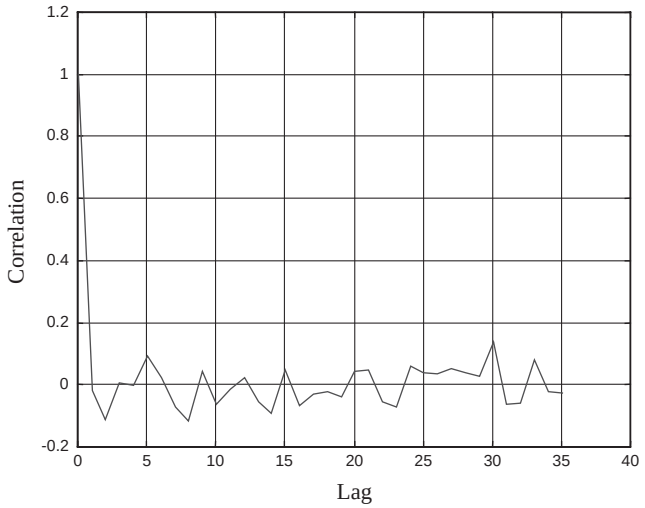

Fig. 7. SACF of $x_1(k)$.

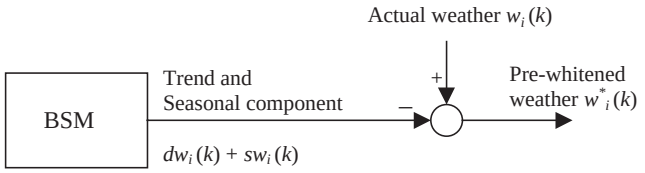

Fig. 8. Pre-whitening a weather variable.

to $s_i(k)$ but not to $x_i(k)$ and so cannot be used to forecast $x_i(k)$. Thus the seasonal component in the weather variables must be removed. As pointed out by Harvey [13] in the general case, this may be achieved by filtering the causal variable with the same model used to produce the residual. That is, the trend, $dw_i(k)$, and seasonal components, $sw_i(k)$, of the weather variable, $w_i(k)$, are removed using the BSM to give a weather residual $w_i^*(k)$ (Fig. 8). The weather residual $w_i^*(k)$ is also called the *pre-whitened weather variable*.

4.2. Input selection

Input selection forms perhaps the most important step in model building [21]. Inclusion of non-causal variables leads to poor model generalisation. In addition, reducing the dimensionality of the inputs aids training of neural networks [21].

The input variables available to choose from are:

- $\boldsymbol{t}_i(k)$, a vector of pre-whitened temperature from hour i to hour i-23 on day k,
- $\boldsymbol{ws}_i(k)$, a vector of pre-whitened wind speed from hour i to hour i-23 on day k and
- $\boldsymbol{h}_i(k)$, a vector of pre-whitened humidity from hour i to hour i-23 on day k.

Thus there are 72 variables in the set of *possible* inputs, $[\boldsymbol{t}_i(k)\boldsymbol{ws}_i(k)\boldsymbol{h}_i(k)]$. Ideally, neural networks with all possible combinations of the inputs should be constructed and the best selected. However, this is not possible for two reasons:

1. the computational expense of a neural network prohibits more than a few combinations being tested and
2. the number of possible combinations of the inputs ($72! = 6 \times 10^{103}$) is too large to implement with any model.

To increase the number of combinations that can be tested, a linear regression model (RM) [13], 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 has the form:

$$x_i(k) = a_{i,1}u_{i,1}(k) + a_{i,2}u_{i,2}(k) + \cdots + a_{i,n}u_{i,n}(k) + \varepsilon_i(k) \tag{9}$$

where $x_i(k)$ is the residual to be forecast, $u_{i,j}$ is the jth input for model i (i.e. for hour i), $a_{i,j}$ is the coefficient applied to that input (calculated by least squares) and n is the number of inputs used.

Note that the input selection procedure must be carried out for each parallel model. A subscript is thus used to indicate the hour index of the particular model (e.g. RM_i refers to a regression model for hour i).

Four methods are now evaluated for input selection.

Method 1:

Method 1 performs input selection using the following algorithm:

For all inputs:
  Train a linear regression model.
  Calculate the *T-ratio* [16] of all the coefficients. This is the ratio of the variance of $a_{i,j}$ to the amplitude of $a_{i,j}$. A high T-ratio for $a_{i,j}$ implies that $u_{i,j}$ is of little use in forecasting x_i.
  Order the inputs with increasing value of T-ratio.
For number of inputs NINP = 1 to 72:
  Select the first NINP inputs.
  Train a linear regression model for these inputs, for each training bootstrap (see Table 2 and Fig. 3).
  Calculate each of the Minimum Absolute Errors (MAE's)[1], on each of the bootstrap validation sets.
  Next NINP

In summary, the technique uses the T-ratio to order the inputs so that 72 combinations of the inputs can be evaluated (Fig. 9).

[1] 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* (explained in Step 9).

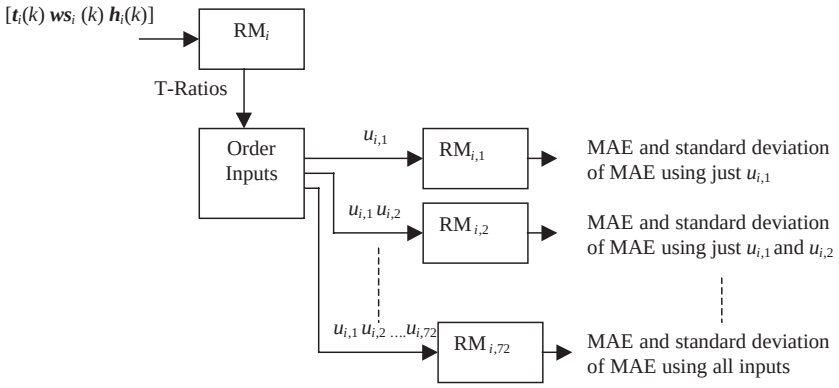

Fig. 9. A block diagram of method 1 for input selection for hour i model.

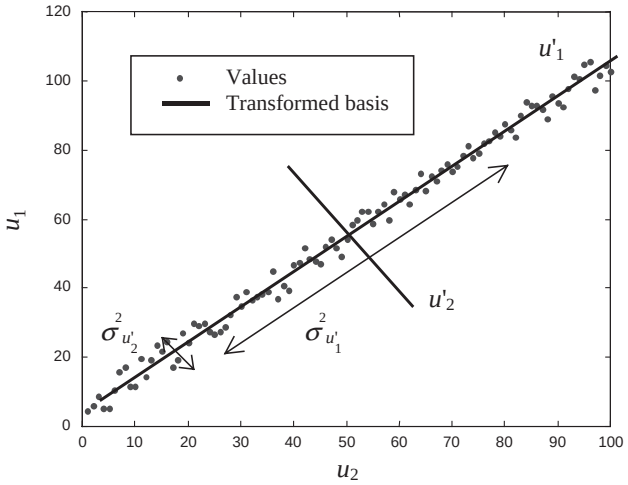

Fig. 10. An example of two variables transformed using PCA.

Method 2:

One disadvantage of Method 1 is that it is susceptible to *collinearity* in the inputs (a high degree of correlation between any of the inputs) [26]. If two inputs are highly correlated then the coefficients attached to those inputs from step 3 will have smaller values than if one was excluded. This, in turn, increases their T-ratio values and can mistakenly push these inputs down the priority list. One means of reducing the collinearity in the input data is to use principal component analysis (PCA) [26].

PCA is a technique used for input dimension reduction. Consider, for example, the case where just two highly correlated input variables are available $u_1(t)$ and $u_2(t)$ (Fig. 10). PCA transforms these variables into a set of orthogonal variables $u'_1(t)$ and $u'_2(t)$

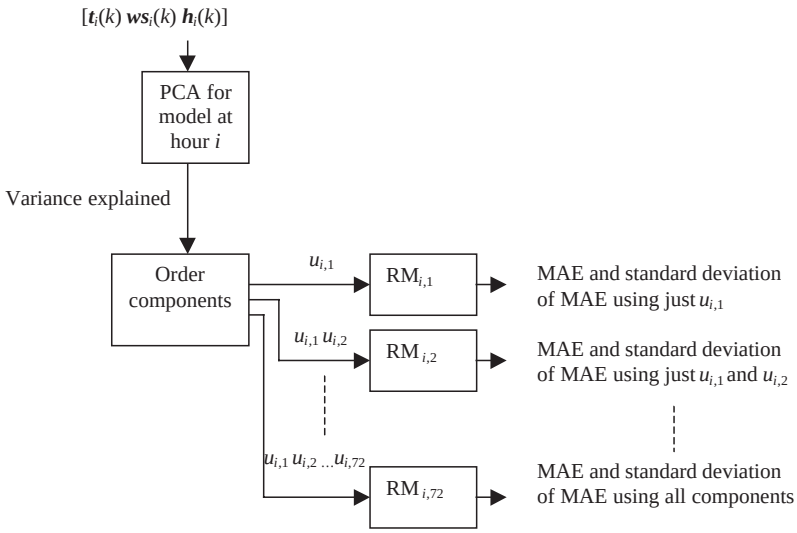

Fig. 11. A block diagram of method 2 for input selection for hour i model.

such that each variable represents the coefficient along a basis vector in characteristic directions of the original data set [26] (Fig. 10). Thus, the transformed variables are not collinear.

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 Fig. 10), with the first component containing the highest amount of information [26] (Fig. 10). As can be seen from Fig. 10, $u_2'(t)$ accounts for very little information and could be discarded.

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 72
    Select first NCOMP components.
    Train a linear regression model for these components, for each training bootstrap (see Table 2 and Fig. 3).
    Calculate each of the Minimum Absolute Errors (MAE's), on each of the bootstrap validation sets.
    Next NCOMP

Fig. 11 gives an overview of Method 2.

Method 3:

One difficulty with 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

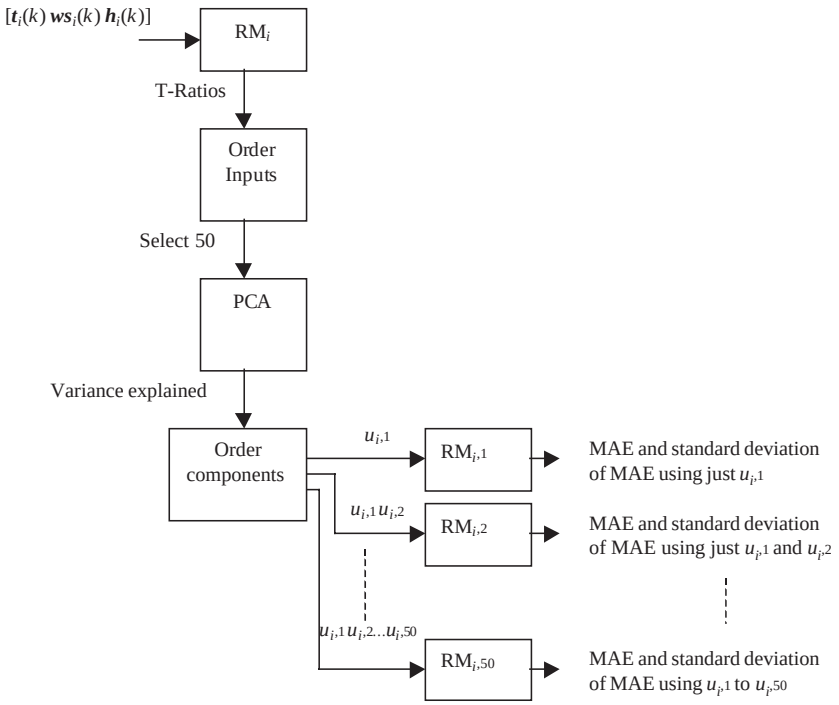

Fig. 12. A block diagram of method 3 for input selection for hour i model.

to the output data. For example, the first component may have the highest level of variance explained with respect to the input data while 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* [16] 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 2 and Fig. 3).
  Calculate each of the Minimum Absolute Errors (MAE's), on each of the bootstrap validation sets.
  Next NCOMP

Fig. 12 gives an overview of Method 3.

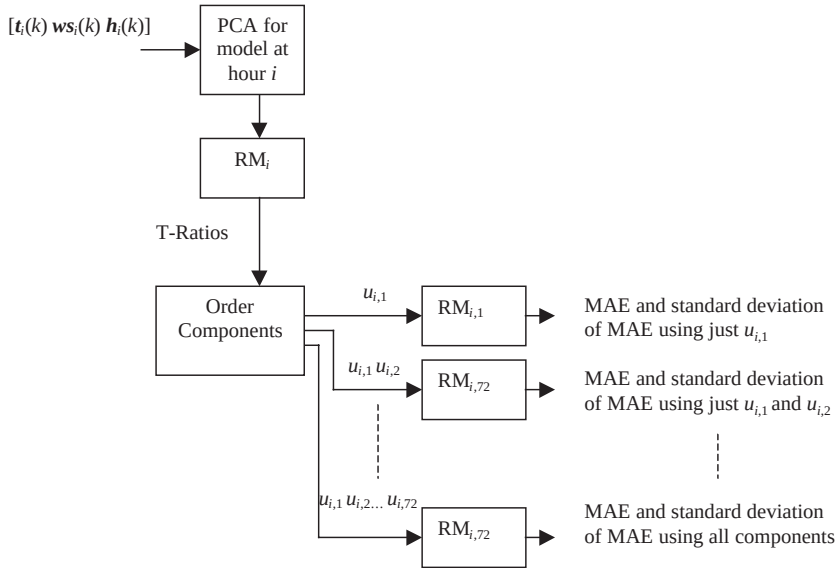

Fig. 13. A block diagram of method 4 for input selection for hour i model.

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* [16] of all the coefficients.
  Order the transformed components with increasing value of T-ratio.
For number of inputs NCOMP = 1 to 72:
  Select first NCOMP components.
  Train a linear regression model for these components, for each training bootstrap (see Table 2 and Fig. 3).
  Calculate each of the Minimum Absolute Errors (MAE's), on each of the bootstrap validation sets.
  Next NCOMP

Fig. 13 gives an overview of Method 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. An example is shown in Fig. 14. The optimum MAE's, and the

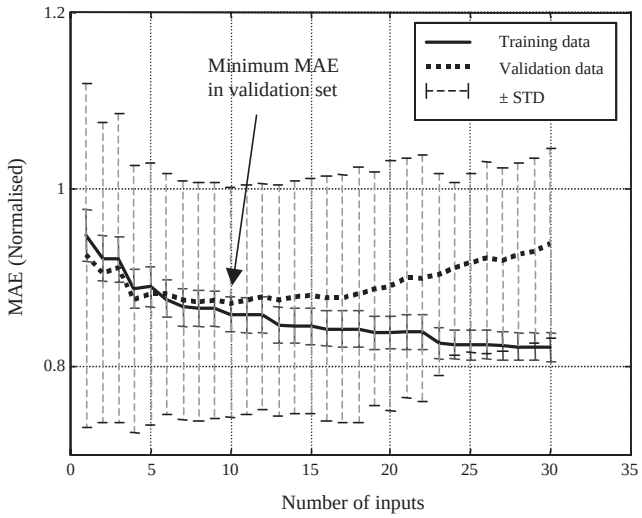

Fig. 14. Bootstrapped MAE for method 2 (6 p.m. series).

Table 4
Input selection MAE (normalised)

	Method 1	Method 2	Method 3	Method 4
Mean MAE (validation sets).	0.90	0.86	0.90	1.00
Standard deviation of MAE (validation sets)	0.13	0.13	0.13	0.15
Mean MAE (training sets).	0.83	0.83	0.87	0.96
Standard deviation of MAE (training sets)	0.03	0.02	0.03	0.06

associated standard deviations, for each hour of the day, are then averaged to give a summary of the results (Table 4). As can be seen, Method 2 achieves the lowest MAE in both the validation and training sets. Additionally, Method 2 provides the lowest standard deviations of the MAE's in both the training and validation sets (indicating more confidence in the results) and is thus selected as the optimum input selection method.

The optimum number of components used in Method 2 is found to vary from 6 to 15 depending on the hour of the day. The *mode* (an integer value mid-way between the maximum and minimum) is 10 and due to the computational expense of calculating the topology of neural networks, 10 components are used for all hours of the day.

Additionally, it was found in [10] that there is a non-linear relationship between $x_i(k)$ and $x_i(k-1)$, $x_i(k-2)$. These inputs are therefore included in the neural network. Fig. 15 shows an overview of the parallel models which produce an estimate of $x_i(k)$ as $x_i^{nn}(k)$.

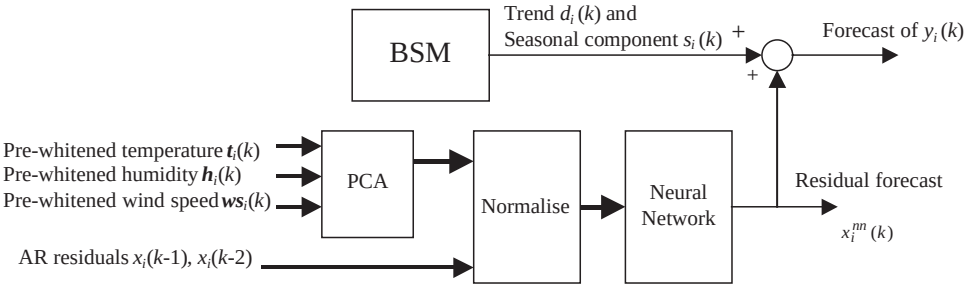

Fig. 15. Overview of a parallel model.

4.3. Training the neural network models

A multi-layer perceptron network using the back-propagation learning algorithm [4] is used. It was found that use of a single hidden layer resulted in an excessive number of neurons and that networks with 2 hidden layers are superior. Similar results have been observed in other studies as detailed in [14].

Each network consists of 2 hidden layers with *tan sigmoid* activation functions and a linear activation function in the output layer. The input data is normalised between ± 1 so that the *tan sigmoid* activation functions are not driven into saturation [4], with a resulting speed up in training, since the high gain (gradient) of the neuron characteristic is used.

Each model is trained using early stopping [14] in which training ceases when the sum squared error of predictions in the validation set reaches a minimum (an example is shown in Fig. 16). 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 (NN) [14] and assists in conjunction with topology determination and input selection, in achieving a parsimonious network.

The topology of a neural network determines the degrees of freedom available to model the data [14]. If the neural network is too simple then the network will not be able to learn the function relating the input to the output [14]. An over-complex network will learn the noise in the data and will not be able to generalise [14].

In order to determine the correct topology, 50 NN architectures were examined, using 1–5 and 1–10 nodes in the first and second hidden layers, respectively. 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. To perform this for each partitioned series would require training 1200 (50×24) neural networks, which is too expensive computationally, so the network topology is refined for the 6 p.m. series (as a representative model) and applied to the others.

Tables 5 and 6 shows the average training and validation mean absolute percentage error (MAPE) for each network topology, respectively. The MAPE is the preferred error measure in the field of short-term load forecasting as it allows electric utilities of different sizes to compare forecasting accuracy, which a measure such as the mean

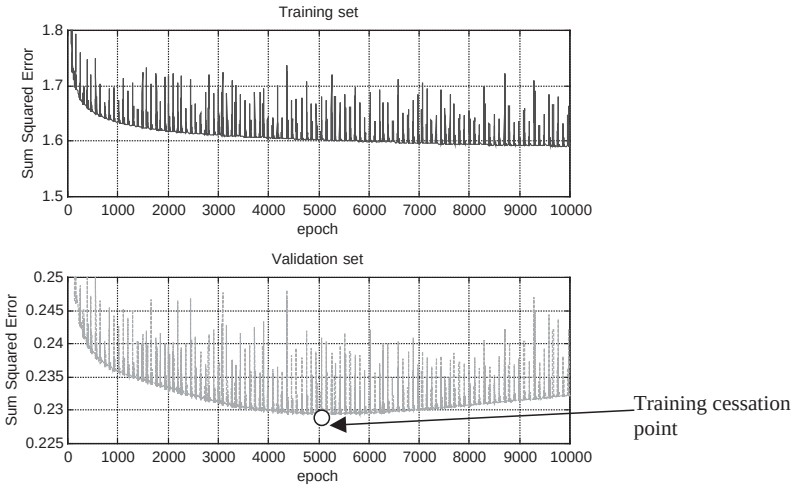

Fig. 16. An example of early stopping (Architecture: 4×4).

Table 5
Average *training* MAPE for differing NN topologies (6 p.m. series)

#Nodes	Layer 2 : 1	2	3	4	5	6	7	8	9	10
Layer 1 : 1	1.85	1.83	2.00	2.13	2.07	2.09	2.19	2.14	2.17	2.23
2	1.92	1.87	1.90	1.86	1.87	1.86	1.89	1.91	2.04	1.91
3	1.80	1.78	1.88	1.77	1.77	1.75	1.77	1.84	1.82	1.90
4	1.79	1.87	1.86	**1.75**	1.74	1.77	1.92	1.77	**1.73**	1.73
5	1.79	1.75	1.72	1.71	1.70	1.73	**1.70**	1.73	1.79	1.69

Table 6
Average *validation* MAPE for differing NN topologies (6 p.m. series)

#Nodes	Layer 2 : 1	2	3	4	5	6	7	8	9	10
Layer 1 : 1	1.42	1.43	1.60	1.68	1.65	1.62	1.69	1.73	1.81	1.83
2	1.90	1.92	1.53	1.55	1.56	1.58	1.52	1.54	1.77	1.64
3	1.44	1.47	1.53	1.48	1.50	1.50	1.51	1.60	1.59	1.68
4	1.42	1.49	1.48	**1.45**	1.50	1.48	1.63	1.48	**1.43**	1.51
5	1.47	1.46	1.48	1.52	1.53	1.47	**1.44**	1.47	1.62	1.53

squared error (MSE) or MAE would not allow. In order to calculate the MAPE on the overall load forecast, $y_i^{nn}(k)$, the trend and seasonal components, which are forecast by the preliminary AR model (Section 4.1), must be re-introduced (Fig. 15):

$$y_i^{nn}(k) = d_i(k) + s_i(k) + x_i^{nn}(k), \tag{10}$$

where $y_i^{nn}(k)$ is forecast of the load on hour i of day k produced by the parallel model for hour i.

Neural network topology selection is based on two competing criteria:

- failure to stop early during training is used as an indication of too simple an architecture and
- the objective is to favour networks with less complexity.

Examination of Table 6 shows that an architecture of 1×1 (1 nodes in hidden layer 1 and 1 node in hidden layer 2) appeals as it has the best performance (1.42%) over the validation set. However the performance of this architecture in the training set (1.85%) is relatively poor (Table 5). In addition, 1×1 networks fail to stop early during training. This architecture is therefore deemed to be too simple.

Failure to stop early during training also occurs with architectures of 1 or 2 nodes in either the first or second hidden layers. Thus these architectures are also eliminated from the selection. Of the remaining networks, topologies 4×4 (4 nodes in hidden layer 1 and 4 nodes in hidden layer 2), 5×7 and 4×9 achieved the lowest MAPE's (Tables 5 and 6). Topology 4×4 is chosen using the second criteria. Finally, Fig. 16 demonstrates the training cessation point for one of the networks trained with this architecture.

5. Sequential model

The sequential model (SM) ignores the hour of the day dependence which, as pointed out in Section 1.1, can decrease the model performance. However, though points are serially correlated in this model, some extra information with regard to the behaviour at individual hours can be included using the multi-time scale (MTS) technique of Murray et al. [24]. In this formulation, a linear auto-regressive (sequential) model is encouraged to achieve intermediate targets through adjustment of the initial conditions of the model.

With the MTS formulation, the SM is restricted to a linear state space model and a BSM is used, where the load t hours from the start of the sequential series is modelled using a trend and a seasonal component as

$$y_m(t) = d_m(t) + s_m(t) + x_m(t), \tag{11}$$

where $d_m(t)$ is the trend component, $s_m(t)$ is the seasonal component and $x_m(t)$ is the SM residual t hours form the start of the data.

Note that the variables in this model are indexed with the number of hours from the start of the data, t, as opposed to indexing by hour i on day k as in the case of the parallel models. This is because the model is sequential and cannot easily be indexed in the same manner as the parallel models.

As with the preliminary (BSM) model in Section 4.1, the trend component is modelled using an integrated random walk [13], with the seasonal component modelled

using a differenced periodic random walk [13] as

$$\theta_m(t) = \begin{bmatrix} d_m(t) \\ \dot{d}_m(t) \\ - - - - - - - \\ s_m(t) \\ s_m(t-1) \\ . \\ . \\ s_m(t-(24-2)) \end{bmatrix} = \left[\begin{array}{cc|ccccc} 1 & 1 & 0 & 0 & . & . & 0 \\ 0 & 1 & 0 & 0 & . & . & 0 \\ - & - & - & - & - & - & - \\ 0 & 0 & -1 & -1 & . & . & -1 \\ 0 & 0 & 1 & 0 & . & . & 0 \\ . & . & . & . & . & . & 0 \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & . & 1 & 0 \end{array}\right] \times \begin{bmatrix} d_m(t-1) \\ \dot{d}_m(t-1) \\ - - - - - - \\ s_m(t-1) \\ s_m(t-2) \\ . \\ . \\ s_m(t-(24-1)) \end{bmatrix} + \begin{bmatrix} 0 \\ \eta_{dm}(t-1) \\ - - - \\ \eta_{sm}(t-1) \\ 0 \\ . \\ . \\ 0 \end{bmatrix}, \tag{12}$$

where $\theta_m(t)$ is the state vector, $\dot{d}_m(t)$ is the rate of change of the trend and $\eta_{dm}(t)$ and $\eta_{sm}(t)$ are errors in the estimates of the model states. The seasonal length in this case is 24, since there are 24 h in each day. Eq. (12) may be expressed in matrix form [13] as

$$\theta_m(t) = \Phi_m \theta_m(t-1) + \eta_m(t-1), \tag{13}$$

where Φ_m is the state transition matrix and $\eta_m(t)$ a vector of error terms. The load may be extracted from the state vector by using the *observation matrix*, H_m, defined [13] as

$$y(t) = H_m \theta_m(t) + x_m(t) = [1 \quad 0 \quad 1 \quad 0 \quad \cdots \quad 0 \quad 0 \quad 0]\theta_m(t) + \varepsilon(t), \tag{14}$$

where $\varepsilon(t)$ is the modelling error.

For the MTS formulation [24], the intermediate targets are forecasts of:

1. The load at the overnight minimum at 5 a.m.,
2. The load the lunchtime peak at 1 p.m.,
3. The load at 2 p.m.,
4. The load at 6 p.m.,
5. The load at 12 p.m. for days $k, k+1, k+2$ (also called the *end* points), and

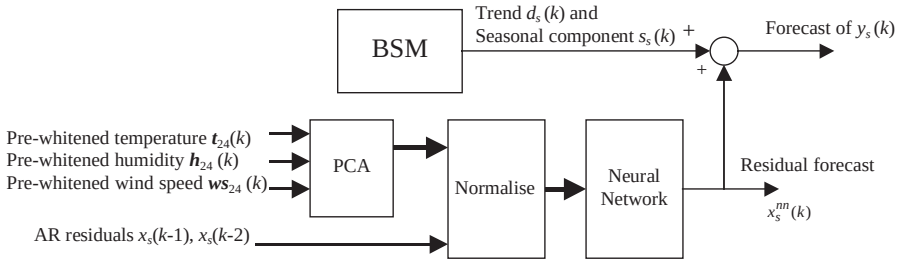

Fig. 17. Overview of the end-sum model.

6. The integrated daily consumption, in MWh, for days k, $k+1, k+2$ (also called *daily sums*).

Targets 1 to 5 represent forecasts for individual hours, while the targets in 6 essentially represent the sum of the previous 24 points as discussed in Section 5.2.

5.1. Individual hour targets

The forecasts for these target points (known as *cardinal points*) are provided by the parallel models, as discussed in Section 4. A selection of the parallel models already examined are used to provide forecasts for the 5 a.m, 1 p.m., 2 p.m., 6 p.m. and 12 p.m. points, while further parallel models are used to determine forecasts for the 12 p.m. value 1 and 2 days in advance. As described in Section 4, all the cardinal point models are BSM+neural network models, which are driven by a variety of weather inputs. Since the overall sequential model is purely autoregressive, this provides an important mechanism for including the influence of weather variables in the sequential model, albeit via target adjustment.

5.2. The daily sum model

The integrated daily load for day k, $y_s(k)$, is defined as

$$y_s(k) = \sum_{i=1}^{24} y_i(k). \tag{15}$$

This series is modelled using a similar structure to the parallel models of Section 4 (Fig. 17). In accordance with this philosophy, $y_s(k)$ is first modelled using a preliminary AR model similar to $(5,6,7)$ with:

$$y_s(k) = d_s(k) + s_s(k) + x_s(k), \tag{16}$$

where $d_s(k)$ is the trend component for day k, $s_s(k)$ is the seasonal component for day k and $x_s(k)$ is the daily sum residual for day k. The daily sum residual is then modelled using a neural network with the same structure as that described in

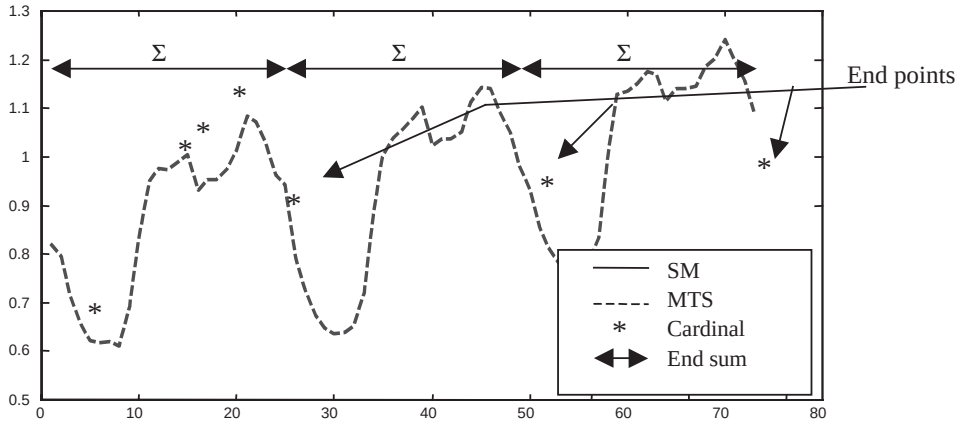

Fig. 18. Diagram showing MTS technique.

Section 4. Specifically, the daily sum neural network has:

- $\boldsymbol{t}_{24}(k), \boldsymbol{ws}_{24}(k)$ and $\boldsymbol{h}_{24}(k)$ as causal variables,
- inputs selected using Method 2 with 10 components retained, as in the parallel models, and
- a topology with 4 nodes in the first and second hidden layers.

The neural network then produces a forecast of the daily sum residual, $x_s^{nn}(k)$, which is used to produce a forecast of $y_s(k)$ by re-introducing the trend and seasonal component as

$$y_s^{nn}(k) = d_s(k) + s_s(k) + x_s^{nn}(k), \tag{17}$$

where $y_s^{nn}(k)$ is the end sum forecast of $y_s(k)$.

5.3. The multi-time scale technique

An example of how the MTS technique adjusts a sequential model forecast of the load 3 days ahead using the cardinal points and daily sum targets, is shown in Fig. 18. The initial conditions of the SM, which are adjusted by the MTS technique, are the states of the state vector $\theta_m(t)$.

The MTS technique partitions the state vector into states which are *fixed* at the forecasting origin and states which are *freed* (i.e. free to be adjusted in order to achieve point and sum targets). Eq. (13) may be expressed in terms of the fixed and freed states [24] as

$$\theta_m(t+1) = \begin{bmatrix} \theta_{m1}(t+1) \\ \theta_{m2}(t+1) \end{bmatrix} = \Phi_m \theta_m(t) = [\Phi_{m1} \quad \Phi_{m2}] \begin{bmatrix} \theta_{m1}(t) \\ \theta_{m2}(t) \end{bmatrix}, \tag{18}$$

where $\theta_{m1}(t)$ and $\theta_{m2}(t)$ are vectors of fixed and *freed* states at the forecasting origin, t, respectively and Φ_{m1} and Φ_{m2} are the partitions of the state transition matrix associated

with $\theta_{m1}(t)$ and $\theta_{m2}(t)$, respectively. The SM may be used to generate forecasts p steps ahead by repeated use of Eq. (13) as

$$\theta_m(t+p) = \begin{bmatrix} \theta_{m1}(t+p) \\ \theta_{m2}(t+p) \end{bmatrix} = (\Phi_m)^p \theta_m(t) = [\Phi_{m1}(p) \quad \Phi_{m2}(p)] \begin{bmatrix} \theta_{m1}(t) \\ \theta_{m2}(t) \end{bmatrix}, \quad (19)$$

where $\Phi_{m1}(p)$ and $\Phi_{m2}(p)$ are the partitions of $(\Phi_m)^p$ associated with $\theta_{m1}(t)$ and $\theta_{m2}(t)$. However, this solution does not take into account the desired targets, so the $\theta_{m2}(t)$ are now adjusted in order to attempt to achieve this. The MTS technique calculates the adjusted states $\theta^*_{m2}(t)$ by formulating the problem as a set of over-determined equations in $\theta^*_{m2}(t)$, made up of three types of soft constraints [24], namely:

1. *a smoothing constraint* in which the MTS forecast deviation from the SM forecast by $e_1, \ldots, e_{72}$ is to be minimised from $t+1$ to $t+72$,
2. *a cardinal point constraint* in which deviation of the 7 cardinal point forecasts, $\hat{y}_{cp1}, \ldots, \hat{y}_{cp7}$, of the load from the SM forecasts by $e_{cp1}, \ldots, e_{cp7}$ is to be minimised at times $t+t_1, \ldots, t+t_7$ [2] respectively, and
3. *a daily sum constraint* in which the deviation of the forecasts of the daily sum for days 1 to 3, $\hat{y}_{s1}, \hat{y}_{s2}$ and $\hat{y}_{s3}$ from the sum of the SM forecasts over those days by e_{s1}, e_{s2} and e_{s3} is to be minimised.

The complete set of constraints, represented as a set of over-determined equations is:

$$\begin{matrix} \text{Constraint 1} \\ \\ \\ \\ \\ \text{Constraint 2} \\ \\ \\ \\ \text{Constraint 3} \\ \\ \\ \\ \end{matrix} \left\{ \begin{matrix} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix} \right. \begin{bmatrix} H_m \Phi_m \theta_m(t) - H_m \Phi_{m1}(1) \theta_{m1}(t) \\ \vdots \\ H_m (\Phi_m)^{72} \theta_m(t) - H_m \Phi_{m1}(72) \theta_{m1}(t) \\ \hat{y}_{cp1} - H_m \Phi_{m1}(t_1) \theta_{m1}(t) \\ \vdots \\ \hat{y}_{cp7} - H_m \Phi_{m1}(t_7) \theta_{m1}(t) \\ \hat{y}_{s1} - \sum_{j=1}^{24} H_m \Phi_{m1}(j) \theta_{m1}(t) \\ \vdots \\ \hat{y}_{s3} - \sum_{j=49}^{72} H_m \Phi_{m1}(j) \theta_{m1}(t) \end{bmatrix}$$

[2] For example the first cardinal point is at 5 a.m. and thus $y(t+t_1)$ is equal to $y_5(k)$.

$$= \begin{bmatrix} H_m\Phi_{m2}(1) \\ \vdots \\ H_m\Phi_{m2}(72) \\ H_m\Phi_{m2}(t_1) \\ \vdots \\ H_m\Phi_{m2}(t_7) \\ \sum_{j=1}^{24} H_m\Phi_{m2}(j) \\ \vdots \\ \sum_{j=49}^{72} H_m\Phi_{m2}(j) \end{bmatrix} \theta^*_{m2}(t) + \begin{bmatrix} e_1 \\ \vdots \\ e_{72} \\ e_{cp1} \\ \vdots \\ e_{cp7} \\ e_{s1} \\ \vdots \\ e_{s3} \end{bmatrix} W, \tag{20}$$

where W is a diagonal weight matrix which allows Eq. (20) to be solved using weighted least squares [24]. This is advantageous, since it allows the technique to assign more significance to the achievement of particular targets. For a full derivation of Eq. (20), see [24]. The values in the weight matrix, W, are determined by minimising the MAPE in the training set with respect to the weights [10]. The optimisation routine used is the Nelder Mead Simplex Algorithm [3].

Once the freed states have been adjusted, a p-step ahead forecast is generated [24] via:

$$\theta^*_m(t+p) = \begin{bmatrix} \theta_{m1}(t+p) \\ \theta^*_{m2}(t+p) \end{bmatrix} = (\Phi_m)^p \theta^*_m(t) = [\Phi_{m1}(p) \quad \Phi_{m2}(p)] \begin{bmatrix} \theta_{m1}(t) \\ \theta^*_{m2}(t) \end{bmatrix} \tag{21}$$

6. Results

The performance of the sequential (*using the MTS technique*) and parallel models are evaluated using the novelty set data, which has up to now been excluded from the analysis. The MAPE's achieved by the sequential model and the parallel models in the novelty set are shown in Fig. 19 for each hour of the day. The preliminary Linear Auto-Regressive models (Section 4.1), used by each parallel model, forecasts the load using only the trend and seasonal components. The performance of these models on the novelty set is included as a baseline to determine the improvement in performance when the neural networks are used to forecast the preliminary Linear Auto-Regressive model residuals. The most significant differences between the performance of the sequential model and parallel models are at midnight (0 a.m.), 9 a.m. and 6–8 p.m. 9 a.m. and 6–8 p.m. are also the times at which load was shown to possibly have independent components in Section 3. In spite of the fact that the forecasts from the parallel model

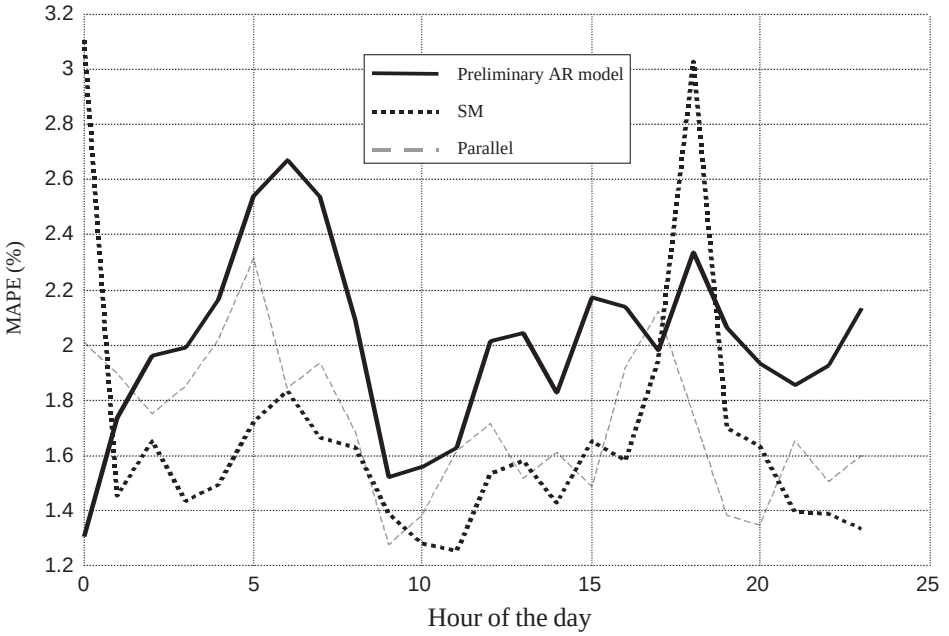

Fig. 19. MAPE as a function of the hour of day in the novelty set.

Table 7
Summary of results

Model	Parallel	Sequential	Preliminary linear AR	Composite
Training set MAPE	1.64%	1.90%	2.31%	1.73
Novelty set MAPE	1.71%	1.67%	2.00%	1.57

for 6 p.m. are used as a cardinal point in the MTS technique, the performance by the sequential model at 6 p.m. (Fig. 19) is nevertheless poor. The large MAPE for the sequential forecasts for midnight is unexplained.

Given that the sequential model is inferior to the parallel model for forecasting the load at 9 a.m., 6–8 p.m. and midnight, the forecasts at these times need not be used. Parallel models exclusively can be used to forecast the load at these times. A composite model using forecasts from the sequential model except at midnight, 9 a.m. and 6–8 p.m., where parallel forecasts are used, gives a better result than either sequential or parallel individually (Table 7).

To summarise the results, an average of the 24 MAPE's, one for each hour of the day (shown in Fig. 19), are used to generate a *global* (i.e. a single figure for comparison) MAPE. The global MAPE's achieved in both the training and novelty data sets are shown in Table 7. Although the parallel models are superior in the training set, their performance is inferior to the sequential model performance in the novelty set. This shows that the generalisation properties of the parallel models are not as good as those

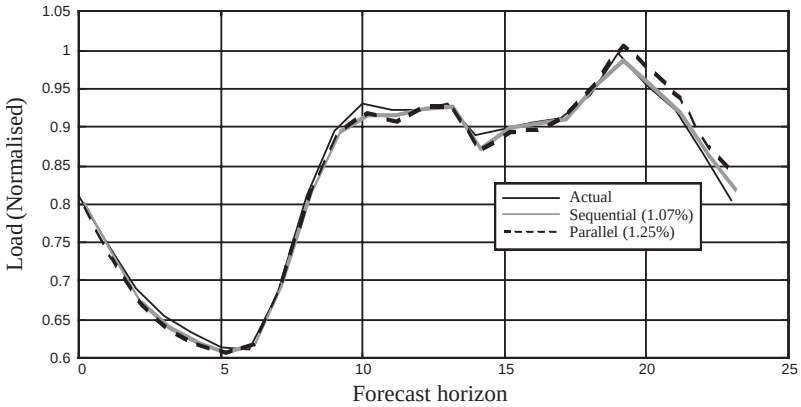

Fig. 20. Sample 1-day ahead forecast (MAPE's shown in legend).

of the sequential model. Possible reasons for this include that fact that the topologies of the parallel models are sub-optimal (Section 4.3) and the sequential model takes serial correlations into account, which may give it better inherent robustness.

The composite model shows a consistent result in both data sets and a lower MAPE than both the sequential or parallel models in the novelty set. A sample forecast is shown in Fig. 20.

As can be seen, both the sequential model and parallel models produce a good forecast.

7. Conclusion

This paper has examined whether a sequential or parallel approach to load forecasting is appropriate in the Irish case. In testing the data for hour of the day dependence (Section 3), it was seen that the load at 4 particular hours in the day appeared to have independent components and may be best modelled separately. The sequential approach was subsequently shown to be inferior to the parallel approach for forecasting these hours. By incorporating several parallel neural network models into the sequential model via the multi-time scale technique the validity of the comparison was improved.

The largest problem with the parallel models used was shown to be the prohibitive computational expense of calculating optimal topologies for all 24 parallel models. However, as the number of parallel models used in the composite model is far less than 24 (5 cardinal point models, 1 sum model and 3 further parallel models for the 'independent' hours) the computational expense is significantly reduced. The composite model also gives the best forecasting results, giving an acceptable compromise between accuracy and computational complexity.

Several methods for input selection were examined in Section 4.2. It was found that removing the collinearity from the input data using PCA and ordering the components by variance explained in the input (Method 2) gave the best result. However, even when the inputs have been transformed using PCA, selecting which components to retain was not a straightforward task. Surprisingly, selection of the components using

the T-ratio as in Method 4 was found to give inferior results to all the other methods. Interestingly Method 4 was inferior to Method 1, which did not use PCA at all. Thus, PCA alone cannot be attributed to the success of Method 2.

References

[1] E.H. Barakat, M.A. Qayyum, M.N. Hamed, S.A. Al Rashed, Short-term peak demand forecasting in fast developing utility with inherit dynamic load characteristics. Part I. Application of classical time-series methods, IEEE Trans. Power Syst. 5 (3) (1990) 813–819.
[2] G.E.P. Box, G.M. Jenkins, Time Series Analysis: Forecasting and Control, Holden Day, San Francisco, 1970.
[3] B.D. Bunday, Basic Optimisation Methods, Edward Arnold, London, 1984.
[4] A. Chihocki, R. Unbehauen, Neural Networks for Optimisation and Signal Processing, Wiley, New York, 1993.
[5] T.W.S. Chow, C.T. Leung, Neural networks based short-term load forecasting using weather compensation, IEEE Trans. Power Syst. 11 (4) (1996) 1736–1742.
[6] J. Connor, L.E. Atlas, D.R. Martin, Recurrent networks and NARMA modelling, Adv. Neural Inf. Process. Syst. 4 (1992) 301–308.
[7] G.A. Darbellay, M. Slama, Forecasting the short-term demand for electricity, do neural networks stand a better chance? Int. J. Forecast. 16 (2000) 71–83.
[8] M.M. Elkateb, K. Solaiman, Y. Al-Turki, A comparative study of medium-weather-dependant load forecasting using enhanced artificial/fuzzy neural network and statistical techniques, Neurocomputing 23 (1998) 3–13.
[9] D. Fay, J.V. Ringwood, M. Condon, M. Kelly, Comparison of linear and neural parallel time series models for short term load forecasting in the Republic of Ireland, in: 35th Universities Power Engineering Conference, Belfast, UK, 6th–8th September 2000, Queens University, Belfast (Not paginated on CD-ROM).
[10] D. Fay, J.V. Ringwood, M. Condon, M. Kelly, A data fusion model for Irish electricity load forecasting, in: Irish Signal and Systems Conference, Maynooth, Ireland, 25th–27th June 2001, Techman, Maynooth, 2001, pp. 135–140.
[11] N.P.A. van Giersbergen, J.F. Kiviet, How to implement the bootstrap in static or stable dynamic regression models: test statistic versus confidence region approach, J. Econometrics 108 (1) (2002) 133–156.
[12] P.C. Gupta, Adaptive short-term forecasting of hourly loads using weather information, in: D.W. Bunn, E.D. Farmer (Eds.), Comparative Models for Electrical Load Forecasting, Wiley, New York, 1985, pp. 42–56.
[13] A.C. Harvey, Forecasting, Structural Time Series Models and the Kalman Filter, 4th Edition, Cambridge University Press, UK, 1994.
[14] S.H. Hippert, C.E. Pedriera, R.C. Souza, Neural networks for short-term load forecasting: a review and evaluation, IEEE Trans. Power Syst. 16 (1) (2001) 44–55.
[15] D.G. Infield, D.C. Hill, Optimal smoothing for trend removal in short term electricity demand forecasting, IEEE Trans. Power Syst. 13 (3) (1998) 1115–1120.
[16] L.J. Kazmier, N.F. Pohl, Basic Statistics for Business and Economics, McGraw-Hill, Singapore, 1987.
[17] A. Khotanzad, M.H. Davis, A. Abaye, D.J. Maratululam, An artificial neural network hourly temperature forecaster with applications in load forecasting, IEEE Trans. Power Syst. 11 (1996) 870–876.
[18] S.J. Kiartzis, A.G. Bakirtzis, V. Petridis, Short-term load forecasting using neural networks, Elec. Power Syst. Res. 33 (1995) 1–6.
[19] V.S. Kodogiannis, E.M., Anagnostakis, A study of advanced learning algorithms for short-term load forecasting, Eng. Appl. Artif. Intell. 12 (1999) 159–173.
[20] K.Y. Lee, Y.T. Cha, J.H. Park, Short-term load forecasting using an artificial neural network, IEEE Trans. Power Syst. 7 (1) (1992) 124–132.
[21] D.W. Long, M. Brown, C. Harris, Principal component analysis in time-series modelling, in: 35th Universities Power Engineering Conference, Belfast, UK, 6th–8th September 2000, Queens University, Belfast (Not paginated on CD-ROM).

[22] C.N. Lu, H.T. Wu, S. Vemuri, Neural Network based short term load forecasting, IEEE Trans. Power Syst. 8 (1) (1993) 336–342.

[23] J. Mitchell, Comparing feedforward neural network models for time series prediction, in: Proceedings of Neural Research and Applications, Belfast, UK, 25th–26th June 1992, Queens University, Belfast, pp. 175–182.

[24] F.T. Murray, J.V. Ringwood, P.C. Austin, Integration of multi-time-scale models in time series forecasting [electricity consumption], Int. J. Syst. Sci. 31 (10) (2000) 1249–1260.

[25] R. Ramanthan, R. Engle, C.W.J. Granger, F.V. Araghi, Short-run forecasts of electricity loads and peaks, Int. J. Forecast. 13 (1997) 161–174.

[26] A.C. Rencher, Methods of Multivariate Analysis, Wiley, New York, 1995.

[27] S. Vemuri, W.L. Huang, D.J. Nelson, On-line algorithm for forecasting hourly loads of an electric utility, IEEE Trans. Power Appar. Syst. PAS-100 (1981) 3775–3784.

[28] J. Vermaak, E.C. Botha, Recurrent neural networks for short-term load forecasting, IEEE Trans. Power Syst. 13 (1) (1998) 126–132.

Damien Alan Fay was born in 1973 in Dublin, Ireland. He received an honours degree in Electronic Engineering from the National University of Ireland, Dublin, in 1995. In 1997 he received a 1st class honours Masters of Engineering at Dublin City University. From 1997 to 1998 he worked for Alsthom ltd. UK, researching models for steel rolling processes and differential positioning systems for naval vessels. In 1998 he took a research position at Dublin City University where he is currently a Ph.D. student. His main research interests are forecasting techniques, non-linear systems modelling, data fusion techniques and modelling of steel rolling processes.

John Ringwood was educated at Dublin Institute of Technology, Ireland and at Strathclyde University in Scotland, where he was awarded the Ph.D. in 1985. He was a Senior Lecturer at Dublin City University until 2000, when he was appointed Professor and Head of the Department of Electronic Engineering at NUI Maynooth, Ireland. He has held visiting positions at Massey University and the University of Auckland in New Zealand. His research interests focus on the development and application of modelling and control systems techniques. John has acted as a control systems consultant to a number of companies, both in Ireland and abroad. He is a Fellow of the IEE, a Fellow of the IEI and a Chartered Engineer. John is also the current Dean of the Engineering Faculty at NUI Maynooth.

Marissa Condon received the B.E. degree in Electronic Engineering from the National University of Ireland, Galway in 1995, and was awarded the Ph.D. degree in 1998 by the National University of Ireland. She subsequently worked in the National Grid of the Electricity Supply Board of Ireland. In March 2000, she took up a lecturing post at the School of Electronic Engineering in Dublin City University. Her research interests are in the fields of transient analysis and simulation of high-frequency circuits and the development of algorithms suitable for Computer Aided Design (CAD) packages. In addition, she is currently working in the area of non-linear model reduction and the development of suitable macromodels for large-scale circuits and systems operating at high frequencies. She also has interests in electrical load forecasting and neural networks. She served on the Steering committee of the Universities Power Engineering Conference from 1999–2000. She won the prize at the UPEC'96 conference in Crete for the best paper.

Michael Kelly was educated at Dublin Institute of Technology where he was awarded a first class honours degree in Electrical Engineering in 1992. Since then he has worked for Irelands Transmission System Operator (TSO), ESB National Grid. He has worked as a senior engineer in their National Control Centre. Since 1998 he has been manager of the TSO's Generation Analysis Section with responsibility for long term electricity demand forecasting, generation adequacy assessment and market modelling.