Available online at www.sciencedirect.com
ScienceDirect
Neurocomputing 70 (2006) 139–154
NEUROCOMPUTING
www.elsevier.com/locate/neucom

Wavelet-based nonlinear multiscale decomposition model for electricity load forecasting

D. Benaouda[a,*], F. Murtagh[b], J.-L. Starck[c], O. Renaud[d]

[a]Department of Computer Science, College of Information Technology, Universiti Tenaga Nasional, Km7, Jalan Kajang-Puchong, 43009 Kajang, Selangor, Malaysia
[b]Department of Computer Science, Royal Holloway, University of London, Egham, Surrey TW20 0EX, England, UK
[c]DAPNIA/SEDI-SAP, CEA-Saclay, 91191 Gif sur Yvette, France
[d]Faculté de Psychologie et Sciences de l'Education, Université de Genève, 40 Bd. Du Pont d'Arve, 1211 Genève 4, Switzerland

Received 20 June 2005; received in revised form 17 January 2006; accepted 23 April 2006
Communicated by A. Zobaa
Available online 6 June 2006

Abstract

We propose a wavelet multiscale decomposition-based autoregressive approach for the prediction of 1-h ahead load based on historical electricity load data. This approach is based on a multiple resolution decomposition of the signal using the non-decimated or redundant Haar à trous wavelet transform whose advantage is taking into account the asymmetric nature of the time-varying data. There is an additional computational advantage in that there is no need to recompute the wavelet transform (wavelet coefficients) of the full signal if the electricity data (time series) is regularly updated. We assess results produced by this multiscale autoregressive (MAR) method, in both linear and non-linear variants, with single resolution autoregression (AR), multilayer perceptron (MLP), Elman recurrent neural network (ERN) and the general regression neural network (GRNN) models. Results are based on the New South Wales (Australia) electricity load data that is provided by the National Electricity Market Management Company (NEMMCO).

Keywords: Wavelet transform; Load forecast; Scale; Resolution; Time series; Autoregression; Multi-layer perceptron; Recurrent neural network; General regression neural network

1. Introduction

Many power systems not only are being pushed to their limits to meet their customers' demands, but also spend a lot of resources in their operation scheduling. System security is becoming an ever more important issue in modern electrical power systems. Furthermore, power systems need to operate at even higher efficiency in a deregulated electricity market whereby the participating companies such as electricity generators and retailers have to compete in order to maximize their profits. A forecast that exceeds the actual load may lead to extra power being generated and therefore may result in excessive investment in electric plant that is not fully utilized. On the other hand, a forecast that is too low may lead to some revenue loss from sales to neighboring utilities. Hence, accurate electricity load forecasting (LF), including very short-term, short-term, mid-term, and long-term, plays a central role in ensuring adequate electricity generation to meet the customer's demands in the future. LF also helps to build up cost-effective risk management plans for the participating companies in the electricity market. Consequently, good operational planning and smart management decision-making such as economic maintenance scheduling of generation capacity, scheduling of fuel purchase, system security assessment, ability to avoid unnecessary start-ups of generating units, planning the scheduling of peaking power, buying or selling electricity at best price, and scheduling of ancillary services, can all then be carried out based on LF which forecasts the load of a few minutes, hours, days, weeks, months ahead. The aim of LF is to

*Corresponding author. Tel.: +603 8928 7286; fax: +603 8921 2161.
E-mail address: Djamel@uniten.edu.my (D. Benaouda).

0925-2312/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2006.04.005

predict future electricity demand based on historical load data, and currently available data.

Traditionally forecasting methods based on statistical linear regression such as autoregressive (AR) and AR moving average (ARMA) models [6,19,41], have been used for LF [16,31,35]. Time series models [18], and expert system-based methods [30] have also been used for LF. See also [46] for weather sensitive (WS) and non-weather sensitive (NWS) LF methods using regression techniques where WS and NWS load components can be separated by using known weather load models, and [7,8] for LF using automatic adaptive, univariate, and multivariate methods. In recent years, other methods based on artificial intelligence, have shown promising results. The feed-forward artificial neural network (ANN) or multilayer perceptron (MLP)-based methods have received considerable attention for LF. The ANNs as supervised models have been used to deal with the nonlinearity and non-stationarity in electricity load prediction and have produced good and satisfactory results [9,10,13,21,22,25–28,39]. ANNs are well suited to modeling the complex, nonlinear relations involved in LF. These complex relationships are modeled through a process of learning from examples in training, consisting of learning from examples of past load behavior. During the process, ANNs develop internal relationships and use these relationships to synthesize the load forecast. Good reviews on LF solving can be found in [17,21,31,36].

In this article, we explore how the use of linear and nonlinear regression fed with wavelet-transformed data can aid in capturing useful information on various time scales. We also show that multiresolution-based AR approaches outperform the traditional single-resolution approach, and even the well-known nonlinear-based neural network methods (MLP and general regression neural network) to modeling and forecasting.

Wavelet transforms provide a useful decomposition of the time series, in terms of both time and frequency. They have been used effectively for image compression, noise removal, object detection and large-scale structure analysis, among other applications [44,45].

We use the Haar à trous wavelet transform throughout this article. The original signal data can be expressed as an additive combination of the wavelet coefficients at different resolution levels. We introduced the undecimated Haar wavelet transform in [50], and this method was also used in [42]. This choice of wavelet transform was motivated by the fact that the wavelet coefficients are computed only from data obtained previously in time, and the choice of an undecimated wavelet transform avoids aliasing problems.

The wavelet transform has been proposed for time series analysis in many papers, including filtering and forecasting, in recent years. See, for financial time series prediction [1,42,44], wavelet transformed data with use of a neural network [4,50], LF prediction by neuro-fuzzy-wavelet combination [5,23,24,33,34,49], web traffic forecasting [2,3], and Kalman filtering [11]. See also [12,42] which relate the wavelet transform to a multiscale AR type of transform. Wavelet networks are supervised neural networks with (often but not always, orthonormal) wavelet functions replacing, as basis functions, the widely used sigmoid (or other) transfer functions [48] used in ANNs.

2. Wavelets

Our task is to consider the approximation of a time series at coarser and coarser resolution, summarized in a multiresolution decomposition. The individual time series resulting from the decomposition, taken together, can provide a detailed picture of the underlying processes. Nonetheless the current state of these processes may not suffice and some further information about the recent past may be required to make some valuable statements about the near future.

Using a bank of filters with varying frequencies and widths is a difficult task from the optimization point of view. The wavelet transform provides a sound mathematical principle for designing and deploying filters which provides trade-offs between the objectives of efficiency and effectiveness, while retaining immediate relationships with the time series and hence facilitating interpretation and understanding of the data and the analysis methodology. We use a set of filters obtained by rescaling several times a single function, often called a *mother wavelet*, by compressing and expanding it in the time domain, and the outputs of which are the *wavelets*.

The Daubechies and Morlet wavelet transforms have been increasingly adopted by signal- and image-processing researchers. While Daubechies wavelets exhibit good trade-off between parsimony and information richness, it was reported [29] that identical events across the observed time series can appear in so many different fashions that most prediction models are unable to recognize them well. Morlet wavelets, on the other hand, have a more consistent response to similar events but have the weakness of generating many more inputs than the Daubechies wavelets for the prediction models. As already noted, our chief consideration regarding the choice of mother wavelets include (i) aliasing, implying preference for a redundant wavelet transform algorithm, and (ii) a wavelet function which respects the asymmetric nature of a time varying signal, leading to our use of the Haar wavelet function.

2.1. The "A Trous" wavelet decomposition

The continuous wavelet transform of a continuous function produces a continuum of scales as output. On the other hand, input data is usually discretely sampled, and furthermore a dyadic or two-fold relationship between resolution scales is both practical and adequate. The latter two issues lead to the discrete transform.

The output of a discrete wavelet transform can take various forms. Traditionally, a triangle (or pyramid in the case of 2-dimensional images) is often used to represent all

that is worth considering in the sequence of resolution scales. Such a triangle comes about as a result of decimation or the retaining of one sample out of every two. The major advantage of decimation is that just enough information is kept to allow exact reconstruction of the input data. Therefore decimation is ideal for an application such as compression. It can be easily shown too that the storage required for the wavelet-transformed data is exactly the same as is required by the input data. The computation time for many wavelet transform methods is also linear in the size of the input data, i.e. $O(n)$ for n-length input time series.

With the decimated form of output it is less easy to visually or graphically relate information at a given time point at different scales. More problematic is the lack of shift invariance. This means that if we had deleted the last few values of our input time series, then the output wavelet transformed, decimated, data would be quite different from heretofore. One way to bypass this problem at the expense of greater storage requirements is by means of a redundant or non-decimated wavelet transform.

A redundant transform based on an n-length input time series, then, has an n-length resolution scale for each of the resolution levels that we consider. It is easy under these circumstances to relate information at each resolution scale for the same time point. We do have shift invariance. Finally, the extra storage requirement is by no means excessive.

The first redundant, discrete wavelet transform to be described now is one used by [4]. The successive resolution levels are formed by convolving with an increasingly dilated wavelet function, which looks rather a Mexican hat (central bump, symmetric, two negative side lobes). Alternatively these resolution levels may be constructed by (i) smoothing with an increasingly dilated scaling function looking rather like a Gaussian function defined on a fixed interval (support)—this function is a B_3 spline; and (ii) taking the difference between successive versions of the data which are smoothed in this way.

The à trous wavelet transform [40] allows the filter outputs to be interpreted in a meaningful way. The à trous wavelet can be described simply as follows. First, perform successive convolutions with the discrete low-pass filter h:

$$C_{i+1}(k) = \sum_{l=-\infty}^{+\infty} h(l) C_i\left(k + 2^i l\right), \tag{1}$$

where the finest scale is the original series: $C_0(t) = X(t)$ (see e.g. [47]). The increase in distances between the samples points (i.e. $2^i l$) explains why the name "à trous" (with holes) has been applied to this method. The low-pass filter, h, is a B_3 spline, defined as $\left(\frac{1}{16}, \frac{1}{4}, \frac{3}{8}, \frac{1}{4}, \frac{1}{16}\right)$. This is of compact support (necessary for a wavelet transform), and is point-symmetric. The latter, however, does not allow for the fact that time is fundamentally asymmetric.

Now, from the sequence of smoothing of the signal, we take the difference between successive smoothed versions to obtain the wavelet coefficients w_i:

$$w_i(k) = C_{i-1}(k) - C_i(k). \tag{2}$$

The latter provide the detail signal, or wavelet coefficients, which we hope in practice will capture small features of interpretational value in the data.

It is easy to show that we have the following expansion of the original data:

$$X(t) = C_p(t) + \sum_{i-1}^{p} w_i(t). \tag{3}$$

For a fixed number of scales (p), the computational complexity of the above algorithm is $O(n)$ for an n-length input. It therefore has more favorable computational cost than the Fast Fourier Transform (FFT).

If the background or residual vector C_p is sufficiently smooth, its one-step ahead prediction may become trivial, e.g. by linear extrapolation or, even more simply, a carbon copy (C_{t-1} as a predictor of C_t) of it.

We now consider the issue of prediction. We can think of successive convolutions as a moving average of increasingly distant points. At time t, we have the observations $X(t)$, $X(t-1), \ldots, X(1)$ and we are seeking an accurate estimate of $X(t+k)$, where k is the look ahead period.

If the resolution scales, w_i, C_p, are physically interpretable (as they may possibly be) then the independent predictions are similarly open to interpretation. The fact that the reconstruction (Eq. (3)) is additive allows us to form a consensus prediction also in an additive manner.

A hybrid strategy can also be adopted in regard to exactly what is combined to yield an overall prediction, i.e. we can test a number of short-memory and long-memory predictions at each resolution level, and retain the method, which performs best.

2.2. The Haar "a trous" wavelet transform

The à trous wavelet transform with a wavelet function related to B_3 spline function, as described above, is not appropriate for a directed (time-varying) data stream. To cater for the requirement that future data values cannot be used in the calculation of the wavelet transform, we use the Haar à trous wavelet transform, introduced in [50].

The Haar wavelet transform was first described in the early years of this century and is described in almost every text on the wavelet transform. As already noted, the asymmetry of the wavelet function used makes it a good choice for edge detection, i.e. localized jumps. The usual Haar wavelet transform, however, is a decimated one. We now develop a non-decimated or redundant version of this transform. This will be an à trous algorithm, with a B_1 or "top hat" scaling function, as used by the Haar wavelet transform.

The non-decimated Haar algorithm is exactly the same as the à trous algorithm described heretofore, except that the low-pass filter h, $\left(\frac{1}{16}, \frac{1}{4}, \frac{3}{8}, \frac{1}{4}, \frac{1}{16}\right)$, is replaced by the simpler

filter $(\frac{1}{2},\frac{1}{2})$. There, h is now non-symmetric. Consider the creation of the first wavelet resolution level. We have C_1 created from C_0 (original signal) by convolving the latter with h. Then:

$$C_1(k) = \frac{1}{2}[C_0(k) + C_0(k-1)],$$
$$w_1(k) = C_0(k) - C_1(k).$$

More generally:

$$C_{i+1}(k) = \frac{1}{2}\left[C_i(k) + C_i(k-2^i)\right],$$
$$w_{i+1}(k) = C_i(k) - C_{i+1}(k). \quad (4)$$

At any time point, k, we never use information (time-wise) after k in calculating the wavelet coefficient.

The Haar à trous transform provides a convincing and computationally very straightforward solution to troublesome time series boundary effects at the time point t. Obviously we do not care about time point 1, since this is far back in the time series. More experimental results with this redundant transform can be found in [37,50].

Fig. 1 shows the transform on a sample set of New South Wales (NSW) electricity load data. The elementwise sum of scales 1–7, plus the smooth trend, gives the original data set. Note the following: (i) all wavelet scales are of zero mean; and (ii) the smooth trend plot is very often much larger-valued (as it is the case here) compared to the max–min ranges of the wavelet coefficients.

Fig. 2 shows which time steps of the input signal are used to calculate the last wavelet coefficient in the different scales. A wavelet coefficient at a position t is calculated from the signal samples at positions less than or equal to t, but never larger.

3. Forecasting using wavelet decomposition

We used the Haar à trous wavelet decomposition, described above, of the signal for forecasting. The forecasting problem considered is the determination of one hour-ahead electricity forecast. Instead of using the vector of past electricity load observations $X = (X_1, \ldots, X_N)$ to forecast X_{N+1}, its wavelet transform is used. Half hourly and hourly load data over a 6-year period from NSW (Australia) were used. From a typical daily load graph as shown in Figs. 3 and 4 we note a definite pattern based on the season of the year, holiday season, and day of the week. The load also follows a set of patterns within any day and depending on the time of the day. Fig. 4 shows that there is an enormous regularity in load data.

Renaud et al. [32,37] give the necessary wavelet coefficients at each scale (j) that will be used for the forecast at time $N+1$ as having the form $w_{j,N-2^j}(k-1)$ and $C_{J,N-2^j}(k-1)$ for positive values of k shown in Fig. 5.

3.1. Linear and stationary-based models

Assume a signal $X = (X_1, \ldots, X_N)$ and assume we want to forecast X_{N+1}. We use the coefficients $w_{j,N-2^j}(k-1)$ for $k = 1, \ldots, A_i$ and $j = 1, \ldots, J$ and $C_{J,N-2^j}(k-1)$ for

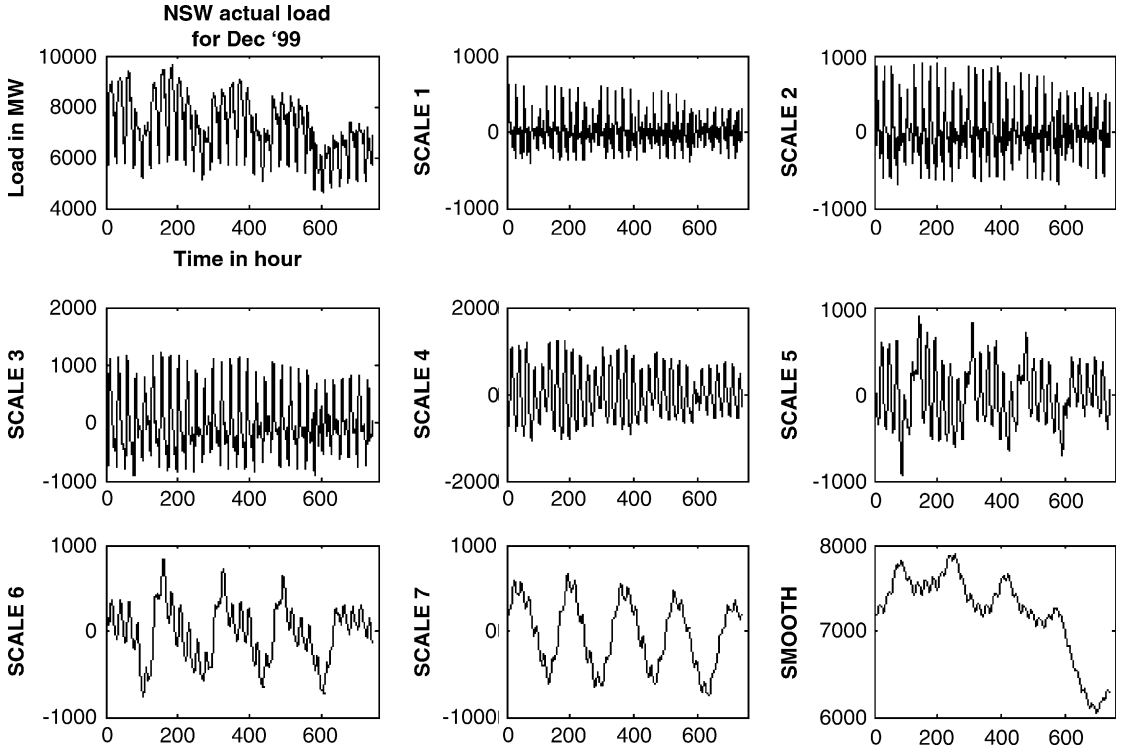


Fig. 1. Haar à trous wavelet transform of a sample set, 744-valued, electricity hourly load.

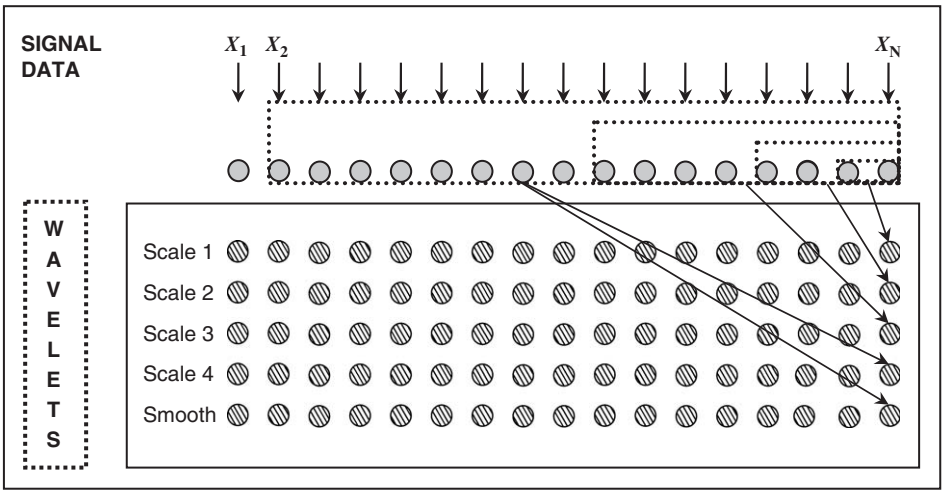


Fig. 2. Redundant Haar à trous wavelet transform—this shows which time steps of the signal data are used to compute the last wavelet coefficients at each different scale.

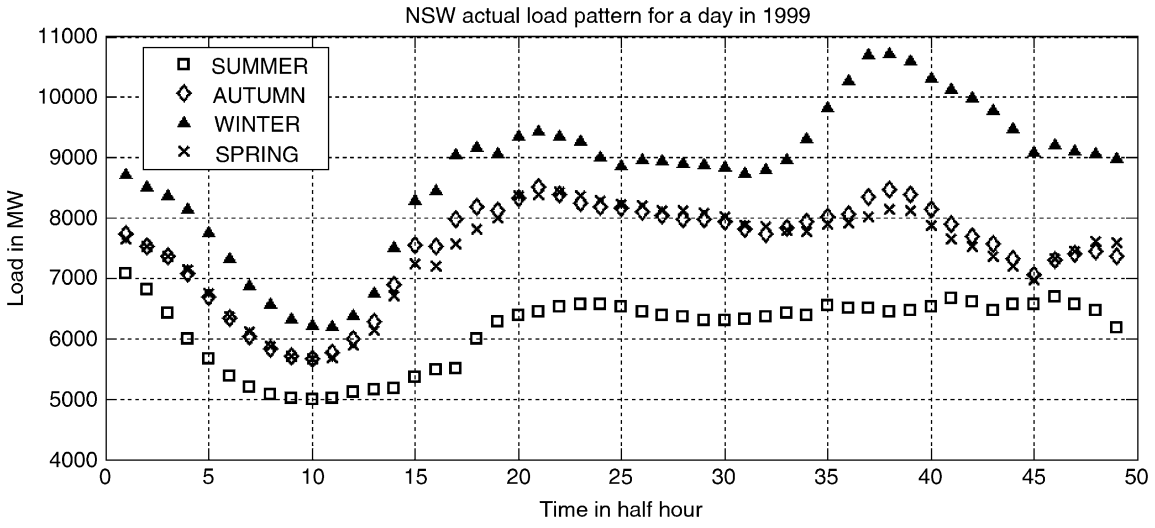


Fig. 3. Daily load pattern in different seasons.

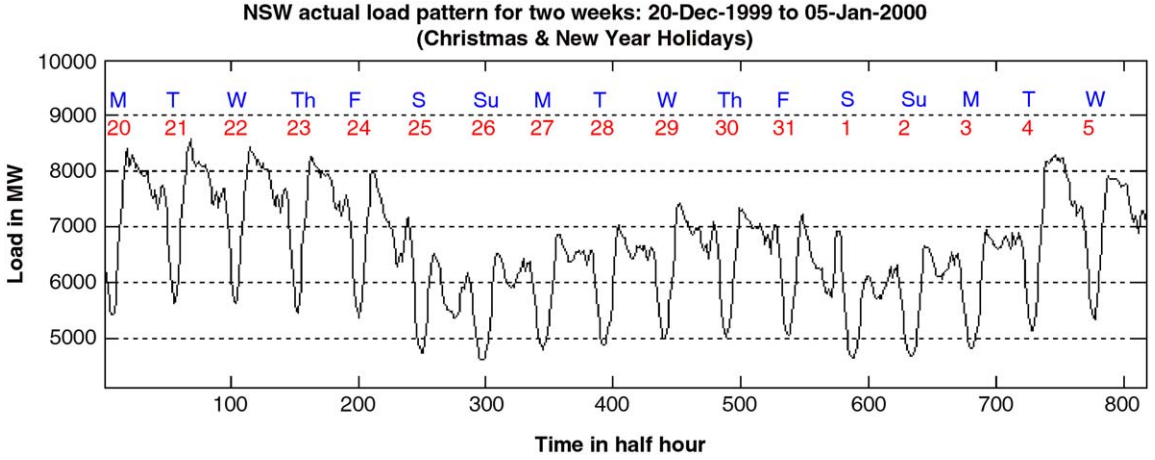


Fig. 4. Fortnightly load pattern around Christmas and New Year Holidays.

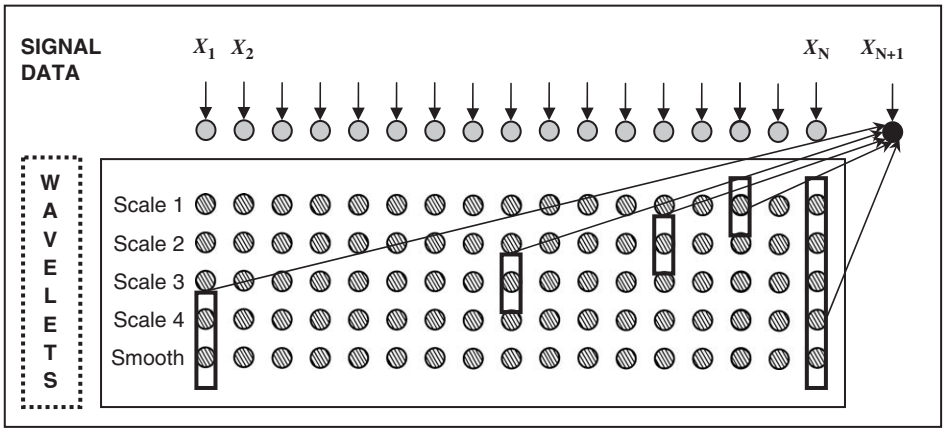


Fig. 5. Ten wavelet coefficients, MAR(2) with 4 wavelet scales plus the smoothed array, that are used for the forecast of the next value X_{N+1}.

$k = 1, \ldots, A_{J+1}$. See Fig. 4 when $J = 4$ and $A_j = 2$ for $j = 1, \ldots, J+1$.

A one-step forward forecast of a linear AR model or AR(p) process is written:

$$\widehat{X}_{N+1} = \sum_{k=1}^{p} \widehat{\beta}_k X_{N-(k-1)}, \tag{5}$$

where $\left(\widehat{\beta}_k\right)$ are ordinary least-squares estimators obtained by minimizing the residual sum of squares with respect to $(\beta_1, \ldots, \beta_p)$

$$\mathrm{RSS}_p = \sum_{N=p}^{n} \left(X_{N+1} - \sum_{k=1}^{p} \beta_k X_{N-(k-1)} \right)^2. \tag{6}$$

In order to use the wavelet decomposition, we consider multi-resolution AR forecasting (MAR) [37,38]:

$$\widehat{X}_{N+1} = \sum_{j=1}^{J} \sum_{k=1}^{A_j} \hat{a}_{j,k} w_{j,\, N-2^j(k-1)} + \sum_{k=1}^{A_{J+1}} \hat{a}_{J+1,k} C_{J,\, N-2^j(k-1)}, \tag{7}$$

where $w = (w_1, \ldots, w_j, C_j)$ represents the Haar à trous wavelet transform of X, i.e. $X = \sum_{j=1}^{J} w_j + C_J$.

Fig. 5 shows which wavelet coefficients are used for the forecasting using $A_j = 2$ for all resolution levels j, and a wavelet transform with five scales (four wavelet scales plus the smoothed array). In this case we can see that only 10 coefficients are used, including coefficients that take into account low-resolution information. This means that long-term forecasting can easily be achieved, either by increasing the number of scales in the wavelet transform, or by increasing the AR order in the last scale, but with a very limited additional number of parameters.

To further link this method with forecasting based on classical AR, note that if on each scale the lagged coefficients follow an AR(A_j) model, the addition of the forecasts on each scale would lead to the same forecasting Eq. (7).

To estimate the $p = \left(\sum_{j=1}^{J+1} A_j\right)$ unknown coefficient vectors (β) whose variables are the ($a_{j,k}$) described above, we used the least-squares method: minimizing the sum of squares of the differences between the forecast value in Eq. (7) and the actual value X_{N+1} over all the values of N in the training sample time, which leads to solving of the normal equations: $Z \times \hat{\beta} = X$, where Z is the $n \times p$ matrix composed of the n sample of the input variables $\left[w_{j,N-2^j(k-1)},\ C_{J,N-2^J(k-1)}\right]$, and the observed (actual) $n \times 1$ vector is X. The AR order at the different scales must be defined. We used the Bayesian Information Criterion (BIC), to select the parameters A_j independently on each scale; AIC and AICC methods can also be used [6,41].

The BIC criterion is defined as follows [41]:

$$\widehat{\sigma}_p^2 = \mathrm{mse}_p = \frac{\mathrm{RSS}_p}{n-p} \text{ (Mean Square Error)},$$

$$\mathrm{BIC}_p = \ln\left(\widehat{\sigma}_p^2\right) + \frac{p \log_{10}(n-p)}{n-p}. \tag{8}$$

The value of p yielding the minimum BIC furnishes the best model.

3.2. Non-linear and non-stationary-based models

To go beyond the MAR forecasting models described previously, we can use any type of forecasting, linear or nonlinear, that uses the previous signal data $\left(X_{N-p}, \ldots, X_N\right)$ and generalize it through the use of the wavelet coefficients $w_{j,N-2^j(k-1)}$ and the smoothed coefficients $c_{J,N-2^J(k-1)}$, $k \leqslant N$ instead. In this article, we feed a MLP neural network with these coefficients as inputs, use one or more hidden layer(s), and obtain $\widehat{X}_{N+1}$ as the forecasted output (see Fig. 6). To obtain the optimal weights (parameters) we can either use an efficient evolutionary programming algorithm [15] or a gradient-based back-propagation algorithm [20] to train the network. We assume a standard MLP with a linear transfer

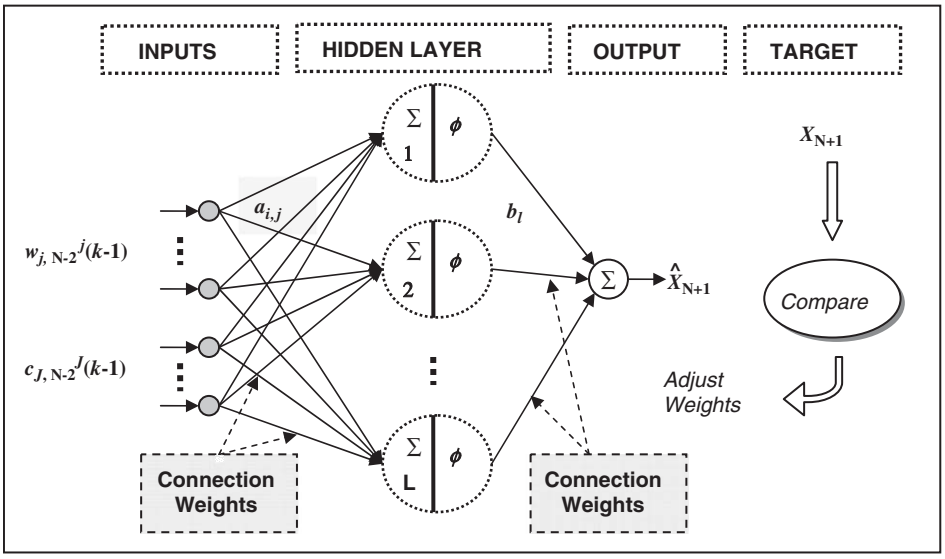


Fig. 6. Wavelet based multilayer perceptron (MLP) neural network architecture.

function at the output neuron, and L hidden layer neurons with sigmoid transfer function (ϕ), this is written as

$$\hat{X}_{N+1} = \sum_{l=1}^{L} b_l \times \phi\left(\sum_{j=1}^{J}\sum_{k=1}^{A_j} \hat{a}^l_{j,k} w_{j,N-2^j(k-1)} + \sum_{k=1}^{A_{J+1}} \hat{a}^l_{J+1,k} C_{J,N-2^j(k-1)}\right). \quad (9)$$

4. Implementation of a neuro-wavelet hybrid method

In this section we describe our prototype to capture the overall context for deploying the forecasting engines that will be evaluated below.

4.1. System design

The quite complex processing chain in electricity LF is best handled by a computer system designed for the job. The system provides an electricity load forecast solution for participating companies' usage in order to not only bid strategically into the electricity market but also to maximize their profits. All forecasts provided are on the scale of very short-term, short-term, mid-term, and long-term forecasts. Such a system must be designed with essential elements as shown in Fig. 12. (Our emulation of this computing platform used Microsoft Windows XP and a graphical user interface designed using MATLAB 7.1 software.) The user, in the control center, must define the problem by choosing the appropriate hour intervals, day, and forecaster model. The results of running the model are displayed graphically. There are three main functional modules to the system. These are shown in Fig. 13 and are described in detail in the following subsections.

4.2. Load and weather data exploration and analysis

The purpose of load and weather visualization is to enable the user in the control center to become familiar with the data before making explicit choices. The ability to plot load/weather variables against time (half-hour, hour), and cross-plots of one variable against another, are standard. The system allows also zoom in/out a certain area in the plot. This provides functionality for deciding whether some load, weather (temperature, relative humidity, wind speed, precipitation), demographic, economic variables in a cross-plot, say, are correlated so these variables can be selected as the model's inputs. Principal components analysis is available for any selection of samples, loads, and weather data in order to decorrelate and compress the data. Finally, the wavelet transforms of the data help to capture small features of interpretational value in the data.

4.3. Data selection

Explicit selections of day/date interval, wavelet scale, weather condition variables, if available, must be made. However, weather variables must be carefully selected based on the data exploration and analysis phase.

4.4. Neuro-wavelet hybrid forecaster

The wavelet transform load/weather database is used to prepare samples for training and testing the selected model. The operator may now select the forecaster model, train, and test the model. If the model's testing performance is not met or is un-satisfactory then alternative data selection must be made. Otherwise, the trained model is run on a specific new sample to give a forecasted load. A range of different models, spanning linear single/muli-resolution

AR through to nonlinear feed-forward MLP neural networks, including hybrid models, are available.

5. Experimental results

In this work the MATLAB 7.1 high-level programming language was used to implement the proposed forecasting methods, the single resolution AR model, multiscale wavelet-based AR model (MAR), the MLP neural network, the Elman recurrent neural network (ERN) and the general regression neural network (GRNN). The simulations were run on a Microsoft Windows XP-based platform (Intel Pentium processor 1400 MHz, and 256 MB of RAM). The fast Levenberg–Marquardt back-propagation training algorithm was used to train the neural network on the relatively large training data sets. The above models were assessed on the NSW (Australia) real load data from electricity market data, which is available publicly from the National Electricity Market Management Company (NEMMCO).

Table 1 shows the list of time periods that were used for the load forecast in each load season. The historical load data for the period described in Table 1 was used to train the above models, and then employed to forecast the test load data.

The parameter settings for training the models above, including MLP through the Levenberg–Marquardt back-propagation algorithm are described in Table 2. The MLP's training process was stopped after 1 h, 13 min, and 28 s (CPU time) in 500 epochs with Mean Squared Error (MSE) = 0.005, MSE, the network's total error. On the other hand, the GRNN's smoothing or spread factor was set to 50. Table 3 shows a comparative study between the different models used with respect to their CPU computing time for 1-h ahead hourly load forecast.

For comparison, the MAR model's results were compared with the ones produced by the basic single AR model [41], Levenberg–Marquardt back-propagation-based MLP, or a feed-forward neural network [20], the ERN [14] and the GRNN [43]. We also trained the MLP, ERN, and GRNN with the Haar à trous wavelet-transformed data whereby the results were greatly improved.

Table 1
New South Wales (NSW) load data used both to train and test the proposed forecasting models

Time period (date)		Sample size (hourly load)
Historical load data		
1 January 1999–31 December 2001		26297
Test load data		
Summer load	1 January 2002	24
	2 January 2002	24
	3 January 2002	24
Autumn load	9 April 2002	24
	11 April 2002	24
	12 April 2002	24
	25 April 2002	24
Winter load	14 June 2002	24
Spring load	4 October 2002	24
Summer load	25 December 2002	24
	26 December 2002	24
	27 December 2002	24
	28 December 2002	24
	29 December 2002	24
	30 December 2002	24
	31 December 2002	24

5.1. Input data preparation and training

The input data consists of historical electricity load which was collected over 3 years (January 1, 1999–December 31, 2001): 26297 hourly load values, to train the above models; and data of 1 year (January 1, 2002–December 31, 2002) is used for testing, see Table 1. For testing we chose days from different seasons, including January 3, 2002; April 11, 2002; and other dates (see Table

Table 2
Model architectures and parameter settings for 1-h ahead hourly load forecasting

	AR	MAR	MLP*w*	MLP	GRNN	GRNN*w*	ERN*w*
No. of input nodes	27	28	28	20	20	28	28
No. of hidden nodes	—	—	10	10	—	—	10
No. of output nodes	1	1	1	1	1	1	1
GRNN's spread factor	—	—	—	—	50	50	—
Total training patterns	26297	26297	26297	26297	26297	26297	26297
Epochs	—	—	500	500	—	—	500
Training stopping threshold	—	—	0.001	0.001	—	—	0.001
Levenberg–Marquardt's learning parameter settings							
Mu	—	—	0.001	0.001	—	—	0.001
Mu_dec	—	—	0.1	0.1	—	—	0.1
Mu_inc	—	—	10	10	—	—	10
Mu_max	—	—	10^{10}	10^{10}	—	—	10^{10}

Table 3
CPU computing time for 1-h ahead hourly load forecast

Model	Training						Testing
	Input Size	Total Samples	CPU Time (H:M:S)	Epochs	MSE/STDEV	Performance Goal	CPU Time (s)
MLP*w* (28-10-1)[a]	28	26297	01:13:28/01:38:28[b]	500	0.005[c]	0.001	12
MLP (20-10-1)	20	26297	00:45:33	500	0.009	0.001	12
GRNN*w* (28-1)[a]	28	26297	13 s/25 mins[b]	—	—	—	3
GRNN (25-1)	20	26297	5 s	—	—	—	1
AR(27)	27	26297	3 s	—	154.5[d]	—	1
MAR(7)/Scale = 3	28	26297	5 s/25 mins[b]	—	184.9[d]	—	1
ERN*w* (28-10-1)[a]	28	26297	05:46:51/06:11:51[b]	500	0.003[c]	0.001	11

[a]Input neurons are being fed by wavelet transformed data.
[b]This is an execution time when generating, on the fly, the training and testing file from a wavelet-transformed file where scale = 3 and order = 7. If the train-test file is separately generated by the data preparation module, the execution time becomes very much faster.
[c]MSE: mean-squared error.
[d]STDEV: standard deviation.

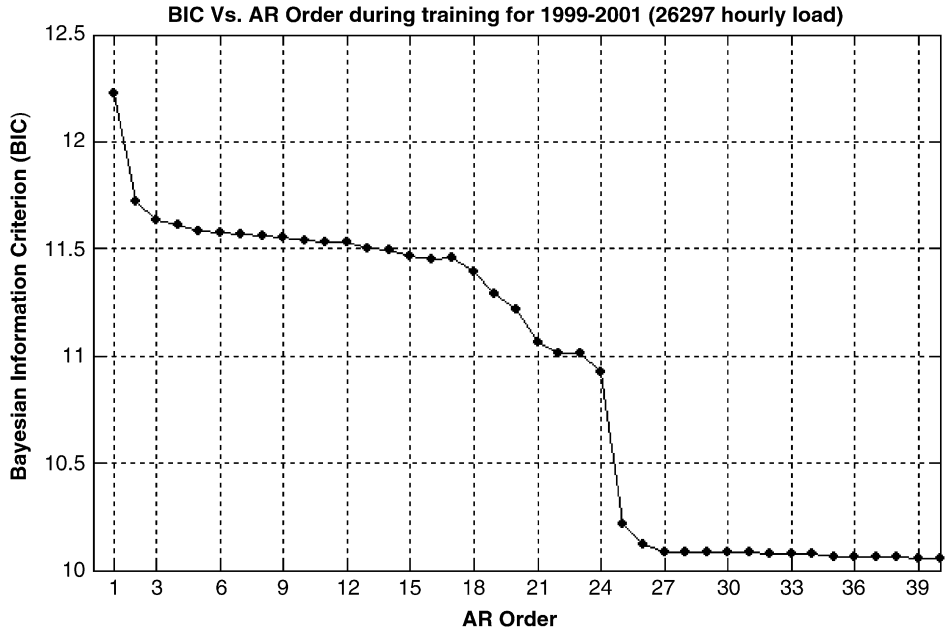


Fig. 7. Plot of BIC against AR order.

captions: 4–19). The forecasting architectures used including MLP, ERN, and GRNN were defined from the training data; but data from the test period was used—of necessity—to get the 1-h ahead forecast.

In our training and testing approach we therefore take 3 years as the period during which training takes place. The test intervals are chosen from within the following year. Procedurally, our motivation was to achieve algorithmic robustness and generality of application.

The best linear AR model found, non-multiresolution AR(27), based on the BIC [41], is shown in Fig. 7. An MAR(7), with 3 wavelet resolution scales, model provided better results, in terms of absolute percentage error (APE) (%) and mean absolute percentage error (MAPE) (%) performance measures described below, compared to AR and other non-linear approaches such as MLP (20-10-1), back-propagation, 3 layers, 20 input units, 10 hidden units, 1 output unit, and GRNN (20-1), 3 layers, 20 inputs, 1 output, which both give best performance on the training data set during the learning process. We also ran MLP, ERN and GRNN on the Haar à trous wavelet transformed data, MAR(7) with 3 wavelet scales, and these are respectively denoted as MLP*w*, the wavelet-based nonlinear model (see Fig. 6), ERN*w* and GRNN*w*. According to simulation results, MLP*w*, ERN*w*, and GRNN*w* outperformed, respectively MLP, ERN and GRNN when fed with the original data. Furthermore, MLP*w* outperformed MAR(7) in some testing days, but ERN*w* outperformed both of them in most testing days.

So the ERN*w* is either the best predictor model, and MLP*w* is either the second best predictor model or the third best after the MAR model.

The disadvantage of the ERN*w* model is that it is very slow during training compared to MLP*w*. Therefore, in regard to neural network regression there is a potential trade-off between effectiveness (in testing) and efficiency (in training).

In the comparisons of model performance, the load forecast accuracy is determined in terms of two performance measures, which are adopted here as follows:

APE, which is defined as

$$\mathrm{APE}(\%) = \frac{|\mathrm{Actual}(i) - \mathrm{Forecast}(i)|}{\mathrm{Actual}(i)} \times 100.$$

MAPE, which is defined as follows:

$$\mathrm{MAPE}(\%) = \frac{1}{m}\sum_{i=1}^{m} \frac{|\mathrm{Actual}(i) - \mathrm{Forecast}(i)|}{\mathrm{Actual}(i)} \times 100,$$

where m is the total number of hours predicted, Actual(i) is the actual load for the hour i, and Forecast(i) is the predicted load for the hour i.

5.2. One-hour ahead LF

As shown in Tables 4–19, AR, MAR, MLP, ERN and GRNN models, as described previously, were used to forecast 1-h ahead electricity load. In addition, MLP, ERN and GRNN were also used to forecast 1-h ahead load using the wavelet transformed data, rather than the original data, and are denoted MLP*w*, ERN*w* and GRNN*w*, respectively. The MLP, MLP*w*, ERN*w*, and GRNN networks used are 3-layer architecture, denoted as input-hidden-output. The input variables and the number of hidden neurons (for the networks) were selected give the best performance on the training data set during the models' training process, and are defined in Table 2.

- Fig. 8 shows simultaneous plots of the actual load versus, respectively MAR, ERN*w*, MLP*w*, and AR-based forecasted load for a day of April 11, 2002.

Table 4
One-hour ahead hourly load forecasting results for a day of January 2002

Forecasted day of January 2, 2002 (summer load)		
Model	Max APE (%)	MAPE(%)
ERN*w* (28-10-1)[a]	1.7718	0.7913
MLP*w* (28-10-1)[a]	2.7982	1.0959
MLP (20-10-1)	3.6196	1.4387
MAR(7)/Scale = 3	4.9997	2.0315
AR(27)	7.2477	2.1377
GRNN*w* (28-1)[a]	10.5848	2.3465
GRNN (20-1)	17.4264	3.5483

[a]Input neurons are being fed by wavelet transformed data.

Table 5
One-hour ahead hourly load forecasting results for a day of January 2002

Forecasted day of January 3, 2002 (summer load)		
Model	Max APE (%)	MAPE(%)
ERN*w* (28-10-1)[a]	1.4522	0.6615
MLP*w* (28-10-1)[a]	2.8509	1.0565
MAR(7)/Scale = 3	4.3313	1.2987
MLP (20-10-1)	3.2233	1.4394
AR(27)	6.1069	1.7461
GRNN*w* (28-1)[a]	5.8470	1.7542
GRNN (20-1)	6.9039	1.9481

[a]Input neurons are being fed by wavelet transformed data.

Table 6
One-hour ahead hourly load forecasting results for a day of April 2002

Forecasted say of April 9, 2002 (autumn load)		
Model	Max APE (%)	MAPE(%)
ERN*w* (28-10-1)[a]	1.6740	0.7142
MLP*w* (28-10-1)[a]	2.2404	0.8156
MAR(7)/Scale = 3	2.1809	0.9299
MLP (20-10-1)	3.1668	1.1838
AR(27)	3.5009	1.3518
GRNN*w* (28-1)[a]	6.1072	2.2547
GRNN (20-1)	8.2099	2.7960

[a]Input neurons are being fed by wavelet transformed data.

Table 7
One-hour ahead hourly load forecasting results for a day of April 2002

Forecasted day of April 11, 2002 (autumn load)		
Model	Max APE (%)	MAPE(%)
ERN*w* (28-10-1)[a]	1.5503	0.6419
MAR(7)/Scale = 3	2.1211	0.8397
MLP*w* (28-10-1)[a]	3.8964	0.9990
AR(27)	3.6043	1.1490
MLP (20-10-1)	4.2261	1.2613
GRNN*w* (28-1)[a]	8.4860	1.4545
GRNN (20-1)	8.5824	2.2723

[a]Input neurons are being fed by wavelet transformed data.

- Fig. 9 shows plots of the difference between actual load and forecasted load ERN*w*, MLP*w*, MAR, AR, and MLP, respectively for a day of April 11, 2002.
- Fig. 10 shows simultaneous plots of the actual load versus, respectively ERN*w*, MLP*w*, MAR, MLP-based forecasted load for a day of January 3, 2002.
- Fig. 11 shows plots of the difference between actual load and forecasted load ERN*w*, MLP*w*, MAR, AR, and MLP, respectively, for a day of January 3, 2002.

Table 8
One-hour ahead hourly load forecasting results for a day of April 2002

Forecasted day of April 12, 2002 (autumn load)		
Model	Max APE (%)	MAPE(%)
ERN*w* (28-10-1)[a]	1.9989	0.6509
MAR(7)/Scale = 3	2.6122	0.8547
MLP*w* (28-10-1)[a]	3.5656	0.9861
AR(27)	4.2940	1.1861
MLP (20-10-1)	4.8677	1.7523
GRNN*w* (28-1)[a]	7.9124	2.0437
GRNN (20-1)	6.8812	2.7866

[a]Input neurons are being fed by wavelet transformed data.

Table 9
One-hour ahead hourly load forecasting results for a day of April 2002

Forecasted day of April 25, 2002 (autumn load)		
Model	Max APE (%)	MAPE(%)
ERN*w* (28-10-1)[a]	1.8642	0.8978
MLP*w* (28-10-1)[a]	3.5822	1.0622
MAR(7)/Scale = 3	9.6323	1.8131
MLP (20-10-1)	7.3447	1.8118
AR(27)	8.8854	2.1253
GRNN*w* (28-1)[a]	13.4215	2.4916
GRNN (20-1)	27.4576	4.1453

[a]Input neurons are being fed by wavelet transformed data.

Table 10
One-hour ahead hourly load forecasting results for a day of June 2002

Forecasted day of June 14, 2002 (winter load)		
Model	Max APE (%)	MAPE(%)
MAR(7)/Scale = 3	2.6519	0.7268
ERN*w* (28-10-1)[a]	2.2798	0.8758
MLP*w* (28-10-1)[a]	2.8113	0.9873
AR(27)	3.9263	1.0227
MLP (20-10-1)	3.0813	1.0651
GRNN*w* (28-1)[a]	5.3814	1.4188
GRNN (20-1)	5.3557	1.5850

[a]Input neurons are being fed by wavelet transformed data.

Table 11
One-hour ahead hourly load forecasting results for a day of October 2002

Forecasted day of October 4, 2002 (spring load)		
Model	Max APE (%)	MAPE(%)
ERN*w* (28-10-1)[a]	2.0637	0.9470
MAR(7)/Scale = 3	3.2292	0.9775
MLP*w* (28-10-1)[a]	2.6576	1.1013
MLP (20-10-1)	3.4377	1.2450
AR(27)	5.8894	1.2727
GRNN*w* (28-1)[a]	4.9900	1.3258
GRNN (20-1)	3.1987	1.5926

[a]Input neurons are being fed by wavelet transformed data.

Table 12
One-hour ahead hourly load forecasting results for a day of December 2002

Forecasted day of December 25, 2002 (summer load)		
Model	Max APE (%)	MAPE(%)
ERN*w* (28-10-1)[a]	2.2842	0.8766
MLP*w* (28-10-1)[a]	5.4097	1.5101
MLP (20-10-1)	4.9054	1.6144
AR(27)	6.0200	1.8189
MAR(7)/Scale = 3	5.9334	1.9296
GRNN*w* (28-1)[a]	6.0036	2.2763
GRNN (20-1)	17.3829	7.2458

[a]Input neurons are being fed by wavelet transformed data.

Table 13
One-hour ahead hourly load forecasting results for a day of December 2002

Forecasted day of December 26, 2002 (summer load)		
Model	Max APE (%)	MAPE(%)
ERN*w* (28-10-1)[a]	1.5327	0.7326
MLP*w* (28-10-1)[a]	3.2157	0.9777
MAR(7)/Scale = 3	3.1300	1.2386
AR(27)	4.0390	1.2248
MLP (20-10-1)	4.1756	1.4303
GRNN*w* (28-1)[a]	7.9571	1.6544
GRNN (20-1)	9.6433	2.7961

[a]Input neurons are being fed by wavelet transformed data.

- Tables 4–19 describe a summary of testing results on selected days (in different seasons) based on six different models. It also shows that ERN*w* is the best predictor model.

6. Conclusions and discussions

In this work a linear multi-resolution autoregressive (MAR) method has been proposed based on a wavelet transform to forecast 1-h ahead load of the New South Wales electricity market. This wavelet transform is the redundant Haar à trous wavelet transform which decomposes the signal data into multiple resolution scales and has the advantage of being shift-invariant. It is also easy to implement and is computationally efficient.

The simplest MAR method exhibits higher ability of generalization than the single AR method. Unlike AR, the MAR's forecasting method uses a small number of wavelet coefficients of the decomposition of the past values on each scale. MAR also outperformed the ordinary nonlinear methods such as multilayer perceptron (MLP) and general regression neural network (GRNN). Importantly—and

Table 14
One-hour ahead hourly load forecasting results for a day of December 2002

Forecasted day of December 27, 2002 (summer load)		
Model	Max APE (%)	MAPE(%)
ERN*w* (28-10-1)[a]	1.9445	0.7798
MLP*w* (28-10-1)[a]	3.8414	1.4384
MAR(7)/Scale = 3	6.4441	1.3008
AR(27)	4.8717	1.3858
GRNN*w* (28-1)[a]	3.4406	1.6047
GRNN (20-1)	4.5541	1.7044
MLP (20-10-1)	3.5918	1.7864

[a]Input neurons are being fed by wavelet transformed data.

Table 15
One-hour ahead hourly load forecasting results for a day of December 2002

Forecasted day of December 28, 2002 (summer load)		
Model	Max APE (%)	MAPE(%)
ERN*w* (28-10-1)[a]	1.9236	0.8607
MAR(7)/Scale = 3	2.9988	0.9770
MLP*w* (28-10-1)[a]	3.8690	1.0958
AR(27)	3.4147	1.1782
MLP (20-10-1)	4.2976	1.2814
GRNN*w* (28-1)[a]	4.6527	1.5319
GRNN (20-1)	6.0422	1.7356

[a]Input neurons are being fed by wavelet transformed data.

Table 16
One-hour ahead hourly load forecasting results for a day of December 2002

Forecasted day of December 29, 2002 (summer load)		
Model	Max APE (%)	MAPE(%)
ERN*w* (28-10-1)[a]	1.8752	0.6877
MLP*w* (28-10-1)[a]	3.7942	1.1175
MAR(7)/Scale = 3	3.9535	1.1826
AR(27)	4.1432	1.2026
MLP (20-10-1)	5.3583	1.2474
GRNN*w* (28-1)[a]	3.7675	1.4413
GRNN (20-1)	4.2903	1.8044

[a]Input neurons are being fed by wavelet transformed data.

Table 17
One-hour ahead hourly load forecasting results for a day of December 2002

Forecasted day of December 30, 2002 (summer load)		
Model	Max APE (%)	MAPE(%)
ERN*w* (28-10-1)[a]	2.0600	0.8085
MLP*w* (28-10-1)[a]	3.8465	1.2096
MLP (20-10-1)	3.6679	1.4462
MAR(7)/Scale = 3	5.5673	1.5180
AR(27)	4.8247	1.7223
GRNN*w* (28-1)[a]	7.0936	1.9498
GRNN (20-1)	12.4620	2.6853

[a]Input neurons are being fed by wavelet transformed data.

Table 18
One-hour ahead hourly load forecasting results for a day of December 2002

Forecasted day of December 31, 2002 (summer load)		
Model	Max APE (%)	MAPE(%)
ERN*w* (28-10-1)[a]	1.7766	0.8697
MAR(7)/Scale = 3	4.1551	1.2867
MLP*w* (28-10-1)[a]	4.9693	1.4381
AR(27)	5.7725	1.4960
MLP (20-10-1)	8.3699	1.5680
GRNN*w* (28-1)[a]	9.0740	2.7500
GRNN (20-1)	7.6383	3.5047

[a]Input neurons are being fed by wavelet transformed data.

Table 19
One-hour ahead hourly load forecasting results for a day of January 2002

Forecasted day of January 1, 2002 (summer load)		
Model	Max APE (%)	MAPE(%)
ERN*w* (28-10-1)[a]	1.7558	0.9245
MAR (7)/Scale = 3	5.4161	1.8235
MLP (20-10-1)	4.1662	1.8230
MLP*w* (28-10-1)[a]	4.5350	1.8912
AR(27)	5.7911	2.1467
GRNN*w* (28-1)[a]	10.1240	2.8003
GRNN (20-1)	7.3357	3.5240

[a]Input neurons are being fed by wavelet transformed data.

this constitutes the core of our new results presented here—a hybrid of MAR with ERN and MLP, denoted as ERN*w*, and MLP*w*, respectively, gave even better forecasting results. For these neuro-wavelet approaches, the MLP and ERN inputs were those same wavelet coefficients as used by MAR. We also combined MAR with GRNN, denoted as GRNN*w*, and the results improved over the GRNN fed by the original data values.

The experiments show that ERN*w* outperforms all of them, and MLP*w* and MAR alternatively outperform each other, but both outperformed the linear AR, nonlinear MLP, GRNN*w* and GRNN. On the other hand, the experiments also show that GRNN*w* outperformed the normal GRNN.

Although the experimental results have shown that the wavelet-based nonlinear models, ERN*w* and MLP*w*, were mostly performing better than the wavelet-based multi-resolution linear model (MAR), they are very much slower to train, especially when trained with a relatively huge data set. So the MAR model should be used first due to its

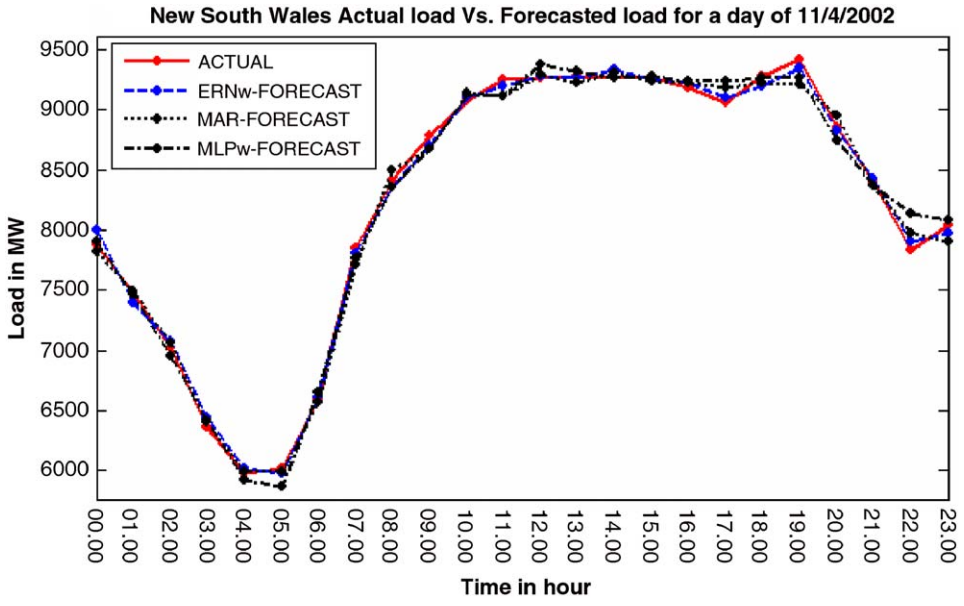


Fig. 8. Example of daily electricity load forecasting for 1 day in April 2002.

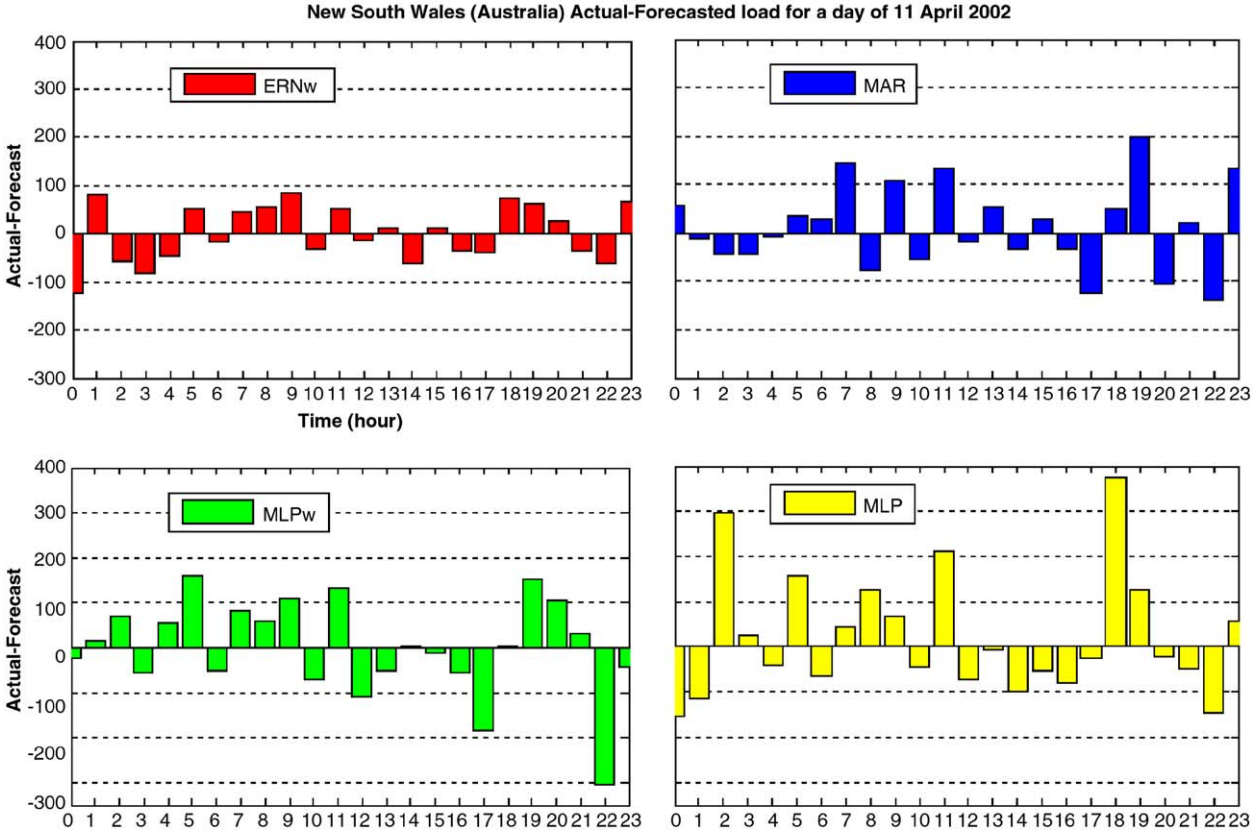


Fig. 9. The difference between the actual and forecasted load by ERN*w*, MAR, MLP*w*, and MLP, respectively for a day of April 11, 2002.

simplicity, and computational efficiency, but ERN*w*, and MLP*w* may be useful to add further accuracy in the prediction results. Consequently all of MAR, MLP*w*, and ERN*w* models are worthy of consideration for use in practice. This points to the interest of a system such as that described in Section 4 above, to avail of these three

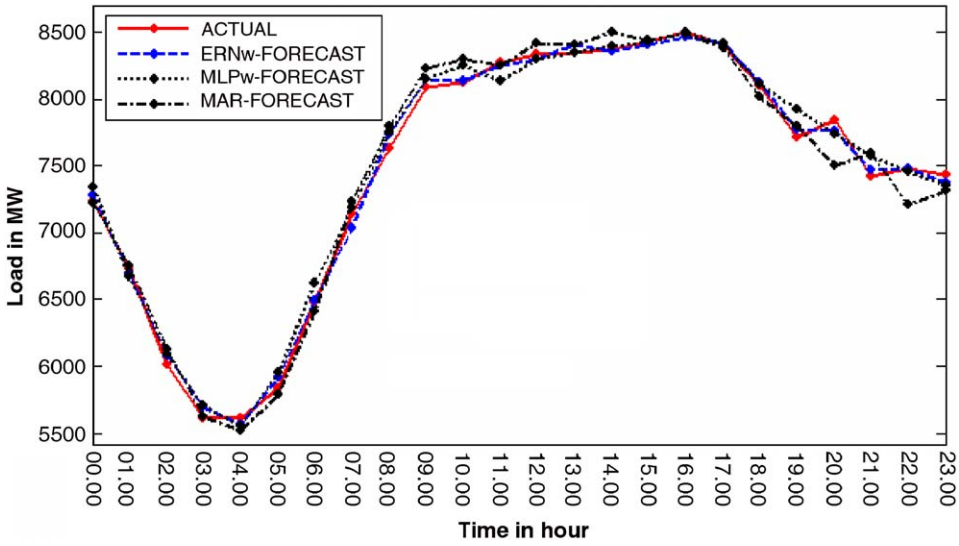


Fig. 10. Example of daily electricity load forecasting for 1 day, January 3, 2002.

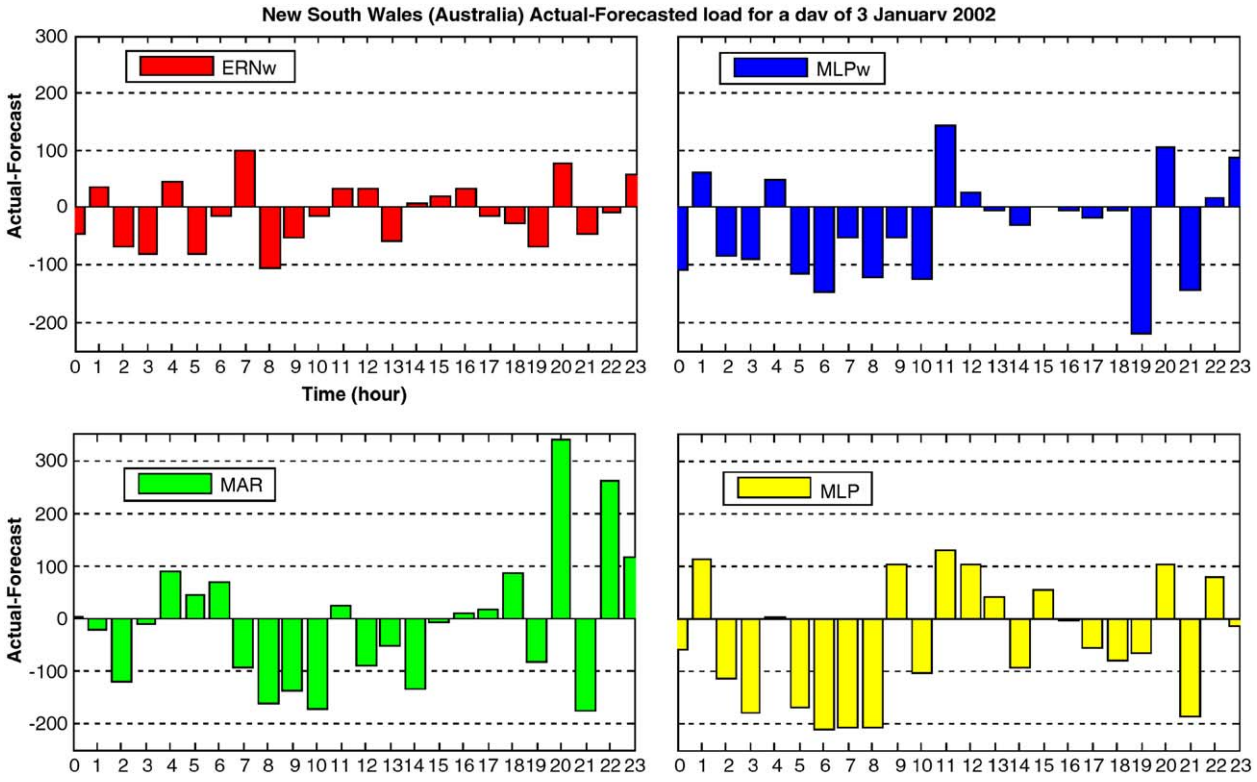


Fig. 11. The difference between the actual and forecasted load by ERN*w*, MLP*w*, MAR, and MLP, respectively for a day of January 3, 2002.

different methods, with effectiveness/efficiency trade-offs, to get an accurate load forecast.

In the near future, we intend to use these three methods on the prediction of the daily and weekly peak load due to its importance in the electricity generation scheduling. In fact, the accuracy of peak load forecast is crucial to any electric utility, since it is important to determine the number and type of generator (coal, gas, hydro) to dispatch. The peak load information will be part of the input of the optimal dispatch program for daily generation scheduling. If the input to this program is inaccurate it could cost the utility thousands of dollars as they would

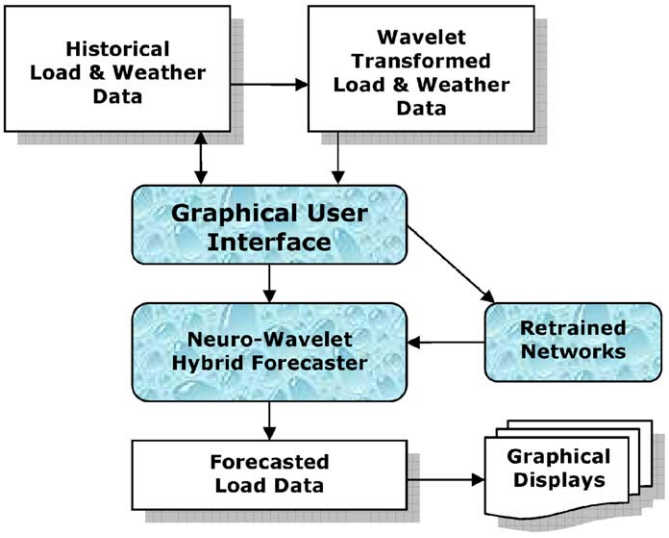


Fig. 12. Overall process flow of the computer system to derive load forecast from historical load data.

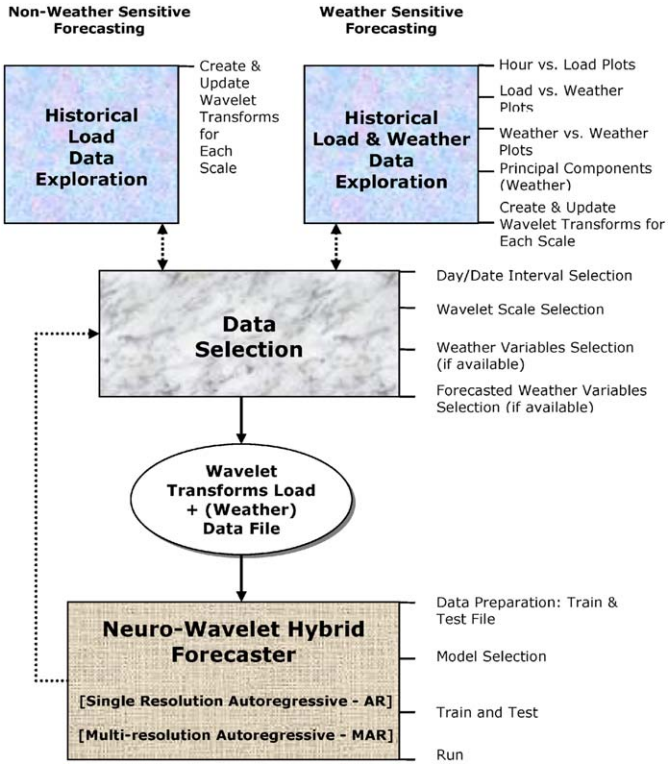


Fig. 13. Functional process flow of the way historical load data is explored, selected, and forecasted in our model.

have to reconfigure the system dispatch manually in real time. The only issue with the peak load forecast is the amount of historical data available, since the methods above require a reasonable sample of daily/weekly/monthly/annual peak load data for training that need to be collected over the years, at least with 10–15 years load data Figs. 12 and 13.

Acknowledgment

We are very grateful to NEMMCO for the electricity load and price data.

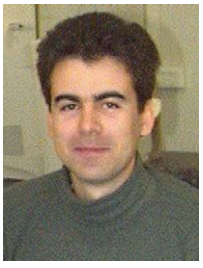
References

[1] A. Aussem, F. Murtagh, Combining neural network forecasts on wavelet-transformed time-series, Connect. Sci. 9 (1997) 113–121.
[2] A. Aussem, F. Murtagh, A neuro-wavelet strategy for web traffic forecasting, J. Off. Stat. 1 (1998) 65–87.
[3] A. Aussem, F. Murtagh, Web traffic demand forecasting using wavelet-based multiscale decomposition, Int. J. Comput. Intell. Syst. 16 (2001) 215–236.
[4] A. Aussem, J. Campbell, F. Murtagh, Wavelet-based feature extraction and decomposition strategies for financial forecasting, Int. J. Comput. Intell. Finance 6 (1998) 5–12.
[5] Z. Bachir, M.E. El-Hawary, Short term load forecasting by using wavelet neural networks, Can. Conf. Electr. Comput. Eng. (2000) 163–166.
[6] P.J. Brockwell, R.A. Davis, Time Series: Theory and Methods, Springer, Berlin, 1991.
[7] D.W. Bunn, Forecasting loads and prices in competitive power markets, Proc. IEE 88 (2) (2000) 163–169.
[8] D.W. Bunn, E.D. Farmer, Comparative Models for Electrical Load Forecasting, Wiley, New York, 1985.
[9] S.T. Chen, D.C. Yu, A.R. Moghaddamjo, Weather sensitive short-term load forecasting using non-fully connected artificial neural network, IEEE Trans. Power Syst. 7 (3) (1992) 1098–1105.
[10] T.W.S. Chow, C.T. Leung, Neural networks based short term load forecasting using weather compensation, Proc. IEEE (1996) 1736–1742.
[11] R. Cristi, M. Tummula, Multirate, multiresolution, recursive Kalman filter, Signal Process. 80 (2000) 1945–1958.
[12] K. Daoudi, A.B. Frakt, A.S. Willsky, Multiscale autoregressive models and wavelets, IEEE Trans. Inform. Theory 15 (1999) 828–845.
[13] X.Y. Dong, X. Li, Z. Xu, K.L. Teo, Weather dependent electricity market forecasting with neural networks, wavelet and data mining techniques, Proc. of AUPEC '03, No. 72, Christchurch, New Zealand, 2003.
[14] J.L. Elman, Finding structure in time, Cogn. Sci. 14 (1990) 179–211.
[15] D.B. Fogel, L.J. Fogel, V.W. Porto, Evolutionary methods for training neural networks, Proceedings of the Conference on Neural Networks for Ocean Engineering, Washington, DC, 1991, pp. 317–328.
[16] N.J. Francisco, C. Javier, et al., Forecasting next-day electricity prices by times series models, IEEE Trans. Power Syst. 17 (12) (2002) 342–348.
[17] G. Gross, R. Galiana, Short-term load forecasting, Proc. IEEE 75 (12) (1987) 1558–1572.
[18] M.T. Hagan, S.M. Behr, The time series approach to short-term load forecasting, IEEE Trans. Power Syst. 2 (3) (1987) 785–791.
[19] A.C. Harvey, Forecasting, Structural Time Series Models, and the Kalman Filter, Cambridge University Press, Cambridge, 1990.
[20] S. Haykin, Neural Networks: A Comprehensive Foundation, second ed., Prentice-Hall, Englewood Cliffs, NJ, 1999.
[21] H.B. Hipert, C.E. Pedreira, et al., Neural-networks for short term load forecasting: a review and evaluation, IEEE Trans. Power Syst. 16 (2001) 44–55.
[22] K.-L. Ho, Y. Yih, C.-C. Yang, Short term load forecasting using a multi layer neural network with adaptive learning algorithm, Proc. IEEE (1992) 141–149.
[23] C.-M. Huang, H.-T. Yang, Evolving wavelet-based networks for short-term load forecasting, IEE Proc. Gener. Transm. Distrib. 148 (3) (2001) 222–228.

[24] V. Iyer, C.C. Fung, T. Gedeon, A fuzzy-neural approach to electricity load and spot-price forecasting in a deregulated electricity market, TENCON '3, Conference on Convergent Technologies for Asia-Pacific Region, 4 (2003) 1479–1482.
[25] B.S. Kermanshahi, C.H. Poskar, G. Swift, W. Buhr, A. Silik, Artificial neural network for forecasting daily loads of a Canadian electric utility, Proc. IEEE (1993) 302–307.
[26] A. Khotanzad, R. Afkhami-Rohani, T.-L. Lu, A. Abaye, M. Davis, D.J. Maratukulam, ANNSTLF-A neural-network-based electric load forecasting system, IEEE Trans. Neural Networks 8 (4) (1997) 835–846.
[27] A. Khotanzad, R.-C. Hwang, A. Abaye, D.J. Maratukulam, An adaptive modular artificial neural network hourly load forecaster and its implementation at electric utilities, IEEE Trans. Power Syst. 10 (3) (1995) 1716–1722.
[28] A. Khotanzad, R. Afkhami-Rohani, D.J. Maratukulam, ANNSTLF-artificial neural network short-term load forecaster-generation three, IEEE Trans. Power Syst. 13 (4) (1998) 1422–1423.
[29] T. Masters, Neural Novel and Hybrid Algorithms for Time Series Prediction, Wiley, New York, 1995.
[30] I. Moghram, R. Saifur, An expert system based algorithm for short term load forecasting, IEEE Trans. Power Syst. 3 (2) (1988) 392–399.
[31] I. Moghram, R. Saifur, Analysis and evaluation of five short-term load forecasting techniques, IEEE Trans. Power Syst. 4 (4) (1989) 1484–1491.
[32] F. Murtagh, J.L. Starck, O. Renaud, On neuro-wavelet modeling, Decis. Supp. Syst. 37 (2004) 475–484.
[33] A. Oonsivilai, M.E. El-Hawary, Wavelet neural network based short-term load forecasting of electric power system commercial load, Proc. IEEE Can. Conf. Electr. Comput. Eng. (1999) 1223–1228.
[34] S.E. Papadakis, J.B. Theocharis, S.J. Kiartzis, A.G. Bakirtzis, A novel approach to short-term load forecasting using fuzzy neural networks, IEEE Trans. Power Syst. 13 (2) (1998) 480–492.
[35] A.D. Papalexopoulos, T.C. Hesterberg, A regression-based approach to short-term system load forecasting, IEEE Trans. Power Syst. 5 (4) (1990) 1535–1547.
[36] D.C. Park, M.A. El-Sharkawi, R.J. Marks, Electricity load forecasting using an artificial neural network, IEEE Trans. Power Syst. 6 (2) (1991) 442–449.
[37] O. Renaud, J.L. Starck, F. Murtagh, Prediction based on a multiscale decomposition, Int. J. Wavelets, Multiresolution Inform. Process. 1 (2) (2003) 217–232.
[38] O. Renaud, J.L. Starck, F. Murtagh, Wavelet-based combined signal filtering and prediction, IEEE Trans. Syst., Man, Cybernet 35 (6) (2005) 1241–1251.
[39] T. Senjyu, H. Takara, K. Uezata, T. Funabashi, One-hour-ahead load forecasting using neural network, IEEE Trans. Power Syst. 17 (1) (2002) 113–118.
[40] M.J. Shensa, Discrete wavelet transforms: wedding the à trous and Mallat algorithms, IEEE Trans. Signal Process. 10 (1992) 2461–2482.
[41] R.H. Shumway, D.S. Stoffer, Time Series Analysis and Its Applications, Springer, Berlin, 1999.
[42] S. Soltani, D. Boichu, P. Simard, S. Canu, The long-term memory prediction by multiscale decomposition, Signal Process. 80 (2000) 2195–2205.
[43] D.F. Specht, A general regression neural network, IEEE Trans. Neural Networks 2 (2) (1991) 568–576.
[44] J.L. Starck, F. Murtagh, Astronomical Image and Data Analysis, Springer, Berlin, 2002.
[45] J.L. Starck, F. Murtagh, A. Bijaoui, Image and Data Analysis: The Multiscale Approach, Cambridge University Press, Cambridge, 1998.
[46] R.L. Sullivan, Power System Planning, McGraw-Hill, New York, 1977.
[47] M. Vetterli, J. Kovacevic, Wavelets and Subband Coding, Prentice-Hall, Englewood Cliffs, NJ, 1995.
[48] Q. Zhang, A. Benveniste, Wavelet networks, IEEE Trans. Neural Networks 3 (1992) 889–898.
[49] B.L. Zhang, Z.Y. Dong, An adaptive neural-wavelet for short term load forecasting, Int. J. Electr. Power Syst. Res. 59 (2001) 121–129.
[50] G. Zheng, J.L. Starck, J.G. Campbell, F. Murtagh, Multiscale transforms for filtering financial data streams, J. Comput. Intell. Finance 7 (1999) 18–35.

Djamel Benaouda holds a Ph.D. from Université Joseph Fourier, Grenoble, France. Previous posts have included Senior Consultant in Software and Internet Technology in Chicago, and Research Scientist appointments with Ericsson Mobile Systems (Dublin, Ireland), University of Ulster (Northern Ireland, UK), University of Reading (England, UK), and Systems Simulation Ltd. in London. His research interests include intelligent systems using machine learning and wavelet methods, e-Learning, e-Commerce, and e-Business. He is a Senior Lecturer at the Department of Computer Science, College of Information Technology, Universiti Tenaga Nasional, Malaysia.

Fionn Murtagh holds a Ph.D. from Université P.&M. Curie, Paris VI, France, and Habilitation from Université L. Pasteur, Strasbourg, France. Previous posts have included Senior Scientist with the Space Science Department of the European Space Agency, and visiting appointments with the European Commission's Joint Research Centre, and the Department of Statistics, University of Washington. He is Professor of Computer Science and Head of the Computer Science Department at Royal Holloway, University of London. He is Editor-in-Chief of the Computer Journal, a Member of the Royal Irish Academy, and a Fellow of the British Computer Society.



Jean-Luc Starck has a Ph.D. from Université Nice-Sophia Antipolis and Habilitation from University Paris XI. He was a visitor at the European Southern Observatory (ESO) in 1993 and at Stanford's Statistics department in 2000 and 2004. He has been a Researcher at CEA since 1994. His research interests include image processing, multiscale methods and statistical methods in astrophysics. He is also co-author of two books entitled Image Processing and Data Analysis: the Multiscale Approach (Cambridge University Press, 1998), and Astronomical Image and Data Analysis (Springer, 2002).

Olivier Renaud received the M.Sc. degree in Applied Mathematics and the Ph.D. degree in Statistics from Ecole Polytechnique Fédérale (Swiss Institute of Technology), Lausanne, Switzerland. He is currently Maître d'Enseignement et de Recherche Sup. in Data Analysis, University of Geneva. He earned a 1-year fellowship for Carnegie-Mellon University, Pittsburgh, PA, and was also visiting scholar at Stanford University, Stanford, CA for 1 year. His research interests include non-parametric statistics, wavelet-like methods and machine learning.