On Neuro-Wavelet Modeling

F. Murtagh (1), J.L. Starck (2) and O. Renaud (3)
(1) School of Computer Science, Queen's University Belfast,
Belfast BT7 1NN, Northern Ireland, UK
(2) DAPNIA/SEI-SAP, CEA-Saclay, 91191 Gif sur Yvette, France
(3) Faculté de Psychologie et Sciences de l'Education,
40 Bd. du Pont d'Arve, 1211 Genève 4, Switzerland
Corresponding author: F. Murtagh, f.murtagh@qub.ac.uk

April 17, 2003

Abstract

We survey a number of applications of the wavelet transform in time series prediction. The Haar à trous wavelet transform is proposed as a means of handling time series data when future data is unknown. Results are exemplified on financial futures and S&P500 data. Nonlinear and linear multiresolution autoregression models are studied. Experimentally, we show that multiresolution approaches can outperform the traditional single resolution approach to modeling and prediction.

Keywords: à trous wavelet transform, Haar wavelet transform, time series forecasting, feature selection.

1 Introduction

Neural networks, as supervised machine learning methods, provide a valuable framework for the representation of relationships present in data. Nonetheless, the choice of input data is not a trivial matter when difficult noisy data is handled. Data preprocessing and data selection remain essential steps in the knowledge discovery process for real-world applications and, when correctly carried out, greatly improve the network's ability to capture valuable information. 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.

Wavelet transforms provide a useful decomposition of a signal, or time series, so that faint temporal structures can be revealed and handled by nonparametric models. They have been used effectively for image compression, noise removal, object detection, and large-scale structure analysis, among other applications [15, 16].

We use the *à trous* redundant wavelet transform throughout this paper. The original signal can be expressed as an additive combination of the wavelet coefficients at the different resolution levels. We introduced the undecimated Haar wavelet transform in [21], and this method was used also in [14]. This choice of wavelet transform was motivated by

the fact that the wavelet coefficients are calculated 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 in recent years. Much of this work has focused on periodogram or scalogram analysis of periodicities and cycles. For financial time series prediction, [15, 10, 14] discussed the use of the wavelet transform in the case where the market can be modeled by a fractional Brownian motion (fBm), a $1/f$ fractal process, which implies the persistence of correlations over large periods of time. Wavelets would appear to be very appropriate for analyzing non-stationary signals [17], and a link between wavelets and the difference operator was made in [18].

Several approaches have been proposed for time-series filtering and prediction by the wavelet transform, based on a neural network [21, 5], Kalman filtering [7], or an AR (autoregressive) model [14]. See also [8] which relates the wavelet transform to a multiscale autoregressive type of transform. Wavelet networks are neural networks (supervised mapping networks) with wavelet functions replacing the more usual sigmoid transfer functions [19, 20].

2 Wavelets for Feature Discovery

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.

A naive approach would consist of using a bank of filters with varying frequencies and widths. Unfortunately, choosing the proper filters is a difficult task: we may be tempted to feed an excessive number of inputs to the predictor model so as not to discard important features which we aim to capture, or vice-versa. The wavelet transform provides a sound mathematical principle for designing and spacing filters which provide trade-offs between these objectives, while retaining immediate relationships with the time series. These principles define 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, the outputs of which are the wavelets.

Among all wavelets proposed in the literature, Daubechies and Morlet wavelet transforms have been increasingly adopted by signal and image processing researchers. While Daubechies wavelets exhibit a good trade-off between parsimony and information richness, it was reported [9] 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 model. As already noted, our chief considerations regarding the choice of mother wavelet are (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 use of the Haar wavelet function.

3 The À 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 wavelet 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 an n-length input time series.

A major disadvantage of the decimated form of output is that we cannot simply – visually or graphically – relate information at a given time point at the different scales. With somewhat greater difficulty, however, this goal is possible. What is not possible is to have shift invariance. This means that if we had deleted the first few values of our input time series, then the output wavelet transformed, decimated, data would not be the same as heretofore. We can get around this problem at the expense of a greater storage requirement, 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 redundant, discrete wavelet transform to be described now is one used by us in [2]. The successive resolution levels are formed by convolving with an increasingly dilated wavelet function which looks rather like a Mexican sombrero (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 in fact 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 [13] allows the filter outputs to be interpreted in a meaningful way. The à trous wavelet transform can be described simply as follows. First, perform successive convolutions with the discrete low-pass filter h:

$$c_{i+1}(k) = \sum_{l=-\infty}^{+\infty} h(l) c_i(k + 2^i l) \tag{1}$$

where the finest scale is the original series: $c_0(t) = x(t)$. The increase in distances between the sampled 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 $(\frac{1}{16}, \frac{1}{4}, \frac{3}{8}, \frac{1}{4}, \frac{1}{16})$. This is of compact support (necessary for a wavelet transform), and is point-symmetric. The latter

does not allow for the fact that time is a fundamentally asymmetric variable, and we will return to this important issue shortly.

Now, from the sequence of smoothings 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.

Clearly, in prediction studies, very careful attention must be given to the boundary. We took an intuitively acceptable approach by taking (for a time series of size n) $c(n+k) = c(n-k)$.

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, the computational complexity of the above algorithm is $O(n)$ for an n-length input. It therefore has more favorable computational cost than the FFT (fast Fourier transform).

If the background or residual vector c_p is sufficiently smooth, its one-step ahead prediction may become trivial, e.g. by a linear approximation or, even more simply, a carbon copy ($x(t-1) \longrightarrow x(t)$) of it.

We now consider the issue of prediction. We can think of the 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 lookahead period.

If the resolution scales, w_i and c_p, are physically interpretable (as they may possibly be) then the independent predictions are similarly open to interpretation. The fact that the reconstruction (Eqn. 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.

Another idea is to use Eqn. 3 for defining features. This is to take the feature vector at time point t as $\{w_1(t), w_2(t), \ldots, w_p(t), c_p(t)\}$. For time-varying data, we will use this approach in a novel way. Say that we are considering the situation up to time-point $t = t^0$. Remember that we are using wrap-around to define the wavelet transform at this problematic "boundary" of our data. Then we use $x(t^0)$ as our feature vector.

4 The Haar À Trous Wavelet Transform

The à trous wavelet transform with a wavelet function related to a 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 [21].

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, but with a different pair of scaling and wavelet functions compared to those used previously.

The non-decimated Haar algorithm is exactly the same as the à trous algorithm, except that the low-pass filter h, $(\frac{1}{16}, \frac{1}{4}, \frac{3}{8}, \frac{1}{4}, \frac{1}{16})$, 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 by convolving the latter with h. Then:

$$c_1(k) = \frac{1}{2}(c_0(k) + c_0(k-1))$$

and

$$w_1(k) = c_0(k) - c_1(k)$$

More generally,

$$c_j(k) = \frac{1}{2}(c_{j-1}(k) + c_{j-1}(k-2^j))$$

$$w_j(k) = c_j(k) - c_{j-1}(k)$$

At any time point, k, we never use information 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 time point t (obviously we do not care about time point 1, since this is far back in the time series). Experimental results with this redundant transform can be found in [21, 12].

Figure 2 shows the transform on a set of financial futures, which will be further studied below. The elementwise sum of scales 1 through 8, plus the smooth trend, gives the original data set. We do not show the original data set because it simply looks like a noisy version of the smooth trend plot. Note the following: (i) all wavelet scales are of zero mean; and (ii) the smooth trend plot is very often much larger-valued (as is the case here) compared to the max-min ranges of the wavelet coefficients.

Figure 3 shows which pixels 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.

5 Choice of Wavelet Features

Aussem et al. [2] considered two types of feature resulting from the use of a redundant à trous wavelet transform. The objective was to make five-day ahead estimates of the S&P500 index.

In addition to the à trous wavelet transform, requiring knowledge of future values, a time-varying version was used which is limited to data values up to time point t. The handling of the data boundary at time point t becomes crucially important under these circumstances. Discussion of this algorithm can be found in [21]. We do not recommend it, and so we will not further discuss it here.

The two types of feature considered in [2] were as follows:

1. Decomposition-based approach: Wavelet coefficients at a particular time point were taken as a feature vector.

2. Scale-based approach: Modeling and prediction were run independently at each resolution level, and the results were combined.

We recall, for the decomposition-based approach, that the à trous redundant transform is an additive decomposition, which is non-orthogonal. In [2] and also in [3], the wavelet coefficients were made robust by capping the extreme values, and they were also linearly rescaled into the unit interval. The final smooth trend component from the wavelet transform was not used. Among performance criteria used were normalized mean squared error (NMSE), and direction variation symmetry (DVS), the latter being the percentage of correctly predicted direction variations with respect to the target variable. We found that the decomposition-based approach performed somewhat worse on NMSE, and a good deal better on DVS, relative to a window or time-lagged vector alternative. The latter, based only on the target variable was the following mapping: $y_{t-5}, y_{t-4}, \ldots, y_t \longrightarrow y_{\text{target}}$. Two criticisms can be leveled at this outcome: (i) the wavelet transform used does not take account of the time-varying nature of the data; and (ii) the decomposition based on resolution scale is acceptable if inherent, natural scales are found in reality, but if not then other decompositions, in particular an orthogonal or whitening one, may be more worthwhile.

In the scale-based approach, we found that the wavelet coefficients at higher frequency levels (i.e., lower scales) provided some benefit for estimating variation at less high frequency levels. Table 1 summarizes what we did, and the results obtained. DRNN is the dynamic recurrent neural network model used. The DRNN model is endowed with internal memory so that additional information on the past time series is used. The architecture is shown in Figure 4. The memory order of this network is equivalent to applying a time-lagged vector of the same size as the memory order. Hence the window in Table 1 is the equivalent lagged vector length.

In Table 1, NMSE is normalized mean squared error, DVS is direction variation symmetry (see above), and DS is directional symmetry, i.e. the percentage of correctly predicted directions with respect to the target variable. Once the univariate predictions were obtained for each series, the S&P500 target estimate was calculated as a simple sum, given the additive decomposition involved in the wavelet transform used.

Consider the modeling and prediction at scale 2. We determine the target w_2 value based on (de facto in the DRNN): $w_2(t-15), w_2(t-14), w_2(t-13), \ldots w_2(t)$ combined into one input vector with $w_1(t-15), w_1(t-14), w_1(t-13), \ldots w_1(t)$. The curve of w_2 is smoother than w_1 and therefore there is less information for the network to retrieve from w_2 when predicting w_1. On the other hand, as was done by us, use of w_1 for the prediction of w_2 *is* of benefit, since the more noisy and irregular the data the more demanding the prediction task and the more useful the neural network. The same reasoning applies to a trivial carbon copy estimate, i.e. target $= x(t)$, which performs better and better as the scale is increased, i.e. as the data becomes smoother. On the final smooth trend curve, resid(t) in Table 1, a crude linear extrapolation estimate, i.e. target $= x(t) + 5(x(t) - x(t-1))$, was found to be even more favorable than the neural network solution. This linear estimate was retained in place of the neural network estimate.

When all wavelet estimates were recombined, the resulting NMSE value was 0.72, the DS was 73% and the DSV 60%. These results are significantly better than the direct

or feature vector approach in terms of MSE and sign prediction. Further discussion of these results can be found in [2]. For further work involving the DRNN neural network architecture and wavelet processed data, see [1, 4].

In the work described, using S&P500 data, we have used a wavelet transform decomposition (i) for feature extraction, and (ii) as a powerful problem decomposition methodology. The latter gave the best results. We will now look further at the selection of wavelet coefficients from the sequence of resolution levels.

6 Parsimonious Features from Haar À Trous Transform

In the last section, we summarized work which used windows of time-lagged vectors at each wavelet resolution scale. From Table 1, we saw how these windows were of effective length 10, 15, 20, 25 for wavelet scales 1, 2, 3 and 4. The rationale behind these window lengths was relatively ad hoc, based on the knowledge that longer window lengths were needed by higher scales. Figure 3 is illustrative of our real needs, however. It is seen that wavelet scales 1, 2, 3, and 4 in reality require consideration of window lengths 2, 4, 8 and 16. Given the redundancy inherent in the Haar à trous transform, the mapping scheme illustrated in Figure 5 is both adequate and parsimonious in terms of inputs to be considered. Figure 5 illustrates the order 2 case.

We next study how we can allow for adaptivity in the numbers of wavelet coefficients selected from different resolution scales and then used in the predictions. Let the window size at scale j be denoted A_j. If the original signal shows high frequency variation, then we take A_j to be larger for small j, and to be 1 or 0 for larger j. We adopt a converse approach if the signal has $1/f$ behavior.

Assume a signal $X = (X_1, \ldots, X_n)$. A one-step forward prediction of a linear autoregressive or AR(p) process is written $\hat{X}_{n+1} = \sum_{k=1}^{p} \hat{\phi}_k X_{n-(k-1)}$. In order to use a wavelet decomposition, we consider AR multiscale prediction [12]:

$$\hat{X}_{n+1} = \sum_{j=1}^{J} \sum_{k=1}^{A_j} \hat{a}_{j,k} w_{j,n-2^j(k-1)} + \sum_{k=1}^{A_{J+1}} \hat{a}_{J+1,k} c_{J,n-2^J(k-1)} \tag{4}$$

where $\mathcal{W} = w_1, \ldots, w_J, c_J$ represents the Haar à trous wavelet transform of X ($X = \sum_{j=1}^{J} w_j + c_J$). For example, choosing $A_j = 1$ for all resolution levels, j, leads to the prediction

$$\hat{X}_{n+1} = \sum_{j=1}^{J} \hat{a}_j w_{j,n} + \hat{a}_{J+1} c_{J,n}. \tag{5}$$

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

To further link this method with a prediction based on a regular AR, note that if on each scale the lagged coefficients follow an AR(A_j) model, the addition of the predictions on each level would lead to the same prediction formula (4).

Generalizing formula (4) to the nonlinear case leads to the learning of a mapping of the form presented in mapping formula (6). We assume a standard multilayer perceptron with a linear transfer function at the output node, and L hidden layer neurons.

$$\hat{X}_{N+1} = \sum_{l=1}^{L} \hat{a}_l g \left(\sum_{j=1}^{J} \sum_{k=1}^{A_j} \hat{a}_{j,k} w_{j,N-2^j(k-1)} + \sum_{k=1}^{A_{J+1}} \hat{a}_{J+1,k} c_{J,N-2^J(k-1)} \right) \tag{6}$$

where (composition of) nonlinear function g is used (e.g. sigmoidal in a feedforward multilayer perceptron).

7 Scale-Dependent Numbers of Wavelet Coefficients

We used a set of financial futures in order to exemplify results. We used approximately half as a training set, and the remainder as a hold-out test set.

We consider the following alternatives:

1. Multiresolution autoregression, using a range of model orders. MAR(1), with reference to Figure 5, used the 5 components of the Haar à trous transform at time step t in order to predict the value at time step $t+1$. A linear model of order 2, as represented in Figure 5, is denoted MAR(2).

2. A linear autoregression model again with a range of model orders. A model of order 4 is denoted AR(4), i.e. $\{x(t-4), x(t-3), x(t-2), x(t-1)\} \longrightarrow x(t)$. A nonlinear autoregression model of order 4 is denoted NAR(4).

No data rescaling or standardization was carried out. Use of classical and well-understood function learning methods was motivated by our objectives of: (i) Clearly separating the use of a wavelet transform for feature finding from the linear or nonlinear function mapping, in order to focus on the properties of the wavelet features; and (ii) Targeting a generic modeling, and analysis (interpolation or extrapolation), pipeline, which is robust and scalable.

The data set consisted of 6160 daily highs of a set of financial futures dating from a period up to the late 1990s. Linear regression offers the following properties:
(i) Very straightforward in regard to rescaling or normalization of input and output: regression weights are determined automatically.
(ii) Deterministic, and no dependence on initialization.
(iii) The inverse wavelet transform is a linear operation. This points towards acceptability of linear regression as opposed to the need for a nonlinear mapping method (neural network) in order to demonstrate the capability of the MAR model.
(iv) Downweighting of less relevant independent variables is very clear.

The training set was the first 3080 observations. Based on the use of all remaining 3080 observations as the test set, we examined a range of model orders in order to select

MAR(2) and AR(3) for further study. MAR(2) (linear multiresolution autoregression, cf. Figure 5) implies 10 independent variables if we assume 4 resolution scales, and AR(3) implies a sliding window of 3 independent variables. MSE (mean squared error) results obtained are shown in Table 2. Figure 1 illustrates the fit between predicted and target in the case of the MAR(2) model.

For small number of predicted values, i.e. limited extrapolation into the future, we see that the MAR model outscores the AR model.

8 Preprocessing the Signal by Filtering

We have not discussed noise filtering of the input data signal in this article. Options which can be considered here are as follows.

1. Data smoothing by means of a moving average leads to linear or nonlinear ARMA modeling.

2. Mentioned in [2] is the trivial omission of mediocre forecasts on the more high frequency (or lowest resolution) scales of a wavelet transform.

3. Described in [21] was the use of an entropy principle to optimally trade-off signal entropy and what may be taken as noise entropy, in the context of additive Gaussian noise. Other noise models are catered for e.g. by using an Anscombe variance stabilizing transform in the case of Poisson noise, or combined low valued Poisson and Gaussian, or by assuming locally Gaussian properties in the case of non-stationary noise [11].

4. In further work we intend to develop an adaptive noise-removing filter, which respects the time-varying nature of the data by being based on the Haar à trous wavelet transform, and which will cater well for regimes of non-stationarity and varying volatility.

9 Conclusions

Based on S&P500 and futures data, we have examined a new wavelet transform. Its disadvantages – asymmetry, lack of smoothness properties – have been taken into consideration as well as its advantages – most importantly, respect for temporal data.

Our approach to modeling and prediction has been based on pattern finding in the time series data. We examined how wavelet coefficients could provide useful features.

In doing so, we saw how varying numbers of wavelet coefficients were selected at different resolution scales. Therefore we studied a direct selection of different numbers of coefficients at different scales.

Our results outperformed other methods. In near future work, we intend to further exploit the selective use of wavelet coefficients at different resolution scales.

Acknowledgements

We are grateful to M. Savage for the futures data.

References

[1] A. Aussem and F. Murtagh, Combining neural network forecasts on wavelet-transformed time series, Connection Science 9, 113–121, 1997.

[2] A. Aussem, J. Campbell and F. Murtagh, Wavelet-based feature extraction and decomposition strategies for financial forecasting, Journal of Computational Intelligence in Finance 6, 5–12, 1998.

[3] A. Aussem and F. Murtagh, A neuro-wavelet strategy for Web traffic forecasting, Journal of Official Statistics 1, 65–87, 1998.

[4] A. Aussem and F. Murtagh, Web traffic demand forecasting using wavelet-based multiscale decomposition, International Journal of Intelligent Systems 16, 215–236, 2001.

[5] Z. Bashir and M.E. El-Hawary, Short term load forecasting by using wavelet neural networks, Canadian Conference on Electrical and Computer Engineering, 163–166, 2000.

[6] V. Bjorn, Multiresolution methods for financial time series prediction, Proceedings of the IEEE/IAFE 1995 Conference on Computational Intelligence for Financial Engineering, 97, 1995.

[7] R. Cristi and M. Tummula, Multirate, multiresolution, recursive Kalman filter, Signal Processing 80, 1945–1958, 2000.

[8] K. Daoudi, A.B. Frakt and A.S. Willsky, Multiscale autoregressive models and wavelets, IEEE Transactions on Information Theory 45, 828–845, 1999.

[9] T. Masters, Neural, Novel and Hybrid Algorithms for Time Series Prediction, Wiley, New York, 1995.

[10] J. Moody and W. Lizhong, What is the 'true price'? State space models for high frequency FX data, Proc. IEEE/IAFE 1997 Conference on Computational Intelligence for Financial Engineering (CIFEr), 150–156, 1997.

[11] MR, MR/1, MR/2, MR/3 and MR/Finance, Multiresolution Software Environment, www.multiresolution.com, 1998.

[12] O. Renaud, J.L. Starck and F. Murtagh, Prediction based on a multiscale decomposition, report, 2002.

[13] M.J. Shensa, Discrete wavelet transforms: wedding the à trous and Mallat algorithms, IEEE Transactions on Signal Processing 40, 2464–2482, 1992.

[14] S. Soltani, D. Boichu, P. Simard and S. Canu, The long-term memory prediction by multiscale decomposition, Signal Processing 80, 2195–2205, 2000.

[15] J.L. Starck, F. Murtagh and A. Bijaoui, Image and Data Analysis: The Multiscale Approach, Cambridge University Press, 1998.

[16] J.L. Starck and F. Murtagh, Astronomical Image and Data Analysis, Springer-Verlag, 2002.

[17] E.G.T. Swee and S. Elangovan, S., Applications of symmlets for denoising and load forecasting, Proceedings of the IEEE Signal Processing Workshop on Higher-Order Statistics, 165–169, 1999.

[18] K. Xizheng, J. Licheng, Y. Tinggao and W. Zhensen, Wavelet model for the time scale, in: Proceedings of the 1999 Joint Meeting of the European Frequency and Time Forum, 1999 and the IEEE International Frequency Control Symposium, 1999, Vol. 1, 177–181, 1999.

[19] Q. Zhang and A. Benveniste, Wavelet networks, IEEE Transactions on Neural Networks 3, 889–898, 1992.

[20] Q. Zhang, Using wavelet network in nonparametric estimation, IEEE Transactions on Neural Networks 8, 227–236, 1997.

[21] G. Zheng, J.L. Starck, J.G. Campbell and F. Murtagh, Multiscale transforms for filtering financial data streams, Journal of Computational Intelligence in Finance 7, 18-35, 1999.

Short biographies

Fionn Murtagh holds BA and BAI degrees in mathematics and engineering science, and an MSc in computer science, all from Trinity College Dublin, Ireland, a PhD in mathematical statistics from Université P. & M. Curie, Paris VI, France, and an 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 at Queen's University Belfast. 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 University Nice-Sophia Antipolis and an 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. 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 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 one-year fellowship for Carnegie-Mellon University, Pittsburgh, PA, and was also visiting scholar at Stanford University, Stanford, CA for a year. His research interests include non-parametric statistics, wavelet-like methods and machine learning.

Table Captions

- Table 1: Neural network architectures. From left to right are shown: the network input vector, output forecast, architecture, memory order of the input-to-hidden connections (or window size), out-of-sample DS, DVS and NMSE. From [2].
- Table 2: Training set: series of 3080 future highs. Number of predicted values from hold-out test set: $1, 2, \ldots 3079$. AR(3) and MAR(2) models used. Values are mean square errors.

Figure Captions

- Figure 1: Financial futures, 6160 successive daily highs. Right: forecast minus actual for the second half of the data set.
- Figure 2: Haar à trous wavelet transform of a set of 6160 financial futures. Daily highs are used.
- Figure 3: Pixels of the input signal which are used to calculate the last wavelet coefficient on the different scales.
- Figure 4: Example of DRNN architecture 1–5–1. Input links have varying delays, which can be expressed as an input window size. The output provides the forecast. The hidden units, here 5 units, were fully connected with connections of memory order 1.
- Figure 5: Wavelet coefficients that are used for the prediction of the next value.

	Input	Output	Size	Window	DS	DVS	NMSE
DRNN 1	$w_1(t)$	$w_1(t+5)$	1:6:1	10	46%	55 %	1.10
DRNN 2	$w_1(t)$ $w_2(t)$	$w_2(t+5)$	2:6:1	15	58%	53%	1.03
DRNN 3	$w_1(t)$ $w_2(t)$ $w_3(t)$	$w_3(t+5)$	3:5:1	20	66%	47 %	0.87
DRNN 4	$w_1(t)$ $w_2(t)$ $w_3(t)$ $w_4(t)$	$w_4(t+5)$	4:5:1	25	86%	67%	0.40
DRNN 5	resid(t)	resid(t+5)	1:4:1	30	–	83 %	$2 * 10^{-3}$

Table 1: Neural network architectures. From left to right are shown: the network input vector, output forecast, architecture, memory order of the input-to-hidden connections (or window size), out-of-sample DS, DVS and NMSE. From [2].

NPred	AR(3)	MAR(2)
1	4856	4855
2	909	882
3	730	713
4	551	536
5	458	443
6	389	378
7	334	325
8	307	302
9	273	270
10	247	244
11	234	233
3079	193.1	193.3

Table 2: Training set: series of 3080 future highs. Number of predicted values from hold-out test set: $1, 2, \ldots 3079$. AR(3) and MAR(2) models used. Values are mean square errors.

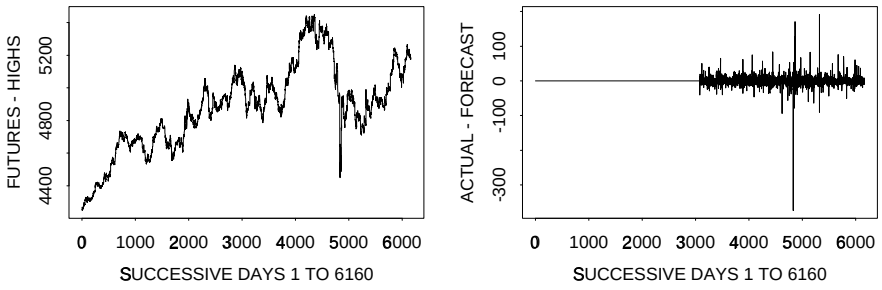

Figure 1: Financial futures, 6160 successive daily highs. Right: forecast minus actual for the second half of the data set.

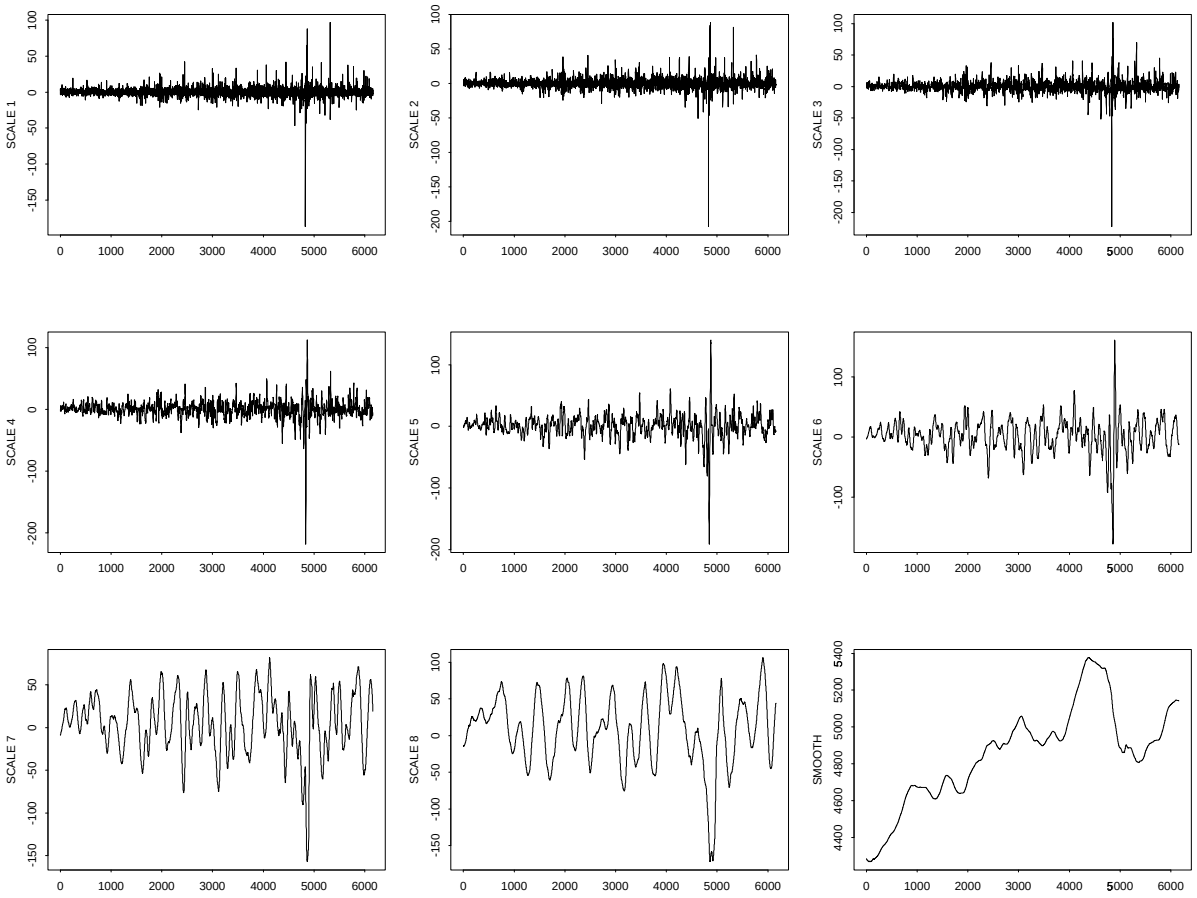

Figure 2: Haar à trous wavelet transform of a set of 6160 financial futures. Daily highs are used.

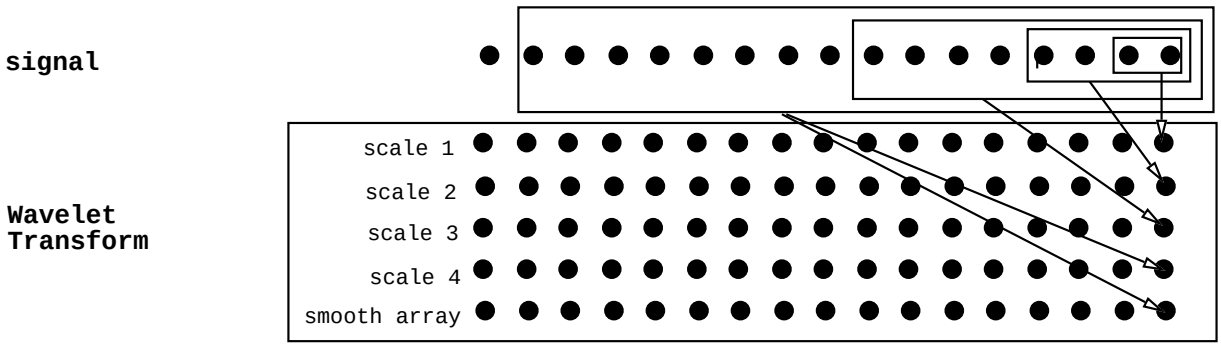

Figure 3: Pixels of the input signal which are used to calculate the last wavelet coefficient on the different scales.

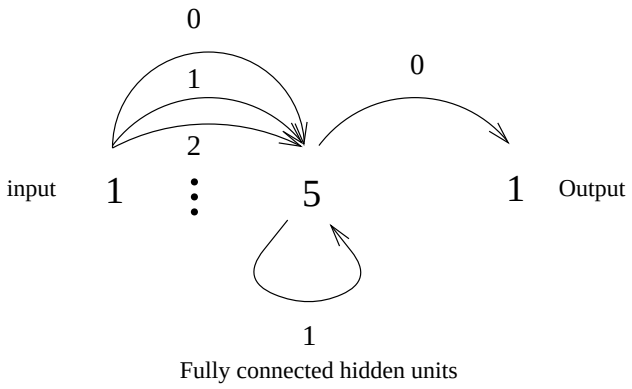

Figure 4: Example of DRNN architecture 1–5–1. Input links have varying delays, which can be expressed as an input window size. The output provides the forecast. The hidden units, here 5 units, were fully connected with connections of memory order 1.

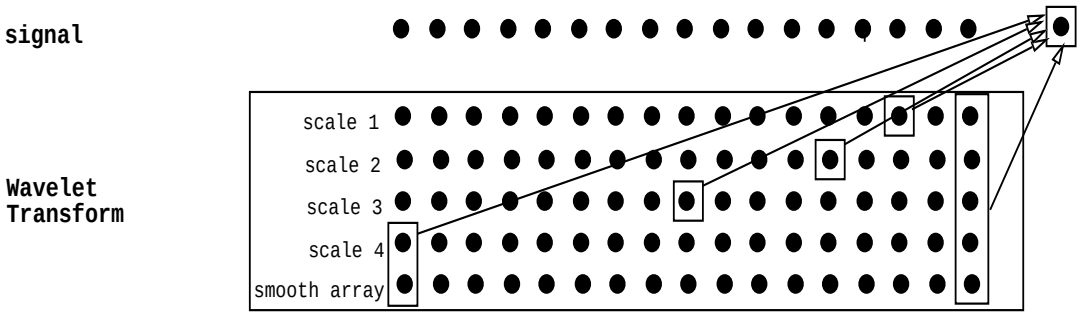

Figure 5: Wavelet coefficients that are used for the prediction of the next value.