Multiresolution Forecasting for Futures Trading Using Wavelet Decompositions

Bai-Ling Zhang, Richard Coggins, *Member, IEEE*, Marwan Anwar Jabri, *Senior Member, IEEE*, Dominik Dersch, and Barry Flower, *Member, IEEE*

***Abstract*—In this paper, we investigate the effectiveness of a financial time-series forecasting strategy which exploits the multiresolution property of the wavelet transform. A financial series is decomposed into an over complete, shift invariant scale-related representation. In transform space, each individual wavelet series is modeled by a separate multilayer perceptron (MLP). To better utilize the detailed information in the lower scales of wavelet coefficients (high frequencies) and general (trend) information in the higher scales of wavelet coefficients (low frequencies), we applied the Bayesian method of automatic relevance determination (ARD) to choose short past windows (short-term history) for the inputs to the MLPs at lower scales and long past windows (long-term history) at higher scales. To form the overall forecast, the individual forecasts are then recombined by the linear reconstruction property of the inverse transform with the chosen autocorrelation shell representation, or by another perceptron which learns the weight of each scale in the prediction of the original time series. The forecast results are then passed to a money management system to generate trades. Compared with previous work on combining wavelet techniques and neural networks to financial time-series, our contributions include 1) proposing a three-stage prediction scheme; 2) applying a multiresolution prediction which is strictly based on the autocorrelation shell representation, 3) incorporating the Bayesian technique ARD with MLP training for the selection of relevant inputs; and 4) using a realistic money management system and trading model to evaluate the forecasting performance. Using an accurate trading model, our system shows promising profitability performance. Results comparing the performance of the proposed architecture with an MLP without wavelet preprocessing on 10–year bond futures indicate a doubling in profit per trade ($AUD1753:$AUD819) and Sharpe ratio improvement of 0.732 versus 0.367, as well as significant improvements in the ratio of winning to loosing trades, thus indicating significant potential profitability for live trading.**

***Index Terms*—Autocorrelation shell representation, automatic relevance determination, financial time series, futures trading, multilayer perceptron, relevance determination, wavelet decomposition.**

I. Introduction

During the last two decades, various approaches have been developed for time series prediction. Among them linear regression methods such as autoregressive (AR) and autoregressive moving average (ARMA) models have been the most used methods in practice [18]. The theory of linear models is well known, and many algorithms for model building are available.

Linear models are usually inadequate for financial time series as in practice almost all economic processes are nonlinear to some extent. Nonlinear methods are widely applicable nowadays with the growth of computer processing speed and data storage. Of the nonlinear methods, neural networks have become very popular. Many different types of neural networks such as MLP and RBF have been proven to be universal function approximators, which make neural networks attractive for time series modeling, and for financial time-series forecasting in particular.

An important prerequisite for the successful application of some modern advanced modeling techniques such as neural networks, however, is a certain uniformity of the data [14]. In most cases, a stationary process is assumed for the temporally ordered data. In financial time series, such an assumption of stationarity has to be discarded. Generally speaking, there may exist different kinds of nonstationarities. For example, a process may be a superposition of many sources, where the underlying system drifts or switches between different sources, producing different dynamics. Standard approaches such as AR models or nonlinear AR models using MLPs usually give best results for stationary time series. Such a model can be termed as global as only one model is used to characterize the measured process. When a series is nonstationary, as is the case for most financial time series, identifying a proper global model becomes very difficult, unless the nature of the nonstationarity is known. In recent years, local models have grown in interest for improving the prediction accuracy for nonstationary time series [25].

To overcome the problems of monolithic global models, another efficient way is to design a hybrid scheme incorporating multiresolution decomposition techniques such as the wavelet transform, which can produce a good local representation of the signal in both the time domain and the frequency domain [13]. In contrast to the Fourier basis, wavelets can be supported on an arbitrarily small closed interval. Thus, the wavelet transform is a very powerful tool for dealing with transient phenomena.

There are many possible applications of combining wavelet transformations into financial time-series analysis and forecasting. Recently some financial forecasting strategies have been discussed that used wavelet transforms to preprocess the

Manuscript received August 1, 2000; revised February 5, 2001. This work was supported by the Australian Research Council and Crux Financial Engineering Pty. Ltd.

B.-L. Zhang and R. Coggins are with the Computer Engineering Laboratory (CEL), School of Electrical and Information Engineering, University of Sydney, NSW 2006, Australia.

M. A. Jabri is with the Computer Engineering Laboratory (CEL), School of Electrical and Information Engineering, University of Sydney, NSW 2006, Australia and also with the Electrical and Computer Engineering Department, Oregon Graduate Institute, Beaverton, OR 97006 USA.

D. Dersch and B. Flower are with the Research and Development, Crux Financial Engineering, NSW 1220, Australia.

Publisher Item Identifier S 1045-9227(01)05016-0.

data [1], [2], [19], [27]. The preprocessing methods they used are based on the translation invariant wavelet transform [7] or *à trous* wavelet transform [4], [23].

In this work, we have developed a neuro-wavelet hybrid system that incorporates multiscale wavelet analysis into a set of neural networks for a multistage time series prediction. Compared to the work in [11], our system exploits a shift invariant wavelet transform called the autocorrelation shell representation (ASR) [4] instead of the multiscale orthogonal wavelet transform as was originally presented in [13]. It is cumbersome to apply the commonly defined DWT for real-time time series applications due to the lack of shift invariance, which plays an important role in time series forecasting. Using a shift invariant wavelet transform, we can easily relate the resolution scales exactly to the original time series and preserve the integrity of some short-lived events [2].

Basically, we suggest the direct application of the *à trous* wavelet transform based on the ASR to financial time series and the prediction of each scale of the wavelet's coefficients by a separate feedforward neural network. The separate predictions of each scale are proceeded independently. The prediction results for the wavelet coefficients can be combined directly by the linear additive reconstruction property of ASR, or preferably, as we propose in this paper, by another NN in order to predict the original time series. The aim of this last network is to adaptively choose the weight of each scale in the final prediction [11]. For the prediction of different scale wavelet coefficients, we apply the Bayesian method of automatic relevance determination (ARD) [16] to learn the different significance of a specific length of past window and wavelet scale. ARD is a practical Bayesian method for selecting the best input variables, which enables us to predict each scale of wavelet coefficients by an appropriate neural network, thus simplifying the learning task as the size of each network can be quite small.

Comparing the previous work on applying wavelet techniques together with connectionist methods to financial time series in [1], [2] our contributions consist of 1) applying some three-stage prediction schemes; 2) a multiresolution prediction which is strictly based on the autocorrelation shell representation; 3) selecting relevant MLP inputs from the overcomplete shell representation using the Bayesian technique ARD; and 4) demonstrating performance using a realistic money management system and trading model.

This paper is organized as follows. In the next section, we briefly describe the wavelet transform and the autocorrelation shell representation. The principle of the Bayesian method of ARD is also introduced. Section III presents our hybrid neuro-wavelet scheme for time-series prediction and system details. The simulation results and performance comparison over different data sets using a realistic trading simulator are summarized in Section IV followed by discussions and conclusions in Section V.

II. Combining Bayesian and Wavelet Based Preprocessing

A. Discrete Wavelet Transform and Autocorrelation Shell Representation

Generally speaking, a wavelet decomposition provides a way of analysing a signal both in time and in frequency. If f is a function defined on the whole real line, then, for a suitably chosen mother wavelet function ψ, we can expand f as

$$f(t) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} w_{jk} 2^{j/2} \psi(2^j t - k) \tag{1}$$

where the function $\psi(2^j t - k)$ are all orthogonal to one another. The coefficient w_{jk} conveys information about the behavior of the function f concentrating on effects of scale around 2^{-j} near time $k \times 2^{-j}$. This wavelet decomposition of a function is closely related to a similar decomposition [the discrete wavelet transform (DWT)] of a signal observed at discrete points in time.

The DWT has the property of being very good at compressing a wide range of signals actually observed in practice—a very large proportion of the coefficients of the transform can be set to zero without appreciable loss of information, even for signals that contain occasional abrupt changes of level or other behavior. It is this ability to deal with heterogeneous and intermittent behavior that makes wavelets so attractive. Classical methods of signal processing depend on an underlying notion of stationarity, for which methods such as Fourier analysis are very well adapted.

One problem with the application of the DWT in time-series analysis is that it suffers from a lack of translation invariance. This means that statistical estimators that rely on the DWT are sensitive to the choice of origin. This problem can be tackled by means of a *redundant* or *nondecimated* wavelet transform [7], [21]. A redundant transform based on an n-length input time series has an n-length resolution scale for each of the resolution levels of interest. Hence, information at each resolution scale is directly related at each time point. To accomplish this, we use an *à trous* algorithm for realizing shift-invariant wavelet transforms, which is based on the so-called autocorrelation shell representation [21] by utilizing dilations and translations of the autocorrelation functions of compactly supported wavelets. The filters for the decomposition process are the autocorrelations of the quadrature mirror filter coefficients of the compactly supported wavelets and are symmetric.

By definition, the autocorrelation functions of a compactly supported scaling function $\phi(x)$ and the corresponding wavelet $\psi(x)$ are as follows:

$$\Phi(x) = \int_{-\infty}^{\infty} \phi(y)\phi(y-x)\,dy$$
$$\Psi(x) = \int_{-\infty}^{\infty} \psi(y)\psi(y-x)\,dy \tag{2}$$

The family of functions $\{\tilde{\Psi}_{j,k}(x)\}_{1\le j\le n_0,\, 0\le k\le N-1}$ and $\{\tilde{\Phi}_{n_0,k}(x)\}_{0\le k\le N-1}$, where $\tilde{\Psi}_{j,k}(x) = 2^{-j/2}\Psi(2^{-j}(x-k))$ and $\tilde{\Phi}_{n_0,k}(x) = 2^{-n_0/2}\Phi(2^{-n_0}(x-k))$, is called an autocorrelation shell. Then a set of filters $P = \{p_k\}_{-L+1\le k\le L-1}$ and $Q = \{q_k\}_{-L+1\le k\le L-1}$ can be defined as

$$\frac{1}{\sqrt{2}}\Phi(x/2) = \sum_{k=-L+1}^{L-1} p_k \Phi(x-k)$$
$$\frac{1}{\sqrt{2}}\Psi(x/2) = \sum_{k=-L+1}^{L-1} q_k \Phi(x-k) \tag{3}$$

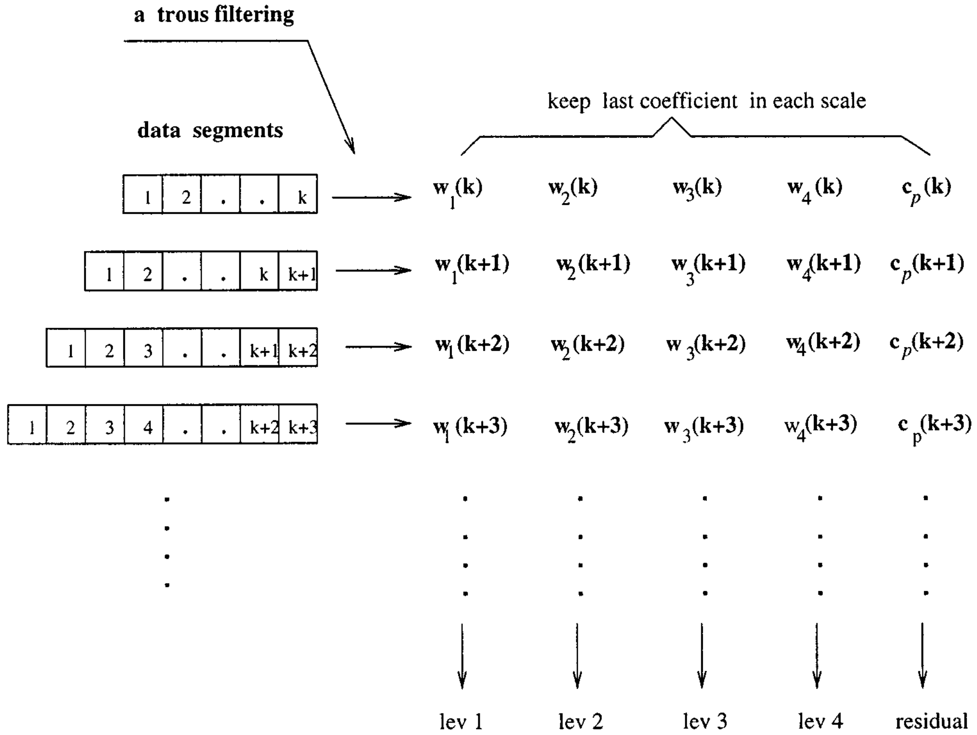

a trous filtering
data segments
keep last coefficient in each scale
1 2 . . k
1 2 . . k k+1
1 2 3 . . k+1 k+2
1 2 3 4 . . k+2 k+3
w1(k) w2(k) w3(k) w4(k) cp(k)
w1(k+1) w2(k+1) w3(k+1) w4(k+1) cp(k+1)
w1(k+2) w2(k+2) w3(k+2) w4(k+2) cp(k+2)
w1(k+3) w2(k+3) w3(k+3) w4(k+3) cp(k+3)
lev 1
lev 2
lev 3
lev 4
residual

Fig. 1. Illustration of the procedure for preparing data in the hybrid neuro-wavelet prediction scheme. Note that each time a segment of the time series is transformed, only the last coefficient is retained.

Using the filters P and Q, we obtain the pyramid algorithm for expanding into the autocorrelation shell

$$\begin{aligned} c_j(k) &= \sum_{l=-L+1}^{L-1} p_l c_{j-1}(k+2^{j-1}l) \\ w_j(k) &= \sum_{l=-L+1}^{L-1} q_l c_{j-1}(k+2^{j-1}l). \end{aligned} \tag{4}$$

As an example of the coefficients $\{p_k\}$, for Daubechies's wavelets with two vanishing moments and $L = 4$, the coefficients are $2^{-1/2}(-\frac{1}{16}, 0, \frac{9}{16}, 1, \frac{9}{16}, 0, -\frac{1}{16})$.

A very important property of the autocorrelation shell coefficients is that signals can be directly reconstructed from them. Given a smoothed signal at two consecutive resolution levels, the detailed signal can be derived as

$$w_j(k) = \sqrt{2}c_{j-1}(k) - c_j(k). \tag{5}$$

Then the original signal $c_0(k)$ can be reconstructed from the coefficients $\{w_j(k)\}_{1\le j\le n_0,\, 0\le k\le N-1}$ and residual $\{c_{n_0}(k)\}_{0\le k\le N-1}$

$$c_0(k) = 2^{-n_0/2}c_{n_0}(k) + \sum_{j=1}^{n_0} 2^{-j/2}w_j(k) \tag{6}$$

for $k = 0, \ldots, N-1$, where $c_{n_0}(k)$ is the final smoothed signal.

At each scale j, we obtain a set of coefficients $\{w_j\}$. The wavelet scale has the same number of samples as the signal, *i.e.*, it is redundant. The set of values of c_{n_0} provide a "residual" or "background." Adding w_j to this, for $j = n_0, n_0 - 1, \ldots,$ gives an increasingly more accurate approximation of the original signal. The additive form of reconstruction allows one to combine the predictions in a simple additive manner.

To make predictions we must make use of the most recent data. To deal with this boundary condition we use the *time-based à trous* filters algorithm proposed in [2], which can be briefly described as follows. Consider a signal $c(1), c(2), \ldots, c(n)$, where n is the present time-point and perform the following steps.

1) For index k sufficiently large, carry out the *à trous* transform (4) on $\{c(1), c(2), \ldots, c(n)\}$ using a mirror extension of the signal when the filter extends beyond k.
2) Retain the coefficient values as well as the residual values for the kth time-point only: $w_1(k), w_2(k), \ldots, w_p(k), c_p(k)$. The summation of these values gives $c(k)$.
3) If k is less than n, set k to $k+1$ and return to Step 1).

This process produces an additive decomposition of the signal $c(k), c(k+1), \ldots, c(n)$, which is similar to the *à trous* wavelet transform decomposition on $c(1), c(2), \ldots, c(k), \ldots, c(n)$. The algorithm is further illustrated in Fig. 1.

B. Application of Automatic Relevance Determination (ARD)

When applying neural networks to time series forecasting, it is important to decide on an appropriate size for the time-window of inputs. This is similar to a regression problem in which there are many possible input variables, some of which may be less relevant or even irrelevant to the prediction of the output variable. For a finite data set, there may exist some random correlations between the irrelevant inputs and the output, making it hard for a conventional neural network to set the coefficients for useless inputs to zero. The irrelevant

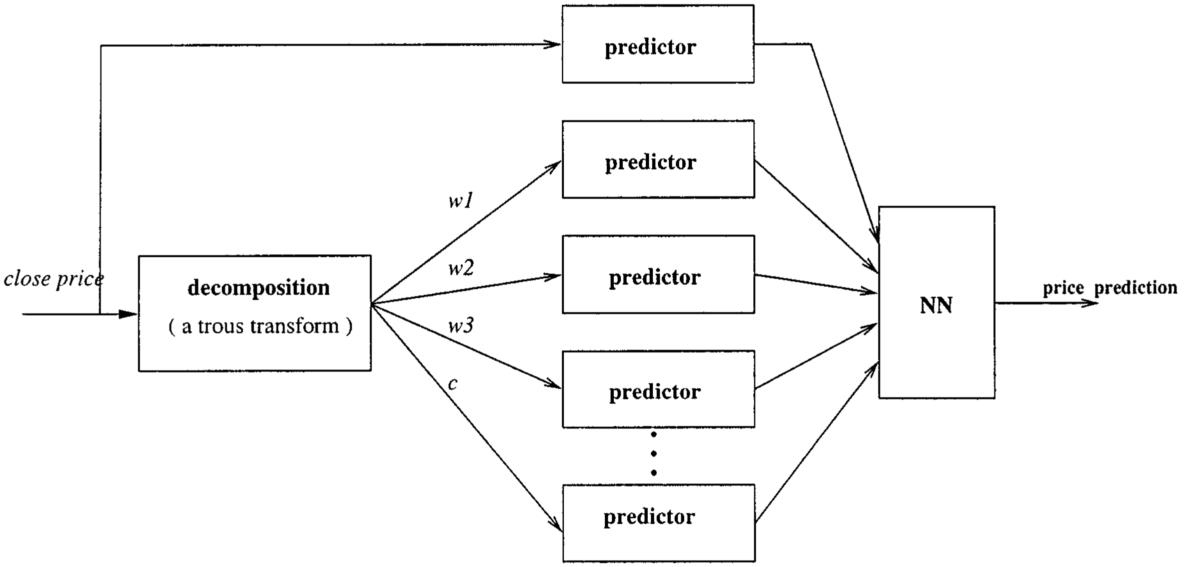


Fig. 2. Overview of the wavelet/neural net multiresolution forecasting system. $w_1, \ldots, w_k$ are wavelet coefficients, c is the residual coefficient series.

inputs however will degrade the model's performance. The ARD method [16] gives us a principled way for choosing the length of past windows to train neural networks. In our hybrid neuro-wavelet scheme, we apply ARD to choose a short-term history for higher temporal resolution (i.e., a higher sampling rate and higher frequencies) while a long-term history is used for lower temporal resolution. Through this, substantial information on both the "detailed" and "general" history of the time-series can be effectively exploited.

ARD stems from a practical Bayesian framework for adaptive data modeling [15], in which the overall aim is to develop probabilistic models that are well matched to the data, and make optimal predictions with those models. Given a data set, neural-network learning can be considered as an inference of the most probable parameters for a model. In most cases, there are a number of advantages of introducing Bayesian optimization of model parameters [5]. In particular, they provide a means to explicitly model prior assumptions by constructing the prior distribution over parameters and model architectures. In neural-network learning problems with high-dimensional inputs, generalization performance can often be improved by selecting those inputs relevant to the distribution of the targets. In the ARD scheme, we define a prior structure with a separate prior variance hyperparameter associated with each input. These hyperparameters correspond to separate weight decay regularisers for each input. In other words, ARD is effectively able to infer which inputs are relevant and then switch the others off by automatically assigning large values to the decay rates for irrelevant inputs, thus preventing those inputs from causing significant overfitting.

The ARD scheme used in this paper approximates the posterior distribution over weights by a Gaussian distribution. Using this approximation, the "evidence" for a nonlinear model can be readily calculated by an iterative optimization to find the optimal values for the regularization parameters. The optimization of these hyperparameters is interleaved with the training of the neural-network weights. More specifically, the parameters are divided into classes c, with independent scales α_c. For a network having one hidden layer, the weight classes are: one class for each input, consisting of the weights from that input to the hidden layer; one class for the biases to the hidden units; and one class for each output, consisting of its bias and all the weights from the hidden layer. Assuming a Gaussian prior for each class, we can define $E_{W(c)} = \sum_{i\in c} w_i^2/2$, then the ARD model uses the prior of equation

$$P(\{w_i\}|\{\alpha_c\}, \mathcal{H}_{ARD}) = \frac{1}{\prod Z_{W(c)}} \exp\left(-\sum_c \alpha_c E_{W(c)}\right). \tag{7}$$

The evidence framework can be used to optimize all the regularization constants simultaneously by finding their most probable value, i.e., the maximum over $\{\alpha_c\}$ of the evidence, $P(D|\{\alpha_c\}, \mathcal{H}_{ARD})$. We expect the regularization constants for irrelevant inputs to be inferred to be large, preventing those inputs from causing significant overfitting.

III. Hybrid Neuro-Wavelet Scheme for Time-Series Prediction

Fig. 2 shows our hybrid neuro-wavelet scheme for time-series prediction. Given the time series $f(n), n = 1, \ldots, N$, our aim is to predict the lth sample ahead, $f(N+l)$, of the series. That is, $l = 1$ for single step prediction; for each value of l we train a separate prediction architecture. The hybrid scheme basically involves three stages, which bear a similarity with the scheme in [11]. In the first stage, the time series is decomposed into different scales by autocorrelation shell decomposition. In the second stage, each scale is predicted by a separate NN and in the third stage, the next sample of the original time series is predicted, using the different scale's prediction, by another NN. More details are expounded as follows.

For time series prediction, correctly handling the temporal aspect of data is our primary concern. The *time-based à trous* transform as described above provides a simple method. Here we set up an *à trous* wavelet transform based on the autocorrelation shell representation. That is, (5) and (6) are applied to successive values of t. As an example, given a financial index with 100 values, we hope to extrapolate into the future with 1 or more than 1 subsequent values. By the *time-based à trous* transform, we simply carry out a wavelet transform on values x_1 to x_{100}. The last values of the wavelet coefficients at time-point $t = 100$ are kept because they are the most useful values for prediction. Repeat the same procedure at time point $t = 101$ and so on. We empirically determine the number of resolution levels J, mainly depending on the inspection of smoothness of the residual series for a given J. Much of the high resolution coefficients are noisy. Prior to forecasting, we get an overcomplete, transformed data set.

In Fig. 3, we show the behavior of the three wavelet coefficients over a 100-day period for a bond rating series. The original time series and residual are plotted at the top and bottom in the same figure, respectively. As the wavelet level increases, the corresponding coefficients become smoother. As we will show in the next section, the ability of the network to capture dynamical behavior varies with the resolution level.

In the second stage, different predictors are allocated for different resolution levels and are trained by the following wavelet's coefficients $w_i^j(t)$, $j = 0, \ldots, J$, $i = 1, \ldots, N$. All the networks used to predict the wavelets' coefficients share the same structure of a feedforward multilayer perceptron (MLP). The network for scale j has D_j input units, one hidden layer with K_j sigmoid neurons, and one linear output neuron. Each neuron in the networks has an adjustable bias. The D_j inputs to the jth network are the previous samples of the wavelets' coefficients of the jth scale. In our implementation, each network is trained by the backpropagation algorithm using the scaled conjugate gradient (SCG) method and a weight decay regularization of the form $(1/\lambda)\sum_i {w_i}^2$ was used [5].

The procedure for designing neural-network structure essentially involves selecting the input layer, hidden layer, and output layer. A basic guideline that should be followed is Occam's razor principle, which states a preference for simple models. The fewer weights in the network, the greater the confidence that over-training has not resulted in noise being fitted. The selection of input layer mainly depends on the considerations of which input variables are necessary for forecasting the target. From the complexity viewpoint it would be desirable to reduce the number of input nodes to an absolute minimum of essential nodes. In this regard, we applied ARD to empirically decide the number of inputs in each resolution level.

The optimum number of neurons in the hidden layer is highly problem dependent and a matter for experimentation. In all of our experiments, we set the number of hidden neurons by using half the sum of inputs plus outputs. Accordingly, for 21 inputs and one output, 11 hidden units are used. It is worthy to note that the selection of input and hidden layer neurons also determines the number of weights in the network and an upper limit on the weight number is dictated by the number of training vectors available. A rough guideline, based on theoretical considerations of the Vapnik–Chervonenkis dimension, recommends that the number of training vectors should be ten times or more the number of weights [3].

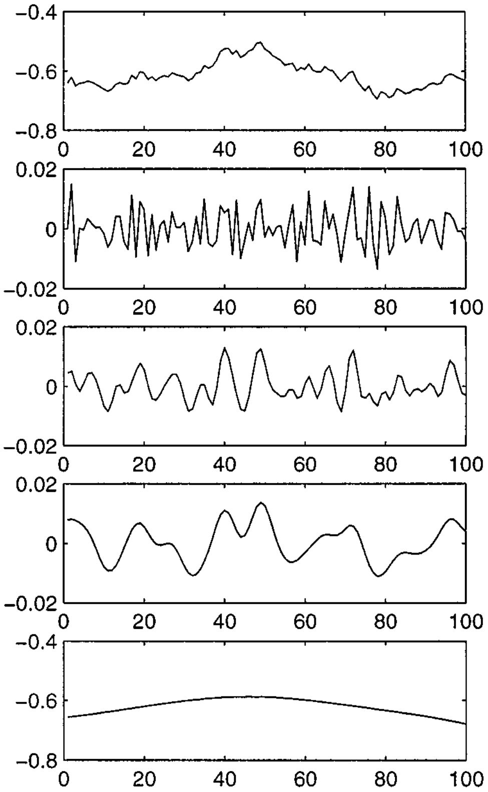


Fig. 3. Illustration of the *à trous* wavelet decomposition of the closing price series. From top to bottom: normalized price, w_1, w_2, w_3 and residual series.

In the third stage, the predicted results of all the different scales $\hat{w}_{N+i}^j(t)$, $j = 0, \ldots, J$ are appropriately combined. Here we discuss four methods of combination. In the first method, we simply apply the linear additive reconstruction property of the *à trous* transform, as expressed in (6). The fact that the reconstruction is additive allows the predictions to be combined in an additive manner. In the following we denote it as method I.

A hybrid strategy can also be empirically applied to determine what should be combined to provide an overall prediction. In the second method, the predicted results of all the different scales are linearly combined by a single-layer perceptron in order to predict the desired following sample of the original time series. In order to improve the prediction accuracy, a multilayer perceptron (MLP) with the same structure as for wavelet coefficients prediction is employed for price series and the corresponding prediction results are incorporated into the third stage, as shown in Fig. 2. For brevity, we call it method II. Depending on the forecasting horizon on the second stage, the number of inputs to the third-stage network is equal to the number of all the prediction outputs in the first stage. For example, if four resolution levels are exploited and an MLP for direct price prediction is incorporated in the second stage, then for forecasting horizon 7, the number of inputs in the third stage perceptron is $6 \times 7 = 42$.

In our experiments we have also applied a third stage MLP in place of the simple perceptron. The number of hidden neurons is also set to half of the sum of the number of inputs and outputs. We denote this as method III for the combination of prediction results from the second stage. For comparison purposes, we trained and tested an MLP on the original time series, denoted as method IV, without the wavelet preprocessing stage.

As pointed out in [3], target selection is an important issue in applying neural networks to financial series forecasting. We follow the guideline suggested by Azoff to minimize the number of targets required for a given problem. A neural network whose output neurons are reduced from two to one, will have half the number of network weights required, with important consequences for the generalization capability of the network. A single output neuron is the ideal, as the network is focused on one task and there is no danger of conflicting outputs causing credit assignment problems in the output layer. Accordingly, we prefer a forecasting strategy which proceeds separately for each horizon in the second stage.

IV. Simulations and Performances

Our simulations involved the closing prices of four different futures contracts: The three-year and ten-year Treasury bonds (**sfe3yb, sfe10yb**) traded on the Sydney Futures Exchange, the Australian US dollar contract (**cmedolaus**) and the Swiss Franc US dollar contract (**cmesfus**) traded on the Chicago Mercantile Exchange. In order to derive a continuous time series from a set of individual futures contracts, special care must be taken at the expiry of a contract. The price change from one contract to the next cannot be directly exploited in a trading system. Instead a contract must be rolled from the expiry month to a forward month. We found that the four securities we are considering are characterized by a price gap at roll over in the range of the close to close price variation. The concatenation of spot month contracts is therefore a reasonable approximation. In Fig. 4, we show the **sfe10yb** closing price over a ten-year period.

We study the approach of forecasting each wavelet derived coefficient series individually and then recombining the marginal forecasts. Our objective is to perform seven days ahead forecasting of the closing price. As a byproduct, the corresponding price changes are simultaneously derived. To compare with other similar work in the literature, we also construct five days ahead forecastes of the relative price change, *i.e.*, the relative difference percent (RDP) between today's closing price and the closing price five days ahead, denoted $RDP(t)$, which is calculated as $RDP(t) = (x(t+5) - x(t))/x(t)$ [2]. The data sets used consist of the date, the closing price $x(t)$ and the target forecast. A separate MLP network for each level of the coefficients series is constructed. The scaled conjugate gradient (SCG) algorithm was used for training. As the residual series are quite smooth, we simply apply linear AR models to them.

At first, the raw price data requires normalizing, a process of standardising the possible numerical range that the input vector elements can take. The procedure involves finding the maximum (max) and minimum (min) elements and then normalizing the price x_i to the range $[-1, 1]$ [3]:

$$\hat{x}_i = 2\frac{x_i - \min}{\max - \min} - 1. \tag{8}$$

Since many of the high resolution coefficients are very noisy, we applied the ARD technique to determine the relevant inputs of the MLPs on different levels. At first, each network had 21 inputs. The ARD scheme was used with a separate prior for each MLP input. Hence, the regularization constants for noisy inputs are automatically inferred to be large. In Table I we give typical results of hyper-parameters α for input variables when applying ARD to MLPs on different levels. From the results we can see that the first two level coefficients are noisy and have little relevance to the target distribution. To exploit this fact to further improve performance and reduce computational complexity, we apply MLPs with variable input sizes to different levels, as shown in Table II.

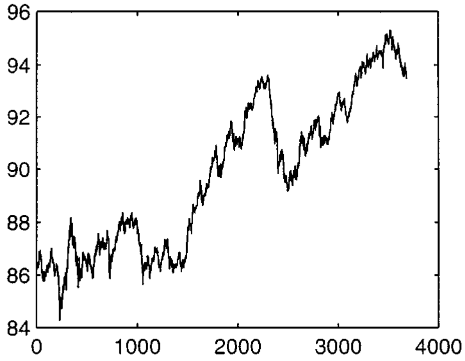


Fig. 4. Closing price for ten-year Treasury bonds traded on the Sydney Futures Exchange.

We decomposed all the time-series into four resolution levels as the residual series become quite smooth. All training sets consist of the first 2000 data values (one closing price per day). For the **sfe3yb** and **sfe10yb**, we use the remaining 600 and 1000 data points for testing, respectively. For the Australian US dollar contract (**cmedolaus**) and the Swiss Franc US dollar contract (**cmesfus**), we use the remaining 600 data points for testing. In Fig. 5, we show the one step ahead predictions for each of the four coefficient series $(\{w_1\}, \{w_2\}, \{w_3\}$ and $\{w_4\})$ and the residual series over a 100 days period on a testing set (from Nov. 15, 1993 to April 13, 1994). As the residual series is very smooth, a simple AR model shows quite satisfactory prediction performance. The ability of the networks to capture dynamical behavior varies with the resolution level [2] and we can observe two facts. First, the higher the scale (e.g., w_5 is "higher" than w_4), the smoother the curve and thus, the less information the network can retrieve. Second, the lower the scale, the more noisy and irregular the coefficients are, thus making the prediction more difficult. The smooth wavelet coefficients at higher scale play a more important role.

In Figs. 6 and 7, we illustrate the forecasting of one day ahead price and price change series (RDP), respectively, using the prediction methods I, II, and IV as previously described. The different prediction methods show quite similar results on the same testing set. But a close inspection reveals a better accuracy resulting from method II, i.e., using a perceptron to combine the prediction results of wavelet coefficients. To quantitatively calculate the prediction performance, we used mean square error (MSE) to describe the forecast performance for price prediction, which is defined as MSE $= (1/N)\sum_{k=1}^{N}(x(k) - \hat{x}(k))^2$, where $x(k)$ is the true value of the sequence, $\hat{x}(k)$ is the prediction. For price change prediction, we used two other measures. The first measure is the normalized mean squared error NMSE $= (1/\sigma^2 N)\sum_{k=1}^{N}(x(k) - \hat{x}(k))^2$, where $x(k)$ is the

TABLE I
HYPER-PARAMETERS α FOR THE MLP NETWORK INPUTS ON DIFFERENT LEVELS FOR THE **sfe10yb** DATA SET. THE ORDER OF THE PARAMETERS ARE FROM PAST TO FUTURE

level 1	720.15	1168.98	2663.91	2420.56	2396.89	4061.82	3649.14
	1013.36	846.29	117.21	276.96	332.98	713.17	365.93
	894.17	1029.06	1895.64	21.52	10.46	6.06	5.25
level 2	712.44	1138.76	1434.47	2636.82	6413.63	82777.96	47055.76
	45475.55	9673.50	725.20	191.87	106.32	37.41	25.13
	18.07	11.67	6.73	4.28	3.41	3.64	3.45
level 3	2.20	3.40	3.09	2.47	4.17	9.29	3.46
	3.46	1.20	1.12	1.01	1.47	1.60	1.34
	2.33	2.56	2.38	1.94	0.99	0.87	0.75
level 4	0.17	0.34	0.55	0.87	2.27	2.50	1.65
	1.72	1.77	2.54	1.32	0.66	1.21	1.54
	1.40	0.79	0.84	1.23	0.18	0.10	0.11

TABLE II
STRUCTURE OF MLPS ON DIFFERENT LEVELS

	level 1	**level 2**	**level 3**	**level 4**	**residual**
#input	3	8	21	21	21

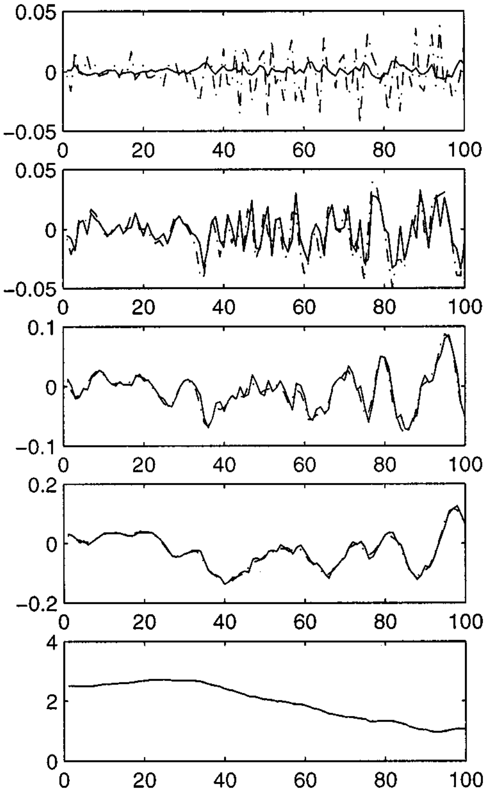


Fig. 5. From top to bottom: one step ahead predictions for the four wavelet coefficient series w_1, w_2, w_3 and w_4 and residual series c, over a 100 days period on the testing set. In each figure, the dashed line is the target series and the solid line is the prediction.

true value of the sequence, $\hat{x}(k)$ is the prediction, and σ^2 is the variance of the true sequence over the prediction period.

A second measure of interest for price change prediction is the directional symmetry (DS), i.e., the percentage of correctly predicted directions with respect to the target variable, defined as $DS = (1/N)\sum_{k=1}^{N} \psi(x(k)\cdot\hat{x}(k))$, where ψ is the Heaviside unit-step function, namely, $\psi(x) = 1$ if $x > 0$ and $\psi = 0$ otherwise. Thus, the DS provides a measure of the number of times the sign of the target was correctly forecast. In other words, $DS = 50\%$ implies that the predicted direction was correct for half of all predictions.

In Table III we used the **sfe10yb** data to compare the four prediction methods with regard to MSE performance for price prediction and NMSE and DS performance for price change predictions, respectively. From the results we can see that all four methods have similar performance with regard to the MSE for price prediction and NMSE and DS for price change prediction and that method II shows better generalization performance.

The evaluation of the overall system is a very important issue. By some performance measures, we can evaluate whether targets have been met and compare different strategies in a trading system. Criteria in setting up a trading strategy will vary according to the degree of risk exposure permitted, so the assessment criteria selected are a matter of choice, depending on priorities.

The most commonly used measure is the Sharpe ratio, which is a measure of risk-adjusted return [22]. Denoting the trading system returns for period t as R_t, the Sharpe ratio is defined to be

$$S_T = \frac{\text{Average}(R_t)}{\text{Standard Deviation }(R_t)} \tag{9}$$

where the average and standard deviation are estimated over returns for periods $t = \{1, \ldots, T\}$.

As another measure of interest we evaluate the quality of our forecasts in a trading simulator. Trading results are simulated using the risk evaluation and money management (REMM) trade simulation environment that has been used in previous simulations [10], [8]. REMM has been developed and tested with the help of expert futures traders. It is currently used by a number of financial institutions to analyze and optimize trading strategies. A description of the functionality of REMM is given in the following. REMM facilitates the testing of a trade entry

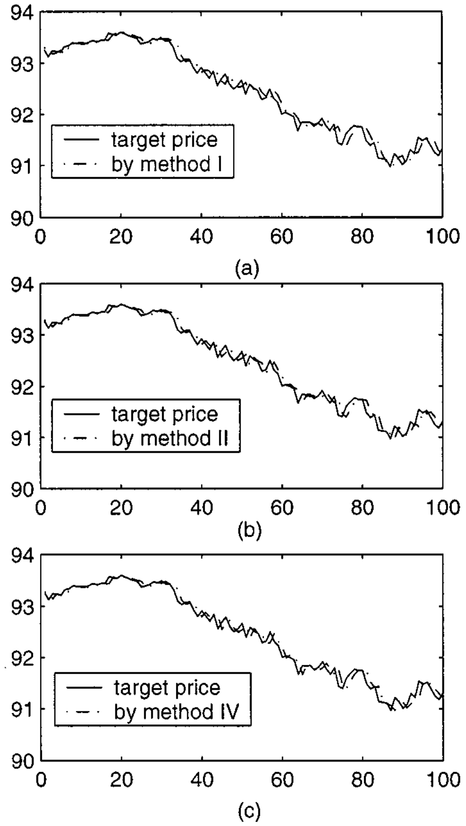


Fig. 6. Demonstration of one day ahead forecasting for the closing price of **sfe10yb** on a testing data set over a 100 days period (from Nov. 15, 1993 to April 13, 1994), using the prediction methods (I, II, and IV). In each figure, the solid line is the target price and the dashed line is the prediction.

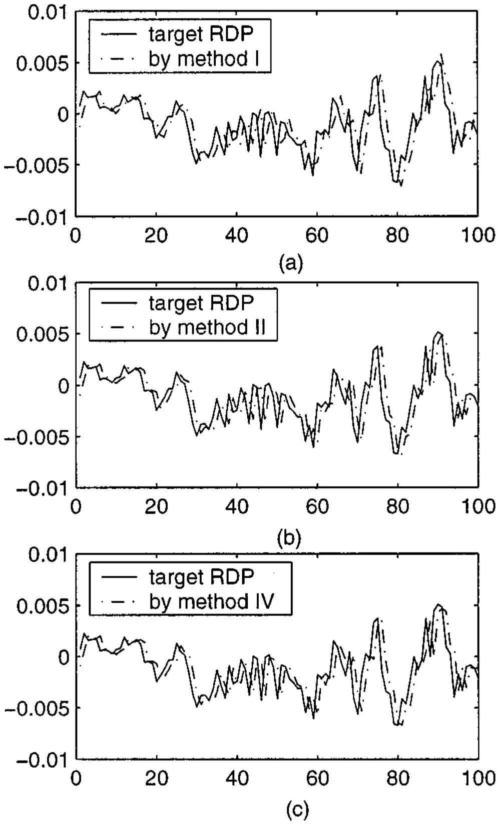


Fig. 7. One step ahead price changes (RDP) forecasts (dashed lines) vs the true RDP series (solid line) for a segment of 100 days in the testing set (from Nov. 15, 1993 to April 13, 1994). See text for explanation for the prediction methods I, II and IV.

strategy by accurately modeling the market dynamics. As an input REMM requires the time and price information and a sequence of trade entry signals. The latter is obtained from the forecaste output of the various prediction systems. Accurate and realistic risk and trade management strategies can be selected to test the quality of the prediction system. This includes the consideration of transaction costs. They are incurred each time a futures contract is purchased or sold. Slippage is a common phenomenon in trading futures. It is the discrepancy between the theoretical entry or exit price and the actual price. In REMM slippage is modeled using a volatility based approach. REMM allows the selection of a number of realistic trade exit strategies like profit take and partial profit take at various target levels, trade expiry and stop loss levels. The exit conditions, e.g., target and stop loss levels, are dynamically adjusted due to changing market conditions. Risk management strategies are implemented by providing trading capital of $1 million and applying risk limits of $10 000 for each trade.

For a given sequence of trade entry signals and a set of risk and trade management parameters the trading system is simulated using a forward stepping approach. At each time step the system is updated by checking for new trade entries and adjusting the exit conditions for open positions caused by the new market price. When an exit condition is satisfied, e.g., due to a target being reached or a stop loss level hit, etc. the open position is novated and the overall portfolio position is updated. More than 50 different performance measures are derived that allow assessment of the quality of the trading system over the given training period. The most relevant measures are listed in the following. The profit per trade is the average profit per trade over the trading period. The win/loss ratio is the ratio of winning trades to loosing trades over the trading period. The Sharpe ratio is the ratio of the annualised monthly return. The worst monthly loss is the total of losses from trades in the worst calendar month. An optimal trading strategy is derived from the training set and applied to the test set.

TABLE III
FOR **sfe10yb** DATA, PREDICTION PERFORMANCES FROM THE FOUR DIFFERENT PREDICTION METHODS

	Price/MSE		RDP/DS		RDP/NMSE	
data set	training	testing	training	testing	training	testing
method I	0.0101	0.0103	77.83%	74.53%	0.4616	0.5195
method II	0.0091	0.0110	78.86%	76.41%	0.4251	0.4815
method III	0.0092	0.0155	78.40%	75.78%	0.4282	0.4799
method IV	0.0090	0.0117	78.60%	76.20%	0.4262	0.4837

Using the REMM simulator, we further compared the profitability related performances of the four forecasting methods, namely, directly summing up the wavelet coefficients predictions from the linear reconstruction property (6) (method I), using a perceptron (method II) or an MLP (method III) to combine the wavelet coefficients prediction and simply applying an MLP without wavelet features involved (method IV). For the ten-year bond contract on the test set (consisting of 1000 days of

TABLE IV

For **sfe10yb** Data, Comparison of the Profitability Related Performances from the Four Different Forecasting Methods

	method I	method II	method III	method IV
data set	training/testing	training/testing	training/testing	training/testing
# Trades	49/71	52/162	57/170	68/190
Profit/Trade	692.78/2002.74	5141.89/1753.30	4542.94/1808.62	4700/819
Profit/Loss	1.1553/1.5269	2.2306/1.6307	2.0801/1.6110	2.1396/1.2870
Sharpe Ratio	0.0009/0.7049	0.6238/0.7321	0.5926/0.3850	0.7186/0.3677
Worst Loss/month	22835/24450	22354/40814	22354/46351	26159/24450

TABLE V

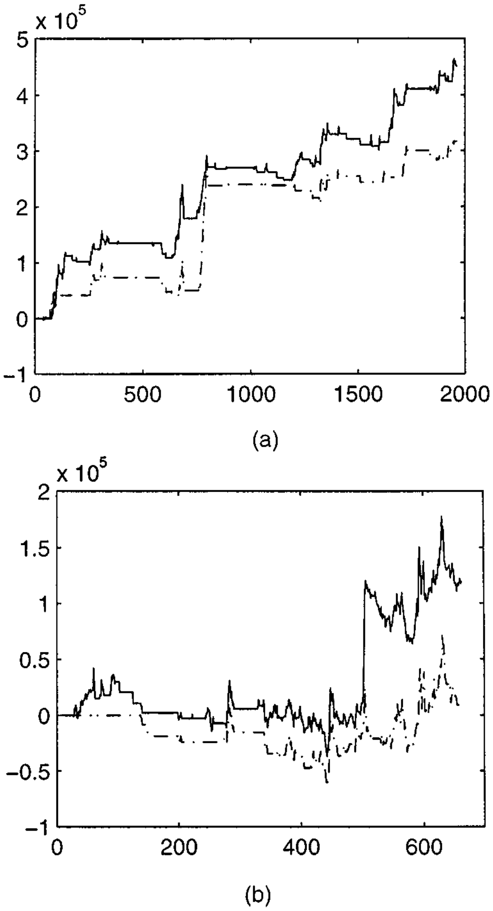
Performance Comparison for Different Data Sets

	sfe10yb	**sfe3yb**	**cmedolsf**	**cmedolaus**
Price/MSE	0.0110	0.0063	1.5100e-05	2.3000e-05
RDP/DS	76.41%	77.24%	80.17%	79.02%
RDP/NMSE	0.4815	0.4589	0.4194	0.4583

data), the measures shown in Table IV were calculated to evaluate the performance of the system under realistic trading conditions. Table IV summarizes the profit per trade, the win/loss ratio, the Sharpe ratio and the worst monthly loss. Each trade is based on a number of contracts determined by the risk per trade.

From Table IV, it is obvious that method II has the highest values of both Sharpe ratio (0.7321) and profit/loss ratio (1.6307), together with a satisfactory trading number and profit per trade. Though a plain MLP (method IV) generates the most trades, it yields the worst performance with regard to the profit per trade, profit-loss ratio and Sharpe ratio. Simply combining wavelet coefficients using (6) (method I) offers reasonable results of profit per trade and profit-loss ratio, but leads to the most conservative trading activity (only 71 trades in more than three years!). Overall, we can recommend method II as a practical forecasting strategy for a trading system.

We have also tested the neuro-wavelet prediction method on the closing prices of other futures contracts: **sfe3yb, cmedolaus** and **cmesfus**. In Table V, we show MSE for price prediction, NMSE and DS for RDP series prediction, all for testing data sets. Profit/loss results are given in Figs. 8 and 9 for the **sef3yb** data and **sef10yb**, respectively. Prediction method I was compared with method IV in Fig. 8 while method II was compared with method I in Fig. 9. From these evaluations, we can conclude that multiscale neural-network architectures generally show better profitability than applying an MLP alone and the hybrid scheme exploiting a second-stage perceptron has best performance.



(a)

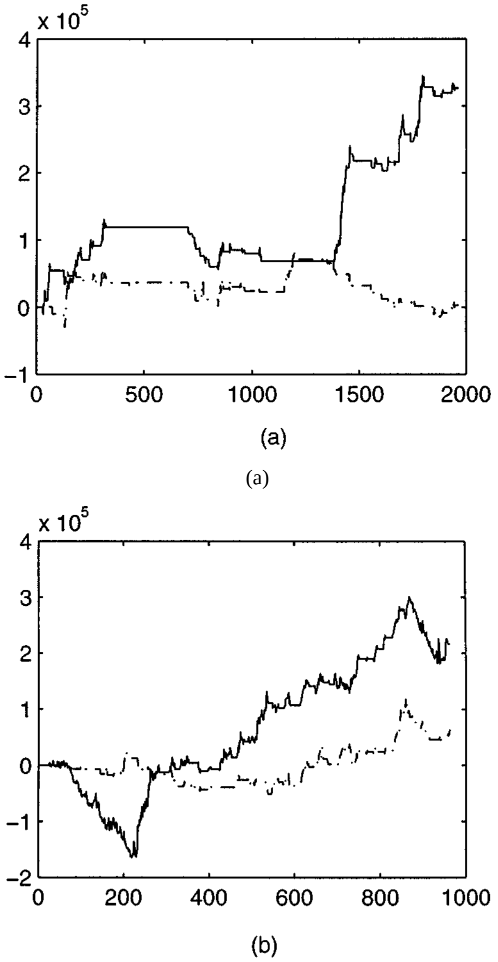
(b)

Fig. 8. Comparison of profit/loss results from applying the neuro-wavelet forecasting scheme method I and the MLP alone (method IV), using the three-year Treasury bonds data **sfe3yb**. The solid line results from method I by applying the linear reconstruction property (6) while the dashed line corresponds to the plain MLP. (a) profit and loss on the training set in \$AUD against trading days. (b) profit and loss for the testing set.

V. Discussion and Conclusion

Forecasting of financial time series is often difficult and complex due to the interaction of the many variables involved. In this paper, we introduced the combination of shift invariant wavelet transform preprocessing and neural-network prediction models trained using Bayesian techniques at the different levels of wavelet scale. We compared this with a conventional MLP by simulation on four sets of futures contract data and determined both forecasting performance and profitability measures based on an accurate trading system model. Our results show significant advantages for the neuro-wavelet technique. Typically, a doubling in profit per trade, Sharpe ratio improvement, as well as significant improvements in the ratio of winning to loosing trades were achieved compared to the MLP prediction.

Although our results appear promising, additional research is necessary to further explore the combination of wavelet techniques and neural networks, particularly over different market conditions. Financial time series, as we have noted, often show considerable abrupt price changes; the extent of outliers often decides the success or otherwise for a given model. While the



(a)

(b)

Fig. 9. Comparison of profit/loss results from applying neuro-wavelet forecasting scheme method I and method II, using the ten-year bond (**sfe10yb**) data. The solid line results from the hybrid architecture using a perceptron for combining the wavelet coefficients (method II) while the dashed line is the simpler architecture (method I), in which the wavelet coefficients are directly summed up. (a) profit and loss on training set in AUD against trading days. (b) profit and loss for the testing set.

prediction performance is improved, the neuro-wavelet hybrid scheme is still a global model, which is susceptible to outliers. Ongoing work includes 1) the integration of the *time-based à trous* filters studied here and mixture of local expert model, which may explicitly account for outliers by special expert networks and 2) direct volatility forecasting by a similar hybrid architecture. Other research areas include the online adaptation of the network models including ARD hyper-parameters, the investigation of wavelet based denoising techniques and solutions to the associated boundary condition problems for the online learning case in order to further improve generalization performance and the investigation of the joint optimization of forecasting and money management systems.

References

[1] A. Aussem and F. Murtagh, “Combining neural networks forecasts on wavelet-transformed time series,” *Connection Sci.*, vol. 9, pp. 113–121, 1997.

[2] A. Aussem, J. Campbell, and F. Murtagh, “Wavelet-based feature extraction and decomposition strategies for financial forecasting,” *J. Comput. Intell. Finance*, pp. 5–12, Mar. 1998.

[3] A. M. Azoff, *Neural Network Time Series Forecasting of Financial Markets*. New York: Wiley, 1994.

[4] G. Beylkin and N. Satio, “Wavelets, their autocorrelation functions and multiresolution representation of signals,” *IEEE Trans. Signal Processing*, vol. 7, pp. 147–164, 1997.

[5] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Oxford Univ. Press, 1995.

[6] ——, “Bayesian methods for neural networks,”, Tech. Rep. NCRG/95/009, 1995.

[7] R. R. Coifman and D. L. Donoho, “Translation-invariant de-noising,” in *Wavelets and Statistics, Springer Lecture Notes*, A. Antoniades, Ed. New York: Springer-Verlag, 1995.

[8] D. R. Dersch, B. G. Flower, and S. J. Pickard, “Exchange rate trading using a fast retraining procedure for generalized radial basis function networks,” in *Proc. Neural Networks Capital Markets*, 1997.

[9] R. J. Van Eyden, *The Application of Neural Networks in the Forecasting of Share Prices*. Haymarket, VA: Finance & Technology Publishing, 1995.

[10] B. G. Flower, T. Cripps, M. Jabri, and A. White, “An artificial neural network based trade forecasting system for capital markets,” in *Proce. Neural Networks Capital Markets*, 1995.

[11] A. B. Geva, “ScaleNet—Multiscale neural-network architecture for time series prediction,” *IEEE Trans. Neural Networks*, vol. 9, pp. 1471–1482, 1998.

[12] A. Kehagias and V. Petridis, “Predictive modular neural networks for time series classification,” *Neural Networks*, vol. 10, pp. 31–49, 1997.

[13] S. G. Mallat, “A theory for multiresolution signal decomposition: The wavelet representation,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 674–693, 1989.

[14] K. R. Müller, J. Kohlmorgen, and K. Plawelzik, “Analysis of switching dynamics with computing neural networks,” Univ. Tokyo, Tech. Rep., 1997.

[15] D. J. C. MacKay, “A practical Bayesian framework for backpropagation networks,” *Neural Comput.*, vol. 4, pp. 448–472, 1992.

[16] ——, “Bayesian nonlinear modeling for the 1993 energy prediction competition,” in *Maximum Entropy and Bayesian Methods, Santa Barbara 1993*, G. Heidbreder, Ed. Dordrecht, The Netherlands: Kluwer, 1995.

[17] T. Masters, *Neural, Novel & Hybrid Algorithms for Time Series Prediction*. New York: Wiley, 1995.

[18] S. Makridakis, S. C. Wheelwright, and R. J. Hyndman, *Forecasting, Methods and Applications*, 3rd ed. New York: Wiley, 1998.

[19] F. F. Murtagh, “Wedding the wavelet transform and multivariate data analysis,” *J. Classification*, vol. 15, pp. 161–183, 1998.

[20] V. Petridis and A. Kehagias, “Modular neural networks for MAP classification of time series and the partition algorithm,” *IEEE Trans. Neural Networks*, vol. 17, pp. 73–86, 1996.

[21] N. Saito and G. Beylkin, “Multiresolution representations using the auto-correlation functions of compactly supported wavelets,” *IEEE Trans. Signal Processing*, 1992.

[22] W. F. Sharpe, “The Sharpe ratio,” *J. Portfolio Management*, pp. 49–58, Fall 1994.

[23] M. J. Shensa, “The discrete wavelet transform: Wedding the á trous and Mallat algorithms,” *IEEE Trans. Signal Processing*, vol. 10, pp. 2463–2482, 1992.

[24] M. R. Thomason, “Financial forecasting with wavelet filters and neural networks,” *J. Comput. Intell. Finance*, pp. 27–32, Mar. 1997.

[25] A. S. Weigend and M. Mangeas, “Nonlinear gated experts for time series: Discovering regimes and avoiding overfitting,” *Int. J. Neural Syst.*, vol. 6, pp. 373–399, 1995.

[26] P. Zuohong and W. Xiaodi, “Wavelet-based density estimator model for forecasting,” *J. Comput. Intell. Finance*, pp. 6–13, Jan. 1998.

[27] Z. Gonghui, J.-L. Starck, J. Campbell, and F. Murtagh, “The wavelet transform for filtering financial data streams,” *J. Comput. Intell. Finance*, pp. 18–35, June 1999.

[28] J. S. Zirilli, *Financial Prediction Using Neural Networks*: International Thomson Computer Press, 1997.

Bai-Ling Zhang was born in China. He received the Bachelor of Engineering degree in electrical engineering from Wuhan Institute of Geodesy, Photogrammetry, and Chartography in 1983, the Master of Engineering degree in electronic system from the South China University of Technology in 1987, and the Ph.D. degree in electrical and computer engineering from the University of Newcastle, Australia in 1999.

From 1998 to 1999, he was a Research Assistant with School of Computer Science and Engineering, University of New South Wales, Australia. From 1999 to 2000, he worked as a Postdoctoral Fellow with the Computer Engineering Laboratory (CEL), School of Electrical and Information Engineering, University of Sydney. Currently, he is a member of the research staff in the Kent Ridge Digital Labs (KRDL), Singapore. His research interests include artificial neural networks, image processing and computer vision, pattern recognition, and time-series analysis and prediction.

Richard Coggins (M'95) received the B.Sc. degree in physics and pure mathematics in 1985 and the B.E. Hons. degree in electrical engineering in 1987 from the University of Sydney, Australia. He received the Ph.D. degree in electrical engineering from the University of Sydney in 1997.

From 1988 to 1990, he worked at Ausonics Pty. Ltd. in the diagnostic ultrasound products group. In 1990, he received the Graduate Management qualification from the Australian Graduate School of Management. He joined the University of Sydney as a Research Engineer in 1990. He is currently a Senior Lecturer at the School of Electrical and Information Engineering at the University of Sydney. He was appointed as a Girling Watson Research Fellow in the Computer Engineering Laboratory in 1997. He was appointed as a Senior Lecturer in 2000. His research interests include machine learning, time series prediction, low-power microelectronics, and biomedical signal processing.

Marwan Anwar Jabri (S'84–M'85–SM'94) was born in Beirut, Lebanon, in 1958. He received the License de Physique and Maitrise de Physique degrees from the Université de Paris VII, France, in 1981 and 1983, respectively. He received the Ph.D. degree in electrical engineering at the University of Sydney in 1988.

He was a Research Assistant during 1984 as part of the Sydney University Fleurs Radiotelescope research team. He was appointed as Lecturer at the University of Sydney in 1988, Senior Lecturer in 1992, Reader in 1994, and Professor in 1996. Since January 2000, he has been the Gordon and Betty Moore endowed Chair Professor at the Electrical and Computer Engineering Department, Oregon Graduate Institute (OGI), Beaverton, and Professor in Adaptive Systems at the University of Sydney, School of Electrical and Information Engineering. He was a Visiting Scientist at AT&T Bell Laboratories in 1993 and the Salk Institute for Biological Studies in 1997. He is author, coauthor, and editor of three books and more than 150 technical papers and is an invited speaker at many conferences and forums. His research interests include digital and analog integrated circuits, biomedical and neuromorphic engineering, and multimedia communication systems.

Dr. Jabri is a recipient of the 1992 Australian Telecommunications and Electronics Research Board Outstanding (ATERB) Young Investigator Medal. He is on the editorial board of several journals. He is member of INNS and a Fellow of the Institute of Engineering Australia.

Dominik Dersch received the masters degree in physics from the Technical University of Munich, Munich, Germany, in 1991 and the Doctorate degree in Natural Science from the Ludwig Maximilians University, Munich, in 1995.

From 1996 to 1997, he was a Research Fellow in the Speech Technology Group at the University of Sydney. From 1997 to 1999, he worked as a Senior Forecasting Analyst in Electricity Trading for Integral Energy Australia. From 1999 to 2000, he was Head of Research and Development of Crux Cybernetics and then of Crux Financial Engineering. He is currently a Senior Quantitative Analyst at HypoVereinsbank in Munich. His research interests include statistical physics, statistics, pattern recognition, classification, and time series analysis. He has worked and published in areas including speech recognition and speech analysis, remote sensing data analysis, medical image processing and classification, and financial time series analysis and prediction. He holds a license as financial advisor with the Sydney Futures Exchange.

Barry Flower (M'96) received the Bachelor of Engineering and Bachelor of Computing Science degrees from the University of New South Wales, Australia, in 1987 and the Ph.D. degree in electronic and computer engineering from the University of Sydney, Australia, in 1995.

From 1990 to 1995, he was a Research Associate and then Girling Watson Research Fellow in the System Engineering and Design Automation Laboratory at the University of Sydney. From 1995 to 2000, he was a Founder and Joint Managing Director of Crux Cybernetics and then Crux Financial Engineering. He is currently Manager of E-Commerce, Strategic Development at Hong Kong and Shanghai Banking Corporation. His research interests include connectionists techniques applied to time series analysis, financial markets, speech recognition, and autonomous robotic systems.