New Mathematics and Natural Computation
Vol. 1, No. 1 (2005) 79–107

World Scientific
www.worldscientific.com

EXTENDED DAILY EXCHANGE RATES FORECASTS USING WAVELET TEMPORAL RESOLUTIONS

MAK KABOUDAN

School of Business, University of Redlands, 1200 East Colton Avenue
Redlands, California 92373, USA
Mak_Kaboudan@Redlands.edu

Applying genetic programming and artificial neural networks to raw as well as wavelet-transformed exchange rate data showed that genetic programming may have good extended forecasting abilities. Although it is well known that most predictions of exchange rates using many alternative techniques could not deliver better forecasts than the random walk model, in this paper employing natural computational strategies to forecast three different exchange rates produced two extended forecasts (that go beyond one-step-ahead) that are better than naïve random walk predictions. Sixteen-step-ahead forecasts obtained using genetic programming outperformed the one- and sixteen-step-ahead random walk US dollar/Taiwan dollar exchange rate predictions. Further, sixteen-step-ahead forecasts of the wavelet-transformed US dollar/Japanese Yen exchange rate also using genetic programming outperformed the sixteen-step-ahead random walk predictions of the exchange rate. However, random walk predictions of the US dollar/British pound exchange rate outperformed all forecasts obtained using genetic programming. Random walk predictions of the same three exchange rates employing raw and wavelet-transformed data also outperformed all forecasts obtained using artificial neural networks.

Keywords: Genetic programming; artificial neural networks; Haar wavelets.

1. Introduction

Studies on forecasting exchange rates tend to focus mostly on obtaining one-step-ahead forecasts that can outperform naive benchmark random walk predictions. From a practical point of view, even if obtained one-step-ahead forecasts outperform random walk predictions, they cannot be delivered before input data needed to construct a model are available. There is a good chance then that one-step-ahead predictions are delivered a bit too late to help in decision making. This is especially true when forecasting high frequency time series such as daily (or intra-daily) exchange rates. This problem motivates producing reliable *extended forecasts* that may be more useful in decision making. "*Extended forecasts*" are defined here as predictions of a variable that continue for a relatively large number of periods (surely >1) beyond currently available data. They are different from multi-step-ahead forecasts obtained using lagged dependent variable solutions. In *extended*

forecasts only actual values of input variables are used to obtain predictions. The thought of producing extended forecasts creates a host of questions that should be addressed. First, how far ahead into the future should an extended forecast be? Second, what modeling or forecast techniques can or should be used? Third, what benchmark forecast would one use in evaluating extended forecast accuracies? And finally, what evaluation criteria should be implemented?

Selection of the *extended* number of periods to forecast should logically depend on the frequency of the data to model and the relative complexity of the dynamics of the system studied. This makes such selection subjective and mainly the outcome of experimentation. In the case of very low frequency observables such as quarterly data, it may be possible to produce a forecast that extends up to eight or even twelve quarters into the future if a system is fairly predictable. For monthly data, it may be possible to produce a forecast that extends up to twelve or even 24 months into the future. For higher frequency data, such as daily, forecasts of 30 to 60 days into the future may be possible. This study focuses on producing extended forecasts of three daily exchange rates: the US dollar/British pound (DBP), US dollar/Japanese yen (DJY), and US dollar/Taiwanese dollar (DTD). Selection of the number of days for which to generate extended forecasts was influenced by the use of wavelet-transformed data as model inputs. To complete wavelet transformations, the series investigated should be of length 2^J, where J is a positive integer. (This matter becomes clear after discussions presented in Secs. 2 and 3 below.) First an attempt to forecast 64 periods ahead failed. It was logical next to try and forecast 32 periods ahead. The results were more encouraging. All models obtained and forecasting techniques used to predict the exchange rates in this paper therefore deliver forecasts of 32 days beyond the last day data is available. The first 16 (of the 32) observations will be used as a validation set. If forecasting that set is successful, confidence in a model or technique and its usefulness for decision making increases. This means that a model that succeeds in reproducing historical data (i.e. data used in fitting or training) best should not be used to forecast future outcomes unless it also happens to be the most successful one in producing validation out-of-sample forecasts. This seems logical since a model that forecasts well for an extended period of time will probably forecast more periods better than forecasts for the same periods obtained from a model whose forecasting ability was never tested.

This study favors use of natural computation algorithms in modeling and forecasting daily exchange rates. Early studies that forecasted exchange rates found no evidence to support the hypothesis that nonlinear generating processes outperform random walk behavior.[1–3] More recent applications of computational algorithms found that artificial neural network and nearest neighbor approaches may deliver more accurate than no-change (random walk) forecasts in a large number of situations.[4–10] Evidence that computational nonlinear techniques may help forecast what is easily characterized as dynamically complex systems motivated investigating the ability of two natural computing algorithms — genetic programming (GP) and artificial neural networks (ANN) — to forecast the three exchange rates.

The techniques are applied to the observed raw data of the three time series and to their wavelet-transformed values. Wavelet transformations decompose each observed sequence into varying scales of temporal resolution. This helps test the hypothesis that fitting or training transformed data using a natural computing algorithm produces at least equally reliable forecasts as those obtained employing the same technique by fitting or training the observed data.

Two benchmark models seem applicable when evaluating *extended* forecasting accuracies. First and for the sake of consistency with established standards, nonlinear natural computation algorithms forecasts are compared with random walk exchange rates forecasts that simply take today's outcome to be tomorrow's forecast or $Y_t = Y_{t-1} + \varepsilon_t$, where $\varepsilon \sim N(0, \sigma^2)$ and $t = 1, \ldots, T$ time periods. The second benchmark forecast is also a random walk model but takes a longer lag length — sixteen days or $Y_t = Y_{t-16} + \varepsilon_t$. Thus the exchange rate sixteen days ahead will be identical to today's exchange rate. It is logical to expect that the one-day-ahead random walk model forecast (RW1) outperforms all competition. It is hoped that at least one of the techniques and configurations used outperforms the sixteen-day random walk model forecast (RW16). It is then logical to assume that techniques and/or configurations that produce better than RW16 *ex post* forecasts should deliver more accurate *ex ante* forecasts that help in decision making. (*Ex post* forecasts are ones completed after actual values of the dependent variable are known to provide for out-of-sample validation. *Ex ante* forecasts are completed before actual values become known.)

Standard forecast evaluation criteria most forecast researchers[5–8] used in the implementation are also being used in this paper. They are

Mean absolute percent error:

$$\text{MAPE} = \frac{100}{F} \sum_{f=1}^{F} |(X_f - \hat{X}_f)/X_f| \tag{1}$$

where X_f = an exchange rate observed values to forecast *ex post*, $\hat{X}_F$ = their forecasted values, and $f = 1, \ldots, F$ periods of *ex post* forecast.

Mean absolute error:

$$\text{MAE} = \frac{1}{F} \sum_{f=1}^{F} |(X_f - \hat{X}_f)|. \tag{2}$$

Root mean squared prediction error:

$$\text{RMSPE} = \sqrt{\frac{1}{F} \sum_{f=1}^{F} (X_f - \hat{X}_f)^2}. \tag{3}$$

Theil's *U* statistics[9–10]:

$$U_1 = \frac{\text{RMSPE}_M}{\text{RMSPE}_{RW1}} \tag{4}$$

where M = model or technique used and RW1 = one-day-ahead random walk prediction. A $U_1 < 1$ implies that M's forecast outperforms that of RW1. For comparison, a U_{16} is also computed. It is defined as:

$$U_{16} = \frac{\text{RMSPE}_M}{\text{RMSPE}_{RW16}} \tag{5}$$

where RW16 = sixteen-day-ahead random walk prediction. A $U_{16} < 1$ implies that M's forecast outperforms that of RW16.

The balance of this paper contains the following: description of the data is in the next section where both the observed raw values as well as their wavelet-transformations are presented. The wavelet used to transform data of the observed exchange rates is described there as well. Section 3 contains a brief review of genetic programming and artificial neural networks. It also contains an explanation of how these two techniques are used to fit and train the wavelet-transformed sets. The results are in Sec. 4. Finally, Sec. 5 concludes.

2. The Data

All data sets used in this study are available via the Internet. They are downloadable from Oanda.com.[11] Employed data of the three exchange rates cover exactly the same period of time. They all start on December 12, 2003 and end on September 8, 2004. Of the 272 observations, 256 were used in fitting and training models. The last 16 observations (August 24–September 8, 2004) were reserved for validation. The objective is to forecast 16 more days (September 9–September 24, 2004).

Before applying any modeling technique, each observed series' predictability was estimated using a GP predictability measure.[12] The measure involves computing a statistic η that approximates the probability that an observed data set is predictable. The test statistic is:

$$\eta = \max\left\{0, \frac{1}{n}\sum_{i=1}^{n}\left(1 - \frac{\text{MSE}_Y}{\text{MSE}_S}\right)_i\right\} \tag{6}$$

where MSE = mean of squared errors (typically used as fitness function) GP models deliver, i is the number of MSE values to compare and average with $i = 1, \ldots, n \geq 30$, Y = observed series, and S = randomly shuffled sequence of the observed Y. The original sequence of series Y is randomly shuffled using Efron's bootstrap method.[13] The idea of the test is rather simple. If the series Y contains a predictable pattern, randomly shuffling the sequence of its observed values will dismember such pattern. If the series Y contains no predictable pattern, randomly shuffling the order of its values will not have any impact. Therefore, if Y were definitely predictable, $\eta \to 0$ while if it were totally random $\eta \to 1$. The statistic provides suggestions about relative predictability given that $0 \leq \eta \leq 1$. When

applied to the three exchange rate series, the results were as follows:

$$\eta_{\mathrm{DBP}} = 1.5\%$$
$$\eta_{\mathrm{DJY}} = 20.4\%$$
$$\eta_{\mathrm{DTD}} = 60.3\%.$$

These measures suggest that GP will probably be the most successful in predicting Taiwan's exchange rate (DTD), may not be able to predict either the Japanese Yen exchange rate (DJY) or the British pound exchange rate (DBP).

Available data will be used to provide six forecasts per exchange rate. Each of the six will be for the same 16 *ex post* and 16 *ex ante* forecast-days. GP will be used to fit observed data and its wavelet-transformed series then provide two forecasts: GP and W-GP. Similarly, ANN will be used to train the observed and wavelet-transformed series then provide two other forecasts: ANN and W-ANN. These four forecasts will then be compared with the two benchmark random walk forecasts RW1 and RW16.

Using wavelet-transformed data to obtain forecasts is not entirely new.[14–17] Applications using wavelets in estimation were reviewed in Ref. 18. In and by itself, wavelet analysis is not a forecasting technique. In wavelet analysis, data can be converted into forms to which it may be easier to fit models. A *wavelet transformation* is a form of data conversion. (See Ref. 19 for transformation forms.) Data conversions are carried out using any of many available wavelets (or newly created ones). The Haar and Daubechies wavelets are perhaps the most popular and are explained by many. (See Refs. 17 and 19, for example.) A wavelet transform is a scaling function used to convert a signal into father and mother wavelets. Father wavelets are representations of a signal's smooth or low-frequency component. Mother wavelets are representations of the details or high-frequency component in a signal. The Haar wavelet is the simplest to use. It is a decimated process where at each level of scaling half the number of observations disappears. When used, first it transforms a series Y_t to obtain mid-point averages (s_1) and mid-point differences (d_1) of consecutive pairs of observations (and not pairs of consecutive points). Averages preserve the main signal while differences capture the series' detailed fluctuations. In turn, mid-point averages (s_1) are transformed to obtain their mid-point averages (s_2) and their mid-point differences (d_2), and so on. Values of the mid-point averages and differences are known as "coefficients" in the literature on wavelets. Obtaining these values is known as a *discrete wavelet transform* process (*dwt*). Alternatively, *dwt* maps a vector of Y_t values to a vector of wavelet coefficients w, or

$$w = \begin{pmatrix} s_J \\ d_J \\ d_{J-1} \\ \vdots \\ d_1 \end{pmatrix} \tag{7}$$

where J is the number of scales or multiresolution components, and

$$\begin{aligned} s_J &= (s_{J,1}, s_{J,2}, \ldots, s_{J,T/2^J})' \\ d_J &= (d_{J,1}, d_{J,2}, \ldots, d_{J,T/2^J})' \\ d_{J-1} &= (d_{J,1,1}, d_{J-1,2}, \ldots, d_{J-1,T/2^J})' \\ \ldots &= \ldots. \\ d_1 &= (d_{1,1}, d_{1,2}, \ldots, d_{1,T/2^J})'. \end{aligned} \tag{8}$$

To transform a series to a maximum J components, that series must be of length $T = 2^J$. For a Haar wavelet, the averages and differences may be computed as follows:

$$\text{The averages:} \quad s_J = (s_{J-1,2t} + s_{J-1,2t-1})/2 \tag{9}$$

$$\text{The differences:} \quad d_{J,t} = (s_{J-1,2t} - -s_{J-1,2t-1})/2. \tag{10}$$

A *dwt* process of the Haar wavelet has a desirable property that may help in forecasting. Given a series' wavelet transformed coefficients s_J and $d_J, \ldots, d_1$ in (8) above, original values of that series can be reconstructed from the transformed data.[20] If a series has $T = 256$, *dwt* with $J = 3$ (for example) delivers four series: s_3 with 32 coefficients, as well as d_3, d_2, and d_1 with 32, 64, and 128 coefficients in each, respectively. For each of the four data sets (s_3, d_3, d_2, and d_1), different models can then be obtained. If the number of observations in s_3 is sufficiently large, the other sets (d_3, d_2, and d_1) must also contain a sufficient number of observations to which one can fit models. Therefore, to obtain meaningful models, the scaling level J must be selected such that s_J contains that number of observations needed to obtain a reasonable equation. That number depends on the technique used.

Basically, the idea of using these four series amounts to adopting a "divide and conquer strategy". Figures 1–3 show the *dwt* (s_3, d_3, d_2, and d_1) and *idwt* (the reconstructed sequence) of DBP, DJY, and DTD with scaling level $J = 3$.

Each component of the transformed sequences should have a different level of complexity. The series containing s_3 data basically represents the clearest signal that may be contained in the observed variable, and therefore is usually the least complex or most predictable. Series d_3 captures what may be viewed as systematic changes in the signal, and therefore may be fairly predictable as well. Series d_2 may be more complex because it captures changes in the averages. Noise is captured mostly in series d_1. For DBP, $\eta = 0.57$, 0.27, 0.06, and 0.00 for series s_3, d_3, d_2, and d_1, respectively. For DJY, $\eta = 0.64$, 0.14, 0.00, and 0.00 for series s_3, d_3, d_2, and d_1, respectively. For DTD, $\eta = 0.83$, 0.04, 0.47, and 0.19 for series s_3, d_3, d_2, and d_1, respectively. Given these complexity indicators, it is clear that GP should produce reasonable fits and forecasts of the father wavelets but only some of the mother wavelets. Fits and forecasts of the original series using the inverse *dwt* process (*idwt*) are easily obtained after modeling each of these series.

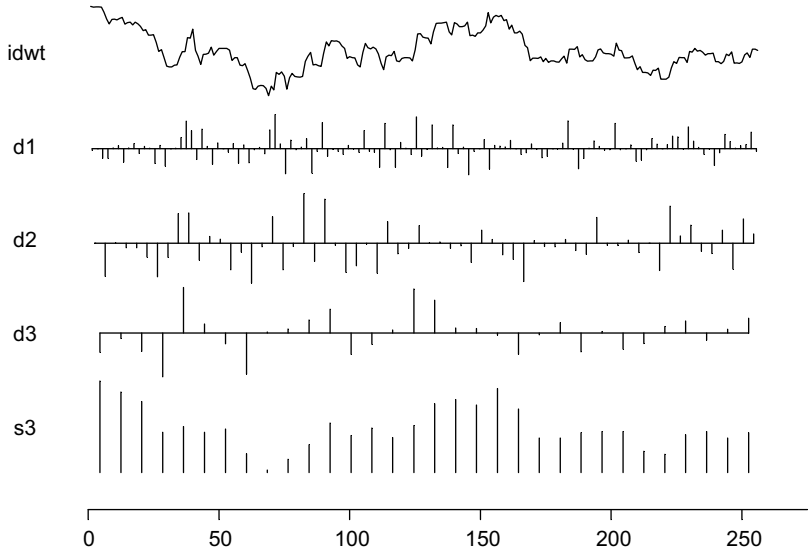


Fig. 1. Plots of *dwt* (s_3, d_3, d_2, & d_1) and inverse (idwt) of DBP. In this graph *idwt* is a reconstruction of the originally observed data.

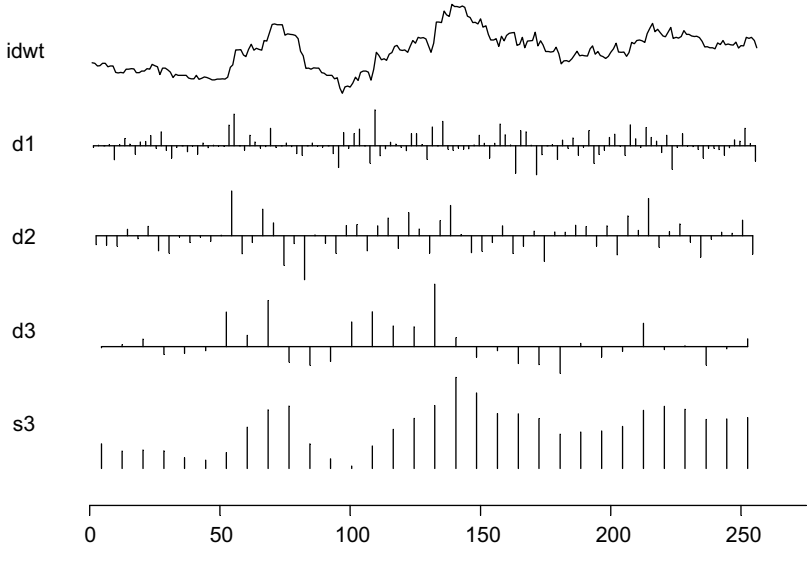


Fig. 2. Plots of *dwt* (s_3, d_3, d_2, & d_1) and inverse (idwt) of DJY. In this graph *idwt* is a reconstruction of the originally observed data.

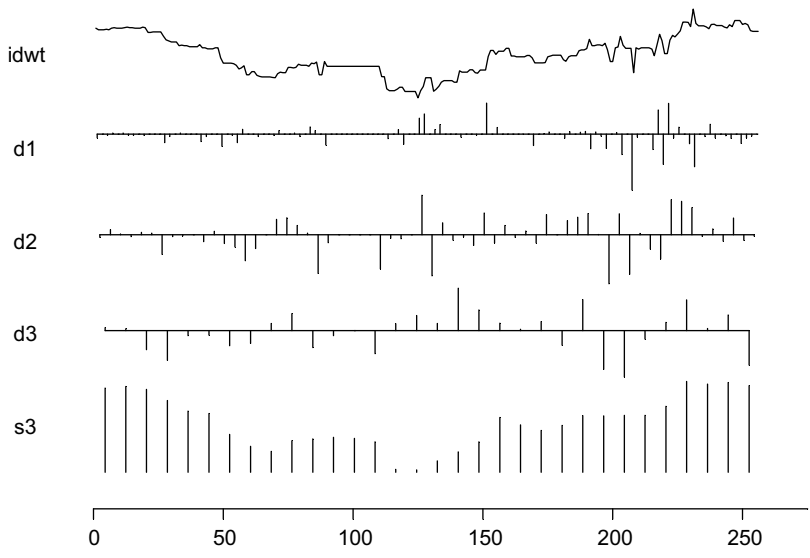


Fig. 3. Plots of *dwt* (s_3, d_3, d_2, & d_1) and inverse (idwt) of DTD. In this graph *idwt* is a reconstruction of the originally observed data.

3. Methodology

Using observed values of each of the exchange rate, equations are evolved and training is completed assuming a delayed autoregressive specification where the dependent variable

$$Y_t = f(Y_{t-L}, Y_{t-L-1}, \ldots, Y_{t-2L}), \tag{11}$$

where L = number of lags and $t = 1, \ldots, T$. Observations $f = T+1, \ldots, T+F$ then identify the *ex post* predictions to validate the efficacy of the forecast. Further and given that actual values at f are known, it is possible to produce *ex ante* predictions for F more periods. This structure means that for any of the untransformed series containing 256 observations, observations $t = 17, \ldots, 240$ are the data to fit and train. Observations $\mathcal{L} = 1, \ldots, L$, where $L = 16$ are lost degrees of freedom that provide the explanatory or lagged dependent variables. It is then possible to forecast $F = 16$ periods ahead using only actual values as inputs. Observations $f = 241, \ldots, 256$ are utilized to validate the *ex post* prediction and observations $f = 257, \ldots, 272$ are the *ex ante* predictions.

The next two subsections contain summary descriptions of the techniques employed (GP and ANN). Readers familiar with them may skip to subsec. 3.3.

3.1. *Genetic programming*

GP is a computerized optimization technique employed to solve diverse problems in different disciplines. Foundations of GP are in Ref. 21 and description of how

GP is used in forecasting and its statistical properties are in Ref. 22. The GP software used in this study is TSGP[22] written in C++ for Windows environment. It takes two types of input files: data files and a configuration file. Data input files contain values of the dependent and each of the independent variables. The configuration file contains execution information such as: name of the dependent variable, number of observations to fit, number of independent variables, number of observations to forecast, number of equation specifications to evolve, and other GP-specific parameters. (See Ref. 23 for more on these parameters and their selection.)

TSGP is designed to randomly assemble an initial population of individual specifications (say 1,000), computes their fitness, and then breeds new equations as members of a new generation with the same population size. An individual specification is represented by a tree consisting of nodes connected with arcs. The inner nodes contain mathematical operators (such as $+, -, *, /$, sin, cos, etc.). A tree continues to grow until end nodes contain explanatory variables or a constant term. A node containing a variable or constant is thus known as a "terminal". A new population is bred using mutation, crossover, and self reproduction. Fitter equations in a population get a higher chance to participate in breeding. In mutation, TSGP is designed to randomly assemble a sub-tree that replaces a randomly selected existing part of a tree. In crossover, randomly selected parts of two existing trees are swapped. In self reproduction, a top percentage of the fittest individuals in an existing population are passed on to the next generation. The idea is to continue generating new populations while preserving good genes. After completing a specified number of generations, the program terminates and saves to an output file the specification that captures the dynamics of a series best. The best equation is then used to forecast that series' future values. Actual and fitted values as well as residuals and standard evaluation statistics (R^2, MSE, and *ex post* prediction MSE) are written to a different output file. Figure 4 contains a representation of the GP architecture used.

GP-resulting specifications may be viewed as coincidental equations that may capture the dynamics of a process. As such, GP is useful only when the interest is in obtaining a "good" forecast. GP models are not helpful in policy-making or conducting "what if" type of analysis, but they may help in planning. Coefficients in GP-evolved equations are not computed. They are computer-generated random numbers (between -128 and 127). This gives GP an advantage over conventional statistical regressions. Given that there are no coefficients to compute, GP is robust against problems of multicollinearity, autocorrelation, and heteroscedasticity. There are also no degrees of freedom lost to compute coefficients. However, because variables and operators selected to assemble the equations are random, while attempting to breed the fittest individual equation, the program occasionally gets trapped in a local minimum MSE rather than a global one within the search space. It is therefore necessary to generate a large number of equations and then select the best one(s) to use in forecasting. For each exchange rate 200 GP searches for a best equation were completed. The best 200 are then sorted to identify the best of the best to use in forecasting. Once the 200 runs were complete, two steps are followed to select the

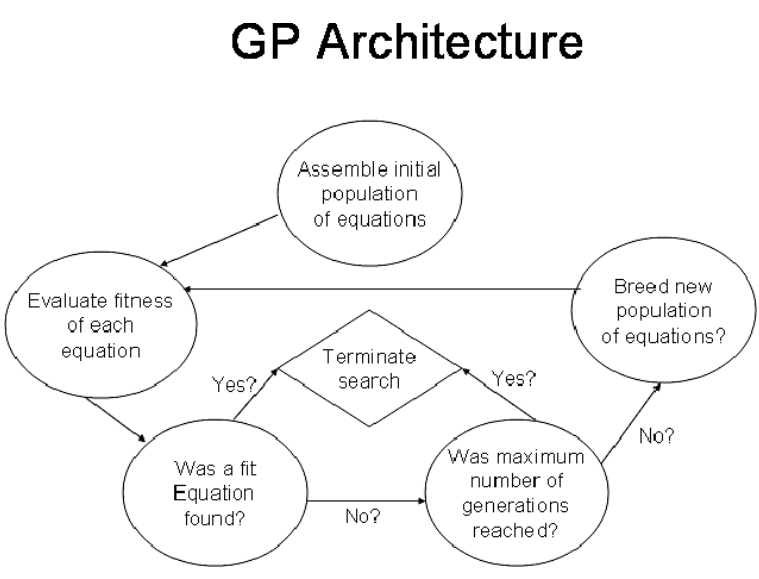


Fig. 4. Flow diagram representing the GP architecture used.

best of the best forecasting model. Summary results of the 200 best models TSGP produces are first sorted according to the lowest historical MSE. Equations with the 10 lowest MSE (where the number 10 is arbitrarily set) are sorted in an ascending order according to *ex post* forecast fitness. (MAPE is used in this study. Prediction MSE may be used instead.) The fittest equation among the 10 (one with the lowest MAPE) is then selected to compute the forecast.

3.2. *Artificial neural networks*

ANN is a well-established information-processing paradigm based on the way the densely interconnected parallel structure of the human brain processes information. The technique can be used to detect structure in time-series. A network is a collection of mathematical models that emulates the nervous systems and draws on the analogies of adaptive learning. Input data is presented to the network that learns to predict future outcomes. Complete description of types of networks to choose from and an explanation of how ANN can be used in forecasting are abundant. (See Ref. 24 for more details.) In time series forecasting, and similar to GP, it is assumed that future values of a series are determined from its own past observations, a standard autoregressive model structure. Figure 5 is a representation of a typical network used in mapping a function similar to that in Eq. (11).

NeuroSolutions[24] is the software used here in training the networks and forecasting the three exchange rates. Basically, two network configurations — multilayer perceptions (MLP) and generalized feed forward (GFF) — were employed in training models to forecast exchange rates. MLP is a layered feed forward network where

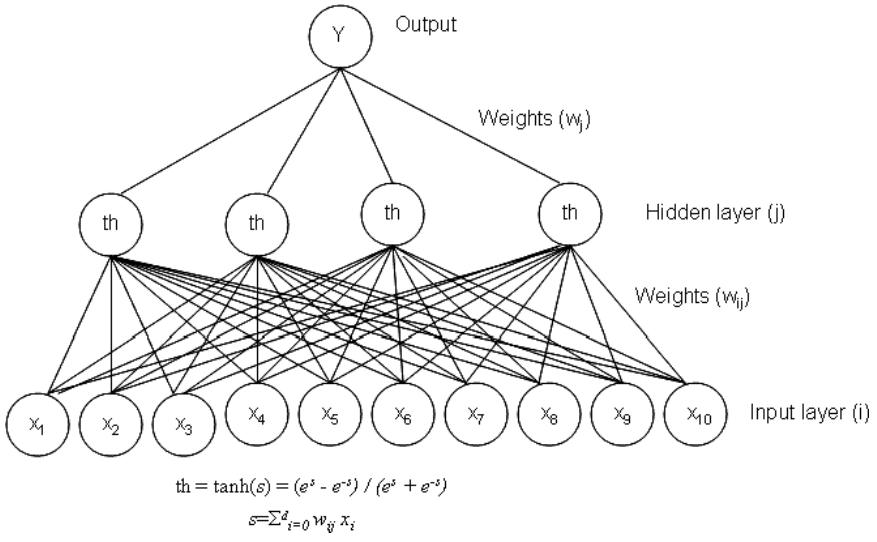


Fig. 5. MLP network used to generate the final forecast.

the first layer contains the inputs. These inputs are the lagged dependent variables. The last layer delivers output or problem solution. The middle layer(s) (which may be one or more) are hidden and contain weights and node biases estimated during a network training process. MLP usually produces good approximations. It is typically trained with static backpropagation. GFF is a generalization of MLP with connections that can jump over one or more layers. Generally, artificial neural networks are well known for their ability to replicate complex dynamics fairly well.

3.3. *Fitting and training of wavelet transformations*

In addition to GP-fitting and ANN-training of the observed exchange rates sequences, GP fits four additional models and ANN trains four additional networks, one for each of the four *dwt* series representing each sequence. Given that each of the four transformations has a different level of complexity, each should have a model with a unique specification. The model specification assumed for s_3 is:

$$s_{3,t} = f(s_{3,t-2}, s_{3,t-3}, s_{3,t-4}, s_{3,t-5}, s_{3,t-6}). \tag{12}$$

According to (12), s_3 is assumed to be a function of five distant (starting with the second lag) but consecutive lagged values. Similarly,

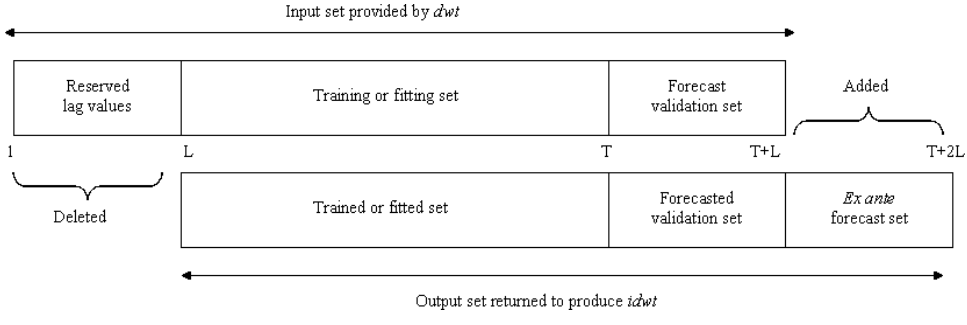
$$d_{3,t} = f(d_{3,t-2}, \ldots, d_{3,t-6}); \tag{13}$$

$$d_{2,t} = f(d_{2,t-4}, \ldots, d_{2,t-10}); \tag{14}$$

$$d_{1,t} = f(d_{1,t-8}, \ldots, d_{1,t-16}). \tag{15}$$

Once the four models are obtained, they are used to compute fitted values and forecasts. Each reconstructed series has the same number of observations as its

original data set except that t is shifted forward to include forecast periods. To obtain fitted and forecasted values, the decimation process to use with *dwt* and its inverse must satisfy the condition of perfect reconstruction. This means that the number of observations in each *dwt* series must remain the same to produce *idwt* — the reconstructed series. Accordingly, to reconstruct the series, the number of observations T is shifted forward by the desired number of observations to forecast *ex ante*. More specifically, s_3 and d_3 originally have 32 observations. Each will lose 1 degree of freedom for the delayed lag start, lose 5 degrees of freedom in lagged dependent variables, and is reduced by 2 more observations to forecast leaving 24 for training or fitting models. The resulting models will then produce the 24 fitted and 4 forecast values. Since in *idwt* reconstruction, the number of observations must be restored to 32 observations, four actual values are added at the front end of the set. The number of observations used to fit and forecast are doubled for d_2, then doubled again for d_1. Using this method, Eqs. (12) and (13) provide a four-step-ahead forecast (two *ex post* and two *ex ante*), Eq. (14) provides a eight-step-ahead forecast, and Eq. (15) provides a 16-step-ahead forecast. The final forecast delivered by the inverse *dwt* is 32 steps ahead. In other words, starting points are shifted forward by the number of steps ahead desired to forecast and the same number of observations are deleted from the historic data used in modeling leaving outcomes with an equal number of observations the *dwt* originally delivered. The following chart explains the process:



Both *dwt* and *idwt* are simple to obtain using ***S+Wavelets***. Here are the commands to obtain the DBP wavelet transformations s_3, d_3, d_2, and d_1:

```
> module(wavelets)
> DBP.dwt <- scan("DBP.txt")
> write((DBP.dwt[["s3"]]),ncol=1,"s3.txt")
> write((DBP.dwt[["d3"]]),ncol=1,"d3.txt")
> write((DBP.dwt[["d2"]]),ncol=1,"d2.txt")
> write((DBP.dwt[["d1"]]),ncol=1,"d1.txt")
```

The files s3.txt, d3.txt, d3.txt, and d1.txt contain the series to fit and train. Fitness and training solutions are then scanned back to replace the original wavelet

transformations s_3, d_3, d_2, and d_1. Here are the commands to replace them before reconstructing the series:

> soluts3 <- scan("DBPs3_Sol.txt")
> solutd3 <- scan("DBPd3_Sol.txt")
> solutd2 <- scan("DBPd2_Sol.txt")
> solutd1 <- scan("DBPd1_Sol.txt")
> DBP.dwt[["s3"]] <- soluts3
> DBP.dwt[["d3"]] <- solutd3
> DBP.dwt[["d2"]] <- solutd2
> DBP.dwt[["d1"]] <- solutd1.

DBPs3_Sol.txt, DBPd3_Sol.txt, etc. are output files either GP or ANN delivers. Before reconstructing the series, values scanned in are then renamed by names originally given to the *dwt* components to overwrite the original "DBP.dwt". Here are the commands to reconstruct and write the GP solution to a text file:

> solution_DBP <- reconstruct(DBP.dwt),
> write(solution_DBP, ncol=1, "Haar_W-GP_Sol_DBP.txt").

If the solution was obtained using ANN, "Haar_W-GP_Sol_DBP.txt" in the last command should be replaced by a different file name indicating that it is an ANN output such as "Haar_W-ANN_Sol_DBP.txt". To obtain solutions for DJY and DTD, the process described above is simply repeated using the appropriate data sets and notations.

4. Results

Although out-of-sample forecasts are the focus of this section, in-sample results must be discussed first. *Ex post* out-of-sample results follow. These consist of forecasts for sixteen periods whose outcome were known at the time this investigation started. Statistical evaluations of the *ex ante* forecasts were also possible and are presented below because actual values of the *ex ante* period became available by the time this research was completed.

4.1. *GP and W-GP*

GP parameters used in evolving models representing all series were identical to perhaps deliver more consistent comparison. Table 1 is a basic Koza Tableau[21] that contains the key parameters used.

Although GP does not produce the best historical fits, it tends to produce great forecasts. GP and *idwt* of wavelet-GP (W-GP) fitted values of the DBP, DJY, and DTD are in Figs. 6–8, respectively. All evolved GP equations are in an Appendix. Table 2 contains related statistics.

The results in Table 2 suggest that (a) GP and W-GP were equally successful in fitting DBP, (b) W-GP fitted DJY better than GP, and (c) GP fitted DTD

Table 1. Basic Koza Tableau containing run parameters.

Parameter	Value
Population size	1000
Maximum number of generations	120
Mutation rate	0.6
Crossover rate	0.3
Cross-self rate	0.1
Operators	+, −, ∗ %, sin, & cos
Selection method	Roulette wheel
Maximum tree depth	100
Fitness measure	MSE

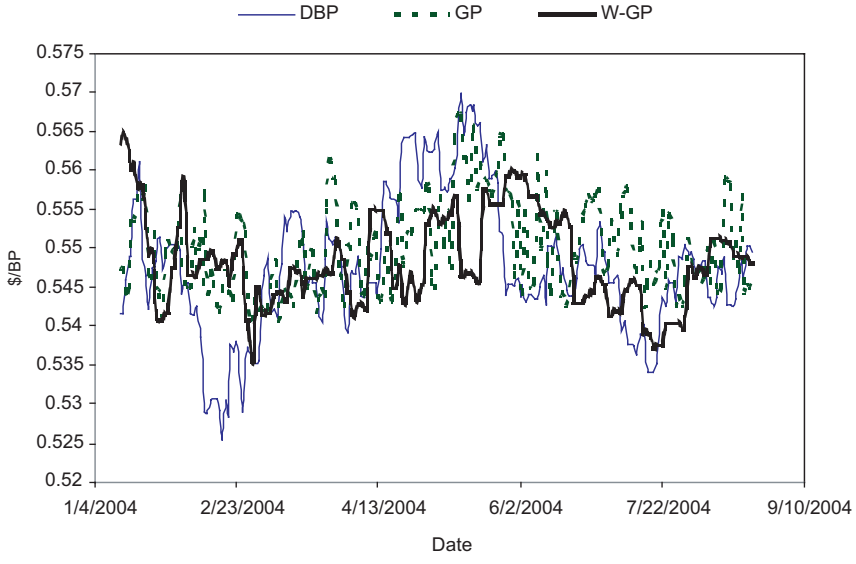


Fig. 6. Actual versus filled DBP exchange rate.

better than W-GP. The standard normal z at the bottom of the table tests equality between means of independent samples. The test statistic is:

$$z = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{16}$$

where $\bar{X}_1$ and $\bar{X}_2$ represent the means of W-GP and GP for the same exchange rate appearing in the same column and the denominator contains their pooled standard error. For each exchange rate fitness measure, the statistic is used to independently

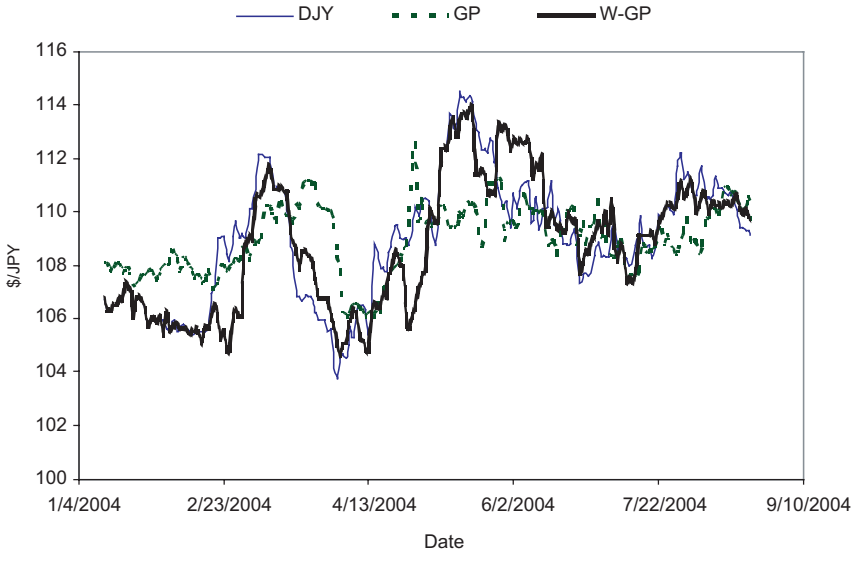


Fig. 7. Actual versus filled DJY exchange rate.

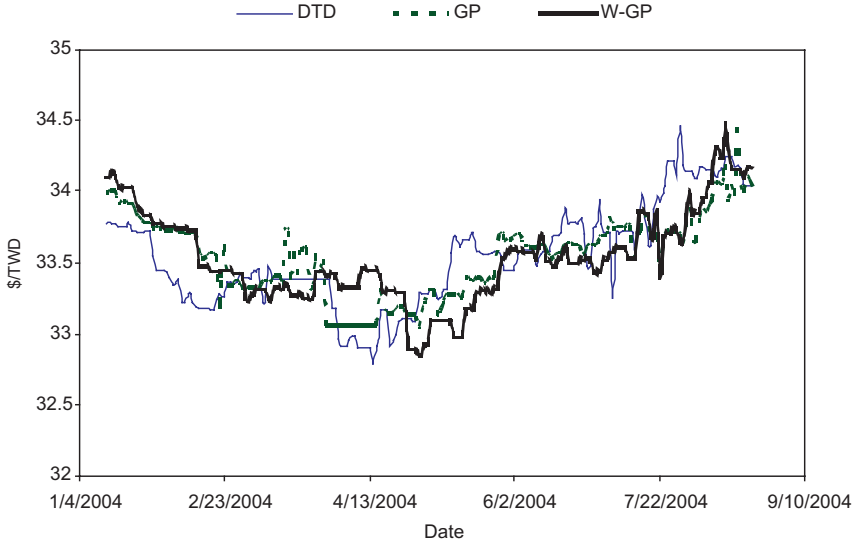


Fig. 8. Actual versus fillet DTD exchange rate.

test the null and alternative hypotheses:

H_o: W-GP MAPE (exchange rate) = GP MAPE (same exchange rate)
H_a: W-GP MAPE (exchange rate) $\neq$ GP MAPE (same exchange rate).

Table 2. GP and W-GP fitting statistics.

	MAPE	MAE	MSE
GP:			
DBP	1.27	0.01	0.01
DJY	1.45	1.58	4.10
DTD	0.55	0.19	0.06
W-GP:			
DBP	1.37	0.01	0.01
DJY	0.90	0.98	1.79
DTD	0.71	0.24	0.08
z (DBP)	1.20	0.95	1.02
z (DJY)	−5.06	−5.48	−5.46
z (DTD)	3.15	3.44	3.40

The results are consistent across but not between the exchange rates. The null is rejected for DJY suggesting that DJY was better fitted by W-GP. For DTD, the results suggest that DTD was better fitted by GP. The null was not rejected for DBP suggesting no difference.

Figures 9–11 show actual values against *ex post* and *ex ante* forecasts. Figure 9 confirms that although GP's *ex post* forecast seems marginally better than that the W-GP *ex post* forecast, both produced terrible *ex ante* forecasts. Figure 10 suggests that W-GP outperformed GP in *ex post* as well as *ex ante* forecasting of DJY. Figure 11 shows GP's relatively better forecast than W-GP of DTD both *ex post* and *ex ante.* GP's success in forecasting DTD is consistent with the predictability results calculated earlier where $\eta = 60.3\%$ and suggests that there seems to be no

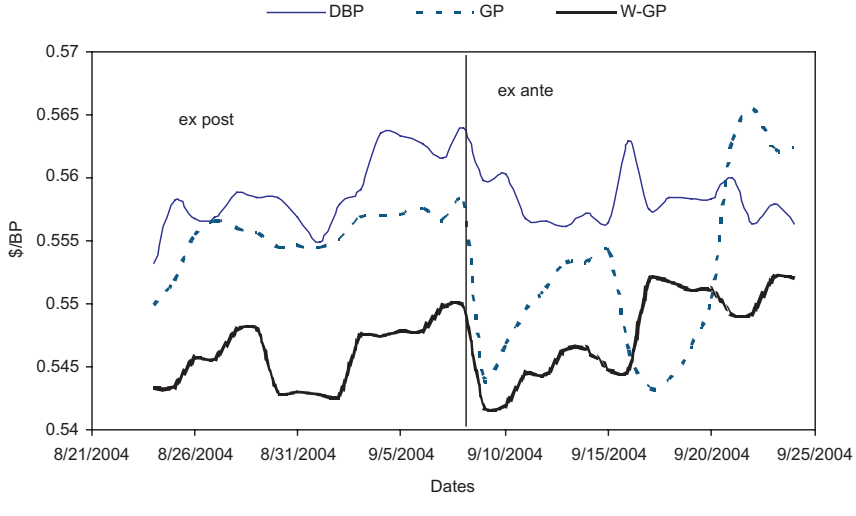


Fig. 9. Actual versus DBP *ex post* and *ex ante* forecasts.

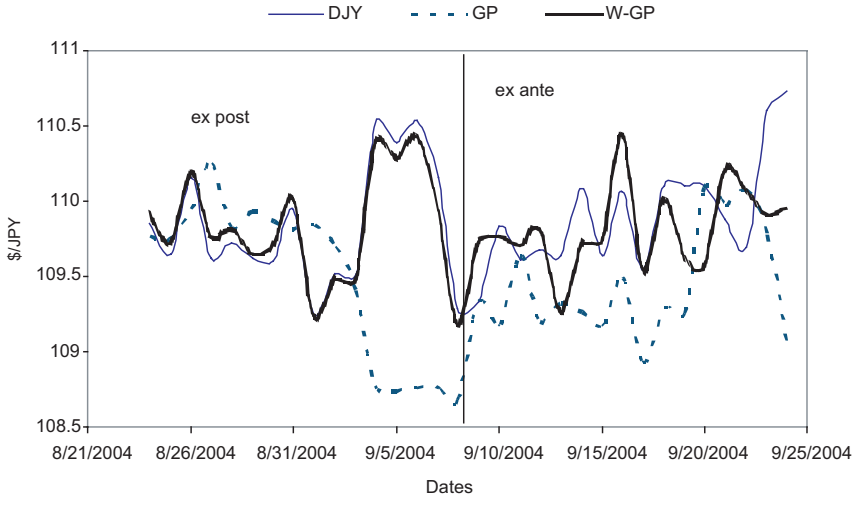


Fig. 10. Actual versus DJY *ex post* and *ex ante* forecasts.

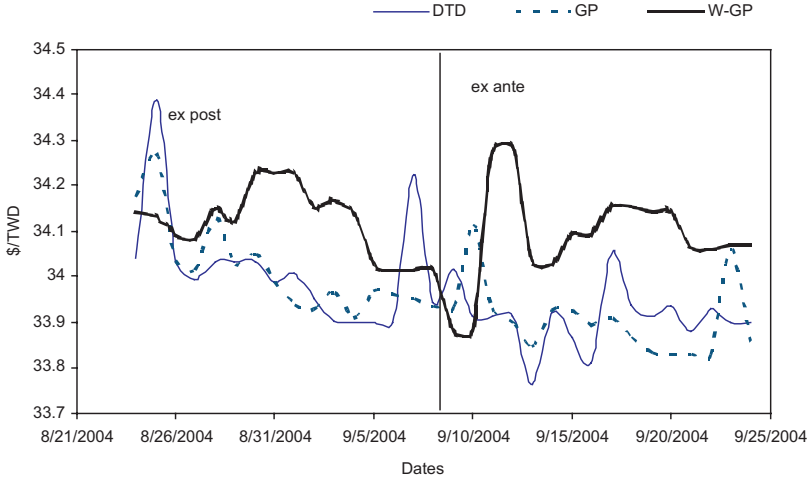


Fig. 11. Actual versus DTD *ex post* and *ex ante* forecasts.

relationship between predictability of observed data and their wavelet-transformed sets. Table 3 presents forecast statistics confirming observations these graphs suggest. Table 3 also provides the statistics for the benchmark comparisons. It shows that GP outperformed RW1 and RW16 in forecasting DTD, while W-GP outperformed RW1 and RW16 in forecasting DJY.

Table 4 contains the *ex ante* forecast statistics using GP and W-GP forecasting. They confirm GP's better ability in predicting DTD and W-GP's better ability in

Table 3. GP and W-GP *ex post* forecast statistics.

	MAPE	MAE	RMSPE	Theil's U_1	Theil's U_{16}
GP:					
DBP	1.50	0.01	0.01	4.35	2.51
DJY	0.87	0.62	1.03	10.04	1.65
DTD	0.18	0.16	0.09	0.55	0.52
W-GP:					
DBP	2.33	0.01	0.01	5.59	3.22
DJY	0.34	0.38	0.48	4.72	0.77
DTD	0.52	0.10	0.19	1.14	1.07

Table 4. GP and W-GP *ex ante* forecast statistics.

	MAPE	MAE	RMSPE	Theil's U_1	Theil's U_{16}
GP:					
DBP	1.54	0.01	0.01	3.49	2.49
DJY	0.49	0.54	0.68	1.76	1.21
DTD	0.24	0.08	0.10	1.00	0.54
W-GP:					
DBP	1.86	0.01	0.01	3.98	2.84
DJY	0.29	0.32	0.39	1.02	0.70
DTD	0.59	0.20	0.22	2.26	1.23

predicting DJY documented in Table 3. They also show that GP still outperforms RW16 in forecasting DTD and W-GP still does better than RW16 in forecasting DJY.

4.2. *ANN and W-ANN*

ANN produces superior historical fits but it tends to produce weaker forecasts. Weaker forecasts occur due to the potential problem of overfitting[7] and incorrect network selection.[5] Forecasts reported here are outcomes of trial and error. Training parameters selected when searching for the best network to forecast each exchange rate variable were modified until the best forecasting model was obtained. Run parameters selected in each case are in Table 5.

Fits of the DBP, DJY, and DTD are in Figs. 12–14, respectively. They also suggest that benefits from wavelet transforming of the observed series were evident in training DJY only. More specifically, Fig. 12 suggests that ANN was more successful than W-ANN in fitting DBP, Fig. 13 suggests that W-ANN fitted DJP better than ANN, and Fig. 14 shows that ANN fitted DTD better than W-ANN. Table 6 contains related statistics. The results in Table 6 are consistent with those GP produced. At the bottom of the table, the standard normal z computations that test equality between means of independent samples confirm rejection of the null

Table 5. Final neural networks architectures selected.

	Network Type	Hidden Layers	Transfer Function*	Maximum Training Epochs	Learning Momentum
ANN:					
DBP	GFF	2	Tan	10000	0.9
DJY	GFF	1	Tan	9000	0.9
DTD	MLP	2	Tan	7500	0.9
W-ANN:					
DBP					
$s3$	MLP	2	Tan	3000	0.7
$d3$	MLP	1	Tan	1800	0.7
$d2$	MLP	2	Tan	9000	0.7
$d1$	MLP	3	Tan	3400	0.9
DJY					
$s3$	MLP	2	Tan	6000	0.9
$d3$	MLP	2	Tan	2500	0.7
$d2$	MLP	2	Tan	3200	0.8
$d1$	MLP	3	Tan	20000	0.7
DTD					
$s3$	MLP	1	Tan	4200	0.8
$d3$	MLP	1	Tan	4000	0.7
$d2$	GFF	2	Tan	16800	0.8
$d1$	MLP	2	Tan	10000	0.7

*Tan = TanAxon.

one way or the other in all three cases. The null hypothesis [H_o: W-ANN MAPE (exchange rate) $\geq$ ANN MAPE (same exchange rate)] is rejected in the case of DJY only. These results suggest that DJY was better fitted by W-GP while DBP and DTD were better fitted by GP.

ANN forecasts are presented in Figs. 15–17. These figures comparing the three exchange rates' actual values with *ex post* and *ex ante* forecasts suggest poor forecasting in all cases. The results reported in Table 7 that contains *ex post* forecast evaluation statistics suggest that RW1 as well as RW16 forecasts outperformed all neural networks forecasts. The *ex ante* forecast results in Table 8 were not much better.

4.3. *RW1 and RW16*

Although presenting the results on the two benchmark forecasts RW1 and RW16 may not be of importance, they are presented here to show that models that deliver reliable *ex post* forecasts will most probably deliver reliable *ex ante* forecasts. Table 9 summarizes the results of all *ex post* forecasts. The four standard performance statistics (MAPE, MAE, RMSPE, and Theil's U) are across from each of the six techniques employed. The last two columns contain the Theil's U_{16} (to have a fair comparison in the case of the 16-period extended forecast) and the ranking of their

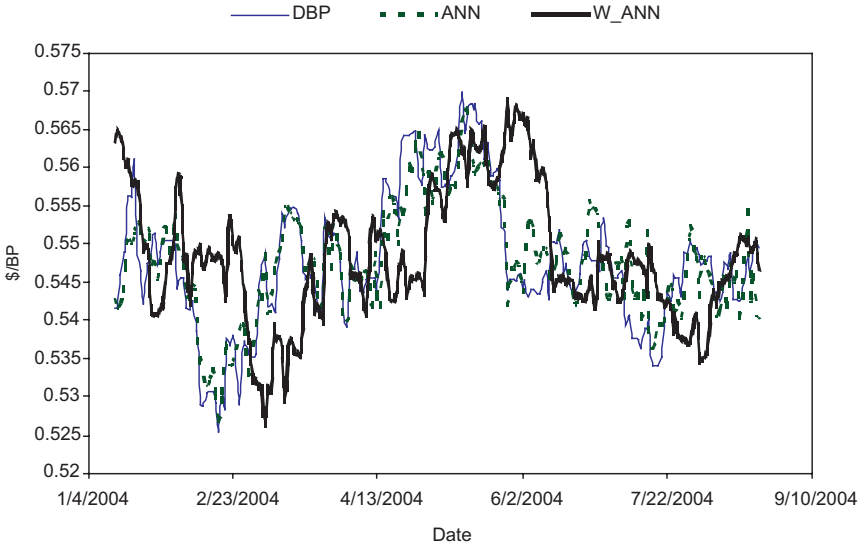


Fig. 12. Actual versus filled DBP exchange rate.

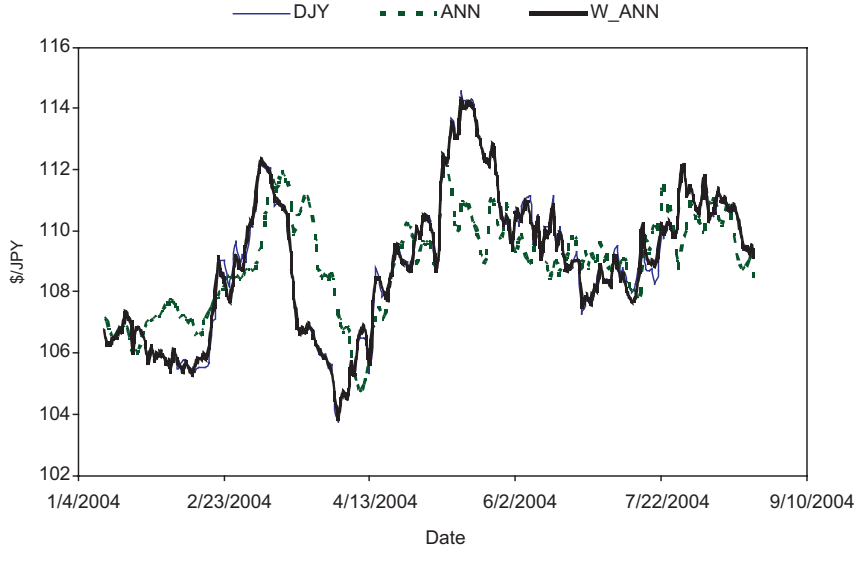


Fig. 13. Actual versus filled DJY exchange rate.

relative RMSPE, where a Rank = 1 belongs to the best forecasting technique, a Rank = 2 belongs to the second best, and so on. Table 10 has identical structure but provides a summary of the outcomes of the *ex ante* forecasts. Interestingly, the ranking columns are identical. More importantly, these tables confirm (i) GP's

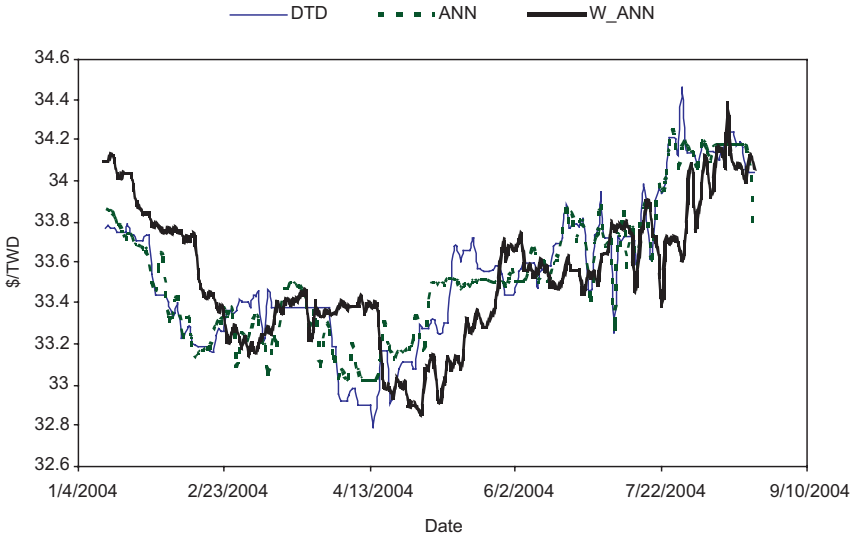


Fig. 14. Actual versus filled DTD exchange rate.

Table 6. ANN and W-ANN fitting statistics.

	MAPE	MAE	MSE
ANN:			
DBP	0.57	0.00	0.00
DJY	1.18	1.29	2.75
DTD	0.25	0.08	0.01
W-ANN:			
DBP	1.52	0.01	0.00
DJY	0.16	0.17	0.05
DTD	0.65	0.22	0.07
z (DBP)	9.31	10.51	10.51
z (DJY)	−9.65	−15.27	−15.25
z (DTD)	9.01	10.87	10.89

superiority in forecasting DTD, and (ii) W-GP outperforming RW16 in the case of DJY.

5. Conclusions

This paper investigated the notion of GP-fitting and ANN-training of different temporal resolutions of an observed set of data to obtain forecasts of daily exchange rates that can be used in decision making. The Haar wavelet transform produced the different temporal resolutions. The Haar wavelet decomposes observed values into low frequency signals and high frequency variations. It was selected because it is

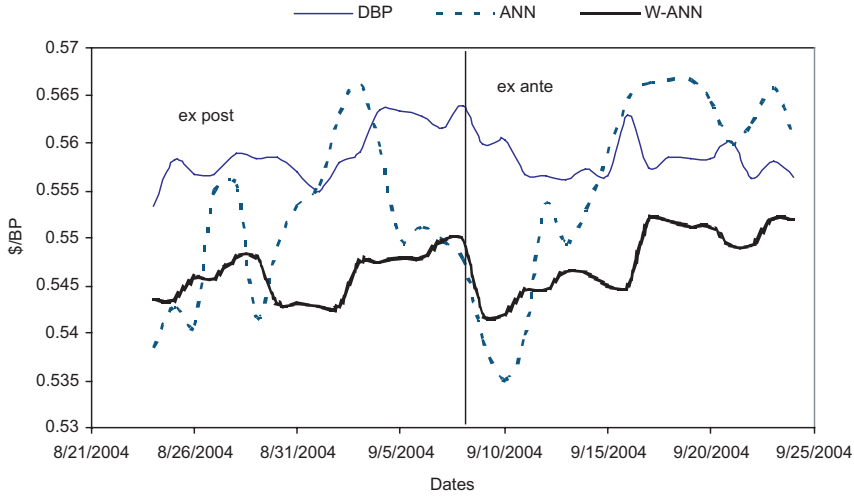


Fig. 15. Actual versus DBP *ex post* and *ex ante* forecasts.

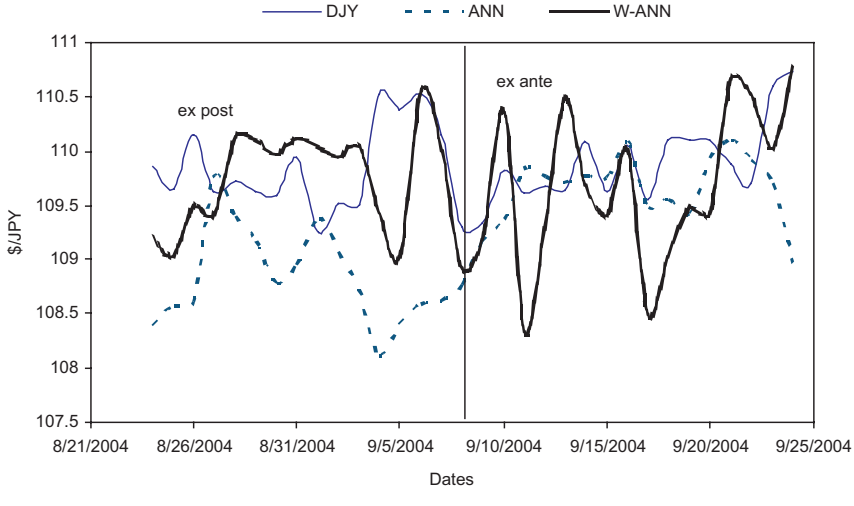


Fig. 16. Actual versus DJY *ex post* and *ex ante* forecasts.

a discrete decimation process that through inversion delivers perfect reconstruction of the signal.

The notion was applied to produce GP and ANN extended forecasts of three exchange rates: US dollar/British pound, US dollar/Japanese yen, and US dollar/Taiwan dollar. Six forecasts of each exchange rate data were obtained using genetic programming, artificial neural networks, one-step-ahead random walk modeling, and finally using sixteen-step-ahead random walk modeling. Genetic

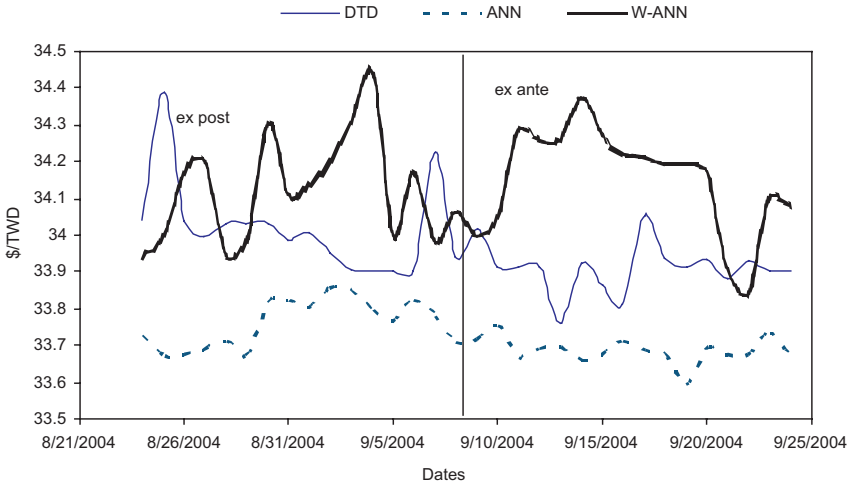


Fig. 17. Actual versus DTD *ex post* and *ex ante* forecasts.

Table 7. ANN and W-ANN *ex post* forecast statistics.

	MAPE	MAE	RMSPE	Theil's U_1	Theil's U_{16}
ANN:					
DBP	1.70	0.01	0.01	4.65	2.68
DJY	0.56	0.96	0.88	8.57	1.41
DTD	0.63	0.21	0.25	1.51	1.43
W-ANN:					
DBP	2.34	0.01	0.01	5.57	3.21
DJY	0.94	1.04	1.24	12.13	1.99
DTD	0.76	0.26	0.30	1.84	1.74

Table 8. ANN and W-ANN *ex ante* forecast statistics.

	MAPE	MAE	RMSPE	Theil's U_1	Theil's U_{16}
ANN:					
DBP	1.46	0.01	0.01	3.74	2.67
DJY	0.34	0.38	0.57	1.49	1.02
DTD	0.66	0.22	0.24	2.45	1.33
W-ANN:					
DBP	1.62	0.01	0.01	3.86	2.76
DJY	0.54	0.60	0.70	1.81	1.24
DTD	0.75	0.25	0.29	3.01	1.64

programming and artificial neural networks were employed once to fit and train the observed data and another to fit and train their wavelet-transformations. Before applying GP or ANN to the data, a genetic programming predictability test was used. It helped formulate some expectations about the relative complexity

Table 9. Summary statistics of all *ex post* forecasts.

	MAPE	MAE	RMSPE	Theil's U_1	Theil's U_{16}	Rank
DBP ($\eta = 1.5\%$):						
GP	1.5031	0.0131	0.0103	4.35	2.51	3
W-GP	2.3334	0.0084	0.0133	5.59	3.22	6
ANN	1.6962	0.0095	0.0110	4.65	2.68	4
W-ANN	2.3351	0.0131	0.0132	5.57	3.21	5
RW1	0.3300	0.0018	0.0024	1.00	0.58	1
RW16	0.6412	0.0036	0.0041	1.74	1.00	2
DJY ($\eta = 20.4\%$):						
GP	0.8700	0.6181	1.0285	10.04	1.65	5
W-GP	0.3420	0.3756	0.4829	4.72	0.77	2
ANN	0.5613	0.9556	0.8777	8.57	1.41	4
W-ANN	0.9414	1.0362	1.2417	12.13	1.99	6
RW1	0.0783	0.0861	0.1024	1.00	0.16	1
RW16	0.4758	0.5228	0.6232	6.09	1.00	3
DTD ($\eta = 60.3\%$):						
GP	0.1779	0.1611	0.0918	0.55	0.52	1
W-GP	0.5181	0.0973	0.1886	1.14	1.07	4
ANN	0.6252	0.2127	0.2504	1.51	1.43	5
W-ANN	0.7555	0.2576	0.3049	1.84	1.74	6
RW1	0.2851	0.0607	0.1656	1.00	0.94	2
RW16	0.4734	0.1762	0.1757	1.06	1.00	3

Table 10. Summary statistics of all *ex ante* forecasts.

	MAPE	MAE	RMSPE	Theil's U_1	Theil's U_{16}	Rank
DBP ($\eta = 1.5\%$):						
GP	1.5390	0.0086	0.0099	3.49	2.49	3
W-GP	1.8556	0.0104	0.0113	3.98	2.84	6
ANN	1.4557	0.0081	0.0106	3.74	2.67	4
W-ANN	1.6160	0.0090	0.0109	3.86	2.76	5
RW1	0.3669	0.0021	0.0028	1.00	0.71	1
RW16	0.5652	0.0032	0.0040	1.40	1.00	2
DJY ($\eta = 20.4\%$):						
GP	0.4877	0.5370	0.6773	1.76	1.21	5
W-GP	0.2938	0.3234	0.3921	1.02	0.70	2
ANN	0.3439	0.3791	0.5739	1.49	1.02	4
W-ANN	0.5424	0.5959	0.6965	1.81	1.24	6
RW1	0.2670	0.2937	0.3856	1.00	0.69	1
RW16	0.4043	0.4450	0.5605	1.45	1.00	3
DTD ($\eta = 60.3\%$):						
GP	0.2390	0.0811	0.0963	1.00	0.54	1
W-GP	0.5882	0.1994	0.2184	2.26	1.23	4
ANN	0.6604	0.2241	0.2363	2.45	1.33	5
W-ANN	0.7468	0.2531	0.2910	3.01	1.64	6
RW1	0.2088	0.0708	0.0965	1.00	0.54	2
RW16	0.3591	0.1217	0.1774	1.84	1.00	3

of the series. The test results helped formulate predictability expectations about the different exchange rates. The US dollar/British pound exchange rate was only 1.5% predictable, the US dollar/Japanese yen was 20.4% predictable, but US dollar/Taiwanese dollar was 60.3% predictable.

Using consistent autoregressive specifications, data of the exchange rates were independently fit using fixed GP parameters and consistent ANN configurations. To evaluate modeling and forecasting accuracy of the computational techniques used, focus was mainly on comparing *ex post* and *ex ante* forecast statistics. According to those statistics, genetic programming forecasted the US dollar/Taiwanese dollar best. Its root mean squared prediction error was about half that of one-step-ahead and sixteen-step-ahead random walk forecasts. Random walk predictions (one- and sixteen-steps-ahead) outperformed genetic programming forecasts obtained for the other two exchange rates as well as all forecasts artificial neural networks produced. The use of wavelet-transformed data helped only in forecasting US dollar/Japanese yen. Only for this exchange rate that genetic programming forecasts using the transformed data outperformed the sixteen-step-ahead random walk predictions.

Although results are mixed, they suggest that the use of wavelet-transformed data as input may help model and forecast complex dynamics. Research involving applying other computational techniques to obtain better extended forecasts using wavelet-transformed data may be more successful. The results reported in this study show that chances are that each dynamical process is probably different and may demand a different technique to produce a reasonable forecast of its future dynamics.

Appendix

GP evolved models of original series:

$$
\begin{aligned}
\mathrm{DBP}_t = {} & \cos(\sin((\sin(\cos(\cos((\mathrm{DBP}_{t-16} \\
& / \cos((\mathrm{DBP}_{t-16} / \cos(\sin((\cos(\cos((\cos(\mathrm{DBP}_{t-29}) * (\cos(\sin(\mathrm{DBP}_{t-19})) \\
& / (\cos(\cos(\mathrm{DBP}_{t-16}))/ -122))))) + (\mathrm{DBP}_{t-23} + \mathrm{DBP}_{t-19})))))))))) \\
& + \mathrm{DBP}_{t-16} / \cos(\sin((\cos(\cos((\cos(\mathrm{DBP}_{t-16}) \\
& * (\cos((\mathrm{DBP}_{t-23} + \mathrm{DBP}_{t-19}))/(\cos(\cos(\mathrm{DBP}_{t-17}))/ -125))))) \\
& + \cos((\sin((\mathrm{DBP}_{t-16}/\mathrm{DBP}_{t-24})) - \mathrm{DBP}_{t-29}))))))))) \\
\mathrm{DJY}_t = {} & (((\mathrm{DJY}_{t-16} - \mathrm{DJY}_{t-22})/(119 - \mathrm{DJY}_{t-26})) \\
& - (((\mathrm{DJY}_{t-30}/ -20)/(\mathrm{DJY}_{t-16} - 116)) \\
& - (((\mathrm{DJY}_{t-16} - \mathrm{DJY}_{t-30})/\mathrm{DJY}_{t-22}) - (((\mathrm{DJY}_{t-30} - 112) \\
& / ((((\mathrm{DJY}_{t-16} - 109) - \mathrm{DJY}_{t-16}) - (-20 - \mathrm{DJY}_{t-16})) \\
& - (120 - \mathrm{DJY}_{t-26}))) - ((127/\mathrm{DJY}_{t-26}) \\
& - (((\mathrm{DJY}_{t-30}/ -21)/(\mathrm{DJY}_{t-16} - 116)) - \mathrm{DJY}_{t-16}))))))
\end{aligned}
$$

$$\begin{aligned}\mathrm{DTD}_t = (&\mathrm{DTD}_{t-16} + ((\cos((\mathrm{DTD}_{t-16} + ((\mathrm{DTD}_{t-16} + (\mathrm{DTD}_{t-16}\\ &+ ((((\mathrm{DTD}_{t-16} + ((\mathrm{DTD}_{t-16} + (((\mathrm{DTD}_{t-26}\\ &* ((\mathrm{DTD}_{t-19} + -71) + -108))/-114)\\ &/ \cos((\mathrm{DTD}_{t-16} + (\mathrm{DTD}_{t-16} + (\mathrm{DTD}_{t-16} + (\mathrm{DTD}_{t-19} + -66)))))))\\ &/ -101)) * (\mathrm{DTD}_{t-19} + -47))/-123)/\cos((\mathrm{DTD}_{t-19} + (\mathrm{DTD}_{t-16}\\ &+(\mathrm{DTD}_{t-16} + (\mathrm{DTD}_{t-19} + -66)))))))/-103)))\\ &* (\mathrm{DTD}_{t-19} + -70))/-106))\end{aligned}$$

GP evolved models of wavelet transformed series:

DBP:

$$\begin{aligned}s3_t = ((&\cos(\sin(-114)) + \cos(\cos((s3_{t-2} - \sin((s3_{t-6} - \cos(((\sin(-114)\\ &+ \sin((s3_{t-6} - \cos(((\sin(-115) + \cos(\sin(-114))) * \sin(s3_{t-6}))))))\\ &* \sin(-115)))))))) * \sin(((s3_{t-6} - \cos((\sin((\sin(\sin(s3_{t-6}))/\cos(s3_{t-2})))\\ &* \sin((\sin(\sin(s3_{t-6}))/\sin(-114)))))) + \sin(\sin(s3_{t-6})))))\\ d3_t = (&(((((\sin(((-9 + d3_{t-3})/(d3_{t-6} * d3_{t-2}))) + (d3_{t-2}/(d3_{t-6} - d3_{t-2})))\\ &+ ((d3_{t-2}/(d3_{t-6} - d3_{t-5})) + \sin((d3_{t-2}/(d3_{t-6} * d3_{t-4})))))\\ &+ ((d3_{t-4}/(d3_{t-4} + d3_{t-3})) * (\cos(d3_{t-3}) + \sin((d3_{t-2}/(d3_{t-6} * d3_{t-4}))))))\\ &+ (d3_{t-2}/(d3_{t-6} - d3_{t-5})))/((-9 * -118)/\cos((d3_{t-2}/(d3_{t-6} - d3_{t-4})))))\\ d2_t = (&((d2_{t-4}/-35) + ((d2_{t-10}/-29)\\ &+ (((d2_{t-7}/-39) + ((((d2_{t-10} - (31 * d2_{t-6}))\\ &+ (d2_{t-7} - (41 * d2_{t-8}))) + ((d2_{t-10} - (49 * d2_{t-8})) + d2_{t-7})) * d2_{t-5}))\\ &+ ((d2_{t-7}/-37) + (\cos(((28 * d2_{t-7})/d2_{t-8})) * (\cos(\cos(\cos((((d2_{t-10}\\ &- (39 * d2_{t-8}))/d2_{t-8})/d2_{t-8})))) * d2_{t-7}))))))\\ d1 = (&d1_{t-9} * d1_{t-9})\end{aligned}$$

DJY:

$$\begin{aligned}s3_t = (&s3_{t-2} - (((((((((s3_{t-5} - s3_{t-4}) + \cos((\cos(s3_{t-5}) + s3_{t-2})))\\ &+ \sin(((s3_{t-5} - s3_{t-4}) + s3_{t-2}))) + \cos(((\cos(s3_{t-6}) + s3_{t-2}) + s3_{t-2})))\\ &+ \cos((s3_{t-5} - s3_{t-4}))) + \sin(((s3_{t-5} - s3_{t-4}) + s3_{t-2})))\\ &+ \sin(((\cos((s3_{t-5} - s3_{t-4})) - s3_{t-4}) + s3_{t-2})))\\ &* (\sin(((s3_{t-5} - \cos(s3_{t-5})) - \cos(s3_{t-5}))) - \cos(s3_{t-2}))))\end{aligned}$$

$$
\begin{aligned}
d3_t = {} & (((\sin(\sin(\sin(\sin(\sin(\sin(\sin(\sin(((d3_{t-3} + d3_{t-3}) * d3_{t-5})))))))))) \\
& + (\sin((d3_{t-6} + (d3_{t-6} + d3_{t-3}))) + \sin(d3_{t-2}))) \\
& * \sin(((d3_{t-3} + d3_{t-3}) * d3_{t-5}))) + (((\cos(d3_{t-6}) + (\cos(d3_{t-6}) + d3_{t-2})) \\
& * (\cos((d3_{t-6} + (\sin(d3_{t-6}) + d3_{t-3}))) * (\cos(d3_{t-3}) * \sin(d3_{t-6})))) \\
& + \cos((d3_{t-6} + (d3_{t-6} + d3_{t-3}))))) \\
d2_t = {} & \sin(\sin(((((d2_{t-5} * d2_{t-10}) + ((((d2_{t-9} + (\cos(((d2_{t-10} * (d2_{t-6} \\
& / \cos((d2_{t-10} + \cos(d2_{t-5}))))) * d2_{t-10}))/d2_{t-10})) + \cos((d2_{t-6}/((d2_{t-5} * \times 7) \\
& * (((d2_{t-5} * d2_{t-10}) * (d2_{t-5} * 7)) * 5))))) + \cos((d2_{t-6} \\
& / (d2_{t-5} * (d2_{t-10} + (d2_{t-5} * d2_{t-10}))))))/5)) - d2_{t-10})/2))) \\
d1_t = {} & (((\cos((d1_{t-12} - (d1_{t-10} + (\sin((d1_{t-10} + (((d1_{t-10} + (\cos(\cos((d1_{t-12} \\
& - (d1_{t-10} + d1_{t-10})))) - d1_{t-12})) - d1_{t-16}) - d1_{t-16}))) - ((\cos((d1_{t-12} \\
& - (d1_{t-10} + (d1_{t-10} - ((d1_{t-14}/d1_{t-10})/\cos(d1_{t-16})))))) - d1_{t-14}) \\
& / (d1_{t-10} - -1)))))) * \sin((d1_{t-14}/d1_{t-10}))) - d1_{t-14})/(d1_{t-10} - -3))
\end{aligned}
$$

DTD:

$$
\begin{aligned}
s3_t = {} & ((\sin(s3_{t-2})/-2) + (((\sin(s3_{t-3})/-6)/(s3_{t-6} \\
& - ((\sin(s3_{t-3})/-5) + s3_{t-2}))) + (((\sin(s3_{t-3})/-29)/\sin(s3_{t-2})) \\
& + (((\sin(s3_{t-3})/-4)/(s3_{t-6} - s3_{t-2})) + (((\sin((((\sin(s3_{t-3})/-7) \\
& / (\sin(s3_{t-2})/-11)) + s3_{t-3}))/s3_{t-5})/(s3_{t-2} - s3_{t-3})) + s3_{t-2}))))) \\
d3_t = {} & (((((d3_{t-2}/((d3_{t-4} * d3_{t-6}) + (((d3_{t-3}/(d3_{t-4}/\cos(d3_{t-4}))) + d3_{t-4}) \\
& * (d3_{t-2} * d3_{t-2}))))/84)/d3_{t-3})/111) + (((d3_{t-2}/d3_{t-3})/80) \\
& + ((((d3_{t-4} * d3_{t-6})/(d3_{t-2}/d3_{t-3})) + ((d3_{t-3}/d3_{t-4}) * (d3_{t-6} * d3_{t-2}))) \\
& + ((d3_{t-4} * d3_{t-6})/\cos(((d3_{t-3}/(d3_{t-4}/\cos(d3_{t-4}))) + d3_{t-4})))))) \\
d2_t = {} & (((((((d2_{t-8} + d2_{t-8}) + d2_{t-4}) * (d2_{t-6} + ((d2_{t-6} + d2_{t-8}) \\
& - ((d2_{t-10} + ((d2_{t-10} + d2_{t-4}) + d2_{t-4}))/d2_{t-10})))) * d2_{t-6}) + d2_{t-4}) \\
& * (d2_{t-6} + (d2_{t-6} + ((d2_{t-10} + d2_{t-4})/((d2_{t-5} + d2_{t-4})/d2_{t-10}))))) \\
& - ((d2_{t-7} * (((((d2_{t-10} + d2_{t-4}) + d2_{t-4}) + d2_{t-4}) + d2_{t-4}) + d2_{t-4})) \\
& / (d2_{t-4}/(d2_{t-5} * d2_{t-5})))) \\
d1_t = {} & ((\cos((((d1_{t-16} * -45) * d1_{t-16})/d1_{t-15}))/((d1_{t-12} * d1_{t-12}) \\
& / \sin((((d1_{t-10}/d1_{t-10}) * (d1_{t-13}/-29)) * d1_{t-8})))) + ((((d1_{t-15} * d1_{t-14}) \\
& / \cos(((d1_{t-16} * -46) * d1_{t-16}))) - (d1_{t-11} * ((d1_{t-12} + d1_{t-15}) \\
& / (d1_{t-11}/d1_{t-8})))) - (d1_{t-11} * ((d1_{t-12} + d1_{t-13})/(d1_{t-11}/d1_{t-12})))))
\end{aligned}
$$

References

1. R. T. Baillie and T. Bollerslev, The message in daily exchange rates: A conditional variance tale, *Journal of Business and Economic Statistics* **7** (1989) 297–306.
2. D. Hsieh, Modelling heteroskedasticity in daily foreign exchange rates, *Journal of Business and Economic Statistics* **7** (1989) 307–317.
3. A. Cecen and C. Erkal, Distinguishing between stochastic and deterministic behaviour in high frequency foreign exchange returns: Can non-linear dynamic help forecasting? *International Journal of Forecasting* **12** (1996) 465–473.
4. C. Kuan and T. Liu, Forecasting exchange rates using feedforward and recurrent neural networks, *Journal of Applied Econometrics* **10** (1995) 347–364.
5. G. Zang and M. Hu, Neural network forecasting of the British pound/US dollar exchange rate, *Omega, International Journal of Management Science* **26** (1998) 495–506.
6. M. Clements and J. Smith, Evaluating forecasts from SETAR models of exchange rates, *Journal of International Money and Finance* **20** (2001) 133–148.
7. G. Zang and V Berardi, Time series forecasting with neural network ensembles: An application for exchange rate prediction, *Journal of Operations Research Society* **52** (2001) 652–664.
8. N. Meade, A comparison of the accuracy of short term foreign exchange forecasting methods, *International Journal of Forecasting* **18** (2002) 67–83.
9. A-S. Chen and M. Leung, Regression neural network for error correction in foreign exchange forecasting and trading, *Computers and Operations Research* **31** (2004) 1049–1068.
10. F. Fernandez-Rodriguez, S. Sosvilla-Rivero and J. Andrada-Felix, Nearest-neighbour predictions in foreign exchange markets, in *Computational Intelligence in Economics and Finance*, eds. S.-H. Chen and P. Wang (Springer-Verlag, Berlin, 2004), pp. 297–325.
11. Oanda.com, http:// www.oanda.com (2004).
12. S.-H. Chen and T.-W. Kuo, Discovering hidden patterns with genetic programming, in *Computational Intelligence in Economics and Finance*, eds. S.-H. Chen and P. P. Wang (Springer-Verlag, Berlin, 2004), pp. 329–347.
13. B. Efron, *The Jackknife, the Bootstrap, and Other Resampling Plans* (Society for Industrial and Applied Mathematics, Philadelphia, 1982).
14. D. Donoho and I. Johnstone, Adapting to unknown smoothing via wavelet shrinkage, *Journal of the American Statistical Association* **90** (1995) 1200–1224.
15. Z. Pan and X. Wang, A stochastic nonlinear regression estimator using wavelets, *Computational Economics* **11** (1998) 89–102.
16. V. Cherkassky and X. Shao, Signal estimation and denoising using VC-theory, *Neural Networks* **14** (2001) 37–52.
17. R. Gençay, F. Selçuk and B. Whitcher, *An Introduction to Wavelets and Other Filtering Methods in Finance and Economics* (Academic Press, San Diego, 2002).
18. G. Lee, Wavelets and wavelet estimation: A review, *Journal of Economic Theory and Econometrics* **4** (1998) 123–157.
19. J. Walker, *A Primer on Wavelets and Their Scientific Applications* (Chapman & Hall/CRC, Boca Raton, 1999).
20. S. Mallat, A theory for multiresolution signal decomposition: The wavelet representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **11** (1989) 674–693.

21. J. Koza, *Genetic Programming* (The MIT Press, Cambridge, MA, 1992).
22. M. Kaboudan, TSGP: A time series genetic programming software, http://newton.uor.edu/facultyfolder/mahmoud_kaboudan/tsgp (2003).
23. M. Kaboudan, Statistical properties of fitted residuals from genetically evolved models, *Journal of Economic Dynamics and Control* **25** (2001) 1719–1749.
24. J. Principe, N. Euliano and C. Lefebvre, *Neural and Adaptive Systems: Fundamentals Through Simulations* (John Wiley & Sons, Inc., New York, 2000).