The European Journal of Finance **1**, 311–323 (1995)

Leading edge forecasting techniques for exchange rate prediction

IAN NABNEY[1], CHRISTIAN DUNIS[2], RICHARD DALLAWAY[3], SWEE LEONG[2] and WENDY REDSHAW[4]

[1]Aston University, [2]Chemical Bank, 125 London Wall, London EC2Y 5AJ[3]Fusion Systems, [4]Summit Financial Systems Limited

This paper describes how modern machine learning techniques can be used in conjunction with statistical methods to forecast short term movements in exchange rates, producing models suitable for use in trading. It compares the results achieved by two different techniques, and shows how they can be used in a complementary fashion. The paper draws on experience of both inter- and intra-day forecasting taken from earlier studies conducted by Logica and Chemical Bank Quantitative Research & Trading (QRT) group's experience in developing trading models. In evaluating different models both trading performance and forecasting accuracy are used as measures of performance. Rule induction is a method for deriving classification rules from data. Logica's data exploration toolkit DataMariner™, which combines rule induction with statistical techniques, has been used successfully to model several exchange rate time series. An attractive feature of this approach is that the trading rules produced are in a form that is familiar to analysts. We also show how DataMariner™ can be used to determine the importance of different technical indicators and to understand relationships between different markets. This understanding can then be used to assist in building models using other analytical tools. Neural networks are a general technique for detecting and modelling patterns in data. We describe the principles of neural networks, the data preprocessing that they require and our experience in training them to forecast the direction and magnitude of movements in time series.

Keywords: machine learning techniques, leading-edge forecasting, rule induction, neural networks.

1 INTRODUCTION

This paper describes how modern machine learning techniques can be used in conjunction with statistical methods to forecast short term movements in exchange rates, producing models suitable for use in trading. It compares the results achieved by two different techniques and shows how they can be used in a complementary fashion. The two techniques used were rule induction, which is a method of extracting classification rules from data, and neural networks, which afford powerful and general methods for nonlinear function modelling.

2 Forecasting Techniques

This section gives a brief introduction to the two forecasting techniques whose application to exchange rate prediction is described in this paper. Both techniques allow the detection and modelling of non-linear effects in data, whereas the majority of conventional statistical methods build linear models (see Chatfield and Collins (1986)).

2.1 Rule induction and DataMariner™

Rule induction is a technique for identifying patterns and relationships in data and expressing them as rules. A rule induction system is given a set of historical examples in each of which a number of attributes are measured and the class or outcome recorded. From these examples the system identifies what the examples in each class have in common; generally the aim is to find the simplest rules that can distinguish between examples from distinct classes. The effectiveness of the rule induction approach is dependent on the quality of the attributes used to discriminate between classes. The rules can be interpreted either as a causal relationship or as a description of the examples in a particular class. They can then be used to classify new examples (see, for instance, Quinlan (1986) Race (1988) and Nabney and Jenkins (1992)).

As an example we consider how rule induction may be applied to forecasting exchange rates with a 24 hour horizon. Each close of day represents an 'example'; its class could be the direction of the price change 24 hours after the time of forecast. The attributes used in the rules could be technical indicators extracted from the time series data, together with useful external indicators (such as secondary market information). So if there were four indicators for detecting trends and four indicators for detecting a ranging market a rule for predicting the direction of price movement could have the following form:

```
IF
  range_4 >= 0.01
  range_1 >= 1.0
  trend_2 >= 0.0005
  trend_3 <= 0.02
THEN
  price_movement = up (0.75)
  price_movement = down (0.25)
```

This rule has isolated circumstances when there is a 75% chance that the exchange rate tomorrow will be higher than today. If the indicators are chosen so that they are familiar to traders or analysts, then these rules can be related to their own experience. Results expressed as descriptive rules are usually easier for most users to interpret. For example, the usefulness, or otherwise, of attributes can easily be assessed by the frequency with which they are used in the rules.

There are a number of rule induction packages currently available. However, these have several drawbacks. First, they are usually based on a particular algorithm, called ID3 (see Quinlan (1986)), which has a number of limitations:

for example, it produces a decision tree, which tends to be more difficult to interpret than modular rules. Second, the tools are generally aimed at supporting the analysis of small amounts of data when applications in finance generally imply the analysis of large data sets. Most seriously of all, there is often a limited range of support tools; unfortunately, rule induction by itself is often not powerful enough to extract all the knowledge from data.

To overcome the drawbacks summarized above, we have used the DATAMARINER™ software which is described in Nabney and Jenkins (1992); this is a set of closely integrated analytical tools with an easy to use graphical interface. At the heart of the data exploration toolkit is a novel rule induction algorithm, designed to overcome some of the drawbacks of ID3 and other techniques (see Nabney and Jenkins (1992)). This algorithm works on a rules-per-class basis, (i.e. for each class in turn), rules are induced to separate examples in that class from examples in all the remaining classes. This produces structured rules directly rather than a decision tree. Three advantages follow from this:

(i) The rules are in a suitable form for understanding a classification; namely a description of each class in the simplest way that enables it to be distinguished from the other classes.
(ii) The rule set is structured in a more modular fashion which enables the user to focus on a single rule at a time to a large extent. As noted by Cendrowska (1988), decision trees can be hard to understand, particularly when the number of nodes is large.
(iii) Empirical results gathered from a number of studies carried out by Logica have shown that DATAMARINER™ generates many fewer rules than the ID3 algorithm, without loss of accuracy.[1]

Many of the other tools in DATAMARINER™ are standard statistical analysis; however, the results of any analysis are always expressed in the form of rules. The most important features of DATAMARINER™ for the purposes of this paper are: (a) the formation of new attributes: ratios and simple linear combinations of existing numeric attributes can be constructed using statistical techniques, and (b) the pruning of rules: to allow rules to generalize well to new data, they can be pruned using a statistically well-founded technique.This prevents the rules from tracking noise in the data they were generated from, a problem called *over-training* (i.e. the equivalent to model overfitting in econometrics).

2.2 Neural networks

Neural networks are a powerful method of modelling complex nonlinear relationships. Like rule induction, a neural network is trained on a set of data and the performance of the trained model is evaluated by testing it on previously unseen data.

[1] A comparative evaluation of DATAMARINER™ was carried out on a range of databases from the repository of Machine Learning Databases maintained at the University of California at Irvine by D. Aha. These databases are drawn from commercial, medical and scientific fields. In nearly every case, the accuracy of the rules generated by Logica's algorithm was as good as or better than the figures quoted in the literature for other rule induction algorithms (see Nabney and Grasl (1991)).

There are a large number of different neural network models. The most commonly used is the multilayer perceptron (MLP) which feeds its inputs along a series of weighted connections and applies nonlinear functions at the nodes (for more details on the MLP and the workings of a neural network model, see, for instance, Rumelhart and McClelland (1986), Pao (1989) and, for an application to exchange rates, Dunis (1995)).

An MLP, like rule induction, requires a set of examples for each of which a number of attributes are measured and some outcome is recorded. However, unlike rule induction, this outcome does not need to be a discrete classification. In fact, neural networks process numeric values, and an MLP can be used to model multiple numeric outputs.

This greater capability gives more choice in how neural networks may be applied to the problem of forecasting exchange rates. At each prediction, the forecast could be either the direction of the price movement (encoded as + 1.0 for up, 0.0 for no change and –1.0 for down) or the new price.

There are powerful theoretical results that state that an MLP, if it is sufficiently large, can approximate any continuous deterministic function (see, for instance, Cybenko (1989), Funahashi (1989) and Hornik *et al.* (1989)). Thus, if we can select the correct inputs for a network, it should be able to model whatever function is required. This implies that provided that a neural network is presented with a sufficiently large time window of previous data, it should be able to extract any 'derived attributes' that are required.

There are two caveats to apply to this, arising from the need to keep the number of inputs reasonably small. First, it may not be feasible to use only a contiguous window of data, since the model may need isolated inputs with very large delays. (For example, the hourly sales forecasts for a retailer will be heavily dependent on the day of the week, which could be represented by the sales from 168 hours earlier.) Second, the system may not be of finite order, which means that to make a prediction, knowledge of all its previous states is needed.

In addition to the above considerations, this capability of approximating any continuous function can be problematic as the underlying behaviour that the network is supposed to learn may change over time. More importantly, the model thus created is a deterministic functional map, hence it is not able to learn patterns arising from truly random perturbations. In fact, when a deterministic process is 'contaminated' by a stochastic perturbation, the neural network will try to learn from spurious relationships arising purely from noise in addition to the 'true' underlying process, which raises the issue of overtraining or noise fitting. It will then only learn the mean of the target data, which may not prove useful when there are outliers in the data.

A technique commonly used to prevent overtraining is the use of a third set of data, the validation set, which is independent of both training and test data. Periodically during training, the error of the neural network on the validation set is measured. The network is trained until this error goes up significantly, and the final network is the one with the lowest error on the validation set. This technique plays the same role as pruning does for rule induction.

The powerful approximation capability of neural networks does have a price. A trained MLP represents a complex mathematical function and it is very difficult to understand in detail the operation of even very small networks. Hence the results are more difficult to interpret than those arising from rule induction.

3 INTER-DAY TRADING MODELS

In a former study, Logica analysed historical data on the close of day prices of three different exchange rates over the same time period (respectively, the US dollar/Deutsche mark, the US dollar/Japanese yen and the British pound/US dollar rates), together with attributes, or 'technical indicators' (such as median prices over a time window, relative strength indices, and so on), which were believed by the client to have good forecasting ability. The objective was to predict the direction of the next price movement based on these attributes.

On presentation to the rule induction system, the class was the direction of the price movement the following day. Of the 1500 examples in each dataset, half were used for training, and half for testing.

The rule induction system achieved an accuracy of between 64% and 69% on this data. A neural network approach using a standard multilayer perceptron (MLP) with the same inputs as for rule induction achieved an accuracy of between 60% and 65%. By way of comparison, when the linear technique of Kalman filtering was applied to this data, the accuracy was in the range 55–60%.[2]

The quality of these results was slightly surprising since all the attributes were numeric, and rule induction has principally been used in domains where there are some discrete attributes. We concluded that this study showed the power of the DataMariner™ approach where rule induction is combined with statistical techniques. The capability of forming linear combinations and ratios of existing attributes during the induction process itself was essential for achieving a high accuracy. Because the data was extremely noisy, the ability to prune the rules at a variety of different levels also improved performance dramatically: pruning was done by assessing the statistical significance of each condition in a rule and then removing those conditions that were likely to have occurred by chance.[3] It seems likely that another reason for the good performance of rule induction was the use of highly predictive technical indicators as inputs to the modelling technique.

Another advantage of rule induction over other techniques on this problem was the extra information that was easily accessible from the rules about the data. For example, the newly constructed attributes for two of the markets were very similar, suggesting a similarity in the underlying mechanisms affecting

[2] For more details on the use of Kalman filtering, see, amongst others, Harvey (1981), pp. 101–19, and Kalman (1960).

[3] In practice, clauses were pruned from a rule starting from the last condition. The Fisher one-tailed probability distribution was used to decide if an individual clause provided a statistically significant increase in discrimination between classes.

prices in those two markets. The usage of attributes in the rule conditions was also interesting: for example, certain attributes were used very frequently, while others were hardly used at all, and could perhaps be replaced by different attributes derived from preprocessing techniques. Finally, some of the attributes constructed as ratio attributes could be used as extra inputs for other techniques, such as neural networks.

4 INTRA-DAY TRADING MODELS

The aim of this study was to determine the potential of machine learning techniques for modelling intra-day movements in exchange rates.

4.1 Analysis procedure

Two markets were studied: US dollar/Swiss franc (USD/CHF), and Deutsche mark/French franc (DEM/FRF). These currencies were chosen because the USD/CHF market has a high volatility, while the DEM/FRF market had relatively low volatility up until the ERM crisis in August 1993. In each case tick data was supplied for the primary currency pair and, in addition, a secondary time series representing the USD/DEM rate current at the time of the trade in the primary currency. The data was collected from the year 1 October 1992 to 1 October 1993. Each data file was approximately 50 Mbyte in size, which meant that extracting indicators often took several hours.

In addition to prediction accuracy, we specified a detailed set of measures to evaluate the trading performance of the models (see Tables 1 and 2).

The first set of experiments involved the use of DataMariner to make twenty minute-ahead predictions. Training data was taken from 11 January 1993 to 9 March 1993 and test data was from 9 March 1993 to 26 April 1993. The trading day was assumed to last from 08:00 to 18:00. Weekends and UK holidays were ignored. The output was a simple prediction that the price movement would be 'up', 'down', or 'no change'. After determining the best algorithm parameters, good results were achieved on USD/CHF (at best, about 50% annual return), and poorer results on DEM/FRF (at best, about 12% annual return).

At this point we decided to make the evaluation criteria more realistic by introducing a slippage of 20 basis points for each position (40 points when squaring off a position).

The existing models were evaluated with this new cost and performed poorly, trading at a loss. This was mainly because they took positions too often (they were being used to take a decision every 10 min). It was therefore decided to use a 60 min predictive horizon. It was also decided to evaluate the performance of neural networks to obtain some comparative results.

With a one hour horizon, it becomes more important to make good use of the available data. It was split into three sets (our standard practice when using neural networks): the training set was from 8 October 1992 to 15 March 1993 (120 days), the validation set from 15 March 1993 to 3 May 1993 (50 days), and the test set from 3 May 1993 to the end of the data (80 days). Data was used

from 00:00 to 21:00 (i.e., from the opening of Tokyo to the close of New York).

Because neural networks are general function modellers, there is more latitude over the selection of inputs from the time series for good performance. Information from the secondary price time series was not included, as the work with DataMariner™ had shown that it was of limited use in forecasting movements in the primary series. The output of the neural network was the forecast price one hour on from the current time.

All the neural networks used in this project were multilayer perceptrons. Systematic experiments were performed to determine a suitable neural network structure and training algorithm parameters (cf. the initial work with DataMariner™). Once the best values had been determined, the network was trained until a minimum in the error was attained. When using the resulting model for trading, the price prediction was banded into 5 classes: 'large up', 'small up', 'no change', 'small down', and 'large down', as shown in Fig. 1.

The 'change threshold' which defined the boundary between small and large movements was generally chosen to be the same as the slippage (0.002). The trading technique that was used is shown in Fig. 2. Here the boxes denote the 'current position' (the model starts square), and the arrows denote the transitions which are carried out when the model makes the prediction given by the attached label.

The use of neural networks immediately led to improved performance on USD/CHF: there was a 55% annualized return on the test data. The performance on DEM/FRF was less good: 23% on the full test data, where most of the profit was made at the ERM crisis. (The annualized gain on the test data before the crisis was 3%).

To improve the DEM/FRF results, changes were made to the input attributes; in particular further indicators were added to the inputs. This improved results on the test data to 37%. Some further experiments with DataMariner™ with a one hour prediction horizon and five classes (as shown in Fig. 1) were carried out; these achieved a 9% annualized return on test data.

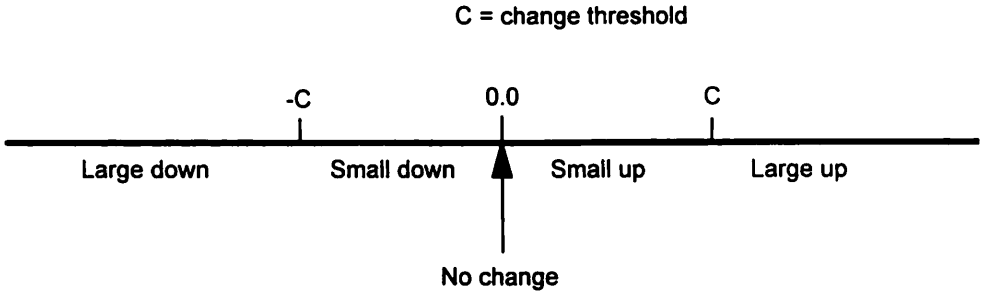


Fig. 1. Prediction classes

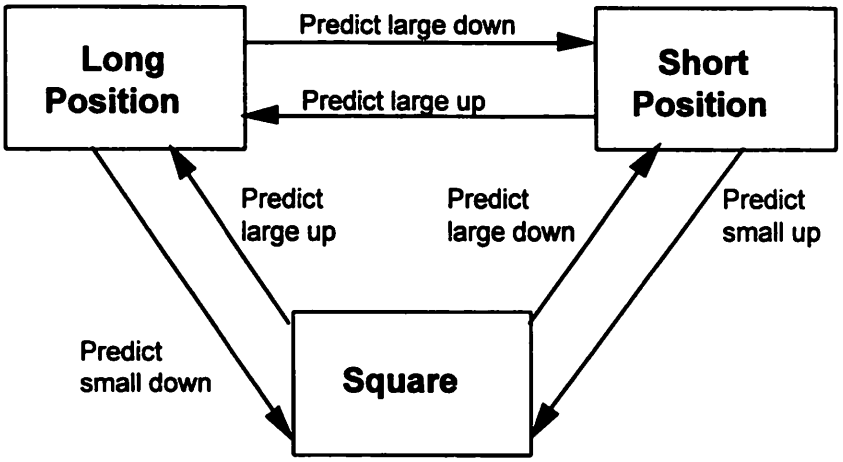


Fig. 2. Trading with a model

4.2 Results

4.2.1 Rule induction

The performance of the rules induced by DataMariner™ was disappointing. For example, the 'best' set of rules on DEM/FRF produced an annualized return of 9.31%, hardly more than the 'risk-free' rate of interest on the French franc or on the Deutsche mark over the same period, that is, respectively, 7.88% and 7.69%. The probable reason for this was that in a short study it was not possible to find the best indicators to derive from the raw price series data. This point is considered further in section 5.

It was found that the correlation of attributes with the classification and their usage in the rules was different for the two markets. For USD/CHF, traditional technical indicators and their ratios were the best predictors for a 20 min horizon. For DEM/FRF, it was found that volume data was important: this was probably because all the training data came from before the ERM crisis, when large movements were scarce and volume was correlated with a change/no change classification.

4.2.2 Neural Networks

The good results achieved with neural networks showed the power of the technique in modelling complex time series data.

Tables 1 and 2 document the results of our USD/CHF and DEM/FRF models. They both detail the number of days for the test period, the average holding period of each position, the standard deviation of the series of gains and losses, the maximum gain, the number of winning trades, the average gain, the percentage of winning trades, the cumulative gain per USD or per DEM, the annualized percentage gain, the root mean squared error, the probability of

Table 1. Neural network results on USD/CHF

NumDays	41	Num. positions	25
Mean time in pos.	28.1 hours	Correct direction	52.5%
StdDev(Gain/Loss)	0.012	StdDev (%ChgFX)	0.165
MaxPosGain	0.046	MaxPosLoss	–0.020
NumGainPos	17	NumLossPos	8
AvgGain	0.0102	AvgLoss	–0.0047
%GainPos	68	%LossPos	32
CumGain	0.1395	%CumGain	9.032
Annual %CumGain	55.07	GainLossRatio	4.61
RMS error	0.00286	Sharpe ratio	4.60
Prob. of losing 100 pips	19.90	MaxDrawDown	0.020 (1 position)

losing 0.01 CHF per USD traded (or 0.01 FRF per DEM traded), the number of positions taken, the percentage of correct directional forecasts by the models, the standard deviation of the underlying exchange rate, the maximum loss, the number of losing trades, the average loss, the percentage of losing trades, the percentage gain over the test period, the gain-to-loss ratio (i.e. the ratio of the average gain weighted by the percentage of winning trades over the average loss weighted by the percentage of losing trades), the Sharpe ratio (a measure of profitability adjusted for risk commonly used by fund managers) and the maximum drawdown of each model (i.e. the largest cumulative loss recorded over the test period).

Table 1 shows the results achieved on USD/CHF with the original set of inputs and a 60 min forecast horizon. Note that the error on the validation set increased rapidly, so that training finished after about 15 000 epochs. This was probably caused by significant differences between training and validation sets,

Table 2. Neural network results on DEM/FRF

NumDays	71	Num. positions	55
Mean time in pos.	15.71 hours	Correct direction	54.5%
StdDev(Gain/Loss)	0.0187	StdDev (%ChgFX)	0.0766
MaxPosGain	0.095	MaxPosLoss	–0.034
NumGainPos	41	NumLossPos	14
AvgGain	0.0103	AvgLoss	–0.0049
%GainPos	74.5	%LossPos	25.5
CumGain	0.3544	%CumGain	10.52
Annual %CumGain	37.05	GainLossRatio	6.13
RMS error	0.00068	Sharpe ratio	6.672
Prob. of losing 100 pips	15.63	MaxDrawDown	0.0455 (3 positions)

suggesting the desirability of using more data for training, either by using more historical data or by oversampling.[4]

Still, with high Sharpe and gain-to-loss ratios, a low probability of losing 0.01 CHF per USD traded and a high annualized return, our USD/CHF trading model appears quite satisfactory.

Graph A1 in Appendix 1 shows the training, validation and test sets for USD/CHF while the graph 2 displays the evolution of the trading performance of the neural network model over the test period.

The results achieved on the whole of the DEM/FRF test data follow in Table 2. These used further technical indicators as additional inputs. Here too the performance sensitivities of our DEM/FRF trading model are quite satisfactory although the maximum drawdown is about ten times the average loss. This occurs during the period of high volatility following the ERM crisis. Graph A3 in Appendix 1 shows the training, validation, and test sets for DEM/FRF. The cumulative gain chart for the test data (A4) shows how the return becomes much more volatile after the ERM crisis. Even in the period following the large jump in the exchange rate, however, the model gives good returns. In the period before the crisis, the cumulative gain is about 0.1, which is an annual rate of about 15.7%.

Encouraging as these results are, there are still improvements that could be made to the forecasting accuracy of the model. It is noticeable that while the errors on the training data are symmetrically distributed about a near-zero mean (see A5), the errors on the test data are biased, with a non-zero mean (see A6).

Given that trading performance is the main means of evaluating models, it may be useful to modify the error function used during training to take into account the eventual classification of the network prediction. (So that output errors that lead to an incorrect classification are penalized more heavily than those that do not.) Although this is attractive, it would reduce the flexibility of using the neural model.

5 CONCLUSIONS

The study carried out on inter-day trading showed that when good indicators are available, rule induction can give results as accurate as other techniques. In addition, the clarity of the rules gave interesting insights into the data, allowing comparisons between the different markets to be drawn.

On our intra-day study, although work remains to be done to improve the models, the neural network approach appeared to offer significantly better performance. The return achieved with the best neural models was very encouraging. Chemical Bank's QRT group is currently developing a trading

[4] In this context, oversampling refers to the technique of generating further time series which overlap with the original one. For example, if the original data was generated every hour on the hour, another time series could be generated every hour on the half hour. As such, this technique is similar to the analysis of panel data (see, amongst others, Chamberlain (1985) and Mátyás and Sevestre (1992)).

system which uses neural models partially based on techniques similar to those used in this study.

One of the most important lessons to be learned from the study was that the way a model is evaluated affects its construction. The current method of trading with the model during evaluation is very simple and inflexible compared to the way that human traders work. It is important to evaluate models in a way that is as similar as possible to the way in which they will be used, whether that is for trading directly or to advise human traders.

DataMariner™ did not achieve as good results as the neural network approach. The most likely reason for this was that the indicators used during the project were suboptimal, since the results improved with the use of different technical indicators.

Although the neural network approach was successful, there are a number of issues that were raised which could be investigated further:

(i) *Network architecture*. The work in this project used a simple multilayer perceptron with a single hidden layer. The largest network performed best, and so it would be advisable to consider larger networks, and also networks with more hidden layers to see if performance could be improved.

(ii) *Input features*. Although the precise form of the features input to a neural network is not as crucial as for rule induction, considerable improvements in accuracy can be achieved by using inputs that contain more information. The improvement achieved on the DEM/FRF results by using additional inputs suggests that there is more that can be done with this data. An alternative is to use recurrent networks, which have an internal memory, so as to avoid having to determine the precise window of past data required to model the system.

(iii) *Confidence intervals*. It would clearly be useful to have error bounds on exchange rate forecasts. There are techniques for estimating confidence intervals on neural network outputs; the most principled of these depend on training the network in a Bayesian framework, as in Williams *et al.* (1995).

(iv) *On-line learning*. The work in this study has built a single model that is then fixed and evaluated on test data. There are neural network models that can be trained on-line, so that changes in the underlying system are tracked. This might be an attractive approach for this application, provided that a suitable validation scheme was developed. A fixed model can be validated by evaluating its performance on recent out-of-sample data; this is not possible if the most recent data is used for adjusting the model on-line.

In both studies, all the training and testing of models was carried out on 'dead' data; using such models in a real time system presents a number of additional problems. During the studies, all the models were developed and tested with data extracts on a stand-alone machine. The intra-day trading system currently being developed takes a live data feed and is integrated with other trading floor systems. In addition, spurious prices are removed and the models have to produce their predictions in real time. The response of the

neural networks described in this paper is sufficiently fast for this not to be a problem.

REFERENCES

Chatfield, C. and Collins, A. J. (1986), *Introduction to Multivariate Analysis*, London Chapman & Hall.

Cendrowska, J. (1988), PRISM: An algorithm for inducing modular rules, *Int. J. of Man-Machine Studies*, **27**, 349–70.

Chamberlain, G. (1985), Panel Data. In Griliches, Z. and Intriligator, M. D. [eds], *Handbook of Econometrics*, Vol. 2. Amsterdam: North-Holland.

Cybenko, G. (1989), Approximation by superposition of a sigmoidal function, *Math. Control, Signals & Systems*, **2**, 303-14.

Dunis, C. (1995), The economic value of leading edge techniques for exchange rate prediction, *Working Papers in Financial Economics*, **5**, 1–7.

Funahashi, K. (1989), On the approximate realization of continuous mapping by neural networks, *Neural Networks*, **2**, 183–92.

Harvey, A. C. (1981), *Time Series Models*, London: Philip Allen.

Hornik, K., Stinchcombe, M. and White, H. (1989), Multilayer feedforward networks are universal approximators, *Neural Networks*, **2**, 359–66.

Kalman, R. E. (1960), A new approach to linear filtering and prediction problems, *J. Basic Engineering*, **1**, 35–45.

Mátyás, L. and Sevestre, P. (eds) (1992), *The Econometrics of Panel Data*. Dordrecht: Kluwer Academic Publishers.

Nabney, I. T. and Grasl, O. (1991), Rule Induction for Data Exploration. In Avignon June '91 Conference on *Expert Systems and their Applications*, **1**, 329–41.

Nabney, I. T. and Jenkins, P. G. (1992), Rule Induction in Finance and Marketing, IBC Conference on *Data Mining in Finance and Marketing*, September.

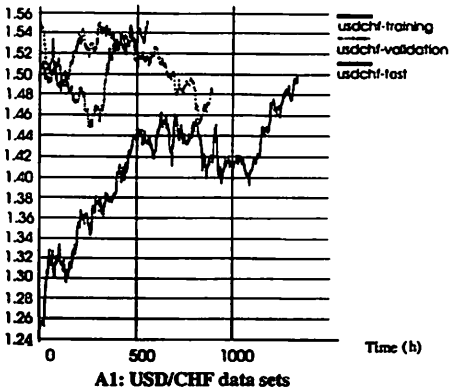
Pao, Y. H. (1989), *Adaptive Pattern Recognition and Neural Networks*. Reading, MA: Addison-Wesley.

Rumelhart, D. E. and McClelland, J. L. (eds.) (1986), *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*. Cambridge, MA: MIT Press.

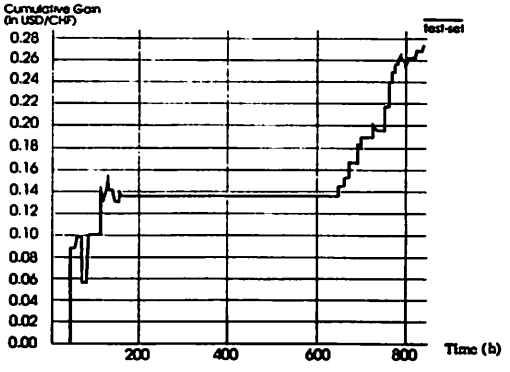
Quinlan, J. R. (1986), Induction of decision trees, *Machine Learning*, **1**, 81–106.

Race, P. R. (1988), Rule induction in investment appraisal, *J. Operational Research Society*, **12**, 1113–23.

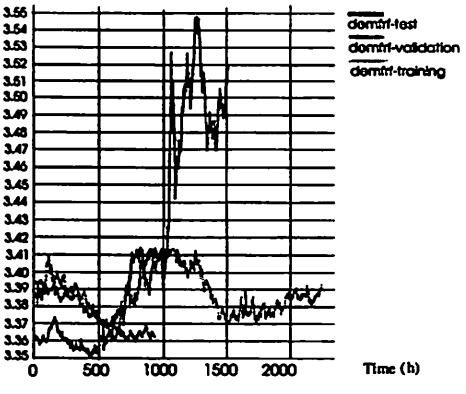
Williams, C. K. I., Qazaz, C., Bishop, C. M. and Zhu, H. (1995), On the relationship between Bayesian error bars and the input data density. Proc. 4th IEE Conference on *Artificial Neural Networks*, Cambridge, 26–28 June, 160–65.



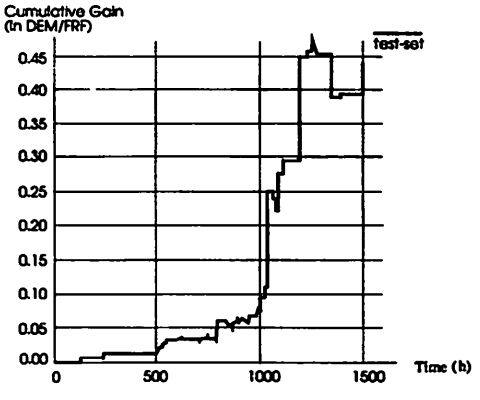
A1: USD/CHF data sets



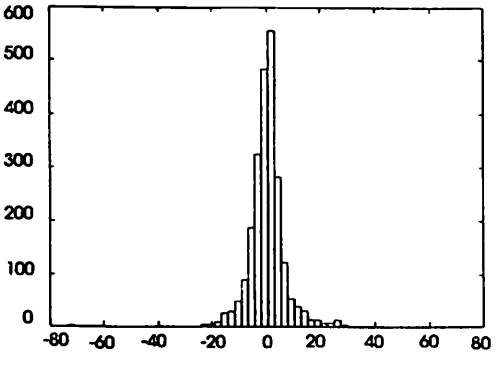
A2: USD/CHF cumulative gain



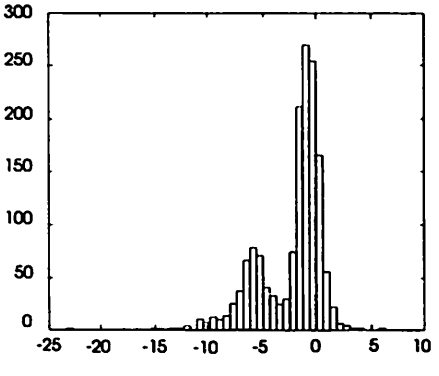
A3: DEM/FRF data sets



A4: DEM/FRF cumulative gain



A5: DEM/FRF neural network error histogram for training data (in pips)



A6: DEM/FRF neural network error histogram for test data (in pips)

Appendix 1