Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Computers & Operations Research 31 (2004) 1049–1068
computers & operations research
www.elsevier.com/locate/dsw

Regression neural network for error correction in foreign exchange forecasting and trading

An-Sing Chen[a], Mark T. Leung[b,*]

[a]Department of Finance, National Chung Cheng University, Ming-Hsiung, Chia-Yi Taiwan, ROC 621
[b]Department of Management Science and Statistics, University of Texas at San Antonio, San Antonio, TX 78249, USA

Abstract

Predicting exchange rates has long been a concern in international finance as most standard econometric methods are unable to produce significantly better forecasts than the random walk model. Recent studies provide some evidence for the ability of using multivariate time series models to generate better forecasts. At the same time, artificial neural networks have been emerging as alternatives to predict exchange rates. In this paper, we propose an adaptive forecasting approach which combines the strengths of neural networks and multivariate econometric models. This hybrid approach contains two forecasting stages. In the first stage, a time series model generates estimates of the exchange rates. In the second stage, General Regression Neural Network is used to correct the errors of the estimates. A number of tests and statistical measures are then applied to compare the performances of the two-stage models (with error-correction by neural network) with those of the single-stage models (without error-correction by neural network). Both empirical and trading simulation experiments suggest that the proposed hybrid approach not only produces better exchange rate forecasts but also results in higher investment returns than the single-stage models. The effect of risk aversion in currency trading is also considered.

Keywords: Neural networks; Multivariate time series; Exchange rate forecasting; Foreign currency; Investment strategies and trading

1. Introduction

The difficulty in predicting exchange rates has been a long-standing problem in international finance as many standard econometric methods are unable to produce significantly better forecasts than the

* Corresponding author. Tel.: +1-210-458-5776; fax: +1-210-458-6350.
E-mail addresses: finasc@ccunix.ccu.edu.tw (A.-S. Chen), mtleung@utsa.edu (M.T. Leung).

doi:10.1016/S0305-0548(03)00064-9

random walk model. Of the various methods of forecasting exchange rates, multivariate time series models have received much attention in the literature and tend to produce the best out-of-sample forecasts. The literature on multivariate time-series models is quite extensive. Specifically in the area of exchange rate forecasting, Canarella and Pollard [1], Wolff [2], Hoque and Latif [3], Liu et al. [4], and Sarantis and Stewart [5] show that multivariate time-series models have some forecasting strength in predicting exchange rates and are able to produce forecasts that are superior to the random walk model. Additional studies using time series techniques on exchange rates include Lothian [6], Joseph [7], and Trapletti et al. [8]. Interested readers can also refer to the exchange rate forecasting book written by Moosa [9].

In the last decade, with the rapid advancement of computer technologies and growing popularity of artificial intelligence, researchers and practitioners have become more inclined to adopt artificial neural network (ANN) as an alternative method in financial forecasting. Although most ANN models share a common goal of performing functional mapping, different network architectures vary greatly in their ability to handle different types of problems. In this study, we select the General Regression Neural Network (GRNN) to conduct currency exchange rate forecasting. It is because GRNN, due to its rather rigid structure, requires relatively less amount of time and effort on training and on the design of architectural construct (see Masters [10]). Our choice is also partly motivated by the promising results reported in Leung et al. [11]. In their paper, performances of GRNN models are compared with those of the more widely used multilayered feedforward network (MLFN), random walk model, and multivariate transfer function.

While GRNN possesses some strength in functional mapping, its intrinsic mathematical operations render it weaker than most econometric models in predicting values in the state space which extends beyond the training data. On the other hand, most econometric models are capable of extrapolating values outside of the existing/known data range during forecasting process. Given this notion, we propose a two-stage model which combines an econometric model with GRNN to overcome this shortcoming and to create synergies in the overall forecasting process. In this two-stage model, an econometric model is used to generate a forecast in the first stage. In the second stage, a GRNN is applied to correct the estimation error in the first stage forecast. It is hoped that this error correction by the neural network can lead to an improvement in the final forecast.

The fundamental concept of the proposed hybrid approach originates from adaptive forecasting which attempts to uncover possible data patterns not captured by a forecasting model in the first pass. During the second pass in the forecasting process, another forecasting model is applied to pick up potential data patterns hidden in the residual term. A vast majority of the models utilized in this adaptive forecasting paradigm have a rather traditional econometric background. Most of these econometric models are linear in their functional forms and hold under specific parametric assumptions. Our proposed hybrid approach deviates from this tradition as it combines the strengths of a neural network and a multivariate econometric model. Furthermore, the approach takes an unconventional step of applying a nonparametric (neural network) model to learn and thus reduce the errors made by various parametric (econometric) models. This approach can also be roughly viewed as nonweighted forecast combination in a sequential manner.

In the current study, adaptive error correction models are used to forecast one-quarter-ahead exchange rates. Their forecasting performances are then compared with several competing models including the ones solely based on single forecasting techniques. Our investigation of GRNN has been motivated by the neural network's attractive properties discussed earlier. In the next section, we will provide the economic rationale associated with currency exchange rate forecasting. In Section 3, we present a brief review and mathematical foundation of GRNN and those econometric techniques used in this research. Then, the data set and our estimation procedures are explained in Section 4. Statistical performances of the two-stage error correction models are reported in Section 5. This section also includes comparisons with the single-stage econometric and GRNN models (i.e., models used to directly forecast the exchange rates). In Section 6, we describe a simulation experiment designed to measure the profit derived from the trading of various currencies. A background of the trading decision rules and a discussion of the trading experiment results are also given. The paper is then concluded in Section 7.

2. Literature review

2.1. Economic background

There are several theories of exchange rate determination normally used in empirical studies: the flexible price monetary model, the sticky price monetary model, the Hooper–Morton model, the portfolio balance model, and the uncovered interest parity model. The flexible price monetary model, the sticky price monetary model, and the Hooper–Morton model are essentially different versions of the monetary approach. Of these three monetary approaches, the Hooper–Morton model is the more general model and encompasses the other two models. Meese and Rogoff [12], Alexander and Thomas [13], Schinasi and Swamy [14], and Meese and Rose [15] are examples of studies which use the monetary approach and the Hooper–Morton model. Sarantis and Stewart [16] provide a good review and description of the main theoretical exchange rate models. Moreover, the study reports a detailed econometric evaluation of all of the theoretical models mentioned above for sterling exchange rates using the cointegration-error correction methodology. Later Sarantis and Stewart [5] show that the exchange rate forecasting models based on the uncovered interest parity (UIP) relationship produce more accurate out-of-sample forecasts. In addition, the empirical study shows that the Bayesian vector autoregression (BVAR) model based on UIP produces the best short-term forecasts, which also outperform the random walk forecasts.

Following the findings from these previous studies, we adopt the UIP relationship as the theoretical basis for our exchange rate forecasting models. The UIP relationship can be cast in the following general forecasting form:

$$\Delta e_{t+1} = F\begin{pmatrix} \Delta e_t, \Delta e_{t-1}, \Delta e_{t-2}, \ldots, \\ \Delta(r_t^* - r_t), \Delta(r_{t-1}^* - r_{t-1}), \Delta(r_{t-2}^* - r_{t-2}), \ldots, \\ \Delta(\pi_t^* - \pi_t), \Delta(\pi_{t-1}^* - \pi_{t-1}), \Delta(\pi_{t-2}^* - \pi_{t-2}), \ldots, \\ \Delta(p_t^* - p_t), \Delta(p_{t-1}^* - p_{t-1}), \Delta(p_{t-2}^* - p_{t-2}), \ldots, \\ \Delta(CA_t/NY_t), \Delta(CA_{t-1}/NY_{t-1}), \Delta(CA_{t-2}/NY_{t-2}), \ldots, \\ \Delta(CA_t^*/NY_t^*), \Delta(CA_{t-1}^*/NY_{t-1}^*), \Delta(CA_{t-2}^*/NY_{t-2}^*), \ldots, \end{pmatrix}, \tag{1}$$

where e is the natural logarithm of the exchange rate, defined as the foreign currency price of domestic currency. r, π, p and (CA/NY) represent the logarithm of nominal short-term interest rate, the expected price inflation rate, the logarithm of the price level, and the ratio of current account to nominal GDP for the domestic economy, respectively. Asterisks denote the corresponding foreign variables. Similar to Fisher et al. [17], the variables (CA/NY) and (CA^*/NY^*) are proxies for the risk premium. Essentially, Eq. (1) is a modified version of the real uncovered interest parity (MUIP) relationship adjusted for risk. Sarantis and Stewart [16] supply a more detailed discussion and investigation of the MUIP relationship. Moreover, Sarantis and Stewart [5] provide the out-of-sample forecasting performance of various multivariate time series and structural models based on the MUIP.

2.2. Neural networks in forecasting exchange rates

Due to their success in financial forecasting, neural networks have been adopted as an alternative method in the prediction of exchange rates. Literature documenting this research effort is quite diverse and involves different architectural designs. A brief survey of research on such designs is presented here. Refenes [18] uses a MLFN in conjunction with a constructive learning algorithm to forecast currency exchange rates. The study also covers a technique for constructing and training a hidden unit with the network architecture. Hsu et al. [19] develop a clustering neural network model to predict the direction of movements in the USD/DEM exchange rate. The experimental results suggest that their proposed model can achieve better forecasting performance relative to other indicators. De Matos [20] compares the strength of a MLFN with that of a recurrent network based on the forecasting of Japanese yen futures. Also, Kuan and Liu [21] provide a comparative evaluation of the performances of MLFN and recurrent network on the prediction of an array of commonly traded exchange rates. Likewise, Tenti [22] applies recurrent neural network models to forecast exchange rates. In a more recent study, Leung et al. [11] compare the forecasting accuracy of MLFN with that of GRNN. The study finds that GRNN possesses a greater forecasting strength relative to MLFN with respect to a variety of currency exchanges. Zhang and Berardi [23] adopt a different approach. Instead of using single network architecture, the paper investigates the use of ensemble methods to exchange rate forecasting. Essentially, the study proposes the use of systematic and serial partitioning methods to build ensemble models consisting of different neural network structures. Their results indicate that the ensemble network can consistently outperform a single network design.

Although there exist only a handful of studies of neural network applications in foreign exchange markets, most of the literature focuses on the forecasting of exchange rate levels. Only a few are devoted to the trading of foreign exchanges and its financial significance. Among these, Shin [24] studies the forecastability of returns on trading currency using neural networks while Refenes and Zaidi [25] focus on the trading of foreign exchange. In their paper, the authors present a hybrid ANN system for managing trading strategies in the foreign exchange markets. They also find that the ANN-guided trading of US/DEM exchange rate results in higher return than other techniques.

Motivated by the findings of some of these previous studies, in this paper, we attempt a hybrid approach which combines the strengths of GRNN and econometric models. In addition to forecasting accuracy, profitability of the hybrid as well as GRNN and other econometric models on currency trading are evaluated. It is hoped that the experimental outcomes can provide some guidelines to the capability of ANN-guided trading mechanisms in financial investment.

3. Methodologies

In this section, we summarize the foundation of the econometric models used in our study. It is followed by a brief description of the architecture and operational logic of the regression neural network. We also show how to combine the strengths of the regression neural network and a selected econometric model within an adaptive forecasting framework.

3.1. Single-stage econometric models

As mentioned earlier, we adopt a number of econometric models to forecast one-month-ahead exchange rates. These econometric models serve two primary purposes. First, they provide a gauge of the forecasting performance when the models are used alone, that is, used in a single-stage forecasting framework. Second, the econometric models yield estimates which are corrected by the regression neural network in our proposed two-stage error correction framework. Since previous studies have provided some evidence of the strengths of multivariate time series models, we take the guidance and narrow our focus on three types of forecasting models—multivariate transfer function (MTF), generalized method of moments (GMM), and Bayesian vector autoregression (BVAR).

3.1.1. Multivariate transfer function (MTF)

A multivariate transfer function model is essentially an ARIMA with added exogenous variables. The general model can be stated as

$$\begin{aligned} y_t = {} & a_0 + a_1 y_{t-1} + \cdots + a_p y_{t-p} + \varepsilon_t + b_1 \varepsilon_{t-1} + \cdots + b_q \varepsilon_{t-q} \\ & + \frac{(\omega_{10} + \omega_{11} L + \cdots + \omega_{1n} L^n)}{(1 - \delta_{11} L - \cdots - \delta_{1m} L^m)} x_{1,t-b} + \frac{(\omega_{20} + \omega_{21} L + \cdots + \omega_{2r} L^r)}{(1 - \delta_{21} L - \cdots - \delta_{2s} L^s)} x_{2,t-b} + \cdots, \end{aligned} \tag{2}$$

where $x_{1,t-b}, x_{2,t-b}, \ldots$ are exogenous variables lagged for b periods, L is the lag operator, y_t is the dependent variable, and ε_t is the residual. The addition of exogenous variables can improve forecasting if they help explain the stock excess returns. Interested readers should refer to Green [26] for a detailed explanation of the technique. Potential exogenous variables for consideration are the factors in the MUIP relationship described earlier.

3.1.2. Generalized method of moments (GMM)

In addition, the generalized method of moments (GMM) with Kalman filter is utilized to generate forecasts. The GMM models used in this study can be summarized in the following way: Let β_t denote the vector of *states* (coefficients) corresponding to the state variables at time t. The measurement equation is the GMM model:

$$y_{t+1} = \boldsymbol{X}_t \beta_t + \mu_t, \tag{3}$$

where y_t is the dependent variable and $\boldsymbol{X}_t$ with m columns is the matrix of independent variables. μ_t is the error term with variance n_t. The state vector follows the process:

$$\beta_t = \beta_{t-1} + \boldsymbol{v}_t \tag{4}$$

with $Var(\boldsymbol{v}_t) = \boldsymbol{M}_t \cdot \mu_t$ and $\boldsymbol{v}_t$ are independent and n_t and $\boldsymbol{M}_t$ (variance of the change in state vector) are assumed known. The Kalman filter recursively updates the estimate of β_t (and its variance) by using the new information in y_t and $\boldsymbol{X}_t$ for each observation. Once we have an estimate of β_{t-1} and its covariance matrix $\boldsymbol{\Sigma}_{t-1}$, then the updated estimate given y_t and $\boldsymbol{X}_t$ is

$$\boldsymbol{S}_t = \boldsymbol{\Sigma}_{t-1} + \boldsymbol{M}_t, \tag{5}$$

$$\boldsymbol{\Sigma}_t = \boldsymbol{S}_t - \boldsymbol{S}_t\boldsymbol{X}_t'(\boldsymbol{X}_t\boldsymbol{S}_t\boldsymbol{X}_t' + n_t)^{-1}\boldsymbol{X}_t\boldsymbol{S}_t, \tag{6}$$

$$\beta_t = \beta_{t-1} + \Sigma_t\boldsymbol{X}_t'\boldsymbol{n}_t^{-1}(y_t - \boldsymbol{X}_t\beta_{t-1}). \tag{7}$$

To use the previous updating equations, we need to supply β_0, the initial state vector; $\boldsymbol{\Sigma}_0$, the initial covariance matrix of the states; n_t, the variance of the measurement equation, and $\boldsymbol{M}_t$, the variance of the change in the state vector. In our study, $\boldsymbol{M}_t$ is set to 0 whereas n_t, $\boldsymbol{\Sigma}_0$, and β_0 are estimated using GMM. Details of the GMM estimation are described in Greene [26].

3.1.3. Bayesian vector autoregression (BVAR)

The BVAR model is essentially a restricted version of the vector autoregression (VAR) model. The VAR model is a general dynamic model with several endogenous variables in the system. In the VAR model, each variable is a linear function of lagged values of all variables in the system where the lag lengths for each variable are identical and every variable in the system is included in each equation. Technically, a VAR model with a lag length of p can be represented as

$$\boldsymbol{Z}_t = \sum_{s=1}^{p} \boldsymbol{\phi}(\boldsymbol{s})\boldsymbol{Z}_{t-s} + \boldsymbol{\mu}_t, \tag{8}$$

$$E(\boldsymbol{\mu}_t\boldsymbol{\mu}_t') = \boldsymbol{\Sigma}, \tag{9}$$

where $\boldsymbol{Z}_t$ is an $(n \times 1)$ vector of variables measured at time period t, $\boldsymbol{\phi}(s)$ is an $(n \times n)$ matrix of the coefficients, p is the lag length of the variables, and $\boldsymbol{\mu}_t$ is an $(n \times 1)$ vector of random disturbances. Instead of eliminating longer lags and/or less important variables, the BVAR technique imposes restrictions on these coefficients based on the assumption that these coefficients are more likely to be near zero than the coefficients on shorter lags and/or more important variables. Nevertheless, the BVAR model also allows the data to override this assumption, if there are strong effects from longer lags and/or less important variables. Interested readers should refer to Hamilton [27] for a more detailed exposition of the technique.

3.2. Single-stage regression neural network model

Although most ANN models share a common goal of performing functional mapping, different network architectures vary greatly in their ability to handle different types of problems. Given its superior forecasting performance reported in Leung et al. [11], general regression neural network (GRNN) is selected as the network architecture in our empirical experiment. Like econometric models, GRNN has two functions. First, it serves as the second stage in the proposed two-stage forecasting framework in which GRNN estimates and corrects the errors of the econometric model estimates. Moreover, GRNN directly performs single-stage neural network forecasting, the results of which provide benchmarks for comparisons with other single-stage and two-stage models.

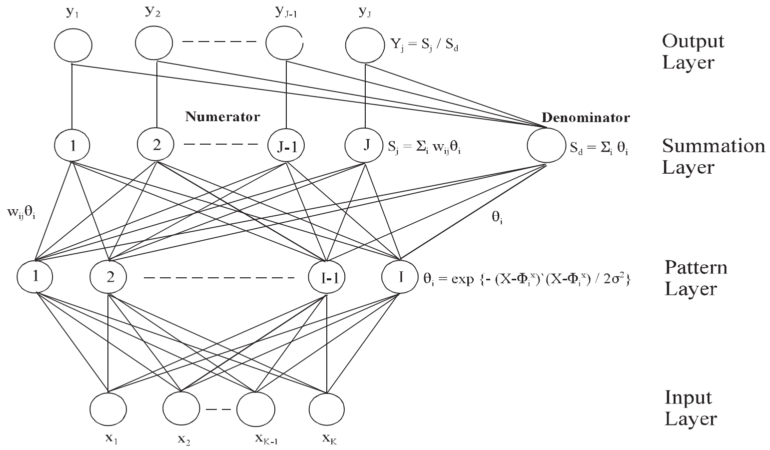

Fig. 1. General GRNN construct. A typical GRNN construct consists of four neuron layers—input, pattern, summation, and output layers. Input layer receives the input vector X and distribute the information to the neurons in the pattern, layer. Each neuron in the pattern layer will then generate an intermediate output θ. The numerator and denominator neurons in the subsequent summation layer will compute the weighted and single arithmetic sums based on the values of θ and the proper output w_{ij} learned through supervised training. The actual mathematical operations performed by these neurons in the hidden layers are illustrated in the diagram. The neurons in the output layer will then carry out divisions of the sums computed by numerator and denominator neurons in the summation layer.

Essentially, GRNN is able to estimate any arbitrary relationship between a given set of input variables and its corresponding outputs. This estimation procedure is carried out by the network during the training process. Upon the completion of training, the deduced relationship is used to compute the (expected value of) output vector based on a given input vector. In the GRNN model, estimation of a dependent variable y with respect to a given vector of independent variables $\boldsymbol{X}$ can be regarded as finding the expected value of y conditional on the value of $\boldsymbol{X}$. Eq. (10) summarizes this statistical concept.

$$E[y|\boldsymbol{X}] = \frac{\int_{-\infty}^{\infty} y f(\boldsymbol{X}, y)\,\mathrm{d}y}{\int_{-\infty}^{\infty} f(\boldsymbol{X}, y)\,\mathrm{d}y}, \tag{10}$$

where y is the output value estimated by GRNN, $\boldsymbol{X}$ is the input vector for the estimation of y, and $f(\boldsymbol{X}, y)$ is the joint probability density function of $\boldsymbol{X}$ and y learned by GRNN from the available training data set.

During the training session, GRNN establishes the joint density function $f(\boldsymbol{X}, y)$ by observing each sample of $(\boldsymbol{X}, y)$ listed in the training data set. After the training is completed, the network estimates the most probable output value $E[y|\boldsymbol{X}]$ when a new input vector $\boldsymbol{X}$ is presented to the network. Since there is no restriction on the functional form or continuity of the joint density function, the regression performed by GRNN can be quite general.

In his original design, Specht [28] replicates this regression process by a special four-layer neural network. A typical GRNN construct is depicted in Fig. 1. In the first layer, which is the input layer, an input vector $\boldsymbol{X}$ is presented to the network for processing. Thus, the number of neurons contained in this layer is equal to the number of elements, K, in the input vector. The input layer merely serves as an input data buffer and does not perform any processing. The input data is then passed on to the second layer, the pattern layer, where each training vector (sample) is represented by a single neuron. Hence, there are I pattern neurons running in parallel if the training data set consists of a total of $i = 1, 2, \ldots, I$ samples. Each individual neuron i then generates an output θ_i based on the input provided by the input layer. The transformation calculated in each of the pattern neurons is essentially a Cacoullos' [29] multivariate extension to the Gaussian function and is shown in Eq. (7):

$$\theta_i = \mathrm{e}^{(\boldsymbol{X}-\boldsymbol{\Phi}_i^x)(\boldsymbol{X}-\boldsymbol{\Phi}_i^x)/2\sigma^2}, \tag{11}$$

where $\boldsymbol{X}$ is the input vector for estimating $\boldsymbol{Y}$, σ is a set of smoothing parameters, and Φ_i^x is the input portion of the ith training vector represented by the ith neuron in the pattern layer.

Every neuron in the first hidden layer is connected subsequently to each of the neurons residing in the third layer, the summation layer, where summations are performed in the neurons. Fig. 1 shows that there are two groups of summation neurons, namely, numerator and denominator neurons. There are J numerator neurons in the summation layer to represent $j = 1, 2, \ldots, J$ elements in the regression output vector $\boldsymbol{Y}$. The group of numerator summation neurons is responsible for computing the weighted sum of the outputs from the pattern neurons. Mathematically, the transformation carried out in the numerator neuron can be written as

$$S_j = \sum_{t=1}^{I} w_{ij}\theta_i, \tag{12}$$

where S_j is the output from the jth numerator neuron, θ_i is the output from the ith neuron in the pattern layer, and w_{ij} is the weight assigned to the connection linking the ith neuron in the pattern layer to the jth neuron in the summation layer.

The network "learns" the values of w_{ij} when it is subject to training. During the supervised training, training samples Φ_i, for $i = 1, \ldots, I$, are presented to the input layer one at a time. A single neuron in the pattern layer is established for each training vector encountered. The network will then assign the weight w_{ij} of the connection linking the ith neuron in the pattern layer to the jth numerator neuron in the summation layer. The assignment of weights follows the values of the output portion of each training vector Φ_i^y such that

$$w_{ij} = y_j^i, \tag{13}$$

where y_j^i is the jth element of the output portion of the training vector Φ_i.

The denominator group in the summation layer has only one neuron. This particular neuron computes the simple arithmetic sum of the output from the pattern neurons:

$$S_d = \sum_{i=1}^{I} \theta_i, \tag{14}$$

where S_d is the output from the denominator neuron. After the network is fully trained, there should be a total of $(I \times J)$ w_{ij}'s connecting the pattern neurons to the numerator summation neuron and another set of I connections with uniform weights of one linking the pattern neurons to the denominator neuron.

The number of neurons in the final layer, the output layer, is equivalent to the number of elements J in the regression output vector $\boldsymbol{Y}$. From an operational point of view, each of the output neurons is a mathematical processor which performs division on the outputs calculated by the numerator and denominator neurons in the summation layer. This division for the output neuron j can be expressed by

$$y_j = \frac{S_j}{S_d} = \frac{\sum_{i=1}^{I} w_{ij}\theta_i}{\sum_{i=1}^{I} \theta_i}. \tag{15}$$

3.3. Two-stage error correction neural network model

A common weakness encountered by many neural network models is the lack of “guessing” ability. In other words, many neural network models are relatively ineffective in making estimates for a state space it has not “seen” or been trained on. On the other hand, most econometric models possess the strength to extrapolate learned/observed results and make estimates for input vectors not included in the training set. Hence, to overcome this weakness of neural network, we propose the error correction neural network (ECNN) for exchange rate forecasting. ECNN is a two-stage forecasting procedure which combines an econometric model and GRNN in an adaptive (sequential) manner. Our conjecture is that this combination can create a synergy effect which further improves the forecasting strength.

In the first stage, an econometric model is used to generate a series of forecasts $e^{\text{model}_\text{f}}$. Let us denote the change in forecast between the current time t and the beginning of the next month as $\Delta e_{t+1}^{\text{model}_\text{f}}$. Then we can generate the forecast residuals, μ_t, defined to be the difference between the actual realized value of Δe_t minus the differenced forecast of $\Delta e_{t+1}^{\text{model}_\text{f}}$:

$$\mu_t = \Delta e_t - \Delta e_t^{\text{model}_\text{f}}. \tag{16}$$

In the second stage of the ECNN model, we train the GRNN using previously generated residual series μ_t as inputs. The trained network is then used to generate a forecast of next period’s expected residual $\mu_{t+1}^{\text{grnn}_\text{f}}$. The differenced ECNN forecast $\Delta e_{t+1}^{\text{ecnn}_\text{f}}$ is then calculated by adding the forecast of next period’s expected residual $\mu_{t+1}^{\text{grnn}_\text{f}}$ to the differenced econometric model forecast for that period $\Delta e_{t+1}^{\text{model}_\text{f}}$. To obtain the final ECNN exchange rate forecast, the series of $\Delta e_{t+1}^{\text{ecnn}_\text{f}}$ is untransformed back to e^{ecnn_f}.

3.4. Random walk model

It is also of interest to compare the results of the random walk model with the forecasts generated from the other models described in this paper. The random walk forecast is just a forecast of no change in the dependent variable. Simply speaking, the random walk model forecast of next month’s exchange rate is just this month’s exchange rate:

$$\hat{e}_{t+1} = \mathrm{e}_t, \tag{17}$$

where $\hat{e}_{t+1}$ is the random walk forecast of the exchange rate for month $(t+1)$.

4. Exchange rate forecasting

4.1. Data

The data set used in this study runs from January 1980 to December 2001 and was obtained from the International Monetary Fund (IMF). The monthly data cover 22 years of observations of all macroeconomic variables in the MUIP relationship (Eq. (1)) and three currency exchange rates—British pound/US dollar, Canadian dollar/US dollar, and Japanese yen/US dollar. In our empirical experiment, the data set is divided into two sample periods—the estimation (in-sample) and the test (out-of-sample) periods. The estimation period covers observations from January 1980 to December 1996 and is used for estimation and refinement of the forecasting model parameters. In order to provide a check for model adequacy and guidance for model selection, we break the estimation period into two sub-periods. The first estimation sub-period represents the 12 years from January 1980 through December 1991 while the second sub-period covers the remaining five years from January 1992 to December 1996. Based on an assessment of performance in the second estimation sub-period, the specification of each model type is selected and subject to out-of-sample testing. This five-year test period runs from January 1997 to December 2001 and is reserved strictly for the purpose of performance evaluation.

Moreover, the input variables in the data set are constructed such that they are all observable before the first day of the month to be forecasted. Constructing the data set in this manner ensures that the estimation of out-of-sample forecasts mimics the forecasting process performed in the real world—only observable data are available to the forecasting model.

4.2. Econometric model estimation

Three separate sets of econometric models (MTF, BVAR, GMM) are estimated for each of the three currencies using estimation samples. The forecasting experiment uses the modified uncovered interest parity (MUIP) described in Sarantis and Stewart [5] to specify the vector $\boldsymbol{Z}_t$. The vector of variables can be represented as

$$\boldsymbol{Z}_t = [\Delta \mathrm{e}_t, \Delta(r_t^* - r_t),\ \Delta(\pi_t^* - \pi_t),\ \Delta(p_t^* - p_t), \Delta(CA_t/NY_t),\ \Delta(CA_t^*/NY_t^*)]', \tag{18}$$

where $\Delta \mathrm{e}_t$ is the change in the natural logarithm of the exchange rate, defined as the foreign currency price of domestic currency. The variables $r, r^*, \pi, \pi^*, p, p^*$, (CA/NY), and (CA^*/NY^*) are as defined previously when the MUIP model was first discussed. The best specification from each set of model type is selected by examining the Theil's U values computed for the second estimation sub-period (from January 1992 to December 1996). This best model specification is then used for out-of-sample forecasting. Details of econometric model selection for ECNN are given in Section 4.4.

The out-of-sample forecasts are then produced for the test period (from January 1997 to December 2001) for each exchange rate. Specifically, after the best model specification has been determined (as described above), data from the entire estimation period, from 1970Q1 to 1989Q4, are used to estimate the initial parameters of each model and generate the first 'true' out-of-sample forecast for January 1997. As new data become observable, they are added to the existing sample while the oldest observation is removed. This process continues through the last period in the out-of-sample forecasting period (December 2001).

4.3. GRNN for single-stage exchange rate forecasting

GRNN is initially trained using data from the first estimation sub-period (from January 1980 to December 1991). After the training process is completed, the trained network is used to generate the forecast for January 1992, the first month in the second estimation sub-period. Then, the forecasting process is rolled over to the subsequent period of February 1992 using the new information released at the end of January 1992. At the same time, the oldest training sample is dropped from the training set. The network is then re-trained by the same procedures described in the last section. This rolling horizon approach parallels the estimation procedure applied to econometric estimation and is similar to the one used by Refenes [18]. The process continues through the last period in the second in-sample estimation sub-period (December 1996). The entire training process is then repeated for different currencies.

After all in-sample forecasts for the second estimation sub-period have been made for each GRNN model specification, Theil's U statistic is used to guide the selection of the best specification for the subsequent out-of-sample forecasting. Afterward, the rolling horizon approach described above is applied to the selected model specification for predicting the out-of-sample test period from January 1997 to January 2001.

4.4. Two-stage ECNN for exchange rate forecasting

In this two-stage framework, an econometric model must first be estimated and then its forecasts are corrected by GRNN. Thus, an econometric model is estimated according to the procedure outlined in Section 4.2. After that, residuals for the in-sample forecasts (from January 1992 to December 1996) are computed. GRNN is applied to estimate and correct the errors as described in Section 3.3. As we move forward, new forecasts are generated from the econometric model and new residuals are produced. Hence, as the data of a month become observable, the residual associated with this month can be generated by subtracting the econometric model forecast for this month from the actual value of Δe_t which is now observable. This newly generated residual μ_t is then added to the information set and the GRNN is retrained using the updated residual series as input. The forecast for the following month's expected residual is then generated using the retrained GRNN. Subsequently, the ECNN forecast for the next month is generated by adding the forecast of next month's expected residual to the original econometric forecast for that month. Like those single-stage models, ECNN in-sample forecasting is repeated for different model specifications. The best model is selected for each model type (MTF, BVAR, GMM) based on Theil's U statistic. It is then used for out-of-sample testing, which follows the paradigm outlined in Sections 4.2 and 4.3. The whole forecasting experiment is repeated for all three currencies.

5. Results

5.1. Performance statistics

Out-of-sample forecasting performance for all two-stage ECNN and single-stage models are tabulated in Tables 1–3, one for each of the three currencies examined. As benchmarks for comparison,

Table 1
Performance statistics for out-of-sample Canadian dollar forecasts (from January 1997 to December 2001) of different models

Model	Root mean squared error (RMSE)	Theil's U statistic	R squared (R^2)	Improvement in U stat over single-stage	Improvement in U stat over GRNN
Single-stage econometric and neural network					
MTF	0.0365	0.8648	0.9383		
BVAR	0.0329	0.8261	0.9507		
GMM	0.0352	0.8594	0.9415		
GRNN	0.0338	0.8317	0.9482		
Two-stage ECNN					
ECNN-MTF	0.0334	0.8489	0.9469	0.0159	−0.0172
ECNN-BVAR	0.0291	**0.8016**	0.9582	0.0245	0.0301
ECNN-GMM	0.0326	0.8285	0.9497	0.0309	0.0032
Random walk	0.0396	1.0000	0.9146		

Note: Theil's U statistic for the random walk model is by definition equal to 1. The best U statistic is in boldface. GMM realizes the greatest improvement in forecasting strength when its errors are corrected by GRNN.

Table 2
Performance statistics for out-of-sample Japanese yen forecasts (from January 1997 to December 2001) of different models

Model	Root mean squared error (RMSE)	Theil's U statistic	R squared (R^2)	Improvement in U stat over single-stage	Improvement in U stat over GRNN
Single-stage econometric and neural network					
MTF	5.8294	0.9283	0.8924		
BVAR	5.6519	0.8968	0.9046		
GMM	5.9847	0.9337	0.8749		
GRNN	5.5240	0.8876	0.9246		
Two-stage ECNN					
ECNN-MTF	5.4133	0.9047	0.9013	0.0136	−0.0171
ECNN-BVAR	5.0645	**0.8713**	0.9183	0.0155	0.0163
ECNN-GMM	5.4648	0.8832	0.8942	0.0255	0.0044
Random walk	6.1830	1.0000	0.8513		

Note: Theil's U statistic for the random walk model is by definition equal to 1. The best U statistic is in boldface. GMM realizes the greatest improvement in forecasting strength when its errors are corrected by GRNN.

corresponding performance statistics for the random walk model are also included. Three descriptive statistics on forecast accuracy reported in the tables are root mean square error (RMSE), Theil's U, and R^2. Theil's U statistic is the ratio of the RMSE of the model forecast to the RMSE of the random walk forecast of no change in the dependent variable. Since the random walk forecast of next quarter's exchange rate is just this quarter's exchange rate, a U statistic of less than 1 implies

Table 3
Performance statistics for out-of-sample British pound forecasts (from January 1997 to December 2001) of different models

Model	Root mean squared error (RMSE)	Theil's U statistic	R squared (R^2)	Improvement in U stat over single-stage	Improvement in U stat over GRNN
Single-stage econometric and neural network					
MTF	0.0397	0.9716	0.5856		
BVAR	0.0372	0.9642	0.5911		
GMM	0.0359	0.9598	0.6014		
GRNN	0.0367	0.9618	0.5977		
Two-stage ECNN					
ECNN-MTF	0.0355	0.9603	0.6003	0.0113	0.0015
ECNN-BVAR	0.0348	0.9579	0.6079	0.0063	0.0039
ECNN-GMM	0.0338	**0.9561**	0.6264	0.0037	0.0057
Random walk	0.0408	1.0000	0.5489		

Note: Theil's U statistic for the random walk model is by definition equal to 1. The best U statistic is in boldface. MTF realizes the greatest improvement in forecasting strength when its errors are corrected by GRNN.

that the model forecast outperforms the random walk model during the forecast comparison period. Likewise, a U statistic in excess of 1 implies the forecasting model did worse than the random walk model. In short, the Theil's U statistic has an advantage over the RMSE when comparing different forecasting models containing the same dependent variable—it provides an immediate comparison of the model forecasts with the random walk forecasts of no change over time.

Based on U statistics and R^2 values, all single-stage and two-stage models generate better forecasts in the out-of-sample test period than the random walk model. Also, the two-stage models are generally better than their single-stage counterparts and the best forecasting model for each currency is a two-stage ECNN model. The model with the best U statistic is highlighted in boldface.

Besides, it can be seen from the results that using the GRNN to correct the estimation errors in the econometric model can lead to improvement in the overall forecast accuracy (U statistic). However, the extent of improvement varies from model to model and from currency to currency. There is no consistent pattern as to which econometric model creates better improvement when we apply the ECNN error correction procedure. Further, it is important to note that not every two-stage ECNN model can outperform the single-stage GRNN model (i.e., exchange rates are directly forecasted by the GRNN alone). For both ECNN-BVAR and ECNN-GMM, using GRNN to correct the first-stage estimation errors can result in more accurate out-of-sample exchange rate forecasts than pure neural network forecasting. On the contrary, this ECNN improvement does not hold for MTF in which the overall accuracy (measured in U statistic) still cannot exceed that of a pure GRNN forecasting model for both the Canadian dollar and the Japanese yen.

5.2. Regression tests

In addition to the more general performance statistics, an array of regression tests are performed to compare the out-of-sample predictive ability of the various exchange rate forecasting models.

Following the methodology used in Liu et al [4], a regression of the following type is used to measure the forecast accuracy of an exchange rate forecasting model:

$$\boldsymbol{Z}_{t+1} - \boldsymbol{Z}_t = \alpha + \beta(\boldsymbol{Z}^e_{t,t+1} - \boldsymbol{Z}_t) + \mu_t. \tag{19}$$

Under this test framework, Z_{t+1} is the actual exchange rate at month $(t+1), Z_t$ is the actual exchange rate at month t, and $\boldsymbol{Z}^e_{t,t+1}$ is the one-month-ahead exchange rate forecast made at time t. Technically, this regression test measures the bias (α) and proportion (β) coefficients of the predicted exchange rate series. If the predicted series follows the true expected value of the exchange rate conditional on the currently available information at time t, regressing $(\boldsymbol{Z}_{t+1} - \boldsymbol{Z}_t)$ on $(\boldsymbol{Z}^e_{t,t+1} - \boldsymbol{Z}_t)$ should produce regression estimates of 0.0 and 1.0 for α and β, respectively. Deviation from these values is evidence of bias and inefficiency in the forecasts. The F-test can be used to test the joint hypothesis that $(\alpha, \beta) = (0, 1)$.

Regression test results on the out-of-sample forecasts from the single-stage econometric and GRNN as well as the two-stage ECNN models are illustrated in Table 4. The F-test results show that the joint hypothesis that $(\alpha, \beta) = (0, 1)$ cannot be rejected for all cases, indicating that all of the forecasts generated by the models used in this study are statistically useful in the prediction of exchange rates. However, it can be observed that the F statistics for the two-stage models are generally less than those for the single-stage counterparts. Also, the use of GRNN to correct the errors originating from a single-stage econometric model can improve the overall accuracy of the forecasts. Similar to the results of performance statistics, the regression tests find that GRNN, when used alone, may or may not outperform some of the two-stage ECNN models. Further, there is no consistent conclusion as to which first-stage econometric model yields better forecastability in the two-stage ECNN framework.

An examination of the bias and proportion coefficients from the regression tests leads to similar conclusions. Essentially, there is no model which consistently outperforms the others in terms of bias and proportion accuracy. Generally, some improvement in bias and/or proportion accuracy can be obtained by using GRNN to correct the estimation errors from single-stage models. Nevertheless, there is no absolute guarantee that the improvement due to ECNN correction can outperform the forecast accuracy of a single-stage GRNN model, although, the improvement usually leads to better results than the single-stage GRNN. This finding justifies the use of GRNN model alone, especially when a forecaster is willing to tradeoff some degree of accuracy for a simpler approach.

6. Currency trading

Although the performance statistics and regression tests provide a fairly reasonable evaluation of the forecasting accuracy of the single-stage and two-stage models, the relative rank of profitability of these models may not be the same in a trading environment. Hence, we conduct a currency trading simulation to measure the financial significance of our models. The trading rules used in the simulation experiment are guided by the exchange rate forecasts estimated by each of the single-stage and two-stage models. Random walk model is excluded from this simulation due to its relatively poor performance statistics and regression test results. In the following sections, we describe the environment and logic of our trading simulation and discuss the outcomes of this trading experiment.

Table 4
Out-of-sample regression tests results for exchange rate forecasting models (January 1997 to December 2001)

Model	α (bias coefficient)	$t(\alpha)$[a]	β (proportion coefficient)	$t(\beta)$[b]	F[c]
Canadian Dollar					
Single-stage econometric and neural network					
MTF	0.0073	1.5681	0.7167	−0.8583	0.8561
BVAR	0.0062	1.3218	0.7968	−0.3704	0.3819
GMM	0.0057	0.8583	0.7691	−0.5934	0.6972
GRNN	0.0074	1.5813	0.7592	−0.6248	0.6912
Two-stage ECNN					
ECNN-MTF	−0.0061	−1.3067	0.7295	−0.8382	0.8235
ECNN-BVAR	0.0051	0.6816	0.8083	−0.3346	0.3264
ECNN-GMM	0.0056	0.8072	0.7782	−0.5719	0.4736
Japanese Yen					
Single-stage econometric and neural network					
MTF	1.2354	1.0135	1.4467	0.9135	0.7931
BVAR	−1.2688	−1.0881	0.3728	−1.4508	0.6723
GMM	−1.3468	−1.1643	0.3136	−1.5943	0.8490
GRNN	1.5234	1.2666	0.4364	−1.2356	0.5232
Two-stage ECNN					
ECNN-MTF	0.8435	0.4813	1.6813	1.5062	0.5270
ECNN-BVAR	−1.0863	−0.5735	0.6136	−0.9762	0.3716
ECNN-GMM	−1.2674	−0.7568	0.5913	−0.8519	0.3942
British Pound					
Single-stage econometric and neural network					
MTF	−0.0073	−0.2035	0.7167	−1.0681	0.3410
BVAR	0.0067	0.9869	0.7968	−0.8984	0.3085
GMM	0.0082	0.3564	0.7891	−0.9732	0.2894
GRNN	0.0074	1.2256	0.7692	−0.8535	0.2971
Two-stage ECNN					
ECNN-MTF	−0.0061	−0.9067	0.8295	−0.6032	0.2642
ECNN-BVAR	0.0026	0.2934	0.8483	−0.3319	0.1973
ECNN-GMM	−0.0037	−0.3942	0.8782	−0.5737	0.1683

Notes: The regression is specified as follows: $Z_{t+1} - Z_t = \alpha + \beta(Z^e_{t,t+1} - Z_t) + \varepsilon_t$, where Z_{t+1} is the actual exchange rate at quarter $t+1$, Z_t is the actual exchange rate at time t, and $Z^e_{t,t+1}$ is the one quarter ahead exchange rate forecast made at time t. BVAR is the Bayesian vector autoregressive model. NAER is the nonlinear adaptive error reduction model. GRNN is the general regression neural network model. None of the reported statistics reported on the table are statistically significant at usual levels indicating that overall out-of-sample forecasts generated by the models are statistically unbiased.

[a]The t-values in parentheses are calculated for the null hypothesis that $\alpha = 0$.

[b]The t-values in parentheses are calculated for the null hypothesis that $\beta = 1$.

[c]F statistics are calculated to test the null hypothesis of $(\alpha, \beta) = (0, 1)$.

6.1. Trading simulation

Our trading simulation covers the out-of-sample period from January 1997 to December 2001, the same five-year period used to measure the forecasting accuracy reported in Section 5. Essentially, at the end of each month, an investor has to select the investment position for the upcoming month. There are three positions an investor can consider—buy the foreign currency and deposit the amount in a foreign money market account (long position), short sell the foreign currency and deposit the US dollars in a domestic money market account (short position), or simply leave the money in the domestic money market account (deposit position). When the investor deposits the foreign currency in a foreign bank account, the investment receives an ongoing money market interest prevailing in that foreign country. Likewise, the investment receives the US money market interest when the investment is deposited in a domestic US account. Moreover, if the investor chooses to buy or short sell the currency, he must close the position at the end of next monthly period, that is, he must sell or buy back the currency to terminate the open position regardless of the exchange rate level at the end of next quarter. Therefore, the objective of our proposed decision rules is to strategically select the investment position at the end of each period in order to maximize the investment return.

Without any loss of generality, the simulation assumes that a fixed amount of \$100 is set for investment purpose at the end of each month. This means that, regardless of the past performance, the investor has to invest exactly \$100 in the foreign currency and/or money markets. Furthermore, the investor cannot withdraw money from the account or close the trading position in the middle of any monthly period. This rationale follows the idea of one-month term deposit (CD).

6.2. Trading rules

The investment returns are determined by three components, namely, appreciation of foreign currency, depreciation of foreign currency, and interest received from money market. Let r and r^* be the monthly domestic (US) and foreign money market rates and, e_t and $E(e_{t+1})$ be the observed exchange rate at the end of month t and the forecast of exchange rate at the end of month $(t+1)$, respectively. Idealistically, an optimal allocation of investment can be made if the domestic and foreign money market rates as well as the observed exchange rate in the current month and the forecast for the next month are known. Nevertheless, the exchange rate forecast is only an estimate and involves uncertainty. Given the notion that an investor is risk averse and can earn riskfree interest from the domestic money market, we need to discount the expected rate of return based on a long or a short position by a risk aversion factor, γ. The risk aversion factor γ can take on any real value greater than negative one ($\gamma > -1.0$). For the risk neutral case, γ is zero. The value of γ is greater than zero ($\gamma > 0.0$) if the investor is risk averse. On the other hand, the value of γ is between zero and negative one ($0.0 > \gamma > -1.0$) if the investor is a risk lover.

Based on this logical framework, we develop a set of risk-adjusted optimal conditions for the currency trading rules. The mathematical expressions for these conditions and their corresponding trading decisions are outlined in Table 5 (derivations of the optimal conditions can be furnished upon request). Intuitively, when γ is equal to zero, the investor is risk neutral and chooses the position which gives the highest expected profit. However, when γ is greater than zero, the risk aversive investor requires an expected profit from the currency trading higher than the riskfree money market interest in order to persuade him to trade the foreign currency. On the contrary, when

Table 5
Optimal conditions for long, short, and deposit currency investment positions

Financial position	Optimal conditions to be satisfied	
Long (buy foreign currency and deposit the investment in foreign money market account)	$\ln\left(\frac{E(e_{t+1})}{e_t}\right) < (r^* - r)(1+\gamma)$ for $(r^* - r)(1+\gamma) < 0$	$\ln\left(\frac{E(e_{t+1})}{e_t}\right) \leqslant \frac{(r^* - r)(1+\gamma)}{2}$ for $(r^* - r)(1+\gamma) \geqslant 0$
Short (short sell foreign currency and deposit the obtained amount in domestic (US) money market account)	$\ln\left(\frac{E(e_{t+1})}{e_t}\right) > \frac{(r^* - r)(1+\gamma)}{2}$	$\ln\left(\frac{E(e_{t+1})}{e_t} > 0\right)$
Simple deposit (deposit the investment in domestic money market account)	$\ln\left(\frac{E(e_{t+1})}{e_t}\right) \geqslant (r^* - r)(1+\gamma)$	$\ln\left(\frac{E(e_{t+1})}{e_t}\right) \leqslant 0$

γ is less than zero, the risk loving investor will trade the riskier currency even when the expected profit from currency trading is less than the riskfree money market interest. Based on the results from our pilot study, three levels of risk aversion ($\gamma = -0.3, 0.0, 1.3$) are chosen for the trading simulation.

6.3. Results of trading experiment

Table 6 shows the average annualized rate of return over the out-of-sample test period, for each of the single-stage and two-stage models. The results indicate that most two-stage ECNN models perform better than their single-stage econometric and GRNN counterparts although the profitability of some ECNN models are not as high as some single-stage models. The findings here confirm the notion that single-stage models still have considerable value in forecasting, which are similar to the results of performance statistics and regression tests. Besides, a forecaster should take into account the tradeoff between the forecast improvement from and the extra computational requirement of conducting the error correction.

It is worthwhile to point out that ECNN-BVAR model generates the highest returns for both Canadian dollar and Japanese yen. For British pound, ECNN-BVAR is not the best performer but an investor can still realize a 0.41% improvement annually by adopting ECNN. On the other hand, MTF is not doing well relative to other single-stage models in British pound forecasting. Nevertheless, using GRNN to correct the error in the second stage yields a 0.61% improvement in annual return, making ECNN-MTF the second best model for guiding pound trading. These observations suggest that the performance rank among the single-stage models does not necessarily hold after the error reduction procedure in ECNN models. Some of the gain in profitability due to this procedure may be too substantial to ignore.

Table 6
Average annualized rates of return for currency trading in the out-of-sample period (from January 1997 to December 2001)

	Risk aversion level (γ)		
	−0.3 (risk loving) (%)	0.0 (risk neutral) (%)	0.3 (risk aversive) (%)
Canadian Dollar			
MTF	6.17	6.17	6.17
BVAR	8.79	8.79	8.79
GMM	8.25	8.25	8.25
GRNN	8.07	8.07	8.07
ECNN-MTF	6.38	6.57	6.57
ECNN-BVAR	9.35	9.35	9.35
ECNN-GMM	8.94	8.61	8.61
Japanese Yen			
MTF	8.23	8.14	8.14
BVAR	7.92	7.92	7.92
GMM	7.08	7.32	7.32
GRNN	8.81	8.81	8.81
ECNN-MTF	8.65	8.65	8.65
ECNN-BVAR	9.50	9.85	9.85
ECNN-GMM	9.28	9.16	9.16
British Pound			
MTF	5.38	5.15	5.15
BVAR	5.23	5.23	5.23
GMM	5.58	5.58	5.58
GRNN	5.15	5.37	5.37
ECNN-MTF	5.76	5.76	5.76
ECNN-BVAR	5.64	5.64	5.64
ECNN-GMM	6.06	6.06	6.06

The table outlines the average annualized return on trading of Canadian dollar, Japanese yen, and British pound with respect to various forecasting model and risk aversion level. The best model for each of the forecasting approach, based on the performance statistics, is selected to guide the currency trading during the out-of-sample period. The risk aversion level represents the conservativeness of investor where $\gamma < 0.0, \gamma = 0.0$, and $\gamma > 0.0$ correspond to risk loving, risk neutral, and risk aversive behavior, respectively.

7. Conclusions

This study introduces a two-stage error correction neural network model to forecast one-month-ahead exchange rates and compares its out-of-sample forecasting performance with a variety of single-stage econometric and neural network models. The economic foundation of the models used in this study is based on the MUIP relationship described by Sarantis and Stewart [5], which shows to produce the best out-of-sample forecasts among several competing models in forecasting bilateral British pound exchange rate. Our experimental study includes also the Canadian dollar and Japanese yen.

From the empirical experiment, the out-of-sample performance statistics and the regression tests indicate that the ECNN models generally outperform the single-stage econometric and GRNN models although the improvement resulting from using the two-stage models varies from currency to currency. However, there is no consistent pattern as to which econometric model should be applied to the first-stage estimation in ECNN model. This suggests that the neural network is able to correct the errors in econometric estimates and, thus, improve the overall forecasting performance.

In this research, we also provide and test a set of investment decision rules guided by the forecasts generated by any singe-stage and ECNN model. A risk aversion factor is also incorporated into the decision rules to account for the preference of investor. Similar to the conclusions drawn by the performance statistics and regression tests, results from the trading simulation show that the currency investments guided by ECNN models usually yield higher average returns than the investments guided by other models. In addition, it is found that the conclusions generally hold for all three currencies examined and are robust to risk aversion level.

Acknowledgements

The authors wish to thank Doug Blocher of Indiana University and Ram Tripathi of the University of Texas for their valuable inputs and comments. Any errors in this paper are our own.

References

[1] Canarella G, Pollard SK. Efficiency in foreign exchange markets: a vector autoregression approach. Journal of International Money and Finance 1987;7:331–46.
[2] Wolff CP. Exchange rate, innovation and forecasting. International Money and Finance 1988;7:49–61.
[3] Hoque A, Latif A. Forecasting exchange rate for the Australian dollar visa vis the US dollar using multivariate time series models. Applied Economics 1993;25:403–7.
[4] Liu TR, Mary EG, Scott HI. The performance of alternative VAR models in forecasting exchange rates. International Journal of Forecasting 1994;10:419–33.
[5] Sarantis N, Stewart C. Structural, VAR and BVAR models of exchange rate determinate: a comparison of their forecasting performance. Journal of Forecasting 1995;14:201–15.
[6] Lothian J. Behavior of real exchange rates. International Journal of Forecasting 1987;3:17–42.
[7] Joseph NL. Model specification and forecasting foreign exchange rates with vector autoregressions. Journal of Forecasting 2001;20:451–84.
[8] Trapletti A, Geyer A, Leisch F. Forecasting exchange rates using cointegration models and intra-day data. Journal of Forecasting 2002;21:151–66.
[9] Moosa IA. Exchange rate forecasting: techniques and applications. New York: Palgrave Macmillan, 2000.
[10] Masters T. Advanced algorithms for neural networks. New York: Wiley, 1995.
[11] Leung MT, Chen AS, Daouk H. Forecasting exchange rates using general regression neural networks. Computers and Operations Research 2000;27:1093–110.
[12] Meese RA, Rogoff K. Empirical exchange rate models of the seventies: do they fit out of sample? Journal of International Economics 1983;14:3–24.
[13] Alexander D, Thomas LR. Monetary/asset models of exchange rate determination: how well have they performed in the 1980's? International Journal of Forecasting 1987;3:53–64.
[14] Schinasi GJ, Swamy PAVB. The out-of-sample forecasting performance of exchange rate models when coefficients are allowed to change. Journal of International Money and Finance 1989;8:373–90.
[15] Meese RA, Rose AK. An empirical assessment on non-linearities in models of exchange rate determination. Review of Economic Studies 1991;58:603–19.

[16] Sarantis N, Stewart C. Monetary and asset market models for sterling exchange rates: a cointegration approach. Economics Discussion Paper 93/1, 1993, Kingston University.
[17] Fisher PG, Tunna SK, Turner DS, Wallis KF, Whitley JD. Econometric evaluation of the exchange rate models of the U.K. economy. Economic Journal 1990;100:1230–44.
[18] Refenes AP. Constructive learning and its application to currency exchange rate forecasting. In: Trippi RR, Turban E, editors. Neural networks in finance and investing. Chicago: Probus Publishing Company, 1993. p. 465–93.
[19] Hsu W, Hsu LS, Tenorio MF. A neural network procedure for selecting predictive indicators in currency trading. In: Refenes AN, editor. Neural networks in the capital markets. New York: Wiley, 1995. p. 245–57.
[20] De Matos G. Neural networks for forecasting exchange rate. M.Sc. thesis, The University of Manitoba, Canada, 1994.
[21] Kuan CM, Liu T. Forecasting exchange rates using feedforward and recurrent neural networks. Journal of Applied Econometrics 1995;10:347–64.
[22] Tenti P. Forecasting foreign exchange rates using recurrent neural networks. Applied Artificial Intelligence 1996;10:567–82.
[23] Zhang GP, Berardi VL. Time series forecasting with neural network ensembles: an application for exchange rate prediction. Journal of the Operational Research Society 2001;52:652–64.
[24] Shin, S.H., Essays on asset return predictability. Ph.D. dissertation, Massachusetts Institute of Technology, 1993.
[25] Refenes AP, Zaidi A. Managing exchange-rate prediction strategies with neural networks. In: Refenes AN, editor. Neural networks in the capital markets. New York: Wiley, 1995. p. 213–9.
[26] Greene WH. Econometric analysis. New York: Macmillan Publishing Company, 1990.
[27] Hamilton JD. Time series analysis. Princeton, NJ: Princeton University Press, 1994.
[28] Specht D. A general regression neural network. IEEE Transactions on Neural Networks 1991;2:568–76.
[29] Cacoullos T. Estimation of a multivariate density. Annals of the Institute of Statistical Mathematics 1966;18: 179–89.