International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 4 (2012) pp. 659-670

Foreign Exchange Rate Prediction using Computational Intelligence Methods

V. Ravi[1*], Ramanuj Lal[2] and N. Raj Kiran[1]

[1]Institute for Development and Research in Banking Technology,
Castle Hills Road # 1, Masab Tank, Hyderabad – 500 057 (AP) India.
*Corresponding Author
rav_padma@yahoo.com; nrajkiran@gmail.com

[2]Department of Physics and Meteorology, Indian Institute of Technology, Kharagpur – 721302, West Bengal, India.
ramanujlal@gmail.com

***Abstract*: This paper presents the application of six nonlinear ensemble architectures to forecasting the foreign exchange rates in the computational intelligence paradigm. Intelligent techniques such as Backpropagation neural network (BPNN), Wavelet neural network (WNN), Multivariate adaptive regression splines (MARS), Support vector regression (SVR), Dynamic evolving neuro-fuzzy inference system (DENFIS), Group Method of Data Handling (GMDH) and Genetic Programming (GP) constitute the ensembles. The data of exchange rates of US dollar (USD) with respect to Deutsche Mark (DEM), Japanese Yen (JPY) and British Pound (GBP) is used for testing the effectiveness of the ensembles. To account for the auto regressive nature of the time series problem, we considered lagged variables in the experimental design. All the techniques are compared with normalized root mean squared error (NRMSE) and directional statistics (D_{stat}) as the performance measures. The results indicate that GMDH and GP based ensembles yielded the best results consistently over all the currencies. GP based ensembling emerged as the clear winner based on its consistency with respect to both D_{stat} and NRMSE, but GMDH outperforms it in one of the currencies (DEM). Based on the numerical experiments conducted, it is inferred that using the correct sophisticated ensembling methods in the computational intelligence paradigm can enhance the results obtained by the extant techniques to forecast foreign exchange rates.**

***KeyWords:* Foreign Exchange Rate forecasting, Computational intelligence, Ensemble, Intelligent techniques, Market risk**

I. Introduction

Foreign Exchange Rate (Forex) markets are one of the most liquid markets in the world. Liquidity implies the ability to be easily converted through an act of buying or selling without causing a significant movement in the price and with minimum loss of value because at any given time there are a large number of buyers and sellers in the market. A crucial factor in maintaining this liquidity is the presence of three types of market players – investors (those who are looking to invest in a currency for long term gains), arbitrators (those who wish to make risk-free profits by exploiting any price mismatch due to market inefficiencies), and speculators (who take bets on direction of price movements). Typically financial institutions would engage in all the three activities, either on behalf of their clients or on their own. Although the percentage price movements and hence marginal gains in Forex markets is very low, the principal amount (also called the Nominal value) of trading runs in trillions of dollars resulting in high absolute profits (or losses!). Traditional daily turnover was reported to be over US$ 3.2 trillion in April 2007 by the Bank for International Settlements [1]. Since then, the market has continued to grow. According to Euromoney's annual FX Poll, volumes grew a further 41% between 2007 and 2008 [2]. In such a situation the profitability of the trader depends upon his ability to predict future rate movements correctly. For large multinational firms, including banks, which conduct substantial currency transfers in the course of business, being able to accurately forecast movements of currency exchange rates can result in substantial improvement in the overall profitability of the firm.

Forecasting has been dominated by linear statistical methods for several decades. Although linear models possess many advantages in implementation and interpretation, they have serious limitations in that they cannot capture nonlinear relationships in the data which are common in many complex real world problems [3]. Approximation of linear models to complicated nonlinear forecasting problems is often not satisfactory. In the early 1980s, Makridakis [4] organized a large-scale forecasting competition (M-competition) in which the majority of commonly used linear forecasting methods were tested using 1001 real-time-series data. The results showed that no single forecasting method is globally the best. According to Zhang *et al.* [5] one of the major reasons for this conclusion is that there is a varying degree of nonlinearity in the data, which cannot be handled properly by linear statistical methods.

The popularity of Artificial Neural Networks (ANNs) and other computationally intelligent methods is derived from the fact that they are generalized nonlinear forecasting models. Palit and Popovic [6] highlight the advantages of the computational intelligent methods as follows: (i) general non-linear mapping between a subset of the past time series values and the future time series values. (ii) the capability of capturing essential functional relationships among the data, which is valuable when such relationships are not *a priori* known or are very difficult to describe mathematically and/or when the collected data are corrupted by noise. (iii) universal function approximation capability that enables modeling of arbitrary nonlinear continuous functions to any degree of accuracy. (iv) capability of learning and generalization from examples using the data-driven self-adaptive approach.

Nevertheless, predicting exchange rate movements is still a problematic task. Most conventional econometric models are not able to forecast exchange rates with significantly higher accuracy. In recent years, there has been a growing interest to adopt the state-of-the-art artificial intelligence technologies to solve the problem. One stream of these advanced techniques focuses on the use of artificial neural networks (ANN) to analyze the historical data and provide predictions to future movements in the foreign exchange market. In this study, we apply different ensemble-based techniques in predicting monthly exchange rates of US dollar with respect to three major foreign currencies – German marks (DEM), British pound (GBP) and Japanese yen (JPY). Six different non-linear ensembles are designed and tested where Backpropagation neural network (BPNN), Wavelet neural network (WNN), Multivariate adaptive regression splines (MARS), Support vector regression (SVR), Dynamic evolving neuro-fuzzy inference system (DENFIS), Group Method of Data Handling (GMDH) and Genetic Programming (GP) constitute the ensembles.

The remainder of this paper is organized as follows. A review of the literature is given in section 2. In Section 3, we give an overview of the different intelligent techniques applied in the exchange rate prediction. Then, a description of the ensemble techniques developed in this paper is provided in section 4. In Section 5, a description of the experimental methodology is presented. Section 6 discusses the results obtained. The paper is then concluded in Section 7.

II. Literature Review

Many research studies have been carried out in the area of exchange rate prediction in the recent years. De Matos [7], as part of his work, compared the strength of a multi-layer feed-forward neural network (MLFN) with that of a recurrent network based on the forecasting of Japanese yen futures. Kuan and Liu [8] provided a comparative evaluation of the performance of MLFN and a recurrent network on the prediction of an array of commonly traded exchange rates. Hsu *et al.* [9] developed a clustering neural network model to predict the direction of movements in the USD/DEM exchange rate. Their experimental results suggested that their proposed model achieved better forecasting performance relative to other indicators. Tenti [10] proposed the use of recurrent neural networks in order to forecast foreign exchange rates. Three recurrent architectures were compared in terms of prediction accuracy of futures forecast for Deustche mark currency. Muhammed and King [11] presented an evolutionary fuzzy network method for prediction in foreign exchange markets. Fuzzy systems not only provided the mechanism to integrate human linguistic knowledge into logical framework but also provided the means to extract fuzzy rules from an observed data set. Genetic Algorithms were used to adapt the parameters of the fuzzy network in order to obtain the best performance. Shazly and Shazly [12] designed a hybrid system combining neural networks and genetic training to the 3-month spot rate of exchange for four currencies: the British pound, the German mark, the Japanese yen and the Swiss franc. The empirical results revealed that the networks' forecasts outperformed predictions made by both the forward and futures rates in terms of accuracy and the direction of change in the exchange rate movement. Also recently, Leung *et al.* [13] in their study compared the forecasting accuracy of MLFN with the general regression neural network (GRNN). Their study showed that the GRNN possessed a greater forecasting strength relative to MLFN with respect to a variety of currency exchanges. Zhang and Berardi [14] investigated the use of neural network combining methods to improve time series forecasting performance of the traditional single keep-the-best (KTB) model. Instead of using single network architecture, their research investigated the use of ensemble methods in exhange rate forecasting. Two general approaches to combining neural netoworks were proposed and examined in predicting the exchange rate between British pound and US dollar. Essentially, the study proposed using systematic and serial partitioning methods to build ensemble models consisting of different neural network strucures. Results indicated that the ensemble netowork could consistently outperform a single network design. Walczak [15] in his study, examined the effects of different sizes of training sample sets on forecasting currency exchange rates. It was shown that those neural networks—given an appropriate amount of historical knowledge—can forecast future currency exchange rates with 60 percent directional accuracy, while those neural networks trained on a larger training set had a worse forecasting performance. In addition to higher-quality forecasts, the reduced training set sizes reduced development cost and time. Hu *et al.* [16] applied a sequential learning neural network, named as Minimal Resouce Allocating Network (MRAN) to forecast monthly exchange rates between US dollar and various other currencies and found the neural network's performance to be better both in terms of forecast and direction accuracy. Also recently, Yu *et al.* [17] proposed a nonlinear ensemble forecasting model integrating generalized linear autoregression (GLAR) with ANN and obtained accurate prediction results and forecasting performances. The proposed model's performance was compared with the individual forecasting methods, as well as the hyrid model and linear combination models and the empirical results showed that the prediction results using the nonlinear ensemble model were better than those obtained using the other models.

Recently, more hybrid forecasting models have been developed that integrate neural network techniues with many conventional forecasting methods such as econometric models and time series models to improve prediction accuracy. Weeding II and Cios [18] constructed a model combining Radial Basis Function (RBF) networks, certainity factors, and Box-Jenkins model. Their experimental results have shown that the combination approach improves the overall reilability of time series forecasting. They discussed three different methods in which the two forecasts can be combined into one hybrid forecast. Similarly, Zhang [19] proposed a hybrid methodology that combined Autoregressive Integrated Moving Average (ARIMA) and ANN models taking advantage of the unique strengths of ARIMA and ANN models in linear and nonlinear modeling. Their experimental results with real data sets indicated that the combined model could be an effective way to improve forecasting accuracy achieved by either of the models used seperately. Chen and Leung [20] proposed an adaptive forecasting approach that comined the strengths of neural networks and multivariate ecoometric models. Their hyrid approach contained 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. 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.

Also, Ince and Trafalis [21] proposed a two-stage forecasting model that incorporated both parametric techniques such as ARIMA and non-parametric techniques such as Support Vector Regression (SVR) and ANN. Their findings showed that the input selection is very important and the SVR technique outperformed the ANN for the input selection methods considered. Lee and Wong [22] investigated the predictive performance of a hybrid multivariate model, using multiple macroeconomic and microstructure of foreign exchange market variables. Conceptually, the proposed system combined and exploited the merit of adaptive learning artificial neural network (ANN) and intuitive reasoning (fuzzy-logic inference) tools. An ANN was employed to forecast a foreign exchange rate movement that was followed by the intuitive reasoning of multi-period foreign currency returns using multi-value fuzzy logic for foreign currency risk management decision-making. Empirical tests with statistical and machine learning criteria revealed plausible performance of its predictive capability.

III. Overview of the Intelligent techniques

The following techniques are applied to predict foreign exchange rates of US dollar in terms of the German Mark, the British Pound, and the Japanese Yen: (i) back-propagation neural network (BPNN), (ii) Wavelet Neural Network (WNN), (iii) Dynamic Evolving Neuro-Fuzzy Inference System (DENFIS), (iv) Multivariate Adaptive Regression Splines (MARS), (v) Support Vector Regression (SVR), (vi) Group Method of Data Handling (GMDH) and (vii) Genetic Programming (GP). As BPNN is very popular, it is not discussed here. All the remaining constituents of the ensembles are described briefly in the subsequent subsections.

A. Wavelet Neural Network

The word *wavelet* is due to Morlet and Grossmann [23] in the early 1980s. Wavelets are a class of functions used to localize a given function in both space and scaling. A family of wavelets can be constructed from a function $\psi(x)$, sometimes known as a "mother wavelet," which is confined in a finite interval. "Daughter wavelets" $\psi^{a,b}(x)$ are then formed by translation (b) and dilation (a). Wavelets are especially useful for compressing image data, since a wavelet transform is in some ways superior to a conventional Fourier transform. An individual wavelet can be defined by [24]:

$$\psi^{a,b}(x) = |\alpha|^{-1/2}\, \psi\left(\frac{x-b}{a}\right)$$

Recently, due to the similarity between the discrete inverse wavelet transform and a one-hidden-layer neural network, the idea of combining both wavelets and neural networks has been proposed. This has resulted in the Wavelet neural network (WNN), a feedforward neural network with one hidden layer of nodes, whose basis functions are drawn from a family of orthonormal wavelets. WNN solves the conventional problem of poor convergence or even divergence encountered in other kinds of neural networks. It can dramatically increase convergence speed [25].

Wavelets, in addition to forming an orthogonal basis, are capable of explicitly representing the behavior of a function at various resolutions of input variables. Consequently, a wavelet network is first trained to learn the mapping at the coarsest resolution level. In subsequent stages, the network is trained to incorporate elements of the mapping at higher and higher resolutions. Wavelet networks employ activation functions that are dilated and translated versions of a single function $\psi : R^d \to R$, where d is the input dimension [26] [27]. This function called the 'mother wavelet' is localized both in the space and frequency domains [24]. The WNN consists of three layers: input layer, hidden layer and output layer. All the units in each layer are fully connected to the nodes in the next layer. The output layer contains a single unit. The WNN implemented here makes use of the Gaussian function as the wavelet activation function. WNN was implemented in ANSI C using Visual Studio 6.0 in Windows environment on a Pentium 4 machine with 256 MB RAM.

$$f(t) = \exp(-t^2)$$

B. Dynamic Evolving Neuro-Fuzzy Inference System (DENFIS)

DENFIS was introduced by Kasabov [28]. DENFIS evolve through incremental, hybrid (supervised/unsupervised), learning, and accommodate new input data, including new features, new classes, etc., through local element tuning. New fuzzy rules are created and updated during the operation of the system. At each time moment, the output of DENFIS is calculated through a fuzzy inference system based on -most activated fuzzy rules, which are dynamically chosen from a fuzzy rule set. A set of fuzzy rules can be inserted into DENFIS before or during its learning process. Fuzzy rules can also be extracted during or after the learning process. DENFIS available in the student version of the NeuCom tool obtained from

(http://www.aut.ac.nz/research/research_institutes/kedri/resea rch_centres/centre_for_data_mining_and_decision_support_systems/neucom.htm) was used in this paper.

C. Multivariate Adaptive Regression Splines (MARS)

Multivariate Adaptive Regression Splines (MARS) was introduced by Friedman [29]. MARS is an innovative and flexible modeling tool that automates the building of accurate predictive models for continuous and binary dependent variables. It excels at finding optimal variable transformations and interactions, the complex data structure that often hides in high-dimensional data. In doing so, MARS effectively uncovers important data patterns and relationships that are difficult for other methods to reveal. MARS available at (http://salford-systems.com/) was used in the paper.

D. Support Vector Regression (SVR)

The SVR is a powerful learning algorithm based on recent advances in statistical learning theory. SVR is a learning system that uses a hypothesis space of linear functions in a high dimensional space, trained with a learning algorithm from optimization theory that implements a learning bias derived from statistical learning theory [30]. SVR has recently become one of the popular tools for machine learning and data mining and can perform both classification and regression. SVR uses a linear model to implement nonlinear class boundaries by mapping input vectors nonlinearly into a high-dimensional feature space using kernels. The training examples that are closest to the maximum margin hyper-plane are called support vectors. All other training examples are irrelevant for defining the binary class boundaries. The support vectors are then used to construct an optimal linear separating hyper-plane (in case of pattern recognition) or a linear regression function (in case of regression) in this feature space. The support vectors are conventionally determined by solving a quadratic programming (QP) problem. SVR has the following advantages:

(i) It is able to generalize well even if trained with a small number of examples.

(ii) It does not assume prior knowledge of the probability distribution of the underlying data set.

SVR is simple enough to be analyzed mathematically. In fact, SVR may serve as a sound alternative combining the advantages of conventional statistical methods that are more theory-driven and easy to analyze and machine learning methods that are more data-driven, distribution-free and robust. Recently, SVR has been used in financial applications such as credit rating, time series prediction and insurance claim fraud detection.

E. Group method of data handling (GMDH)

This is a family of inductive algorithms for mathematical modeling of multi-parametric datasets that features fully-automatic structural and parametric optimization of models. GMDH can find relations in data to select optimal structure of model or network or to increase the accuracy of existing algorithms. This self-organizing approach is different from commonly used deductive modeling. It is inductive as the best solution is found by sorting-out of possible variants and the algorithm itself finds the structure of the model and the laws of the system (http://en.wikipedia.org/wiki/GMDH).

GMDH algorithms inductively sort out gradually complicated polynomial models and select the best solution by means of the *external criterion*. A GMDH model with multiple inputs and one output is a subset of components of the *base function*

$$Y(x_1,...,x_n) = a_0 + \sum_{i=1}^{m} a_i f_i$$

where f are elementary functions dependent on different sets of inputs, a are coefficients and m is the number of the base function components. GMDH algorithm considers various component subsets of the base function called *partial models* and the coefficients of these models are estimated by the least squares method. The number of partial model components is gradually increased to find a model structure with optimal complexity indicated by the minimum value of an *external criterion*. This process is called self-organization of models. GMDH is also known as polynomial neural networks and statistical learning networks due to implementation of the corresponding algorithms in several commercial software products (http://en.wikipedia.org/wiki/GMDH).

F. Genetic Programming

Genetic programming (GP) is a biologically inspired evolutionary algorithm to find computer programs that perform a given task. It is like genetic algorithms (GA) but here each individual is a computer program. It optimizes a population of computer programs according to a fitness landscape based on a program's ability to perform a given computational task. Being computationally intensive in the 1990s GP was mainly used to solve relatively simple problems. But thanks to improvements in GP algorithms and to the exponential growth in CPU power, GP has become more prevalent and has produced many novel and outstanding results in areas such as quantum computing, electronic design, game playing, sorting, searching etc. (http://en.wikipedia.org/wiki/Genetic_programming)

GP evolves computer programs traditionally represented in memory as tree structures which can be easily evaluated recursively (http://en.wikipedia.org/wiki/Genetic_programming). Every tree node has an operator function and every terminal node has an operand, making mathematical expressions easy to evolve and evaluate. The main operators used in GP are crossover and mutation. Crossover is applied on an individual by switching one of its nodes with another node from another individual in the population. With tree-based representation, replacing a node means replacing the whole branch which gives greater effectiveness to the crossover operator. The *children* expressions resulting from crossover are very much different from their initial parents. Mutation affects an individual in the population. It can replace a whole node in the selected individual, or it can replace just the node's information. The imulations were run with Discipulus obtained from http://www.rmltech.com/.

IV. Intelligent Nonlinear Ensembles

The idea behind ensemble systems is to exploit each constituent model's unique features to capture different patterns that exist in the dataset. Both theoretical and empirical works indicate that ensembling can be an effective and efficient way to improve accuracies. Bates and Granger [31] in their seminal work showed that a linear combination of different techniques would give a smaller error variance than any of the individual techniques working in stand-alone mode. Since then, many researchers worked on ensembling or combined forecasts. Makridakis et al. [4] reported that combining several single models has become common practice in improving forecasting accuracy. Then, Pelikan *et al*. [32] proposed combining several feed-forward neural networks to improve time series forecasting accuracy. Some of the ensemble techniques for prediction problems with continuous dependent variable include linear ensemble (e.g., simple average (Benediktsson *et al.* [33]), weighted average (Perrone and Cooper [34]) and stacked regression (Breiman [35] and nonlinear ensemble (e.g., neural-network-based nonlinear ensemble (Yu et al., [17]).

Hansen *et al.* [36] reported that the generalization ability of a neural network system could be significantly improved by using an ensemble of a number of neural networks. The purpose is to achieve improved overall accuracy on the production data. In general, for classification problems, an ensemble system combines individual classification decisions in some way, typically by a majority voting to classify new examples. The basic idea is to train a set of models (experts) and allow them to vote. In majority voting scheme, all the individual models are given equal importance. Another way of combining the models is via weighted voting, wherein the individual models are treated as unequally important. This is achieved by attaching some weights to the predictions given by the individual models and then combining them. Olmeda and Fernandez [37] presented a genetic algorithm based ensemble system, where a GA determines the optimal combination of the individual models so that the accuracy is maximized. Zhou *et al.* [38] carried out a detailed study on ensembling neural networks and proposed that using a set of neural networks to form an ensemble is better than to use all the neural networks. They proposed an approach that can be used to select the neural networks to become part of the ensemble from the available set of neural networks. Genetic algorithm was used to assign weights to the constituent networks.

It is generally the case that for a given dataset one kind of an intelligent technique outperforms the other and the results can be entirely opposite when a different dataset is used. In order not to lose any generality and also to combine the advantages of the intelligent techniques, an ensemble uses the outputs of all the stand-alone intelligent techniques with each being assigned a certain priority level and provides the output with the help of an arbitrator.

An ensemble uses the output obtained from the individual constituents as inputs to it and the data is processed according to the design of the arbitrator. Six different variants of ensembles are designed and employed as shown in Figure1. These include (a) Non-linear ensemble based on BPNN, (b) Non-linear ensemble based on WNN, (c) Non-linear ensemble based on DENFIS, (d) Non-linear ensemble based on MARS, (e) Non-linear ensemble based on GMDH and (f) Non-linear ensemble based on GP.

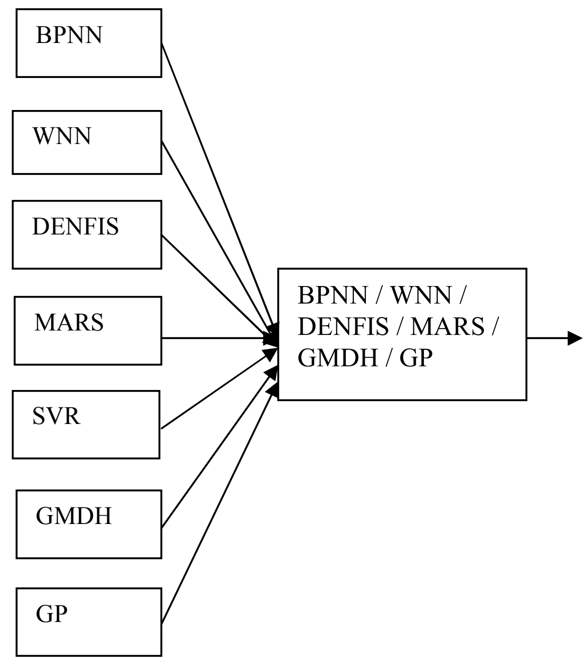

Figure 1. Generic Design of the Ensemble

V. Experimental Design

The foreign exchange data used in our study are obtained from Pacific Exchange Rate Service (http://fx.sauder.ubc.ca/) provided by Prof. W. Antweiler, University of British Columbia, Vancouver, Canada. They consist of monthly US dollar exchange rates with respect to three major currencies - DEM, GBP and JPY. The monthly data from January 1971 to December 2000 (360 observations) is used as the training sample in training the different intelligent techniques that are applied and the monthly data from January 2001 to December 2003 (36 observations) is used as the test sample in comparing the performance of the different intelligent techniques.

Since foreign exchange rate forecasting has only one dependent variable and no explanatory variables in the strict sense and since we have a time-series, we followed the general time series forecasting model in conducting our experiments, which is represented in the following form:

$$X_t = f(X')$$

where X' is vector of lagged variables $\{ x_{t-1}, x_{t-2}, ..., x_{t-p} \}$. Hence the key to finding the solution to the forecasting problems is to approximate the function '*f*'. This can be done by iteratively adjusting the weights in the modeling process.

In their pioneering study of weak-form efficiency in markets, Cornell and Dietrich [39] were the first to use lagged values of the same time-series to predict future currency price movements. An illustration of how training patterns can be designed in the neural network modeling process is provided in Figure 2 (Xu et al. [40]). In this figure, $'p'$ denotes the

	X	Y			
x_1	x_1	x_2	…	x_p	x_{p+1}
x_2	x_2	x_3	…	x_{p+1}	x_{p+2}
x_3	x_3	x_4	…	x_{p+2}	x_{p+3}
.	.	.	…	.	.
.	.	.	…	.	.
.	.	.	…	.	.
x_{t-p}	x_{t-p}	x_{t-p+1}	…	x_{t-1}	x_t

number of lagged variables and $(t-p)$ denotes the total number of training samples. In this representation, $'X'$ is a set of $(t-p)$ vectors of dimension $'p'$ and $'Y'$ is a vector of (t-p) dimensions. Thus, in the transformed data set, $'X'$ and $'Y'$ represent the vector of explanatory variables and dependent variable respectively.

SPSS 14.0 (obtained from http://www.spss.com) was used to find the optimal lag for the given time-series data. We performed the tests of 'auto correlation function' and 'partial auto correlation function' as prescribed by Box-Jenkins methodology in Time series forecasting using SPSS 14.0 software on the data set and found that lag 1 was sufficient DEM and JPY while GBP required lag 2. However, we wanted to investigate whether NRMSE values would improve further when we go for higher lags and we tested from lags 5 to 7 as prescribed by Yu et al. [17]. In view of the foregoing discussion on generating lagged data sets out of the original time series such as this, we created four datasets corresponding to each exchange rate - lag # 1, 5, 6 and 7 respectively for DEM and JPY, lag # 2, 5, 6 and 7 for GBP.

Since it is a time-series data, performing 10-fold cross validation does not make sense, as it involves randomly choosing samples into the folds and then the time aspect of the data gets obscured and overlooked. 10-fold cross validation is extremely powerful and useful in assessing the performance of a model, provided we do not deal with time series or spatial series data. Hence, we carried out hold-out method of testing viz., splitting the data set into 360 training samples and 36 testing samples respectively. In fact, this check is included in many popular commercial data mining / statistical tools. The training data is used to identify the optimal parameters for the model that satisfy the given error criteria and those parameters are the used to forecast values on the test set. The value of Normalized Mean Square Error (NMSE) is used as the measurement criteria.

$$\text{NRMSE} = \sqrt{\frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \bar{y}_i\right)^2}}$$

where n is the number of forecasting observations; y_i is the actual value at period i; $\hat{y}_i$ is the forecasted value of software reliability at period i and $\bar{y}_i$ is the mean.

Clearly, accuracy is one of the most important criteria for forecasting models, but for the business practitioners, the aim of forecasting is to support or improve decisions so as to make money. In exchange rate forecasting, improved decisions often depend on correct forecasting directions between the actual and predicted values, in testing set with respect to directional change of exchange rate movement (expressed in percentages). The ability to forecast movement direction can be measured by Directional change statistics (D_{stat}) developed by Yao and Tan [41] expressed as:

$$D_{stat} = \frac{1}{N}\sum_{i=1}^{N} a_i * 100\%$$

where $a_i = 1$ if $(y_{i+1} - y_i)(\hat{y}_{i+1} - y_i) \geq 0$, and $a_i = 0$ otherwise, y_i is the actual value at period i; $\hat{y}_i$ is the forecasted value of software reliability at period i.

VI. Results and Discussions

For each technique the appropriate parameters, as specified by the algorithm, are tweaked to obtain optimal results. Figures 4-9 depict graphical representations of the forecasting performance achieved through various methods for exchange rates of US dollar with DEM, GBP and JPY using different models over different lags. In Tables 1-6 the results for all the methods and all the different lags are presented. In the tables, second to fifth columns show results for different lags for various methods. The figures in bold in each column denotes the best performance among all the methods for that particular lag. In the sixth column the best performances among all the lags for the corresponding method is presented.

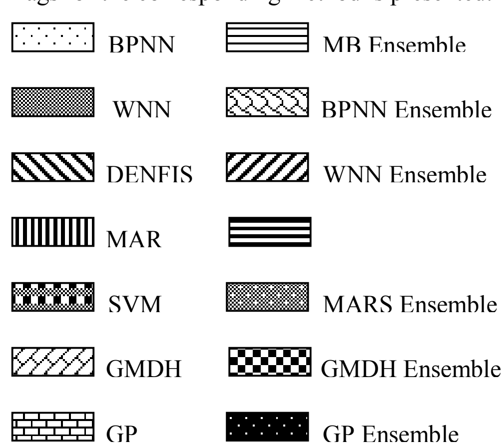

Figure 3. Legend for all the Graphs

Mean Dstat = 70.35112

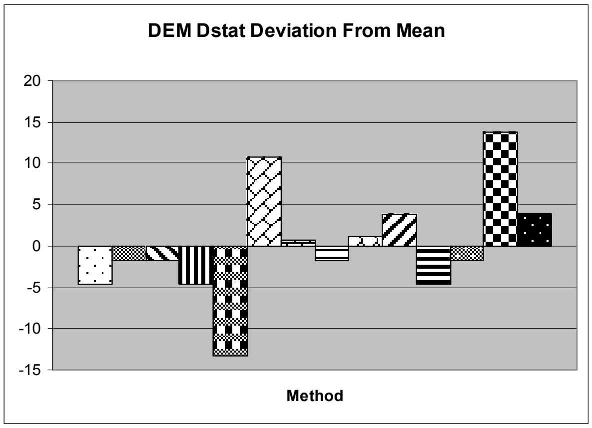

Figure 4. Deviation of D_{stat} values of various methods for DEM from Mean Dstat

Mean NRMSE = 0.20490

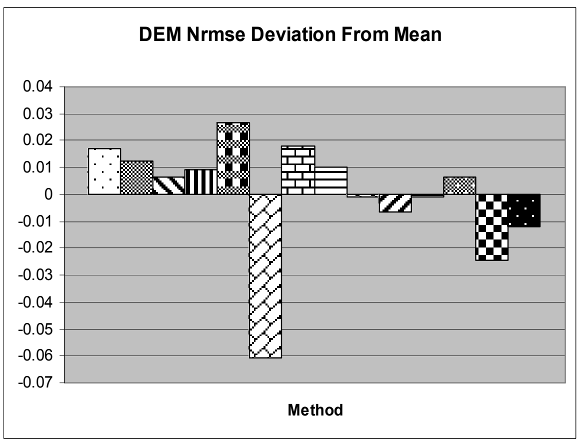

Figure 5. Deviation of NRMSE values of various methods for DEM from Mean Nrmse

Mean D_{stat} = 64.14608

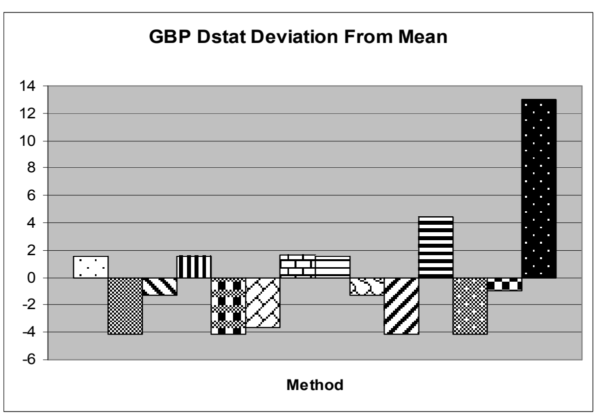

Figure 6. Deviation of D_{stat} values of various methods for GBP from Mean Dstat

Mean NRMSE = 0.27153

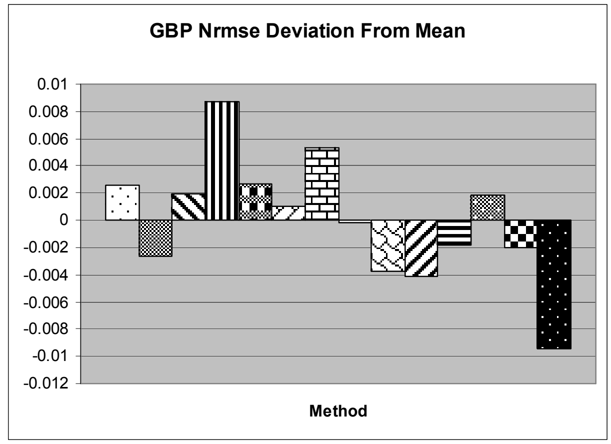

Figure 7. Deviation of NRMSE values of various methods for GBP from Mean Nrmse

Mean D_{stat} = 59.65628

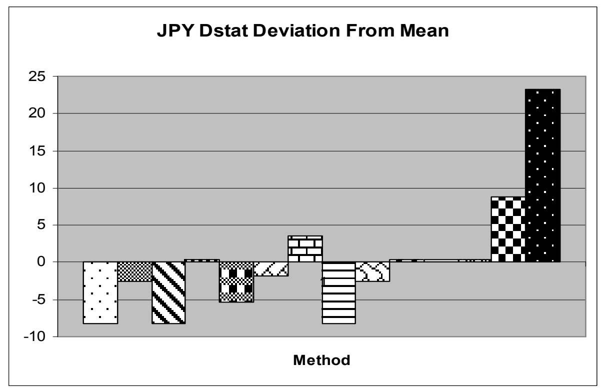

Figure 8. Deviation of D_{stat} values of various methods for JPY from Mean Dstat

Mean NRMSE = 0.44869

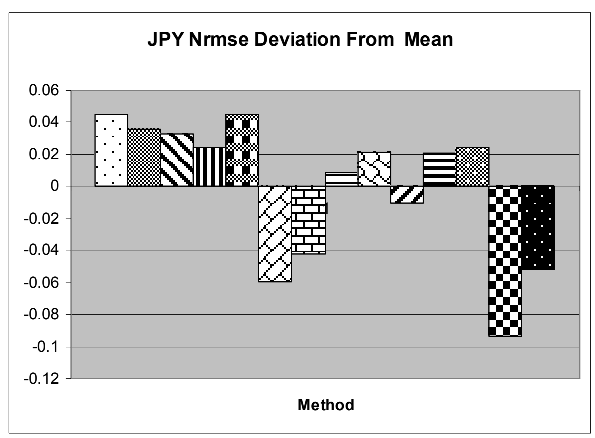

Figure 9. Deviation of NRMSE values of various methods for JPY from Mean Nrmse

Tables 1-3 show the forecasting performance of different techniques in terms of the D_{stat} values over the three currencies – DEM, GBP and JPY respectively over different lags. Also, Tables 4-6 show the forecasting performance of the techniques in terms of the NRMSE values. Results from ARIMA models have also been added for each currency for the sake of comparison. Interesting observations can be drawn from the Tables 1-3. Firstly, there seems to be a correlation between the lag number and the corresponding NRMSE value. In general it can be observed that the NRMSE values decrease with the increase in the lag number. However this is not true for BPNN system for which going to higher lags worsens both the D_{stat} and NRMSE values. This property is in line with Time Series Recency effect propounded by Walczak [15] for Backpropagation networks, where adding extra lags as input worsens the network performance. But importantly the Time Series Recency effect was not observed for other methods. Secondly, the ensemble-based techniques clearly outperformed their stand-alone techniques in terms of NRMSE and D_{stat}.

Table 1. A comparison of D_{stat} values between different techniques for DEM over different lags.

Method	**lag 1**	**lag 5**	**lag 6**	**lag 7**	**Best**
ARIMA (0,1,0)					*42.8571*
BPNN	65.7142	65.7142	62.8571	65.7142	*65.7142*
WNN	65.7142	68.5714	57.1428	68.5714	*68.5714*
DENFIS	60	65.7142	71.4285	68.5714	*71.4285*
MARS	42.8571	62.8571	65.7142	65.7142	*65.7142*
SVM	42.8571	34.2857	54.2857	57.1428	*57.1428*
GMDH	69.2307	71.4285	**77.1428**	81.0810	*81.0810*
GP	68.4210	68.5714	71.4285	71.0526	*71.4285*
MB-ensemble	65.7142	57.1428	65.7142	68.5714	*68.5714*
BPNN-ensemble	65.7142	68.5714	65.7142	71.4285	*71.4285*
WNN-ensemble	65.7142	**74.2857**	74.2857	74.2857	*74.2857*
DENFIS-ensemble	42.8571	68.5714	71.4285	65.7142	*71.4285*
MARS-ensemble	34.2857	65.7142	68.5714	68.5714	*68.5714*
GMDH-ensemble	64.1025	**74.2857**	77.1428	**84.2105**	***84.2105***
GP-ensemble	**74.2857**	68.5714	74.2857	68.5714	*74.2857*

Table 2. A comparison of NRMSE values between different techniques for DEM over different lags.

Method	**lag 1**	**lag 5**	**lag 6**	**lag 7**	**Best**
ARIMA (0,1,0)					*1.1608*
BPNN	0.2218	0.2316	0.2614	0.2360	*0.2218*
WNN	0.2228	0.2224	0.2187	0.2174	*0.2174*
DENFIS	0.2350	0.2152	0.2112	0.2114	*0.2112*
MARS	0.2370	0.2196	0.2151	0.2139	*0.2139*
SVM	0.2380	0.4023	0.2305	0.2314	*0.2305*
GMDH	0.2255	**0.1823**	**0.1740**	**0.1439**	***0.1439***
GP	0.2295	0.2147	0.2027	0.2229	*0.2027*
MB-ensemble	0.2277	0.2369	0.2160	0.2147	*0.2147*
BPNN-ensemble	0.2211	0.2065	0.2045	0.2040	*0.2040*
WNN-ensemble	**0.2185**	0.1964	0.2061	0.1983	*0.1964*
DENFIS-ensemble	0.2463	0.2151	0.2079	0.2040	*0.2040*
MARS-ensemble	0.2375	0.2143	0.2127	0.2111	*0.2111*
GMDH-ensemble	0.2265	0.2022	0.1935	0.1805	*0.1805*
GP-ensemble	0.2238	0.2098	0.1927	0.2075	*0.1927*

Table 3. A comparison of D_{stat} values between different techniques for GBP over different lags.

Method	**lag 2**	**lag 5**	**lag 6**	**lag 7**	**Best**
ARIMA (0,1,1)					*42.8571*
BPNN	65.7142	42.8571	54.2857	54.2857	*65.7142*
WNN	60	62.8571	60	60	*62.8571*
DENFIS	60	68.5714	65.7142	62.8571	*68.5714*
MARS	65.7142	60	60	60	*65.7142*
SVM	60	60	45.7142	54.2857	*60*
GMDH	53.8461	65.7142	65.7142	60.5263	*65.7142*
GP	63.1578	**68.5714**	71.4285	**65.7894**	*71.4285*
MB -ensemble	57.1428	65.7142	60	60	*65.7142*
BPNN-ensemble	60	65.7142	62.8571	62.8571	*65.7142*
WNN-ensemble	57.1428	68.5714	62.8571	60	*68.5714*
DENFIS-ensemble	68.5714	60	57.1428	62.8571	*68.5714*
MARS-ensemble	57.1428	68.5714	62.8571	60	*68.5714*
GMDH-ensemble	60.5263	60	57.1428	63.1578	*63.1578*
GP-ensemble	**74.2857**	**68.5714**	**77.1428**	60	***77.1428***

Table 4. A comparison of NRMSE values between different techniques for GBP over different lags.

Method	lag 2	lag 5	lag 6	lag 7	Best
ARIMA (0,1,1)					*1.06543*
BPNN	0.27408	0.29492	0.30061	0.27559	*0.27408*
WNN	0.26883	0.27070	0.27025	0.27746	*0.26883*
DENFIS	0.27342	0.27540	0.27285	0.27621	*0.27285*
MARS	0.2802	0.28912	0.28564	0.28565	*0.2802*
SVM	0.27419	0.27319	0.28565	0.27501	*0.27319*
GMDH	0.31186	0.25968	0.25804	0.27254	*0.25804*
GP	0.27687	**0.25823**	**0.24286**	0.29346	***0.24286***
MB-ensemble	0.27174	0.27195	0.27132	0.27222	*0.27132*
BPNN-ensemble	0.26891	0.26997	0.26808	0.26779	*0.26779*
WNN-ensemble	0.26886	0.27021	0.26876	**0.26738**	*0.26738*
DENFIS-ensemble	0.26971	0.27830	0.27052	0.27993	*0.26971*
MARS-ensemble	0.27333	0.27548	0.27445	0.28565	*0.27333*
GMDH-ensemble	0.27278	0.26293	0.26063	0.26953	*0.26063*
GP-ensemble	**0.26438**	0.26846	0.26211	0.27502	*0.26211*

Table 5. A comparison of D_{stat} values between different techniques for JPY over different lags.

Method	lag 1	lag 5	lag 6	lag 7	Best
ARIMA (0,1,1)					*45.7142*
BPNN	51.4285	37.1428	42.8571	42.8571	*51.4285*
WNN	57.1428	45.7142	60	48.5714	*60*
DENFIS	51.4285	48.5714	54.2857	42.8571	*54.2857*
MARS	51.4285	51.4285	62.8571	60	*62.8571*
SVM	54.2857	51.4285	45.7142	45.7142	*54.2857*
GMDH	56.4102	54.2857	65.7142	57.8947	*65.7142*
GP	52.6315	62.8571	62.8571	63.1578	*63.1578*
MB – ensemble	45.7142	42.8571	51.4285	51.4285	*51.4285*
BPNN-ensemble	45.7142	40	51.4285	57.1428	*57.1428*
WNN-ensemble	48.5713	54.2857	60	60	*60*
DENFIS-ensemble	42.8571	45.7142	54.2857	60	*60*
MARS-ensemble	51.4285	51.4285	54.2857	60	*60*
GMDH-ensemble	58.9743	65.7142	65.7142	68.4210	*68.4210*
GP-ensemble	**71.4285**	**74.2857**	**71.4285**	**82.8571**	***82.8571***

Table 6. A comparison of NRMSE values between different techniques for JPY over different lags.

Method	lag 1	lag 5	lag 6	lag 7	Best
ARIMA (0,1,1)					*1.6505*
BPNN	0.4933	0.4975	0.4821	0.5176	*0.4821*
WNN	0.4871	0.4841	0.4651	0.4842	*0.4651*
DENFIS	0.4953	0.4736	0.4752	0.4813	*0.4736*
MARS	0.4929	0.4722	0.4733	0.4732	*0.4722*
SVM	0.4936	0.4916	0.4969	0.5016	*0.4916*
GMDH	0.4130	0.4569	0.4426	0.3888	*0.3888*
GP	0.4400	0.4403	**0.4062**	0.4062	*0.4062*
MB-ensemble	0.4882	0.4707	0.4574	0.4590	*0.4574*
BPNN-ensemble	0.4994	0.4681	0.4640	0.4701	*0.4640*
WNN-ensemble	0.4830	0.4596	0.4460	0.4384	*0.4384*
DENFIS-ensemble	0.4960	0.4695	0.4752	0.4692	*0.4692*
MARS-ensemble	0.4929	0.4742	0.4751	0.4732	*0.4732*
GMDH-ensemble	**0.3974**	0.4459	0.4351	**0.3553**	***0.3553***
GP-ensemble	0.47193 3	**0.3968**	0.4100	0.4466	*0.3968*

The best performance of all the networks over all lags is depicted in the form of a bar chart in Fig. 4-9. Fig. 3 depicts the common legend followed in Fig. 4-9. For the sake of clarity and better interpretability we have plotted the deviations of various methods from the mean performance in all figures. This way it is clearer which methods are yielding above average results and which ones are lagging behind. While interpreting the figures it must be borne in mind that for D_{stat} charts higher towers would mean a higher directional accuracy and hence better performance. However, for NRMSE charts lower values would mean better network performance. We can see from the charts that for most cases the GMDH and GP based ensembles performed better than the other ensembles. Fourthly, WNN beats BPNN in most of the cases. Fifth, we see that the sophisticated non-linear ensembles consistently outperformed the simple mean based ensembles (MB-ensemble). Finally, the most important observation is that the GP and GMDH based ensemble outperformed all other techniques in most of the cases except DEM in terms of both D_{stat} and NRMSE.

From the figures we can see that except for a few cases where stand-alone methods perform very well, in most of the cases the ensembling methods give better results. This can be inferred from the observation that most of the figures (Fig. 4, 6, 8) depicting D_{stat} show dips in the first half (corresponding to stand-alone method) and taller bars in the second half (corresponding to ensemble methods). Expectedly, the opposite holds true for NRMSE figures (Fig 5, 7, 9). Based on the D_{stat} measure, we can comprehensively conclude that ensembling is yielding better results than stand-alone techniques. We also observe that ensembling is more time consuming than using intelligent methods in their stand-alone mode because, in general, an ensemble can be constructed only after the results of the constituents are available.

However, it is observed that the gains accrued in the form of improved accuracy more than offset the time lost in ensembling. Further, we point out that, when exchange rate prediction is to be made accurately in an offline manner, then time is no constraint and nonlinear ensemble should be preferred. However, when time is a constraint on-line methods like DENFIS should be preferred, as they need only one-pass or one-iteration to give predictions.

It should be noted that Yu *et al* [17] used the same data sets used in this paper while designing ANN based ensemble with constituents as (i) generalized linear auto regression model (GLAR) (ii) artificial neural network (ANN) (iii) GLAR-ANN hybrid, where the time series is modeled by GLAR and the errors are modeled by ANN (iv) Mean based GLAR-ANN hybrid and (v) Weighted Mean based GLAR-ANN hybrid. Even though Yu *et al.* [17] reported excellent results, our results cannot be compared with theirs because they did not specify which lag was used in the experimental design. Further it was not made clear in the aforementioned paper what data pre-processing steps were followed. It was precisely because of this reason we could not reproduce their results. Further, we claim that our method of ensembling is simpler, more diverse and superior because of the varied types of intelligent techniques used as constituents.

VII. Conclusions

Six nonlinear ensemble architectures are developed to forecast foreign exchange rates in the computational intelligence paradigm. BPNN, WNN, MARS, SVR and DENFIS, GMDH and GP are chosen as the members of the ensembles. The data of exchange rates of US dollar (USD) with respect to Deutsche Mark (DEM), Japanese Yen (JPY) and British Pound (GBP) is used for testing and comparing the performance of the ensembles. Lagged variables are considered throughout the study in order to account for the auto regressive nature of the time series problem. Six different ensembles based on BPNN, WNN, MARS, DENFIS, GMDH and GP were developed. All the techniques are compared with normalized root mean squared error (NRMSE) and directional statistics (D_{stat}) as the performance criteria.

The results indicate that, in terms of both, D_{stat} and NRMSE, GP and GMDH based ensemble consistently outperformed other models over all the currencies. Out of these two although GMDH produced outstanding results for JPY, GP came out to be more consistent of the two when we consider the results of all the three currencies and both the performance measures. We note that though not all ensembling methods are as consistent as GMDH and GP, but based on D_{stat} values, ensembles consistently outperform stand-alone techniques for all the three currencies. This gain in performance has to be weighed against additional computational complexity in making the ensemble. Based on the results, it is inferred that ensembling in the computational intelligence paradigm is a sound alternative to the extent techniques to forecast foreign exchange rates.

References

[1] Triennial Central Bank Survey (December 2007), Bank for International Settlements.

[2] Annual FX poll (May 2008), Euromoney.

[3] CWJ Granger, T. Terasvirta. *Modelling nonlinear economic relationships*, Oxford: Oxford University Press, 1993.

[4] S. Makridakis, A. Anderson, R. Carbone, R. Fildes, M. Hibdon, R. Lewandowski, J. Newton, E. Parzen, R. Winkler. "The accuracy of extrapolation (time series) methods: results of a forecasting competition", *Journal of Forecasting,* pp. 111-153, 1982.

[5] G. Peter Zhang, B. Eddy Patuwo, Michael Y. Hu. "A simulation study of artificial neural networks for nonlinear time-series forecasting", *Computers & Operations Research,* 28, pp.381-396,2001.

[6] A. K. Palit, D. Popovic. *Computational Intelligence in Time Series Forecasting: Theory and Engineering Applications*, Springer-Verlag, 1995.

[7] G. De Matos. "Neural networks for forecasting exchange rate. M. Sc. Thesis", The University of Manitoba, Canada, 1994.

[8] C. M. Kuan, T. Liu. "Forecasting exchange rates using feed-forward and recurrent neural networks", *Journal of Applied Econometrics,* 10, pp. 347–364, 1995.

[9] W. Hsu, L. S. Hsu, M.F. Tenorio. "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. pp. 245–257.

[10] P. Tenti. "Forecasting foreign exchange rates using recurrent neural networks", *Applied Artificial Intelligence*, 10, pp.567–581, 1996.

[11] A. Muhammed, G. A. King. "Foreign exchange market forecasting using evolutionary fuzzy networks", IEEE/*IAFE Conference on Computational Intelligence for Financial Engineering*, pp.213-217, 1996.

[12] M. R. El Shazly, H. E. El Shazly. "Forecasting currency prices using genetically evolved neural network architecture", *International Review of Financial Analysis*, 8, pp.67–82,1999.

[13] M. T. Leung, A. S. Chen, H. Daouk. "Forecasting exchange rates using general regression neural networks", *Computers & Operations Research,* 27, pp.1093–1110, 2000.

[14] G. P. Zhang, V. L. Berardi. "Time series forecasting with neural network ensembles: an application for exchange rate prediction", *Journal of the Operational Research Society,* 52, pp.652–664, 2001.

[15] S. Walczak. "An Empirical Analysis of Data Requirements for Financial Forecasting with Neural Networks", *Journal of Management Information Systems,* 14 (4), pp.203-222, 2001.

[16] M. Hu, P. Saratchandran, S. Narasimhan. "A sequential learning neural network for foreign exchange rate

forecasting", *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 4, pp.3963-3968, 2003.

[17] L. Yu, S. Wang, K. K. Lai. "A novel nonlinear ensemble forecasting model incorporating GLAR and ANN for foreign exchange rates", *Computers & Operational Research*, 32, pp.2523-2541, 2005.

[18] D. K. Weeding II, K. J. Cios. "Time series forecasting by combining RBF networks, certainty factors, and the Box-Jenkins model", *Neurocomputing*, 10, pp.149–168,1996.

[19] G. P. Zhang. "Time series forecasting using a hybrid ARIMA and neural network model", *Neurocomputing*, 50, pp159–175, 2003.

[20] A. S. Chen, M. T. Leung. "Regression neural network for error correction in foreign exchange forecasting and trading", *Computers and Operations Research*, 31(7), pp.1049-1068, 2004.

[21] H. Ince, T. B. Trafalis. "A hybrid model for exchange rate prediction", *Decision Support Systems,* 42, pp.1054-1062, 2006.

[22] V. C. S. Lee, H. T. Wong. "A multivariate neuro-fuzzy system for foreign currency risk management decision making", *Neurocomputing,* 70, pp.941-951, 2007.

[23] A. Grossmann, J. Morlet. "Decomposition of Hardy Functions into Square Integrable Wavelets of Constant Shape", *Siam. J. Math. Anal*, 15(4), pp.723-726, 1984.

[24] V. M. Becccera, R. K. H. Galvao, M. Abou-Seada. "Neural and Wavelet Network Models for Financial Distress Classification", *Data Mining and Knowledge Discovery,* 11, pp.35-55, 2005.

[25] X. Zhang, J. Qi, R. Zhang, M. Li, Z. Hu, H. Xue, B. Fan. "Prediction of programmed-temperature retention values of naphthas by wavelet neural networks", *Computers and Chemistry*, 25, pp.125-133, 2001.

[26] Q. Zhang, A. Benvniste. "Wavelet Networks", *IEEE Transactions on Neural Networks,* 3(6), pp.889-898, 1992.

[27] Zhang Q. Using Wavelet Network in Nonparametric Estimation. IEEE Transactions on Neural Networks 1997; 8(2): 227-236.

[28] N. K. Kasabov. "DENFIS: Dynamic Evolving Neural-Fuzzy Inference System and Its Application for Time-Series Prediction", *IEEE Transactions on fuzzy systems*, 10(2), pp.144-154, 2002 .

[29] J. H. Friedman. "Stochastic Gradient Boosting", *Computational Statistics and Data Analysis,* 38(4), pp.367-378, 1999.

[30] N. Cristianini, J. Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods*, Cambridge University Press, Cambridge, 2000

[31] J. M. Bates, C. W. J. Granger. "The combination of Forecasts", *Operations research Quarterly*, 20, pp.451-468, 1969.

[32] E. Pelikan, C. De Groot, D. Wurtz. "Power consumption in West-Bohemia: improved forecasts decorrelating connectionist networks", *Neural Network World,* 2, pp.701-712, 1992.

[33] J. A. Benediktsson, J. R. Sveinsso, O. K. Ersoy, P. H. Swain. "Parallel Consensual Neural Networks", *IEEE Transactions on Neural Networks,* 8, pp.54-64, 1997.

[34] M. P. Perrone, L. N. Cooper. "When networks disagree: ensemble methods for hybrid neural netwoks", In: Mammone RJ. (Ed.) *Neural Networks for speech and Image processing*, Chapman Hall, pp.126-142, 1993.

[35] L. Breiman. "Bagging predictors", *Machine Learning,* 24, pp.123-140, 1996.

[36] J. Hansen, J. McDonald, J. Slice. "Artificial intelligence and generalized qualitative response models: An empirical test on two audit decision-making domains", *Decision Science,* 23, pp.708–723, 1992.

[37] I. Olmeda, E. Fernandez. "Hybrid classifiers for financial multi-criteria decision making: the case of bankruptcy prediction", *Computational Economics,* 10, pp.317-335, 1997.

[38] Z. H. Zhou, J. Wu, W. Tang. "Ensembling neural networks: Many could be better than all", *Artificial Intelligence*, 137, pp.239-263, 2002.

[39] W. B. Cornell, J. K. Dietrich. "The Efficiency of the Market for Foreign Exchange under Floating Exchange Rates", *The Review of Economics and Statistics*, 60 (1), pp.111-120, 1978.

[40] K. Xu, M. Xie, L. C. Tang, S. L. Ho. "Application of neural networks in forecasting engine systems reliability", *Applied Soft Computing*, 2, pp.255-268, 2003.

[41] J. T. Yao, C. L. Tan. "A case study on using neural networks to perform technical forecasting of forex", *Neurocomputing*, 34, pp.79-98, 2000.

Author Biographies

Vadlamani Ravi (V. Ravi) is an Associate Professor in Institute for Development and Research in Banking Technology (IDRBT), Hyderabad since February 2010. He received his PhD in Soft Computing from Osmania University, Hyderabad and RWTH Aachen, Germany. Earlier, he was an Assistant Professor at IDRBT from 2005-2010 and Faculty at NUS, Singapore from 2002-2005. Prior to that, he was a Scientist in CSIR research Institute for 14 years. During the last 23 years, he published 106 papers in refereed journals/conferences and invited book chapters. He edited Advances in Banking Technology and Management: Impacts of ICT and CRM published by IGI Global, USA. He is a Referee for several international journals and on the Editorial Board of IJIDS, IJDATS, IJISSS, IJWS and IJITPM. He is listed in the Marquis Who's Who in the World in 2009, 2010 and in Engineering in 2011.

Ramanuj Lal holds an Integrated M.Sc in Physics from Indian Institute of Technology, Kharagpur. His research interests include Computational Finance, Algorithmic Trading and Intelligent Techniques. He works for Deep Value Technology Pvt Ltd, India

Nampally Raj Kiran holds an M.Tech (IT) from University of Hyderabad and IDRBT. His research interests included Software Reliability prediction and Neural Networks. He works as a Software Engineer for Emerson, India.