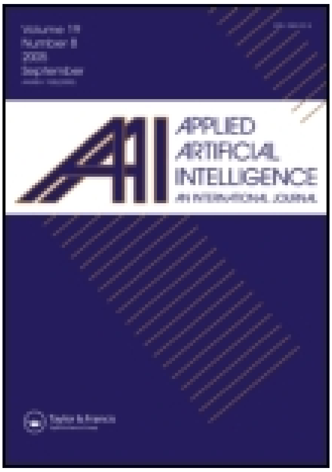

Applied Artificial Intelligence: An International Journal
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/uaai20

Evolving Recurrent Neural Network using Cartesian Genetic Programming to Predict The Trend in Foreign Currency Exchange Rates

Faheem Zafari[a], Gul Muhammad Khan[a], Mehreen Rehman[a] & Sahibzada Ali Mahmud[a]
[a] Centre of Intelligent Systems and Network Research, Electrical Engineering Department, University of Engineering and Technology Peshawar, Peshawar, Pakistan
Published online: 14 Jul 2014.

Applied Artificial Intelligence, 28:597–628, 2014

ISSN: 0883-9514 print/1087-6545 online
DOI: 10.1080/08839514.2014.923174

Taylor & Francis
Taylor & Francis Group

EVOLVING RECURRENT NEURAL NETWORK USING CARTESIAN GENETIC PROGRAMMING TO PREDICT THE TREND IN FOREIGN CURRENCY EXCHANGE RATES

Faheem Zafari, Gul Muhammad Khan, Mehreen Rehman, and Sahibzada Ali Mahmud
Centre of Intelligent Systems and Network Research, Electrical Engineering Department, University of Engineering and Technology Peshawar, Peshawar, Pakistan

□ *Forecasting the foreign exchange rate is an uphill task. Numerous methods have been used over the years to develop an efficient and reliable network for forecasting the foreign exchange rate. This study utilizes recurrent neural networks (RNNs) for forecasting the foreign currency exchange rates. Cartesian genetic programming (CGP) is used for evolving the artificial neural network (ANN) to produce the prediction model. RNNs that are evolved through CGP have shown great promise in time series forecasting. The proposed approach utilizes the trends present in the historical data for its training purpose. Thirteen different currencies along with the trade-weighted index (TWI) and special drawing rights (SDR) is used for the performance analysis of recurrent Cartesian genetic programming-based artificial neural networks (RCGPANN) in comparison with various other prediction models proposed to date. The experimental results show that RCGPANN is not only capable of obtaining an accurate but also a computationally efficient prediction model for the foreign currency exchange rates. The results demonstrated a prediction accuracy of 98.872 percent (using 6 neurons only) for a single-day prediction in advance and, on average, 92% for predicting a 1000 days' exchange rate in advance based on ten days of data history. The results prove RCGPANN to be the ultimate choice for any time series data prediction, and its capabilities can be explored in a range of other fields.*

INTRODUCTION

The last couple of decades have witnessed an exponential increase in the computerization of the world. Forecasting the foreign currency exchange rates is a challenging issue, which is usually done through financial time series. The setup of a time series is very noisy and unstable (Phillip, Tofiki,

Address correspondence to Sahibzada Ali Mahmud, Electrical Engineering Department, University of Engineering and Technology Peshawar, Peshawar 25125, Pakistan. E-mail: sahibzada.mahmud@nwfpuet.edu.pk

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/uaai.

and Bidemi 2011). Although there are several statistical models that are currently used for forecasting the foreign currency exchange rates, they lack flexibility and efficiency to deal with the highly volatile nature of foreign exchange data. Financial time series are usually utilized to tackle the issue of forecasting the foreign currency exchange rate. There is a need for an efficient technique that can be used to deal with the financial time series. Research has indicated that the use of artificial neural networks (ANNs) offer superior performance to other techniques in dealing with time series forecasting (Refenes et al. 1993; Kadilar, Simsek, and Aladag 2009; Tenti 1996). Because of the data-driven and self-adaptive nature of ANNs, they can serve as an alternate for forecasting. They have the capability of learning from the past experience and can obtain the subtle functional relationships that are present in the data. In the presence of adequate observation of a specific problem, ANNs can provide us with the solution, even if the solution is based on information that is difficult to specify (Zhang, Patuwo, and Hu 1998). In this study we have used recurrent Cartesian genetic programming (CGP) to evolve a recurrent neural network (RNN) model, which can be used for forecasting the foreign currency exchange rate. As is clear from the name, the model is recurrent feedback based that is generated using the CGP. Because the forecasting is to be done for a time series, we have used historical data of the currency exchange rates to train the model, so that the model can learn the behavior of the financial time series. Once we trained the network, we then evaluated its performance using various other datasets. The obtained experimental results show that the recurrent-CGP-based artificial neural network (RCGPANN) deals with the unstable financial time data series with extreme efficiency because of its fast learning ability. Experimental results also show that the RCGPANN is superior to all other models utilized up until now in terms of efficiency and the computational cost of implementation.

The rest of this article is organized as follows: "Literature Review" presents a review of related work. "Recurrent Neural Networks" discusses recurrent neural networks, Cartesian genetic programming, and Cartesian genetic programming evolved artificial neural network. "Recurrent Cartesian Genetic Programming evolved Artificial Neural Network" discusses the utilized model. "Experimental Setup" presents the experimental ambience and scenarios in which they were carried out. "Results and Analysis" presents the obtained experimental results. "Conclusion and Future Work" concludes the article and presents future work directions.

LITERATURE REVIEW

There are several models available that have been put forth to deal with time series forecasting. The models differ in their methodology of prediction and can be differentiated from each other on the basis of their ability to predict the foreign exchange rates in the future as well as how easily are they implemented. Despite the promise shown by statistical techniques in dealing with time series forecasting, they are restrained by set of nonlinear data (Kadilar, Simsek, and Aladag 2009: Dablemont et al. 2003). Artificial neural networks have been used successfully for time series forecasting (Tenti 1996) and modeling. They have been used for prediction of financial time series (Bhattacharya, Parlos, and Atiya 2003) along with several other tasks such as prediction of communication network traffic (Bhattacharya, Parlos, and Atiya 2003) and forecasting of river flow (Atiya et al. 1999). A neuroevolutionary technique called Cartesian genetic programming evolved artificial neural network (CGPANN) has also been used in Khan, Khan, and Ullah (2011) to come up with a model that can predict a peak load 24 hours ahead. This network can be used to obtain a significantly unique model for every season because it is trained using annual and quarterly bases. CGPANN is proposed in Khan, Khan, and Miller (2010a) to deal with control problems related to nonlinear Markovian and non-Markovian methods. It is applied to the problem of pole-balancing using both Markovian and non-Markovian cases. The results show that the CGPANN has the tendency to produce a neural architecture along with parameters that will be able to tackle the problems in fewer iterations compared to other neuroevolutionary techniques. The Elman neural network has been used in Marra and Morabito (2005) for forecasting solar activities. An Elman network uses positive feedback in order to build its memory. It does so by adding various recurrent connections. Although the structures are multilayer perceptrons (MLPs), there is a difference present as well. The input layer is made up of the input neurons along with units known as the context units. The context units accumulate the neurons of the hidden layers, which belong to the previous time step. These neurons then serve as inputs of the current time step. The inputs are not supplied with a feedback from the network output. The number of context units is the same as that of the hidden neurons, a result of which the outputs of two Elman networks will be different, even if they have similar weights, biases, and are supplied with the same input at any given time step. The output is different because of the variation in feedback states. A nonparametric method is used in Gradojevic and Yang (2006) to forecast the Canadian and US dollar exchange rate. The results show that ANNs outperform the random walk and linear models in terms of root mean square error (RMSE) and percentage of correctly predicting the variations in the exchange rate. The performance evaluation and comparison of multilayered feed-forward neural network (MLFN) and general regression neural network (GRNN) is presented in Chen and Leung (2005). The ability of the networks to predict the currency exchange correlation is empirically evaluated for both models on the basis of a number of statistical tests such as RMSE, mean absolute error (MAE), and so forth. The experimental results show that the success of neural networks in forecasting relies on the architectural design of the model. The use of market timing tests show that both the models perform efficiently in forecasting the exchange rate correlation. The RNN seems the optimum approach for modeling, and the forecasting accuracy is also the highest for the RNN. Kaashoek and Dijk (2002) have proposed a threefold procedure in order to shrink the network size. This method lacks the numeric intensiveness because the cell contribution relies on the result of just a single optimization procedure while including the variables involved. Because the method is of a descriptive nature, it is beneficial for explanatory analysis of the data. The network, which is obtained through cell pruning, can then be utilized for dynamic analysis and prediction. In Nag and Mitra (2002), a hybrid artificial intelligence method has been used in order to model the diurnal foreign exchange rates. The method is based on the use of ANN along with a genetic algorithm (GA), which provides better performance compared to the fixed-size neural networks. According to Marsh (2000), Markov models can approximate the data properly. As with other linear and nonlinear formulations, Markov models fit the exchange data rate properly. The Japanese yen, British pound sterling and the German deutschemark are used for exchange rates with respect to the US dollar. This model predicts efficiently only within the estimation period used and applying it outside the estimation period will not be satisfactory. The reason behind the ineffective forecasting performance is the instability of the parameters. That's why there is room for further research in order to carry out the necessary incorporations in the model. The accuracy of forecasting by the alternative vector auto regressive models is analyzed in Joseph (2001). The findings conclude that estimation based on the Bayesian method provides superior forecasting compared to the ordinary least square (OLS) method, which becomes even more evident in the case of nonstationary specifications. The predicting ability of the model is weak. Nonparametric neural network regression and RNNs are used in Dunis and Huang (2002) for forecasting the foreign exchange of USD into British pounds and USD into Japanese yen. The RNN seems the optimum approach for modeling, and the forecasting accuracy is also the highest for the RNN. Markov switching models are proposed in Parikakis and Merika (2009) for capturing the volatility dynamics of the exchange rates as well as evaluating their ability to forecast. It is identified that the increase of volatility in four different Euro-based exchange rates is because of the underlying changes in the structure. The findings also show the close relationship between the currencies during periods of high volatility, during which there is a significant increase in cross correlations. The Markov switching Monte Carlo approach proves to be superior to the random walk hypothesis. The use of econometric methodology assists in accurate forecasting of exchange rate movements. The model provides better out-of-sample returns with Euro/US dollar and Euro/British pound; however, it performs poorly when used with Euro/Brazilian real or Euro/Mexican peso. This failure is due to the high volatility in Latin American currencies. In Qian and Rasheed (2010), it is shown that all periods are not equally random. Also, the accuracy was increased up to 67% by collaboration of various models such as the ANN, decision tree, naïve Bayesian classifier, and k-nearest neighbor. The analysis in Skintzi and Sisinis (2007) shows that the generalized autoregressive conditionally heteroskedastic (GARCH) models are better in accounting for the dynamic structure of correlation in the case of bond and stock portfolios, however, the simpler specifications such as the historical mean model deal efficiently with currency portfolios. The predictors in Rivero and Garcia (2005) perform superior to random walk models in terms of prediction error and directional forecasts for forecasting periods of up to five days. The analysis in Farsa and Zolfaghari (2011) is based on the use of the embedding theorem in combination with artificial intelligence and residual analysis. The results show that such a combination can provide better forecasting results for a chaotic time series.

Cartesian Genetic Programming (CGP)

Miller introduced the idea of CGP in 1999. It is the type of genetic programming in which a computer program or a digital code is generated by a two-dimensional graphical representation. It is a highly flexible and efficient technique for genetic programming, which started by evolving an electronic circuit in 1997 (Vasicek and Sakanina 2010; Rothermich and Miller 2002). CGP moves a step further by using arrays and a Cartesian framework, whereas the genetic program relies on the automatic evolution of digital or computer structures (Khan, Khan, and Miller 2010b). Directed acyclic graph formats, which work in the feed forward direction, are used to represent the program in CGP. The grid of programmable nodes are used to represent the two-dimensional graphs of CGP (Rothermich and Miller 2002).

A combination of a fixed number of arrays of integers, which represent the network in the form of functions, inputs, outputs, and their interconnectivity, constitutes the genotype. The network connection that exists between the nodes is limited by the level-back parameter. Nodes can be either active (i.e., when they are major participants in the network producing the output) or they can be the junk nodes that remain inactive. The product of the total rows and columns in the network will make the total number of nodes. Activation functions such as step, tangent-hyperbolic, logical AND, logical OR, and sigmoid can be assigned to the nodes. Weights are assigned to the connections between the nodes. This two-dimensional form of computational nodes, which constitute the network architecture, is termed Cartesian genetic programming (Rothermich and Miller 2002; Miller and Harding 2008). The CGP is an efficient and reliable programming method that can be used to evolve an ANN. The nodes can be implicitly reused because they allowed connection to previous node outputs present in the graph (Gomez and Schmidhuber 2008). CGP is also superior to tree-based programming representations because it can efficiently reuse the noncoding genes and represent various numbers of outputs. The general form of CGP is shown in Figure 1.

Recurrent Neural Networks

RNNs are special neural networks because of their dynamic behavior. They are different from the feed-forward network because of the presence of at least one feedback path. The feedback can be for a layer or a single neuron wherein the output is feedback as the input. The feedback profoundly impacts the network's learning ability. The feedback paths also use branches with unit delay elements, which cause a nonlinear dynamical behavior due to the nonlinear neuron nature (Toha and Tokhi 2008). This nonlinear dynamic plays an important role in the storage function that a recurrent network has (Hayken 1999). Recurrent networks have the tendency to be sensitive and to adapt to the past inputs. The RNNs with the feedback connections and internal dynamic elements are superior to feed-forward networks when it comes to the modeling and control of the nonlinear systems (Linkens and Nyongesa 1996).

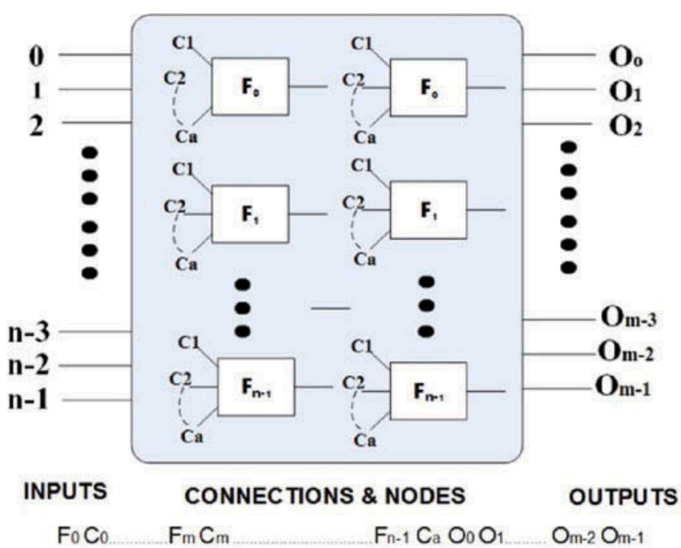

FIGURE 1 General form of Cartesian genetic programming.

For prediction and classification, the RNNs are known for efficiently utilizing the temporal information present in the applied inputs. Once they are trained, the output can then be produced by processing the interrelation between the current inputs and the internal states. The learning process is supervised, during which the target value is used as the second information source. The target values also highlight the relevant interrelations present in the input sequence. The input to the RNNs is in the form of a time series whereas the target can be a trivial sequence based on constant value, or it can be a nontrivial time series. In the case of classification, a constant class label is used as the output, whereas for prediction, another time series constitutes the output (Husken and Stagge 2003). Recently, RNNs with feedback have also been proposed to be used for temporal processing. An RNN known as the prediction recurrent artificial neural network's (PRANN) performance is twofold better than the time-delay (TD) networks. It is an MLP network that has recurrent connections along with a complete backpropagation learning rule. The backpropagation learning rule updates the network's weights. It is based on the recurrent network that was proposed by William and Zipser (1989). It can process time series because it has certain incorporating features. The network utilizes arbitrary dynamics whereas the backpropagation algorithm is used for reducing the output error. The network topology of PRANN has added a linear output node along with a linear hidden node to the already existing RNN proposed by William and Zipser (1989). It can be seen that the weights that connect the nonlinear hidden layer nodes with the output node are not updated when the William and Zipser network is recast into a similar topology such as Madhavan (2011). This is the reason that the PRANN is viewed as the recurrent ANN, which is fit for processing the time series while updating the weights. PRANN outperforms the traditional TD neural networks when used for both linear and nonlinear time series (Madhavan 2011). The RNNs have proved to be better than the feed forward networks because they require less training time and the size of the network is smaller. The results for single-step ahead and multiple-steps ahead forecasting have been better for the RNNs (Kumar, Raju, and Sathish 2004). The dynamic recurrent neural network (DRNN; Aussem, Murtagh, and Sarazin 1995) is another RNN that is obtained by modeling the synapses in the form of autoregressive filters. The DRNNs use a system of nonlinear difference equations of internal variables for approximating the laws governing the time series, which is the reason behind the DRNNs providing history-sensitive forecasts without even using any external memory. The temporal recurrent backpropagation model is used for training the DRNN model. The DRNN has proved to outperform the fuzzy k-nearest neighbors. The RNNs, due to the feedback mechanism, have proved to outperform a number of other models.

The RNNs in Cai et al. (2004) are trained with a training algorithm that relies on the hybrid of evolutionary algorithm (EA) and particle swarm optimization (PSO). This combination of the searching ability of the EA and PSO removes the restriction on the individual's evolution. Indeed, the individual with better performance is able to produce offspring in order to replace the poor-performing individuals. The experimental results showed that the RNNs that are trained using the hybrid algorithm have the capability of predicting the missing values in time series keeping the error at a lesser value compared to the networks that are trained using PSO or EA. Using the boosting algorithm can also improve the results when applied to the RNNs in forecasting the values of a time series as shown by the experiments in Boné, Assaad, and Crucianu (2003). The use of weighted median is also superior to weighted mean when used for combining the learners.

Using the RNNs in combination with symbolic processing can assist in dealing with the problem of high noise and forecasting of nonstationary time series (Giles, Lawrence, and Tsoi 2001). The use of symbolic conversion makes the training of RNNs more effective and the symbolic input assists in extracting the rules from the trained network. The results in Giles, Lawrence, and Tsoi (2001) show that the meaningful symbolic knowledge can be obtained from the noisy time series. It is evident that when a robust learning algorithm in which the outliers are filtered from the data and the parameters are estimated on the basis of the filtered data is applied to an RNN, a robust RNN is produced that has many advantages over the traditional feed forward networks (Connor, Martin, and Atlas 1994). The use of filtered data for training the neural networks provides the network with better forecasting skills compared to the networks that are trained with non-filtered data. Three different recurrent architectures have been analyzed in Tenti (1996). After the analysis a trading plan is also proposed.

Cartesian Genetic Programming Evolved Artificial Neural Network (CGPANN)

The Cartesian genetic programming evolved artificial neural network was proposed by Khan, Khan, and Miller (2010c). It uses CGP representation to define the evolution of ANN. The basic idea behind CGPANN is that the topology, weight, and functions are encoded in one genotype, which is then evolved for the augmented topology, the optimum weights, and functions. The genotype is made up of nodes that correspond to a full ANN. The topological features are random initially, however, some features might be removed, or new features might be added with the passage of time. This happens through the mutation of functions, inputs, weights, connection types, and outputs. At the start, a genotype—known as the parent genotype—is used for representation of the network. The parent genotype is mutated for producing the offspring. The process of producing new offspring continues until an offspring with desired fitness level is attained. The fittest genotype is then used for the desired application. The neuron is the basic functional unit of CGPANN. The number of neurons is dependent on the requirement for the representation of the network. The system output relies on active nodes; the junk neurons do not take part in the output of the network.

Initially, genotype population is generated for network representation. The initial genotype is termed as the parent genotype which produces offspring through mutation; 1+ λ evolutionary strategy is used to produce the offspring, whereas λ shows the number of offspring that are produced. Once the offspring are obtained by the mutation of the parent genotype that is the fittest, then the fittest offspring becomes the parent and is mutated to reproduce new offspring for the next generation. The reproduction of new offspring continues until the desired level of fitness is attained. The application uses the fittest genotype. Figure 3 shows the flow chart, which highlights the generalized approach for CGPANN.

It should be noted that because CGPANN is based on the topology-and-weight-evolving artificial neural network (TWEANN), it tends to be both constructive and destructive (Zhang and Muhlenbein 1993).

RECURRENT CARTESIAN GENETIC PROGRAMMING EVOLVED ARTIFICIAL NEURAL NETWORK (RCGPANN)

The importance of recurrent networks cannot be denied when it comes to dealing with a broader domain of dynamic and nonlinear systems. The RCGPANN presented by Khan, Khan, and Miller 2010a) is the neuroevolutionary algorithm that benefits from the insuperable ability of CGP in the generation of a recurrent artificial neural architecture. It varies from various other classes of CGPANN in that it relies on a feedback mechanism (i.e., the system is fed back with one or many of the outputs). (A neuron structure is

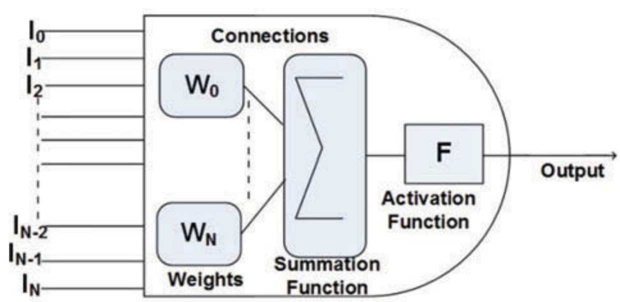

FIGURE 2 Neuron's structure in CGPANN.

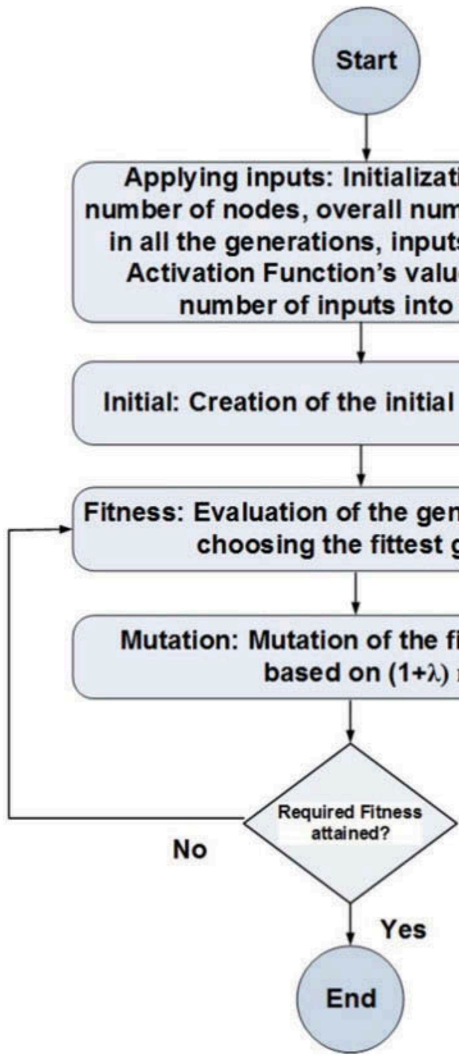

FIGURE 3 Flow chart for CGPANN.

CGPANN is shown in Figure 2.) RCGPANNs are based on the direct encoding method. In direct encoding, the topology functions and weights are encoded in one genotype, which are then evolved in order to obtain the optimum weights, functions, along with an augmented topology. The offsprings are generated by an evolutionary strategy known as the $1+\lambda$ with $\lambda = 9$ evolutionary strategy. As with CGPANN, RCGPANN is also based on TWEANN, hence it is both a destructive and a constructive algorithm. During the evolution of the topological features in RCGPANN, some features are removed and others are added. Mutation is used to evolve the functions, weights, inputs, outputs, and connection types. The connections that are disabled because of mutation are not entirely removed; indeed, there is the probability that they might be re-enabled in generations to follow. The architecture is different compared to the traditional ANN in that the neurons that are derived by the network are not entirely connected, and all the neurons in the input layers are not provided with program inputs. This provides the RCGPANN with the possibility of producing topologies whose hardware implementation and timing is efficient (Refenes et al. 1993). The genotype of RCGPANN is made up of nodes that represent the ANN neurons. The nodes are based on certain inputs, connections, weights, and functions, which can be seen in Figure 4. There are three types of inputs: (1) the program inputs, (2) the inputs that come from the feedback, and (3) previous nodes' inputs. The inputs in the first layer of genotype of the RCGPANN are recurrent and system inputs only; however, presence of recurrent connections for the following layers relies on whether the feedback input is chosen randomly to be used as an node input. A node is considered to be a connected node if the connection has a value of one, and it is considered to be disconnected if its connection is zero. The generation of weights takes place randomly between −1 and +1, however, the weight of the feedback input is always +1. The inputs and weights of all the connected inputs are multiplied and then summed, then they are forwarded to either a linear or a nonlinear function such as a tangent hyperbolic, linear, sigmoid, or step function for the production of outputs at every node. The resultant output can then be used as the input for the next node or system output. The genotype

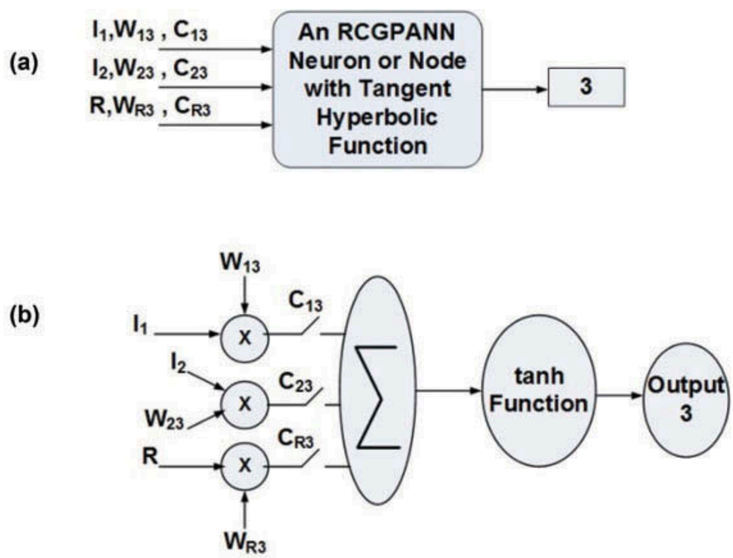

FIGURE 4 (a) 3 inputs into the RCGPANN node; (b) internal view of the RCGPANN node in (a).

output(s) can be any node output(s) or the input(s) of the program. In the case wherein the recurrent input is connected, then the genotype output is used as the feedback to the nodes. The genotype of the RCGPANN is then evolved through mutation generation after generation to achieve the desired fitness. It is worth mentioning that the connections and state unit weights are not mutated. The obtained genotype is then transformed into the artificial neural architecture (Khan, Khan, and Miller 2010a).

Figure 4(a) shows the block representation of a RCGPANN node with 3 inputs (I_1, I_2, R), along with weights (W_{13}, W_{23}, W_{R3}) and the respective connections. C_{13}, C_{23}, C_{R3}. W_{13}, W_{23}, and W_{R3} are the weights associated with the input I_1, I_2, and R, respectively. R is the input that has been fed back from the system output and is known as the recurrent input. The weight W_{R3} is 1 when used for node 3. Initially, R's value is taken to be 0. The internal view of the RCGPANN node can be seen in Figure 4(b). The unconnected three inputs I_1, I_2, and R are, respectively, multiplied with their corresponding weights W_{13}, W_{23}, W_{R3}. Their results are summed and then fed into a sigmoid function, which generates the node 3 output.

Figure 5(a) shows the genotype of a 2 x 2 RCGPANN network, which has 3 inputs (I_1, I_2, R) along with 2 functions (tanh, sigmoid) and an output node "6." The genotype's block diagram is shown in Figure 5(b), and the network's internal view is shown in Figure 5(c). The inputs to node 6 are provided from I_3, I_2, and the feedback value, which is taken as zero for the initial step.

The input I_3 is the node 3 output in which the inputs to the node 3 are I_1, I_2, and system feedback with the tanh function (the system feedback in

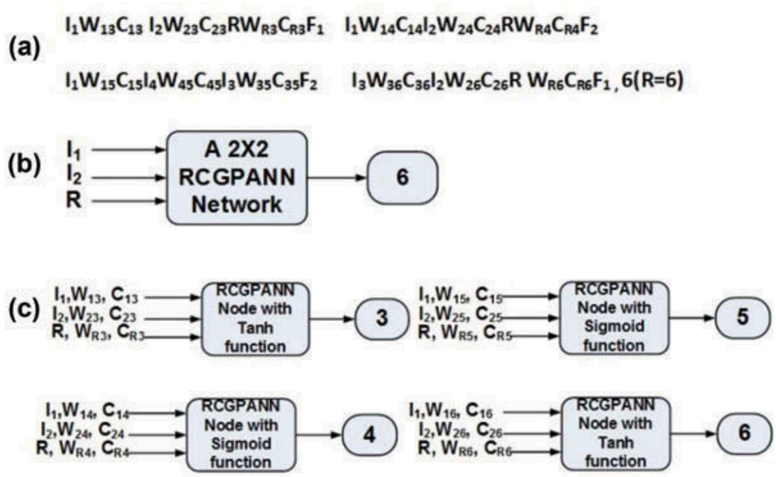

FIGURE 5 (a) A 2 x 2 RCGPANN's genotype; (b) block representation of the genotype in (a); (c) graphical representation of the genotype in (a).

this case is the node 6 output). After computing the node 3 output, the network then processes the result. Once the genotype is obtained, it is then transformed into the neural architecture of Figure 6.

EXPERIMENTAL SETUP

This study uses the historical data of foreign exchange, which is obtained from the Australian Reserve Bank, for training the proposed forecasting model of currency exchange rates. Overall, data of 500 days of the US dollar is used to train ten networks for five independent seeds. At the initial stages of the experiment, a random RCGPANN population was generated. The sigmoid function was used as the activation function. The number of inputs per node was five. A 10% mutation rate (μ_r) was chosen because it results in a better system in less time (Kadilar, Simsek, and Aladag 2009; Huang et al. 2006). We used only a single-row RCGPANN, because the number of generated graphs is infinite. So the number of both nodes and columns was equal. The number of inputs and outputs was fixed at ten for the network. After the mutation of the randomly generated genotype, we obtained a further ten networks on the basis of 1+ λ evolutionary strategy, where λ was nine in this case. The mean absolute percentage error (MAPE) was used to evaluate the fitness of the offspring. The network with the best MAPE parameter was promoted to the next generation. The same network was then utilized for the production of nine more networks by using the mutation process. The

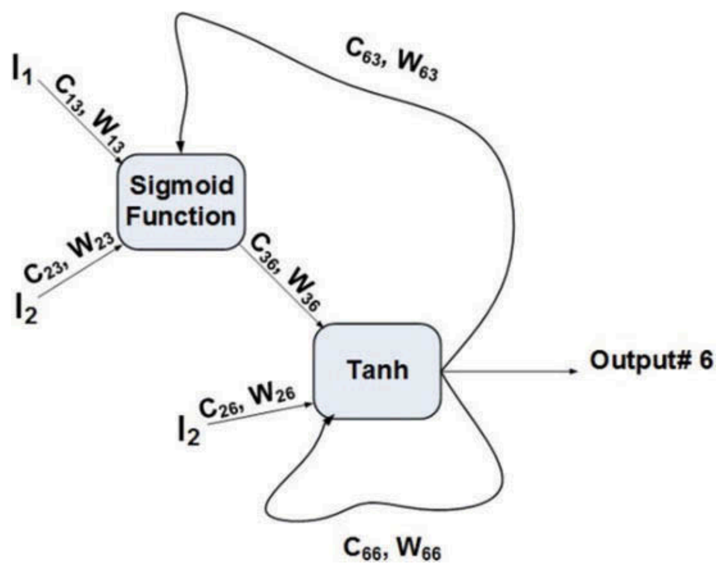

FIGURE 6 Phenotype of the genotype in Figure 5(a).

process kept going until we obtained our desired fitness. During the training phase, we ran the experiment for one million generations.
The mathematical expression for MAPE value is:

$$\mathrm{MAPE} = \frac{1}{N}\sum\nolimits_{i=1}^{N}\frac{\left(L_F^i - L_A^i\right)}{L_A^i} \times 100.$$

Whereas fitness is given by

$$\mathrm{Fitness} = 100 - \mathrm{MAPE}.$$

The forecasted value is represented by L_F; L_A indicates the actual value; and N represents the number of weeks. Internationally, MAPE is used as the standard in performance evaluation of algorithms used for time series prediction whereas fitness or accuracy provides us with the mathematical measure of the system's performance. The proposed method's performance is also presented in terms of the MSE and RMSE as well.

MSE is the methodology used for quantifying the difference of values obtained by an estimator and the actual values of the quantity that is being estimated. Mathematically

$$MSE(X) = \frac{1}{N}\sum\nolimits_{i=1}^{N}\frac{\left(L_F^i - L_A^i\right)^2}{L_A^i},$$

where L_F is the estimated and L_A is the actual value.
The RMSE is the square root of the MSE, that is,

$$RMSE = \sqrt{\frac{1}{N}\sum\nolimits_{i=1}^{N}\frac{\left(L_F^i - L_A^i\right)^2}{L_A^i}}$$

The number of feedbacks was varied. Initially we started with only one feedback, which was then increased to five and finally ten.

Figure 7 presents the structure of the network for various experiments: 7(a) shows the network with only one feedback path, 7(b) shows the network with five feedback paths, and 7(c) shows the network with ten feedback paths. Figure 8 shows the sliding-window mechanism. Initially the data of 10 days is used to forecast the data for day 11 and then the window slides toward day 11 and uses that to forecast for day 12.

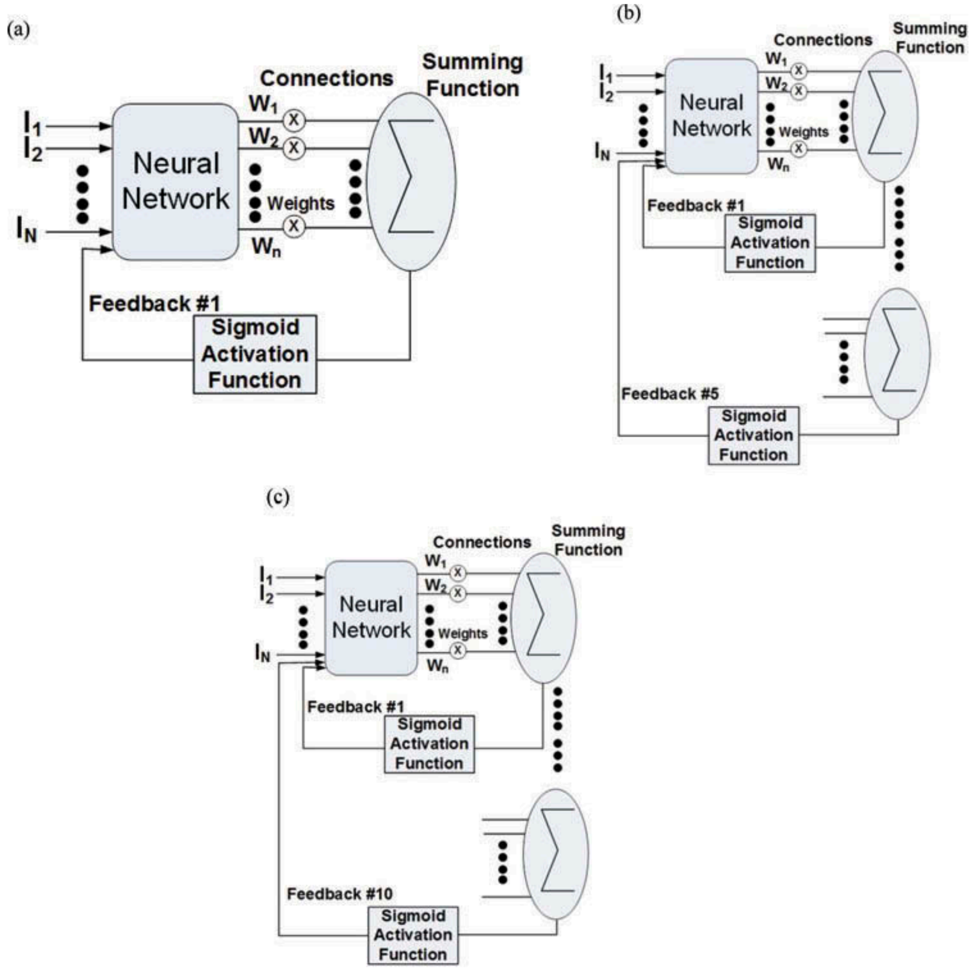

FIGURE 7 (a) Single feedback, (b) five feedbacks, and (c) ten feedbacks.

RESULTS AND ANALYSIS

We tested the forecasting capability of the network on a number of stocks and various sets of data. We also tested the trained network performance by monitoring the historical data spanning 1000 days for a number of foreign currencies. The currencies under observation were Euro, Japanese yen, GB pound, Canadian dollar, New Zealand dollar, Singapore Dollar, Taiwanese dollar, Hong Kong dollar, Chinese yuan, South Korean won, Indonesian rupiah, Malaysian ringgit, and Swiss franc along with trade weight index (TWI) and special drawing rights (SDR). The TWI is the multilateral exchange rate index. It is calculated using the weighted average of exchange rate of home currency against the foreign currency. The weight

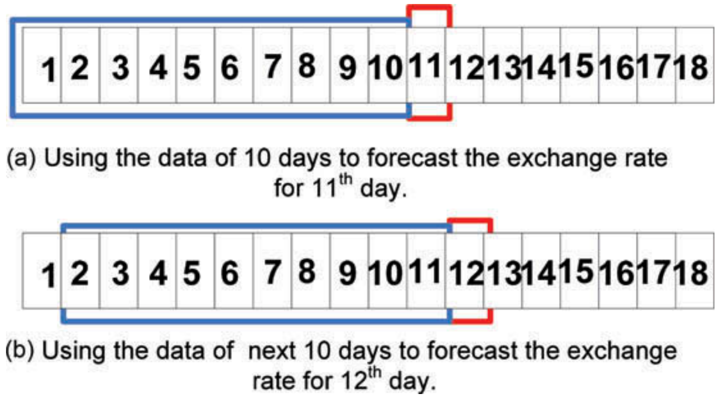

FIGURE 8 The sliding window mechanism.

of the foreign currency depends upon the share of the foreign country's trade with the home country. The SDRs are defined and maintained by the International Monetary Fund. They are based on the additional foreign exchange reserve assets. SDR itself is not a currency, indeed it shows the claim by an IMF member country to a specific currency for which it can be exchanged.

During the test phase, the estimation of data over the testing set with known historical values was executed. Then the network performance was evaluated by comparing the estimated values with the actual values from the data. We present the results for the network trained with single feedback, 5 feedback paths, and 10 feedback paths.

Table 1 provides us the training results for the network. The network was trained for 500 days using USD. It can be seen from the results that increasing the number of feedback paths provides us with higher accuracy with fewer nodes.

The testing results of the network with a single feedback and various numbers of nodes for a number of currencies are presented in terms of accuracy in Tables 2a and 2b for the prediction of the 11th day on the basis of the 10 days of data history for a period of 1000 days. For all the currencies, TWI, and SDR, the most accurate results are with 500 nodes. Because we have used only one feedback, we obtained the most accurate results with 500 nodes. It can be clearly seen that we obtained the most accurate results with 500 nodes for the Indonesian rupiah. Figure 9 shows the pattern for actual and estimated values. Both the waveforms are very close to each other, highlighting the efficiency of RCGPANN in accurate prediction of the values.

The testing results of the network with five feedback paths and various numbers of nodes for a number of currencies are presented in terms of

TABLE 1 Training Results for Various Network Sizes Showing Average Performance in Terms of Accuracy

Training Table			
Nodes	Single Feedback	5 Feedbacks	10 Feedbacks
50	98.34	98.43	98.46
100	98.24	98.45	98.42
150	98.31	98.46	98.44
200	98.37	98.46	**98.46**
250	98.32	**98.46**	98.44
300	98.36	98.46	97.19
350	98.38	98.45	98.46
400	98.37	98.43	98.37
450	**98.45**	98.46	98.45
500	98.43	98.42	98.45

TABLE 2a The Testing Results for Various Currencies in Terms of Accuracy of RCGPANN Predicting the 11th Day on the Basis of 10 Days of Data History with a Single Feedback for a Period of 1000 Days

Nodes	50	100	150	200	250	300	400	450	500
YEN	98.43	98.28	98.39	98.47	98.39	98.65	98.49	98.41	**98.69**
EURO	97.74	97.64	97.72	97.77	97.72	97.80	97.78	97.74	**97.88**
GBP	98.02	97.88	97.98	98.06	97.98	98.19	98.08	97.99	**98.27**
CHF	97.89	97.74	97.85	97.93	97.85	98.18	97.95	97.83	**98.25**
NZD	98.21	98.16	98.20	98.23	98.20	98.19	98.23	98.23	**98.26**
CAD	98.25	98.18	98.23	98.27	98.24	98.28	98.28	98.26	**98.31**
HKD	97.95	97.79	97.91	97.99	97.91	98.39	98.01	97.86	**98.39**
SGD	98.23	98.06	98.18	98.27	98.18	98.55	98.29	98.17	**98.57**
MYR	98.07	97.91	98.03	98.11	98.03	98.44	98.13	98.00	**98.46**
TWD	98.04	97.88	97.99	98.08	97.99	98.32	98.10	97.98	**98.37**
KRW	98.42	98.34	98.40	98.45	98.41	98.52	98.46	98.43	**98.54**
IDR	98.67	98.51	98.63	98.71	98.63	98.85	98.73	98.67	**98.87**
CNY	98.07	97.92	98.03	98.11	98.03	98.40	98.13	98.01	**98.43**

TABLE 2b The Testing Results for TWI and SDR in Terms of Accuracy of RCGPANN Predicting the 11th day on the Basis of 10 Days of Data History with a Single Feedback for a Period of 1000 Days

Nodes	50	100	150	200	250	300	400	450	500
TWI	97.98	97.81	97.93	98.03	97.93	98.44	98.05	97.89	**98.46**
SDR	97.36	97.23	97.32	97.39	97.32	97.74	97.41	97.29	**97.71**

accuracy in Tables 3a and 3b for the prediction of 11th day on the basis of the ten days of data history over a period of 1000 days. EURO and GBP were predicted with highest accuracy using 450 nodes. HKD, TWD, CHF, SGD, MYR, TWI, SDR, CNY, and KRW are predicted with highest accuracy using 50 nodes, whereas 500 nodes presented us with accurate results for IDR, NZD, and CAD. It is evident that we obtained the most accurate

FIGURE 9 Graph showing the proximity between the estimated values and actual values for IDR currency with 98.86% accuracy of a network having 500 nodes and a single-feedback scenario over a period of 1000 days.

TABLE 3a The Testing Results for Various Currencies in Terms of Accuracy of RCGPANN Predicting the 11th Day on the Basis of 10 Days of Data History with a 5-Feedback Scenario for a Period of 1000 Days

Nodes	50	100	150	200	250	300	400	450	500
YEN	**98.70**	98.69	98.69	98.69	98.69	98.69	98.69	98.70	98.69
EURO	97.88	97.88	97.88	97.88	97.88	97.88	97.88	**97.88**	97.88
GBP	98.27	98.27	98.27	98.27	98.27	98.27	98.27	**98.27**	98.27
CHF	**98.26**	98.25	98.25	98.25	98.25	98.25	98.25	98.26	98.24
NZD	98.24	98.26	98.26	98.26	98.26	98.26	98.26	98.25	**98.27**
CAD	98.30	98.31	98.31	98.31	98.31	98.31	98.31	98.31	**98.32**
HKD	**98.43**	98.40	98.41	98.41	98.40	98.40	98.40	98.41	98.39
SGD	**98.58**	98.57	98.57	98.57	98.57	98.57	98.57	98.58	98.56
MYR	**98.48**	98.46	98.47	98.47	98.46	98.46	98.46	98.47	98.46
TWD	**98.38**	98.37	98.37	98.37	98.37	98.37	98.37	98.38	98.36
KRW	**98.54**	98.54	98.54	98.54	98.54	98.54	98.54	98.54	98.54
IDR	98.86	98.87	98.87	98.87	98.87	98.87	98.87	98.86	**98.87**
CNY	**98.45**	98.43	98.44	98.44	98.44	98.43	98.43	98.44	98.43

TABLE 3b The Testing Results for TWI and SDR in Terms of Accuracy of RCGPANN Predicting the 11th Day on the Basis of 10 Days of Data History with a 5-Feedback Scenario for a Period of 1000 Days.

Nodes	50	100	150	200	250	300	400	450	500
TWI	**98.49**	98.47	98.47	98.48	98.47	98.46	98.47	98.48	98.46
SDR	**97.74**	97.72	97.72	97.72	97.72	97.71	97.72	97.73	97.71

result with 500 nodes for the Indonesian rupiah. Analyzing Tables 4a and 4b thoroughly and comparing them with Tables 3a and 3b, it is evident that with the increase in the number of feedback paths we are generally getting higher accuracy with fewer nodes. Figure 10 shows the pattern for actual

TABLE 4a The Testing Results for Various Currencies in Terms of Accuracy of RCGPANN Predicting the 11th Day on the Basis of 10 Days of Data History with a 10-Feedback Scenario for a Period of 1000 Days.

Nodes	50	100	150	200	250	300	400	450	500
YEN	98.69	**98.70**	98.69	98.69	98.69	97.54	98.68	98.64	98.69
EURO	**97.88**	97.88	97.88	97.88	97.88	96.90	97.88	97.79	97.88
GBP	98.27	**98.27**	98.26	98.27	98.26	96.77	98.26	98.20	98.27
CHF	98.25	**98.25**	98.24	98.25	98.24	96.12	98.24	98.17	98.25
NZD	98.26	98.24	98.26	**98.26**	98.25	97.13	98.26	98.17	98.26
CAD	98.31	98.30	98.31	98.31	98.31	97.28	**98.31**	98.27	98.31
HKD	98.40	**98.41**	98.40	98.40	98.39	95.60	98.38	98.35	98.39
SGD	98.57	**98.58**	98.57	98.57	98.57	96.64	98.56	98.52	98.57
MYR	98.46	**98.47**	98.46	98.46	98.45	96.09	98.45	98.41	98.46
TWD	**98.37**	98.38	98.37	98.37	98.37	96.52	98.36	98.31	98.37
KRW	98.54	98.54	98.54	**98.54**	98.54	97.46	98.54	98.50	98.54
IDR	98.87	98.86	98.87	98.87	98.86	97.47	**98.87**	98.84	98.87
CNY	98.43	98.44	98.43	98.43	98.43	96.23	98.42	98.38	98.43

TABLE 4b The Testing Results for TWI and SDR in Terms of Accuracy of RCGPANN Predicting the 11th Day on the Basis of 10 Days of Data History with a 10-Feedback Scenario for a Period of 1000 Days.

Nodes	50	100	150	200	250	300	400	450	500
TWI	98.46	**98.48**	98.46	98.46	98.46	95.78	98.45	98.41	98.46
SDR	97.72	**97.74**	97.72	97.71	97.72	95.17	97.70	97.73	97.72

and estimated values. Both the waveforms are close to each other, highlighting the efficiency of RCGPANN in terms of accurate prediction of the values.

FIGURE 10 Graph showing the proximity between the estimated values and actual values for IDR currency with 98.871% accuracy of a network having 500 nodes and a five-feedback scenario over a period of 1000 days.

The testing results of the network with 10 feedback paths and various numbers of nodes for a range of currencies are presented in terms of accuracy in Tables 4a and 4b for the prediction of 11th day on the basis of the 10 days of data history over a span of 1000 days. It is evident that we obtain the most accurate result with a 400-nodes scenario for the Indonesian rupiah. Analyzing Tables 4a and 4b thoroughly and comparing them with Tables 2a, 2b, 3a and 3b, one can conclude that using ten feedback paths produces more accurate results with fewer nodes. Figure 11 shows the pattern for actual and estimated values. Both the waveforms lie close to each other, highlighting the efficiency of RCGPANN in terms of accurate prediction of the values.

Tables 5a and b present a comparison of RCGPANN with various other networks in terms of accuracy in Table 5a and the MAPE in Table 5b for forecasting the foreign currency exchange rates. RCGPANN is superior to all other models and proves to be better in terms of forecasting accuracy distribution rates. The use of the feedback in RCGPANN results in much higher accuracy along with lower computational cost.

FURTHER ANALYSIS

In order to test the capability of the neural network in terms of prediction for more days in advance, we have evaluated it for predicting 10, 50, 100, 300, 500, and 1000 days in advance based on 10 days of data history. We have selected the best network in all the three scenarios and evaluated for 10, 50, 100, 300, 500, and 1000 days prediction in advance.

FIGURE 11 Graph showing the proximity between the estimated values and actual values for IDR currency with 98.872% accuracy of a network having 400 nodes and 10-feedback scenario for a span of 1000 days.

TABLE 5a Comparison of RCGPANN with Other Contemporary Networks in Terms of Accuracy

Network	Accuracy (%)
Backpropagation Network (Miller et al. 2008)	62.27
Multineural Network (Miller et al. 2008)	66.82
Hybrid (ANN, Decision Tree, Naïve Bayesian Classifier, and K-Nearest Neighbor)	67
Hidden Markov foreign exchange rate forecasting model (HFERFM) (Zhang, Patuwo, and Hu 1998)	69.9
MLP (Taremian 2010)	72
Volterra Network (Taremian 2010)	76
Artificial neural network foreign exchange rate forecasting model (AFERFM) (Zhang, Patuwo, and Hu 1998)	81.2
RCGPANN (Proposed)	98.872

TABLE 5b Comparison of RCGPANN with Other Contemporary Networks in Terms of MAPE

Network	MAPE (%)
Markov model (Refenes et al. 1993)	1.92
Autoregressive integrated moving average (ARIMA) (Phillip, Tofiki, and Bidemi 2011)	1.61
Regression model (Refenes et al. 1993)	1.90
Classification and regression tress (CART) model	1.62
NN model	1.61
RCGPANN (Proposed)	1.12

Table 6a shows the MAPE, MSE, and RMSE results for the proposed model with a single-feedback scenario in predicting 13 different currencies over 1000 days. Table 6b provides the MAPE, MSE, and RMSE results for the proposed network with a single-feedback scenario in forecasting the TWI and SDR.

Tables 6a and 6b indicate that the model with a greater number of nodes in the network tends to perform better in terms of accuracy or MAPE. With increase in the number of nodes, the MAPE value decreases, hence the prediction becomes more accurate. In most of the cases, the network with 450-node and 500-node scenarios produce an optimum network that performs much better. The observations in Table 6 are for predictions spanning 1000 days in advance based on only ten days of data history. The RCGPANN network is highly efficient for prediction, having a shorter time horizon; however, with the increase in time horizon for prediction, the accuracy tends to decrease.

Figure 12 shows the comparison between the actual values and established values of the Taiwanese dollar by using the proposed model of 500 nodes with a single feedback for prediction of 1000 days in advance. From the figure, it is evident that the network follows the trend in currency change the best.

TABLE 6a The MAPE, MSE and RMSE Values for Various Network Sizes with a Single Feedback for Predicting 1000 Days of Data in Advance

Nodes		50	100	150	200	250	300	350	400	450	500
YEN	**MAPE**	45.87	46.63	45.93	44.62	45.62	46.03	44.44	45.11	**12.29**	13.70
	MSE	0.25	0.25	0.25	0.23	0.24	0.25	0.23	0.24	0.02	0.03
	RMSE	0.50	0.50	0.50	0.48	0.49	0.50	0.48	0.49	0.16	0.17
EURO	**MAPE**	47.11	47.86	47.17	45.87	46.86	40.01	45.71	46.35	**14.20**	15.03
	MSE	0.26	0.27	0.26	0.25	0.26	0.20	0.25	0.26	0.03	0.03
	RMSE	0.51	0.52	0.51	0.50	0.51	0.45	0.50	0.51	0.17	0.18
GBP	**MAPE**	56.09	56.85	56.16	54.84	55.84	49.35	54.66	55.31	13.40	**12.42**
	MSE	0.34	0.35	0.34	0.33	0.34	0.27	0.33	0.33	0.03	0.02
	RMSE	0.59	0.59	0.59	0.57	0.58	0.52	0.57	0.58	0.16	0.15
CHF	**MAPE**	57.74	58.49	57.80	56.49	57.49	52.47	56.32	56.98	12.52	**11.56**
	MSE	0.38	0.38	0.38	0.36	0.37	0.31	0.36	0.37	0.02	0.02
	RMSE	0.61	0.62	0.61	0.60	0.61	0.56	0.60	0.61	0.15	0.14
NZD	**MAPE**	28.51	29.25	28.57	27.32	28.27	**20.20**	27.16	27.76	29.45	30.80
	MSE	0.13	0.13	0.13	0.12	0.12	0.07	0.12	0.12	0.13	0.14
	RMSE	0.35	0.36	0.36	0.34	0.35	0.27	0.34	0.35	0.36	0.37
CAD	**MAPE**	31.55	32.48	31.66	30.15	31.34	**16.97**	29.80	29.05	28.94	29.77
	MSE	0.14	0.15	0.14	0.14	0.14	0.04	0.13	0.13	0.12	0.12
	RMSE	0.38	0.39	0.38	0.37	0.38	0.21	0.37	0.37	0.34	0.35
HKD	**MAPE**	60.94	61.71	61.01	59.69	60.70	60.29	59.52	60.19	10.86	**10.10**
	MSE	0.42	0.43	0.42	0.40	0.42	0.41	0.40	0.41	0.02	0.01
	RMSE	0.65	0.65	0.65	0.63	0.64	0.64	0.63	0.64	0.13	0.12
SGD	**MAPE**	57.06	57.82	57.13	55.81	56.81	56.60	55.64	56.30	10.47	**9.51**
	MSE	0.36	0.37	0.36	0.34	0.36	0.35	0.34	0.35	0.02	0.01
	RMSE	0.60	0.61	0.60	0.59	0.60	0.59	0.59	0.59	0.12	0.11
MYR	**MAPE**	58.65	59.41	58.71	57.40	58.40	58.02	57.22	57.89	9.49	**8.95**
	MSE	0.39	0.40	0.39	0.37	0.38	0.38	0.37	0.38	0.01	0.01
	RMSE	0.62	0.63	0.62	0.61	0.62	0.62	0.61	0.61	0.12	0.11
TWD	**MAPE**	57.07	57.83	57.13	55.82	56.82	57.17	55.64	56.31	8.59	**7.96**
	MSE	0.36	0.37	0.36	0.34	0.36	0.36	0.34	0.35	0.01	0.01
	RMSE	0.60	0.61	0.60	0.59	0.60	0.60	0.59	0.59	0.11	0.10
KRW	**MAPE**	35.33	36.08	35.39	34.09	35.08	36.97	33.93	34.58	**25.27**	25.59
	MSE	0.17	0.18	0.17	0.16	0.17	0.18	0.16	0.16	0.09	0.10
	RMSE	0.41	0.42	0.41	0.40	0.41	0.43	0.40	0.41	0.31	0.31
IDR	**MAPE**	42.35	43.11	42.42	41.13	42.11	45.32	40.96	41.62	**13.52**	15.70
	MSE	0.23	0.23	0.23	0.22	0.22	0.25	0.21	0.22	0.02	0.03
	RMSE	0.48	0.48	0.48	0.46	0.47	0.50	0.46	0.47	0.16	0.18
CNY	**MAPE**	57.92	58.68	57.98	56.67	57.67	57.29	56.50	57.16	9.24	**8.79**
	MSE	0.38	0.39	0.38	0.36	0.37	0.37	0.36	0.37	0.01	0.01
	RMSE	0.61	0.62	0.61	0.60	0.61	0.61	0.60	0.61	0.12	0.11

Table 7a presents the MAPE values for a network of 500 nodes with a single-feedback scenario and currencies for predicting 10, 50, 100, 300, 500, and 1000 days in advance.

Table 7b presents the MAPE values for a network of 500 nodes and TWI and SDR, for predicting 10, 50, 100, 300, 500, and 1000 days in advance. It is evident that the model is highly accurate initially for 10 days, but as the number of days increase; generally the MAPE also increases, which causes a decrease in the accuracy of the model.

TABLE 6b The MAPE, MSE and RMSE Values for Various Network Sizes with a Single Feedback for Predicting 1000 Days of Data in Advance

Nodes		50	100	150	200	250	300	350	400	450	500
TWI	**MAPE**	61.56	62.32	61.62	60.30	61.31	58.18	60.13	60.79	10.73	**9.31**
	MSE	0.42	0.43	0.42	0.40	0.41	0.38	0.40	0.41	0.02	0.01
	RMSE	0.65	0.65	0.65	0.63	0.64	0.61	0.63	0.64	0.12	0.11
SDR	**MAPE**	62.77	63.53	62.84	61.52	62.52	58.54	61.34	62.00	12.45	**10.97**
	MSE	0.44	0.45	0.44	0.42	0.43	0.38	0.42	0.43	0.02	0.02
	RMSE	0.66	0.67	0.66	0.65	0.66	0.62	0.65	0.65	0.14	0.13

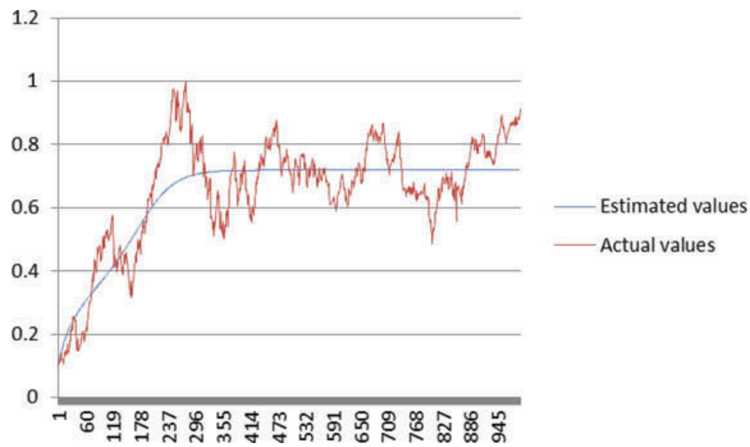

FIGURE 12 The plot for TWD comparison between the real values and the experimental results after 1000 days with 500 nodes and single feedback path.

The network with 5 feedback paths can be seen in Figure 7(b). There are five different feedback paths, which go from the output back into the system as the inputs. Table 8a shows the MAPE, MSE, and RMSE values for various network sizes and 13 different currencies with a 5-feedback scenario evaluated to predict 1000 days' data in advance. Table 8b shows the MAPE, MSE, and RMSE values corresponding to SDR and TWI. The results seem improved with an increase in the number of feedback paths, especially for more prediction in advance.

Table 9a presents the MAPE values for the network size of 500 nodes and different currencies for predicting the number of days in advance. Table 9b presents the corresponding MAPE values for TWI and SDR. It can be clearly seen that the model is highly accurate initially for 10 days, but as the number of days increase; generally the MAPE also increases, which causes a decrease in the accuracy of the model. Figure 13 shows the comparison between the actual values and estimated values, for 1000 days, of TWI by using a network of 500 nodes with a five-feedback paths scenario.

The network with ten feedback paths has been shown in Figure 7(c). There are ten different feedback paths that arise from the system output

TABLE 7a The Average MAPE Values of Different Currencies for Predicting 10, 50, 100, 300, 500, and 1000 Days of Data in Advance Using a Single-Feedback Network and a 500-Node Scenario

	Number of Days					
Currency	10	50	100	300	500	1000
EURO	**8.61**	10.63	12.37	8.45	16.07	15.03
JPY	**2.55**	6.67	6.04	11.53	18.23	13.66
GBP	**5.14**	6.95	10.22	11.87	14.77	12.42
CHF	**6.99**	9.82	7.35	9.52	13.36	11.56
NZD	**7.10**	10.56	7.19	13.26	25.30	30.80
CAD	**2.35**	8.66	15.12	17.90	18.70	29.77
HKD	**2.65**	7.35	6.95	8.63	9.41	10.10
SGD	**2.68**	4.90	4.84	9.31	8.60	9.51
MYR	**2.65**	7.36	6.96	8.85	9.64	8.95
TWD	**1.67**	5.03	6.44	9.96	9.05	7.96
KRW	**3.23**	3.73	4.96	10.31	10.82	25.59
IDR	**3.85**	10.44	15.34	22.90	22.16	15.70
CNY	**2.66**	7.34	6.95	8.84	9.64	8.79

TABLE 7b The Average MAPE Values of TWI and SDR for Predicting 10, 50, 100, 300, 500, and 1000 Days of Data in Advance Using a Single-Feedback Network and 500-Nodes Scenario

	Number of Days					
Currency	10	50	100	300	500	1000
TWI	**2.93**	5.43	5.70	8.81	8.86	9.30
SDR	**3.48**	5.93	5.30	7.70	8.77	10.97

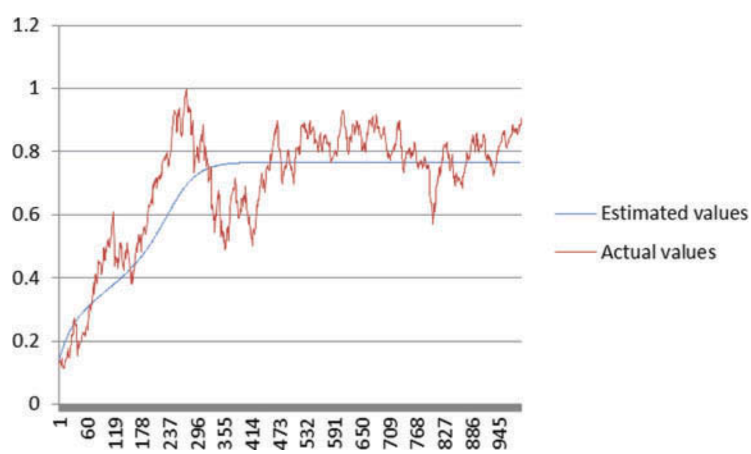

FIGURE 13 The plot for the TWI comparison between the actual values and the estimated data for 1000 days in advance using 10 days of historical data with 500 nodes and a 5-feedback-paths scenario.

TABLE 8a The MAPE, MSE, and RMSE Values for Various Network Sizes and 13 Different Currencies with a 5-Feedback Paths Scenario for Predicting 1000 Days of Data in Advance

Nodes		50	100	150	200	250	300	350	400	450	500
YEN	**MAPE**	14.18	11.66	33.20	32.56	31.85	**11.41**	11.73	12.82	33.33	14.50
	MSE	0.03	0.02	0.14	0.14	0.13	0.02	0.02	0.03	0.14	0.03
	RMSE	0.18	0.15	0.38	0.37	0.36	0.14	0.15	0.16	0.38	0.18
EURO	**MAPE**	16.33	15.63	34.55	33.92	33.15	17.18	15.72	**14.99**	34.63	16.24
	MSE	0.04	0.03	0.15	0.15	0.14	0.04	0.03	0.03	0.16	0.04
	RMSE	0.20	0.19	0.39	0.39	0.38	0.20	0.19	0.18	0.39	0.20
GBP	**MAPE**	15.18	15.60	43.26	42.61	41.84	17.72	15.69	**13.44**	43.36	14.70
	MSE	0.03	0.04	0.21	0.21	0.20	0.05	0.04	0.03	0.21	0.03
	RMSE	0.18	0.19	0.46	0.45	0.45	0.22	0.19	0.16	0.46	0.18
CHF	**MAPE**	12.97	14.39	45.11	44.47	43.72	16.60	14.39	**12.39**	45.21	12.59
	MSE	0.02	0.03	0.24	0.23	0.23	0.04	0.03	0.02	0.24	0.02
	RMSE	0.16	0.17	0.49	0.48	0.47	0.21	0.17	0.15	0.49	0.15
NZD	**MAPE**	32.71	29.76	19.55	19.29	**19.03**	28.95	29.89	30.19	19.56	32.82
	MSE	0.16	0.13	0.07	0.07	0.06	0.13	0.14	0.14	0.07	0.16
	RMSE	0.40	0.37	0.26	0.26	0.25	0.36	0.37	0.37	0.26	0.40
CAD	**MAPE**	32.93	30.11	30.47	31.27	30.65	29.25	30.16	29.45	**28.96**	32.75
	MSE	0.15	0.13	0.13	0.14	0.13	0.12	0.13	0.12	0.12	0.15
	RMSE	0.39	0.36	0.36	0.37	0.36	0.35	0.36	0.35	0.34	0.39
HKD	**MAPE**	**9.24**	12.11	48.37	47.74	47.02	14.84	11.99	10.21	48.48	8.99
	MSE	0.01	0.02	0.27	0.27	0.26	0.04	0.02	0.01	0.27	0.01
	RMSE	0.11	0.15	0.52	0.52	0.51	0.19	0.15	0.12	0.52	0.11
SGD	**MAPE**	11.35	12.70	44.30	43.65	42.94	15.43	12.70	10.33	44.45	**10.91**
	MSE	0.02	0.03	0.23	0.22	0.21	0.04	0.03	0.01	0.23	0.02
	RMSE	0.14	0.16	0.47	0.47	0.46	0.21	0.16	0.12	0.48	0.13
MYR	**MAPE**	9.48	11.09	46.08	45.45	44.72	13.57	11.03	9.26	46.19	**9.20**
	MSE	0.01	0.02	0.25	0.24	0.23	0.04	0.02	0.01	0.25	0.01
	RMSE	0.12	0.14	0.50	0.49	0.48	0.19	0.14	0.12	0.50	0.11
TWD	**MAPE**	9.69	10.54	44.33	43.69	42.98	13.05	10.53	**8.45**	44.48	9.30
	MSE	0.01	0.02	0.22	0.22	0.21	0.04	0.02	0.01	0.23	0.01
	RMSE	0.12	0.15	0.47	0.47	0.46	0.19	0.15	0.11	0.48	0.12
KRW	**MAPE**	29.55	28.33	24.29	23.90	**23.60**	29.88	28.49	26.27	24.45	29.11
	MSE	0.13	0.11	0.09	0.09	0.09	0.12	0.11	0.10	0.10	0.12
	RMSE	0.35	0.33	0.31	0.30	0.30	0.35	0.34	0.32	0.31	0.35
IDR	**MAPE**	15.94	11.38	31.38	30.88	30.16	**8.91**	11.49	14.65	31.35	16.49
	MSE	0.03	0.02	0.13	0.13	0.12	0.01	0.02	0.03	0.13	0.04
	RMSE	0.18	0.14	0.36	0.36	0.35	0.10	0.14	0.17	0.36	0.19
CNY	**MAPE**	9.75	10.93	45.34	44.72	43.99	13.35	10.90	**9.13**	45.46	9.46
	MSE	0.01	0.02	0.24	0.23	0.23	0.04	0.02	0.01	0.24	0.01
	RMSE	0.12	0.14	0.49	0.48	0.48	0.19	0.14	0.12	0.49	0.11

and are fed back into the system as input. Table 10a shows the MSE, MAPE, and RMSE values for 13 various currencies and ten different network sizes with 10 feedback paths to predict 1000 days of data in advance. Table 10b shows the corresponding results for TWI and SDR. The results indicate that using the greater number of feedback paths provides us with the advantage of obtaining more accurate results for predicting more days' data in advance.

TABLE 8b The MAPE, MSE, and RMSE Values for Various Network Sizes for TWI and SDR with 5-Feedback Paths Scenario for Predicting 1000 Days of Data in Advance

Nodes		50	100	150	200	250	300	350	400	450	500
TWI	**MAPE**	9.04	12.17	48.78	48.13	47.41	15.08	12.06	9.80	48.91	8.65
	MSE	0.01	0.02	0.27	0.26	0.26	0.04	0.02	0.01	0.27	0.01
	RMSE	0.11	0.15	0.52	0.51	0.51	0.20	0.15	0.12	0.52	0.11
SDR	**MAPE**	11.35	12.70	44.30	43.65	42.94	15.43	12.70	10.33	44.45	10.91
	MSE	0.02	0.03	0.23	0.22	0.21	0.04	0.03	0.01	0.23	0.02
	RMSE	0.14	0.16	0.47	0.47	0.46	0.21	0.16	0.12	0.48	0.13

TABLE 9a The Average MAPE Values of Different Currencies for Predicting 10, 50, 100, 300, 500, 1000 Days of Data in Advance, Using 5 Feedback Paths and a Network of 50 Nodes Scenario

	Number of Days					
Currency	10	50	100	300	500	1000
EURO	8.79	10.36	11.12	**7.83**	17.23	16.24
YEN	**2.73**	6.62	5.99	8.56	18.18	14.48
GBP	**5.31**	7.21	11.46	15.91	18.77	14.70
CHF	7.17	9.61	**6.62**	12.62	16.69	12.59
NZD	**7.26**	10.13	7.52	11.29	25.94	32.82
CAD	**2.33**	7.47	12.79	17.39	19.93	32.75
HKD	**2.84**	7.27	7.13	9.41	10.35	8.99
SGD	**2.86**	4.83	5.34	12.30	10.99	10.91
MYR	**2.84**	7.28	7.14	9.97	10.78	9.20
TWD	**1.85**	4.98	6.82	11.74	10.67	9.30
KRW	**3.41**	3.90	5.70	11.54	13.13	29.11
IDR	**4.02**	10.43	14.62	18.96	21.49	16.49
CNY	**2.84**	7.27	7.13	9.96	10.77	9.46

TABLE 9b The Average MAPE Values of TWI, SDR, for predicting 10, 50, 100, 300, 500, 1000 Days of Data in Advance Using 5 Feedback Paths and a Network of 500 Nodes Scenario

	Number of Days					
Currency	10	50	100	300	500	1000
TWI	**3.12**	5.32	6.06	11.16	10.93	8.62
SDR	**3.60**	5.78	5.61	10.50	11.43	10.31

All of these values are based on forecasting data of 1000 days in advance. Table 11a presents the MAPE values for the network size of 100 nodes with 10 feedbacks and 13 different currencies for 10, 50, 100, 300, 500, and 1000 days data prediction in advance, whereas Table 11b presents the MAPE values for TWI and SDR. Comparing Tables 11, 9, and 7, it is noted that the increase in number of feedback paths is providing us with better results.

We forecasted the foreign exchange rate for approximately 1000 days based on 10 days of data history, and the results proved that RCGPANN

TABLE 10a The MAPE, MSE, and RMSE Values for Various Network Sizes and 13 Different Currencies with 10-Feedback-Paths Scenario for Predicting 1000 Days of Data in Advance

Nodes		50	100	150	200	250	300	350	400	450	500
YEN	**MAPE**	11.48	10.35	11.62	11.44	12.75	34.40	13.48	**9.23**	11.90	10.72
	MSE	0.02	0.03	0.02	0.05	0.03	0.04	0.02	0.02	0.11	0.02
	RMSE	0.14	0.18	0.14	0.23	0.16	0.19	0.14	0.15	0.33	0.13
EURO	**MAPE**	14.50	**9.11**	13.76	34.64	10.29	26.11	24.22	12.37	44.64	15.83
	MSE	0.04	0.03	0.04	0.09	0.03	0.04	0.04	0.04	0.12	0.04
	RMSE	0.20	0.19	0.19	0.30	0.17	0.21	0.21	0.19	0.35	0.20
GBP	**MAPE**	17.06	15.67	16.46	24.79	**14.51**	16.96	16.24	16.42	30.01	17.23
	MSE	0.05	0.02	0.04	0.13	0.02	0.05	0.07	0.04	0.17	0.05
	RMSE	0.21	0.14	0.20	0.35	0.14	0.23	0.26	0.20	0.42	0.22
CHF	**MAPE**	17.48	**11.92**	16.61	31.15	12.13	20.01	20.21	16.75	38.68	17.58
	MSE	0.04	0.02	0.04	0.13	0.02	0.07	0.07	0.03	0.20	0.05
	RMSE	0.21	0.14	0.19	0.36	0.14	0.27	0.26	0.18	0.44	0.22
NZD	**MAPE**	16.34	**11.39**	15.65	30.72	11.72	23.37	20.17	15.03	40.80	16.27
	MSE	0.13	0.14	0.13	0.05	0.13	0.05	0.11	0.14	0.05	0.13
	RMSE	0.36	0.38	0.36	0.23	0.36	0.23	0.33	0.38	0.22	0.36
CAD	**MAPE**	29.17	31.52	28.79	18.66	29.78	19.43	27.73	30.31	**16.80**	29.15
	MSE	0.12	0.13	0.12	0.12	0.12	0.05	0.11	0.13	0.12	0.13
	RMSE	0.35	0.36	0.35	0.35	0.34	0.22	0.34	0.36	0.35	0.36
HKD	**MAPE**	29.58	30.21	29.27	29.62	28.75	**17.91**	28.16	30.75	29.33	30.19
	MSE	0.03	0.01	0.03	0.16	0.02	0.09	0.09	0.03	0.23	0.04
	RMSE	0.19	0.12	0.17	0.40	0.13	0.30	0.30	0.16	0.48	0.21
SGD	**MAPE**	14.26	**10.16**	13.62	34.69	10.96	27.18	24.24	12.31	44.36	15.80
	MSE	0.04	0.01	0.03	0.16	0.01	0.06	0.09	0.03	0.19	0.05
	RMSE	0.20	0.11	0.18	0.39	0.11	0.25	0.30	0.18	0.44	0.22
MYR	**MAPE**	14.91	**9.41**	14.02	33.77	9.56	21.63	24.26	13.86	40.27	17.06
	MSE	0.03	0.01	0.03	0.16	0.01	0.08	0.09	0.03	0.21	0.04
	RMSE	0.18	0.11	0.17	0.40	0.12	0.28	0.30	0.16	0.46	0.21
TWD	**MAPE**	13.08	**9.10**	12.40	32.62	9.36	24.88	22.63	11.71	42.07	14.84
	MSE	0.03	0.01	0.03	0.13	0.01	0.06	0.08	0.03	0.19	0.04
	RMSE	0.19	0.10	0.17	0.36	0.10	0.25	0.29	0.16	0.43	0.21
KRW	**MAPE**	12.56	**8.03**	11.78	30.93	8.07	22.16	21.73	11.64	40.33	14.72
	MSE	0.12	0.10	0.12	0.13	0.09	0.04	0.14	0.12	0.08	0.13
	RMSE	0.35	0.32	0.34	0.36	0.30	0.21	0.38	0.35	0.29	0.36
IDR	**MAPE**	29.80	25.88	28.84	30.68	24.40	**16.93**	33.17	30.40	23.53	31.36
	MSE	0.01	0.04	0.01	0.07	0.03	0.05	0.02	0.02	0.10	0.01
	RMSE	0.11	0.20	0.11	0.27	0.17	0.21	0.15	0.12	0.32	0.10
CNY	**MAPE**	**9.29**	16.61	9.68	21.22	14.63	17.69	12.54	10.13	27.71	8.44
	MSE	0.03	0.01	0.03	0.15	0.01	0.07	0.09	0.03	0.20	0.04
	RMSE	0.18	0.11	0.17	0.39	0.12	0.27	0.29	0.16	0.45	0.21

is superior in accuracy and implementation cost to other networks. The results tabulated highlight the fact that increasing the number of nodes or using a greater number of feedbacks provides us with accurate results in the long run. Figure 14 shows the comparison between the actual values and estimated values, for 1000 days, of TWI by using a network of 100 nodes with a 10-feedback-paths scenario.

TABLE 10b The MAPE, MSE, and RMSE Values for Various Network Sizes and TWI and SDR with 10-Feedback-Paths Scenario for Predicting 1000 Days of Data in Advance

Nodes		50	100	150	200	250	300	350	400	450	500
TWI	**MAPE**	5.13	6.99	**4.39**	5.13	8.06	3.96	6.01	5.11	4.97	4.66
	MSE	0.04	0.01	0.03	0.16	0.01	0.08	0.09	0.03	0.23	0.05
	RMSE	0.19	0.11	0.18	0.40	0.12	0.29	0.29	0.16	0.48	0.21
SDR	**MAPE**	12.89	8.98	12.18	32.11	**9.06**	24.15	22.25	11.70	41.34	14.70
	MSE	0.04	0.02	0.03	0.16	0.02	0.09	0.09	0.03	0.24	0.05
	RMSE	0.20	0.13	0.19	0.40	0.14	0.31	0.29	0.17	0.49	0.22

TABLE 11a The Average MAPE Values for Various Currencies for 10, 50, 100, 300, 500, and 1000 Days of Data in Advance, Using 5 Feedback Paths and a Network of 500 Nodes Scenario

	Number of Days					
Currency	10	50	100	300	500	1000
EURO	9.63	12.30	14.75	10.07	17.21	15.67
YEN	3.38	8.04	7.59	13.29	19.51	14.36
GBP	6.14	6.96	8.69	9.91	13.77	11.92
CHF	7.97	11.42	9.52	9.04	13.23	11.39
NZD	8.17	12.32	8.69	14.71	26.38	31.52
CAD	3.16	9.76	16.08	18.24	19.06	30.21
HKD	3.51	8.77	8.08	9.43	9.93	10.16
SGD	3.54	6.31	5.10	8.90	8.40	9.41
MYR	3.52	8.78	8.09	9.55	10.10	9.10
TWD	2.51	6.34	6.77	10.10	9.16	**8.03**
KRW	4.03	4.03	4.11	10.13	10.88	25.88
IDR	4.60	11.72	17.34	25.28	23.80	16.61
CNY	3.52	8.77	8.09	9.55	10.10	8.98

TABLE 11b The Average MAPE Values of Different Currencies for 10, 50, 100, 300, 500, and 1000 Days of Data in Advance, Using 5 Feedback Paths and a Network of 500 Nodes Scenario

	Number of Days					
Currency	10	50	100	300	500	1000
TWI	3.85	6.85	6.20	8.86	8.93	9.11
SDR	4.30	7.31	6.15	7.78	8.90	10.80

Figure 15 shows the best RCGPANN network obtained from 500 nodes with 10 feedback paths. From the figure it is evident that the ultimate network uses only two inputs and a single feedback in the final optimum model. Also, only six neurons are active out of 500 evolved, thus resulting in a computationally efficient model. This is the beauty of RCGPANN, which results in this optimum model.

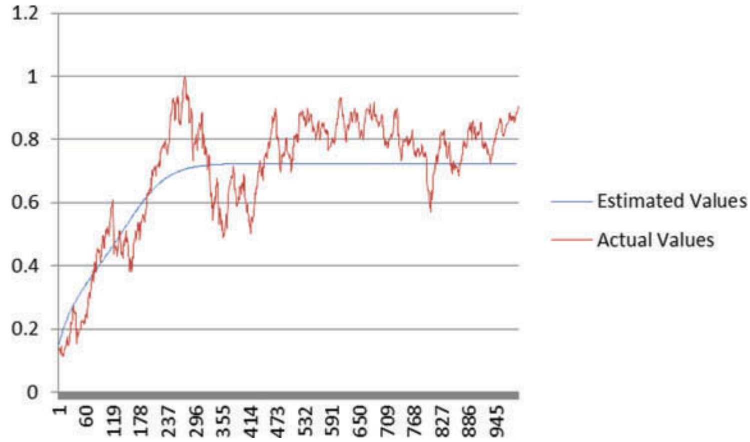

FIGURE 14 The plot for TWI comparison between the actual values and the estimated values for predicting 1000 days of data with 100 nodes and a 10-feedback-paths scenario.

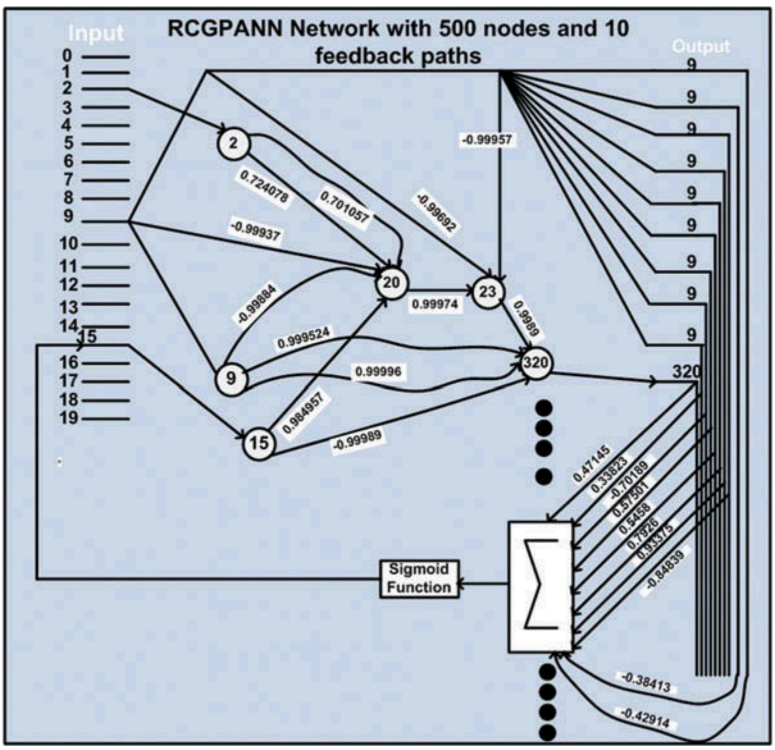

FIGURE 15 The RCGPANN Network with 500 nodes and 10 feedback paths for prediction of TWI.

CONCLUSION AND FUTURE WORK

We have presented a novel algorithm to evolve a recurrent neural network for obtaining a computationally efficient and accurate prediction model. We have trained the network on 500 days of data from an historical data set to predict the 11th day exchange rates based on the 10 days of data history, tested it on thirteen various currencies, the trade weighted index (TWI), and special drawing rights (SDR) to do the same task, producing 98.872 percent accurate results, outperforming all the contemporary networks that exist in the field. We have evaluated three different cases of RCGPANN with single, five, and ten feedbacks and evaluated the system performance with the ten-feedback network performing better than all others. We have evaluated the network further by exploring its capabilities to predict more than a single day's data rate in advance. We have evaluated all the three cases with various network sizes and trained networks on a range of independent seeds. The networks' performances are evaluated to predict 10, 50, 100, 300, 500, and 1000 days of data rates in advance. The results are tabulated and presented, and the detailed analysis demonstrated that RCGPANN performs better than other prediction models, even to predict data rates for more days in advance, especially predicting currencies' exchange rates correctly for up to 1000 days (4 years) in advance with 92 percent accuracy. The results demonstrate that RCGPANN is superior to other networks in terms of accuracy and computational cost. RCGPANN is not restricted to mere foreign exchange rates; indeed it can be applied to various fields in which data can be forecasted on the basis of the historical record. Numerous fields such as load forecasting, river flow forecasting, weather forecasting, and various other fields can be forecasted not only accurately but also computationally efficiently using RCGPANN. RCGPANN is significant in the sense that it utilizes a minimum amount of past data for forecasting. Indeed, RCGPANN has great potential for exploration in a range of fields and will be able to unlock various problems in the upcoming years.

REFERENCES

Atiya, A. F., S. Member, S. M. El-shoura, S. I. Shaheen, and M. S. El-sherif. 1999. A comparison between neural-network forecasting techniques — Case study: River flow forecasting. *IEEE Transactions on Neural Networks* 10(2):402–409.

Aussem, A., F. Murtagh, and M. Sarazin. 1995. Dynamical recurrent neural network towards environmental series prediction. *International Journal of Neural Systems* 6(2):145–170.

Bhattacharya, A., A. G. Parlos, and A. F. Atiya. 2003. Prediction of MPEG-coded video source trace using recurrent neural networks. *IEEE Transactions on Neural Networks* 51(8):2177–2190.

Boné, R., M. Assaad, and M. Crucianu. 2003. Boosting recurrent neural networks for time series prediction. *Artificial neural nets and genetic algorithms*, 18–22. Vienna: Spinger.

Cai, X., N. Zhang, G. K. Venayagamoorthy, and D. C. Wunsch. 2004. Time series prediction with recurrent neural networks trained by a hybrid PSO-EA algorithm. In *Proceedings of the IEEE international joint conference on neural networks*, 2:1647–1652. IEEE.

Chen, A. S., and M. T. Leung. 2005. Performance evaluation of neural network architectures: The case of predicting foreign exchange correlations. *Journal of Forecasting* 24(6):403–420.

Connor, J. T., D. Martin, and L. E. Atlas. 1994. Recurrent neural networks and robust time series prediction. *IEEE Transactions on Neural Networks* 5(2):240–254.

Dablemont, S., G. Simon, A. Lendasse, A. Ruttiens, F. Blayo, and M. Verleysen. 2003. Time series forecasting with SOM and local non-linear models – Application to the DAX30 index prediction. Presented at Proceedings of the 4th Workshop on Self-Organizing Maps, (WSOM'03), 340–345. Hibikino, Japan, September.

Dunis, C. L., and X. Huang. 2002. Forecasting and trading currency volatility: An application of recurrent neural regression and model combination. *Journal of Forecasting* 21(5):317–357.

Farsa, M. A., and S. Zolfaghari. 2011. Residual analysis and combination of embedding theorem and artificial intelligence in chaotic time series forecasting. *Applied Artificial Intelligence: An International Journal* 25(1):45–73.

Giles, C. L., S. Lawrence, and A. C. Tsoi. 2001. Noisy time series prediction using recurrent neural networks and grammatical interference. *Machine Learning* 44(1–2):161–183.

Gomez, F., and J. Schmidhuber. 2008. Accelerated neural evolution through cooperatively coevolved synapses. *Journal of Machine Learning Research* 9:937–965.

Gradojevic, N., and J. Yang. 2006. Non-linear non-parametric non-fundamental exchange rate forecasting. *Journal of Forecasting* 25(4):227–245.

Haykin, S. 1999. *Neural networks: A comprehensive foundation* (2nd ed.), 20–21. Upper Saddle River, NJ: Prentice Hall.

Huang, W., S. Wang, L. Yu, Y. Bao, and L. Wang. 2006. A new computational method of input selection for stock market forecasting with neural networks. In *Proceedngs of the 6th international conference on computational science–Part IV*, 308–315. Berlin, Heidelberg: Springer-Verlag.

Husken, M., and P. Stagge. 2003. Recurrent neural networks for time series classification. *Journal of Neurocomputing* 50:223–235.

Joseph, N. L. 2001. Model specification and forecasting foreign exchange rates with vector auto regression. *Journal of Forecasting* 20(7):451–484.

Kaashoek, J. F., and H. K.V Dijk 2002. Neural network pruning applied to real exchange rate analysis. *Journal of Forecasting* 21(8):559–577.

Kadilar, C., M. Simsek, and C. H. Aladag, 2009. Forecasting the exchange rate series with ANN: The case of turkey. *Istanbul University Econometrics and Statistics e-Journal*, 9(1):17–29.

Khan G. M., S. Khan, and F. Ullah. 2011. Short-term daily peak load forecasting using fast learning neural network. In *2011 International Conference on Intelligent System Design and Applications* 843–848. IEEE.

Khan, M. M., G. M. Khan, and J. F. Miller. 2010a. Efficient representation of recurrent neural networks for markovian/non-markovian non-linear control problems. In *2010 international conference on intelligent system design and applications ISDA'10*, 615–620. IEEE.

Khan, M. M., G. M. Khan, and J. F. Miller. 2010b. Evolution of optimal anns for non-linear control problems using Cartesian genetic programming. The *2010 international conference on artificial intelligence*, 339–346. AIA Press.

Khan, M. M., G. M. Khan, and J. F. Miller. 2010c. Evolution of neural networks using Cartesian genetic programming. In *IEEE Congress on Evolutionary Computation*,1–8. IEEE.

Kumar, D. N., K. S. Raju, and T. Sathish. 2004. River flow forecasting using recurrent neural networks. *Water Resource Managemment* 18(2):143–161.

Linkens, D., and Y. Nyongesa. 1996. Learning systems in intelligent control: An appraisal of fuzzy, neural and genetic algorithm control applications. *IEEE Proceedings of Control Theory and Applications* 134(4):367–385.

Madhavan, P. G. 2011. A recurrent neural network learning algorithm for time series prediction. *Journal of Intelligent Systems* 7(1-2):103–116.

Marra S., and F. C. Morabito. 2005. A new technique for solar activity forecasting using recurrent Elman networks. *IEC Prague* 1(7):68–73.

Marsh, I. W. 2000. High-frequency Markov switching models in the foreign exchange market. *Journal of Forecasting* 19(2):123–134.

Miller, J. F., and S. L. Harding, 2008. Cartesian genetic programming. In *Proceedings of the* 2008 *GECCO conference companion on Genetic and evolutionary computation, ser. GECCO '08*, 2701–2726. New York, NY, USA: ACM.

Nag, A. K., and A. Mitra. 2002. Forecasting daily foreign exchange rates using genetically optimized neural networks. *Journal of Forecasting* 21(7):501–511.

Parikakis, G. S., and A. Merika. 2009. Evaluating volatility dynamics and the forecasting ability of Markov switching models. *Journal of Forecasting* 28(8):736–744.

Philip, A., A. A. Tofiki, and A. A. Bidemi. 2011. Artificial neural network model for forecasting foreign exchange rate. *World of Computer Science and Information Technology Journal* 1(3):110–118.

Qian, B., and K. Rasheed. 2010. Foreign exchange market prediction with multiple classifiers. *Journal of Forecasting* 29(3):271–284.

Refenes, A. N., M. Azema-Barac, L. Chen, and S. A. Karoussos. 1993. Currency exchange rate prediction and neural network design strategies. *Neural Computing & Applications*, 1(1):46–58.

Rivero, S. S., and E. Garcia. 2005. Forecasting the dollar/euro exchange rate: Are international parities useful? *Journal of Forecasting* 24(5):369–377.

Rothermich, J. A., and J. F. Miller. 2002. Studying the emergence of multi-cellularity with Cartesian genetic programming in artificial life. *U.K. Workshop on Computational Intelligence (UKCI'02)*, 397–403, Birmingham, UK.

Skintzi, V.D., and S.X. Sisinis, 2007. Evaluation of correlation forecasting model for risk management *Journal of Forecasting* 26(7):497–526.

Taremian, H. 2010. Stock market value prediction using neural networks. In *2010 international conference on computer information systems and industrial management applications (CISIM)*, 132–136. IEEE.

Tenti, P. 1996. Forecasting foreign exchange rates using recurrent neural networks. *Applied Artificial Intelligence: An International Journal* 10(6):567–582.

Toha, S. F., and M. O. Tokhi. 2008. MLP and Elman recurrent neural network modeling for the TRMS. In *Proceedings of the 7th IEEE international conference on cybernetic intelligent systems*,1–6. IEEE.

Vasicek, Z., and L. Sakanina. 2010. Hardware accelerator of Cartesian genetic programming with multiple fitness units. *Computing and Informatics* 29:1359–1371.

William J. R., and Zipser D. 1989. Experimental analysis of the real time recurrent learning algorithm. *Connections Science*, 1(1):1–25.

Zhang, B., and H. Muhlenbein.1993. Evolving optimal neural networks using genetic algorithms with Occam's razor. *Complex Systems* 7(3):199–220.

Zhang, G., B. E. Patuwo, and M. Y. Hu, 1998. Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting* 14(1):35–62.