

Forex Forecasting: A Comparative Study of LLWNN and NeuroFuzzy Hybrid Model

Puspanjali Mohapatra
(Asst. Prof)
Dept of Computer Science and
Engineering
IIIT Bhubaneswar

Munnangi Anirudh
Dept of Electronics and
Telecommunication
Engineering
IIIT Bhubaneswar

Tapas Kumar Patra
(Reader)
Dept of Instrumentation &
Electronics Engineering
College of Engineering
Technology
Bhubaneswar, India

ABSTRACT

This paper shows how the performance of the basic Local Linear Wavelet Neural Network model (LLWNN) can be improved with hybridizing it with fuzzy model. The new improved LLWNN based Neurofuzzy hybrid model is used to predict two currency exchange rates i.e. the U.S. Dollar to the Indian Rupee and the U.S. Dollar to the Japanese Yen. The forecasting of foreign exchange rates is done on different time horizons for 1 day, 1 week and 1 month ahead. The LLWNN and Neurofuzzy hybrid models are trained with the backpropagation training algorithm. The two performance measures i.e. the Root Mean Square Error (RMSE) and the Mean Absolute Percentage Error (MAPE) show the superiority of the Neurofuzzy hybrid model over the LLWNN model.

General Terms

Time Series Prediction

Keywords

Forex, Foreign Exchange Market., LLWNN, LLWNN based Neurofuzzy hybrid model, Back Propagation training algorithm.

1. INTRODUCTION

The foreign exchange market which is usually referred to as Forex [1] is a large business with a large turnover in which trading takes place round the clock and all over the world. International business is totally controlled by international transactions which are settled in the near future. Therefore the exchange rate forecasting is required to evaluate the foreign denominated cashflows involved in international transactions. It can determine the benefits and risks attached to the international business environment. Currency exchange refers to the trading of one currency against another. Generally consumers consult to the currency exchange to convert their own home currency to the desired foreign currency.

Forex was established in 1971 when floating exchange rates began to materialize. In terms of trading volume, it is the world's largest market with daily trading volumes in excess of \$1.5 trillion U.S. Dollars. It is the most liquid market. The major currencies involved in the Forex transactions are the U.S. Dollar(USD), Japanese Yen(JPY), Euro(EUR), Swiss Franc(CHF), British Pound (GBP), Canadian Dollar(CAD) and Australian Dollar(AUD) where most of the currencies are rated against the USD. Trading between two nondollar currencies occur by first trading one against USD and then trading the USD against the second nondollar currency. Here the exchange rate is referred to as the cross rate.

The Forex economic time series is highly dynamic, nonlinear, complicated, random, chaotic and nonparametric in nature. Further the timeseries is highly nonstationary and have frequent structural breaks[2] and they are also affected by many economic factors, political events, government policies, investors expectations, psychology of traders and investors. The quantity and quality of the data points, the presence of various external issues, inflation rate make the modeling process a very difficult task. These complexities motivated the researchers to use various statistical models like ARMA, ARIMA[3,4], ARCH, GARCH, GARCH-M, EGARCH, IGARCH [5], Box and Jenkins approach[6] along with various soft computing and evolutionary computing methods[7,8]. Artificial Neural Network(ANN), fuzzy set theory, Support Vector Machine(SVM) etc. are considered under soft computing techniques where as the various evolutionary learning algorithms include Genetic algorithm(GA)[9], Ant Colony Optimization(ACO)[10], Particle Swarm Optimization (PSO)[11,12,13], Differential Evolution (DE)[14,15], Bacterial Foraging Optimization (BFO)[16] etc. The Literature survey reveals that different types of ANN's like Radial Basis Function(RBF) [17], Multi Layer Perceptron (MLP) [18], Recurrent Neural Network(RNN)[19], Time delay Neural Network (TDNN) [20], Machine Learning techniques [21], Functional Link Artificial Neural Network(FLANN) [22,23,24], Local Linear Wavelet Neural Network (LLWNN) [25], Evolutionary Neurofuzzy NN[26,27], and various Neurofuzzy hybrid models have been used for time series forecasting. Statistical method can handle only linear data, they become unable to follow the non-linear pattern hidden within the exchange rate data. In this paper, a local linear wavelet neural network is proposed, in which the connection weights between the hidden layer units and output units are replaced by the local linear model. At first the LLWNN model is trained with the backpropagation training algorithm. Later on a LLWNN based neurofuzzy hybrid model is designed and it is also trained with the same back propagation training algorithm. When compared simulation results of Neurofuzzy hybrid model shows its superiority over the LLWNN model. Here LLWNN model can recognize the patterns and adapt themselves to some extent to cope with the changing environment while the Neurofuzzy hybrid model incorporate the human knowledge and expertise for inferencing and decision making.

This paper is organized as follows. Section 2 deals with the basic principle of LLWNN. The basic principle of Neurofuzzy hybrid model is dealt in section 3. Section 4 deals with the BP learning algorithm used for both models. Performance of both models are discussed in section 5. Outputs of both

models are given in section 6. The training and testing results of both models are analysed in section 7. Finally conclusion is given in section 8.

2. BASIC PRINCIPLE OF LOCAL LINEAR WAVELET NEURAL NETWORK (LLWNN)

In Local Linear Wavelet Neural Network (LLWNN) the number of neurons in the hidden layer is equal to the number of inputs and the connection weights between the hidden layer units and output units are replaced by a local linear model. From the literature of LLWNN, It is known that the local linear model provides a more parsimonious interpolation in a high dimensional space providing it's suitability for time series forecasting. This local capacity of the LLWNN model provides some advantages like learning efficiency and the structure transparency. This model suffers from shortcoming that for higher dimensional problems many hidden layer units are needed. Here the LLWNN model is chosen for 1 day, 1 week and 1 month ahead forecasting of Forex. The connection weights between the hidden layer and output layer of conventional wavelet neural networks are replaced with local linear model to form LLWNN model. The architecture of LLWNN model is shown in the Fig. 1.

Generally wavelets are represented in the following form.

$$\psi = \left\{ \psi_i = |a_i|^{-\frac{1}{2}} \psi_i \left(\frac{x-b_i}{a_i} \right); a_i, b_i \in \mathbb{R}^n, i \in \mathbb{Z} \right\} \quad (1)$$

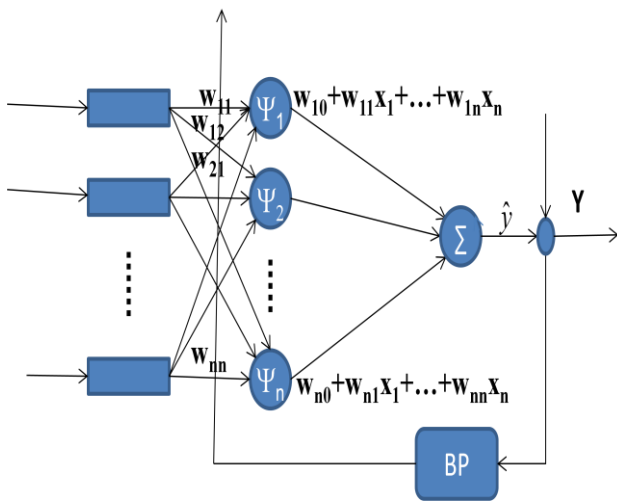


Fig 1: Architecture of LLWNN.

$x = (x_1, x_2, x_3, \dots, x_n)$, $a_i = (a_{i1}, a_{i2}, a_{i3}, \dots, a_{in})$, $b_i = (b_{i1}, b_{i2}, b_{i3}, \dots, b_{in})$ are a family of functions generated from one single function $\psi(x)$ which is localized in both time space and frequency space is called a mother wavelet and the parameters a_i and b_i are called the scale and translation parameters respectively.

The mother wavelet is $\psi(x)$ which is given as

$$\psi(x) = \frac{-x^2}{2} e^{-\left(\frac{x}{\sigma}\right)^2} \quad (2a)$$

$$\psi(x) = e^{-\left(\frac{x-c}{\sigma}\right)^2} \quad (2b)$$

$$\text{where } x = \sqrt{p_1^2 + p_2^2 \dots + p_n^2} \quad (3)$$

Instead of the straight forward weight w_i a linear model $v_i = (w_{i0} + w_{i1}x_1 + \dots + w_{in}x_n)$ is introduced. The activities of the linear models v_i ($i=1,2,\dots,M$) are determined by the associated locally active wavelet functions $\psi_i(x)$ ($i=1,2,\dots,M$), thus v_i is only locally significant.

The standard output of Wavelet Neural Network is

$$f(x) = \sum_{i=1}^M w_i \psi_i(x) = \sum_{i=1}^M w_i |a_i|^{-\frac{1}{2}} \psi_i \left(\frac{x-b_i}{a_i} \right) \quad (4)$$

where Ψ_i is the wavelet activation of i^{th} unit of hidden layer and w_i is the weight connecting the i^{th} unit of hidden layer to the output layer unit.

In LLWNN model the output in the output layer is

$$Y = \sum_{i=1}^M (w_{i0} + w_{i1}x_1 + \dots + w_{in}x_n) |a_i|^{-\frac{1}{2}} \psi \left(\frac{x-b_i}{a_i} \right) \quad (5)$$

3. BASIC PRINCIPLE OF LLWNN BASED NEUROFUZZY HYBRID MODEL

The architecture of proposed LLWNN based Neurofuzzy Hybrid Model is shown in Fig. 2. The Neurofuzzy hybrid system is responsible for the determination of the knowledge base which consists of the following subsystems of developing the membership function, defining a fuzzy reasoning mechanism, the number of rule and rule base. The proposed LLWNN based Neurofuzzy hybrid model uses a LLWNN based combination of input variables. Each fuzzy rule corresponds to a sub-LLWNN, comprising a link. The LLWNN model realizes a fuzzy IF-THEN rule in the following form and the same "P" number of patterns 'Xp' is passed through the linear combiner and multiplied with the weight to generate the partial sum.

Rule j :

IF x_1 is A_{1j} and x_2 is A_{2j} and x_i is A_{ij} and x_N is A_{Nj}

$$\text{THEN } \hat{y}_j = \sum_{k=1}^M w_{kj} \phi_k = w_{1j} \phi_1 + w_{2j} \phi_2 + \dots + w_{Mj} \phi_M \quad (6)$$

where x_i and \hat{y}_i are the input and local output variables, respectively;

A_{ij} is the linguistic term of the precondition part with Gaussian membership function, N is the number of input variables, w_{kj} is the link weight of the local output, ϕ_k is the basis function of input variables, M is the number of basis function, and rule j is the j^{th} fuzzy rule.

The operation functions of the nodes in each layer of the LLWNN model is now described. In the following description, $u^{(l)}$ denotes the output of a node in the l^{th} layer.

Layer 1:

No computation is performed in layer 1. Each node in this layer only transmits input values to the next layer directly.

$$u_i^{(1)} = x_i \quad (7)$$

Layer 2:

Each fuzzy set A_{ij} is described here by a Gaussian membership function. Therefore, the calculated membership value in layer 2 is

$$u_{ij}^{(2)} = \exp\left(-\frac{[u_i^{(1)} - m_{ij}]^2}{\sigma_{ij}^2}\right) \quad (8)$$

where m_{ij} and σ_{ij} are the mean and variance of the Gaussian membership function, respectively, of the j^{th} term of the i^{th} input variable x_i .

Layer 3:

Nodes in layer 3 receive one-dimensional membership degrees of the associated rule from the nodes of a set in layer 2. Here, the product operator described earlier is adopted to perform the precondition part of the fuzzy rules. As a result, the output function of each inference node is

$$u_j^{(3)} = \prod_i u_{ij}^{(2)} \quad (9)$$

where the $\prod_i u_{ij}^{(2)}$ of a rule node represents the firing strength of its corresponding rule.

Layer 4:

Nodes in layer 4 are called consequent nodes. The input to a node in layer 4 is the output from layer 3, and the other inputs are calculated from the LLWNN that has used the function $\tanh(\cdot)$, as shown in Fig. 1. For such a node

$$u_j^{(4)} = u_j^{(3)} \sum_{k=1}^M w_{kj} \phi_k \quad (10)$$

where w_{kj} is the corresponding link weight of the LLWNN and ϕ_k is the functional expansion of input variables.

Layer 5:

The output layer in node 5 acts as the defuzzification layer. So, the final output of the LLWNN model 'y' is expressed as $y = (y_{11} * Fz_{11} + y_{22} * Fz_{22}) / (Fz_{11} + Fz_{22})$

$$(11)$$

Where y_{11} and y_{22} are the output from the LLWNN model and Fz_{11} and Fz_{22} are the output from the NeuroFuzzy hybrid model or output from layer 3.

4. BACK PROPAGATION (BP) LEARNING ALGORITHM

Back propagation algorithm is one of the tested supervised learning algorithm. It minimises the objective function by adjusting the link weights used to develop the models. The gradient of the cost function with respect to that particular weight parameter is calculated and the parameters are updated with the negative gradient.

$$E =$$

$$\frac{1}{2} \left[\begin{array}{c} y(t) - w_{10}\psi_1(x) - w_{11}p_1\psi_1(x) - \dots \\ -w_{20}\psi_2(x) - w_{21}p_2\psi_2(x) - \dots - w_{m0}\psi_m(x) - \\ w_{m1}p_1\psi_m(x) - \dots - w_{mn}p_n\psi_m(x) \end{array} \right]^2 \quad (12)$$

where $y(t)$ is the desired value. A weight updation from i^{th} to $(i+1)^{\text{th}}$ iteration i.e., from $w(t)$ to $w(t+1)$ is given by

$$w(t+1) = w(t) + \Delta w(t) = w(t) + (-w\eta \frac{\partial E(t)}{\partial w(t)}) \quad (13)$$

where $\frac{\partial E}{\partial w}$ for all weights are described by equation (14) to (19).

$$\frac{\partial E}{\partial w_{i0}} = w_{i0} + w\eta * e * \left(\frac{1}{2}\right) * (x_1^2 + x_2^2 + \dots x_n^2) * \exp\left(-((x_1 - c_i)^2 + (x_2 - c_i)^2 + \dots (x_n - c_i)^2)\right) \quad (14)$$

For $\forall j \neq 0$,

$$\frac{\partial E(t)}{\partial w_{ij}} = w_{ij} + w\eta * e * \left(\frac{1}{2}\right) * (x_1^2 + x_2^2 + \dots x_n^2) * \exp\left(-((x_1 - c_i)^2 + (x_2 - c_i)^2 + \dots (x_n - c_i)^2)\right) * x_j \quad (15)$$

i.e.,

$$\frac{\partial E}{\partial w_{10}} = w_{10} + w\eta * e * \left(\frac{1}{2}\right) * (x_1^2 + x_2^2 + \dots x_n^2) * \exp\left(-((x_1 - c_1)^2 + (x_2 - c_1)^2 + \dots (x_n - c_1)^2)\right) \quad (16)$$

$$\frac{\partial E}{\partial w_{12}} = w_{12} + w\eta * e * \left(\frac{1}{2}\right) * (x_1^2 + x_2^2 + \dots x_n^2) * \exp\left(-((x_1 - c_1)^2 + (x_2 - c_1)^2 + \dots (x_n - c_1)^2)\right) x_2 \quad (17)$$

$$\frac{\partial E}{\partial w_{20}} = w_{20} + w\eta * e * \left(\frac{1}{2}\right) * (x_1^2 + x_2^2 + \dots x_n^2) * \exp\left(-((x_1 - c_2)^2 + (x_2 - c_2)^2 + \dots (x_n - c_2)^2)\right) \quad (18)$$

$$\frac{\partial E}{\partial w_{21}} = w_{21} + w\eta * e * \left(\frac{1}{2}\right) * (x_1^2 + x_2^2 + \dots x_n^2) * \exp\left(-((x_1 - c_2)^2 + (x_2 - c_2)^2 + \dots (x_n - c_2)^2)\right) x_1 \quad (19)$$

Other weights are also updated like this. $w\eta$ is the learning rate used in LLWNN model

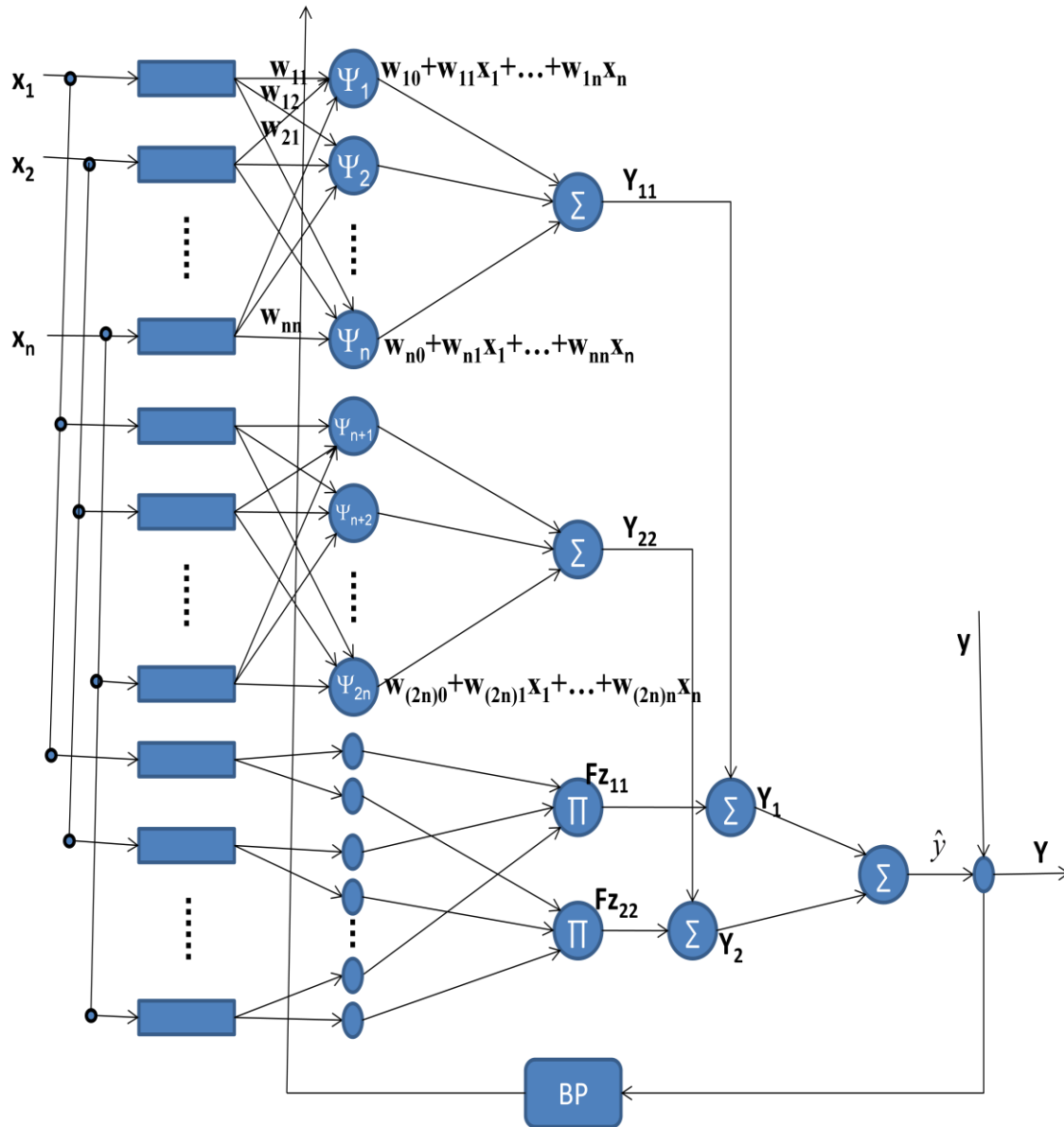


Fig 2: Architecture of proposed LLWNN based NEUROFUZZY Hybrid Model

Table 1.Details of Forex DataSets

FOREX DATA SETS	Total Training Samples	Total Testing Samples	Training Sample	Testing Sample
Dollar to Rupee	1 st -MAY 2008 to 1 st -MAY 2011	1 st -June 2011 to 1 st -June 2012	1095	366
Dollar to Yen	1 st -MAY 2008 to 1 st -MAY 2011	1 st -June 2011 to 1 st -June 2012	1095	366

5. STUDY OF PERFORMANCE OF LLWNN and LLWNN BASED NEUROFUZZY MODEL

The daily Forex data for Dollar to Rupee and Dollar to Yen are considered here as the experimental data. Here the models are forecasting for 1 day, 1 week and 1 month ahead. All the inputs are normalized within a range of [0, 1] using the following formula.

$$X_{norm} = \frac{X_{orig} - X_{min}}{X_{max} - X_{min}} \quad (20)$$

Where X_{norm} → normalized value, X_{orig} → original exchange rate, X_{min} and X_{max} are the daily minimum and maximum prices of the corresponding Forex data.

Details of the FOREX datasets are given in table 1.

Here each Forex dataset is divided into two sets, one for training and one for testing. The duration of the sample, total number of samples, number of training samples and testing samples are given in table 1. Different lagged values are taken as input to the proposed model. The daily, weekly and monthly periodicity and their trend are taken into consideration. However we can also consider other technical indicators as inputs to the models. Here in this paper simple moving average is used as the technical indicator for both the models. Training of the LLWNN and Neurofuzzy models are carried out using the BP algorithm given in Section 4 and the optimum weights are obtained. Then using the trained model, the forecasting performance is tested using test patterns for 1 day, 1 week, and 1 month ahead. The Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE) are used to measure the performance of the proposed models.

The MAPE is defined as

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100 \quad (21)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (22)$$

where y_i and \hat{y}_i are the desired and predicted value respectively. 'N' represents the total number of samples under test.

6. MODEL OUTPUT

The performance of BP based LLWNN and Neurofuzzy model is given in Table 2 and Table 3. Model outputs for various time horizons during training and testing are presented here from fig. 3-16.

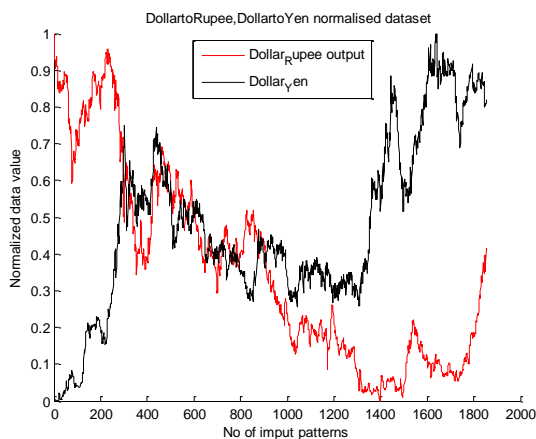


Fig 3: Two Normalized Forex indices: Dollar to Rupee, Dollar to Yen

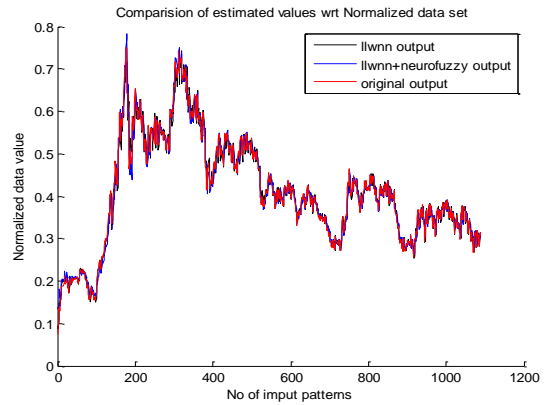


Fig 4: one day ahead prediction during training (Dollar to Rupee)

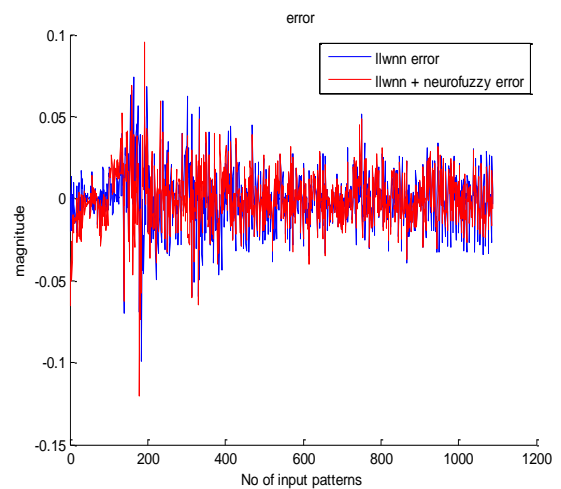


Fig 5: Error during 1 day ahead training (Dollar to Rupee)

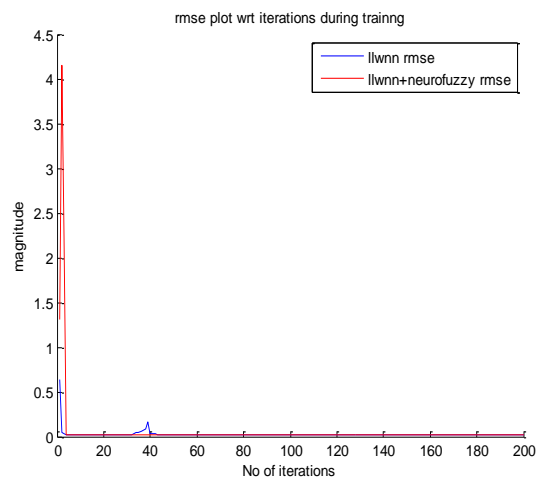
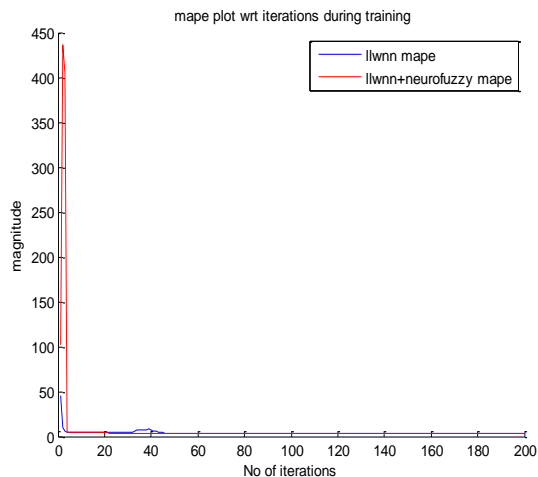
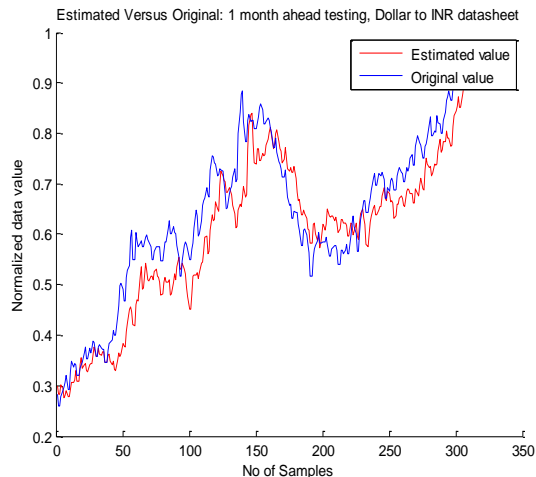


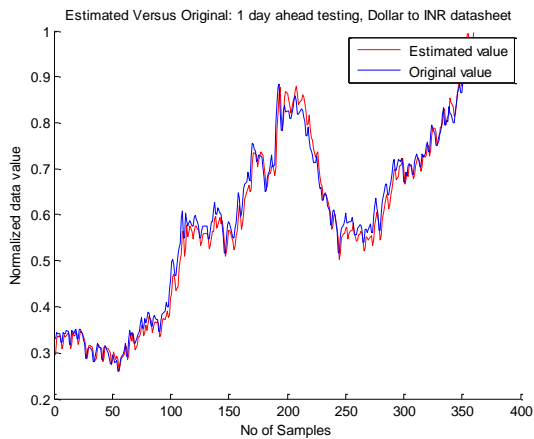
Fig 6: RMSE during 1day ahead training (Dollar to Rupee)



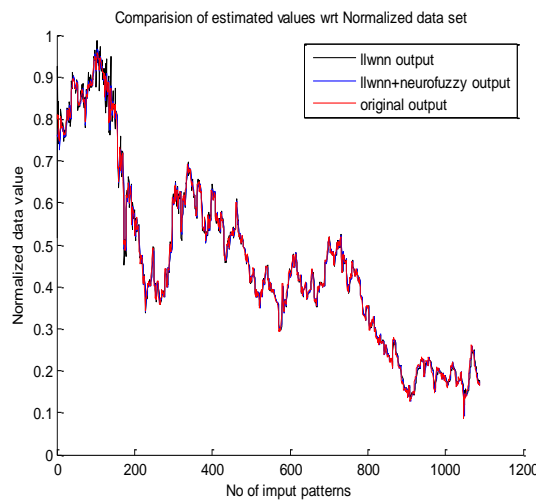
**Fig 7: MAPE during 1day ahead training
 (Dollar to Rupee)**



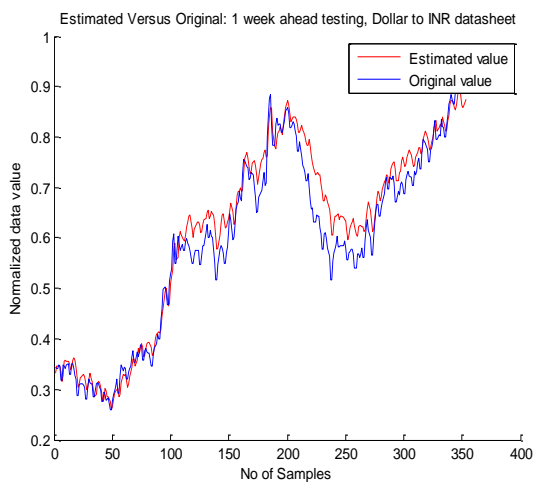
**Fig 10: One month ahead prediction during testing
 of Neurofuzzy Hybrid Model (Dollar to Rupee)**



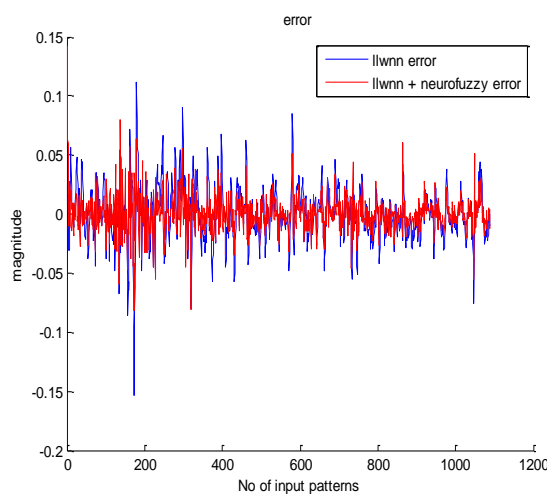
**Fig 8: One day ahead prediction during testing
 of Neurofuzzy Hybrid Model (Dollar to Rupee)**



**Fig 11: One day ahead prediction during training
 (Dollar to Yen)**



**Fig 9: One week ahead prediction during testing
 of Neurofuzzy Hybrid Model (Dollar to Rupee)**



**Fig 12: Error during 1 day ahead training
 (Dollar to Yen)**

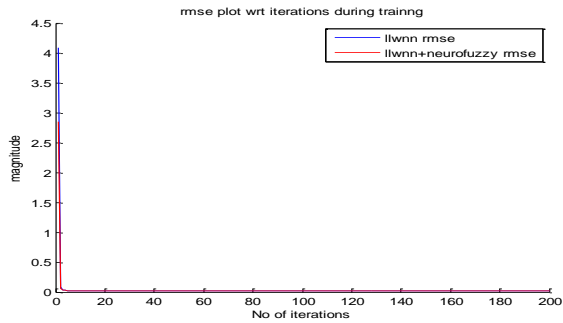


Fig 13: RMSE during 1day ahead during training (Dollar to Yen)

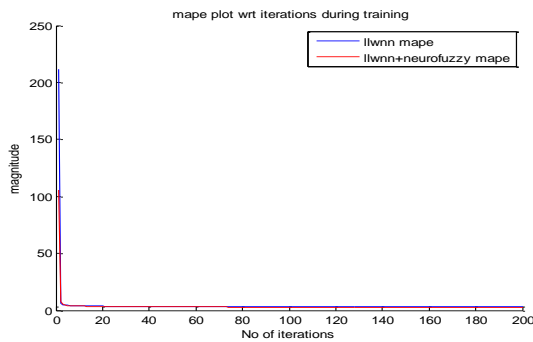


Fig 14: MAPE for 1day ahead during training (Dollar to Yen)

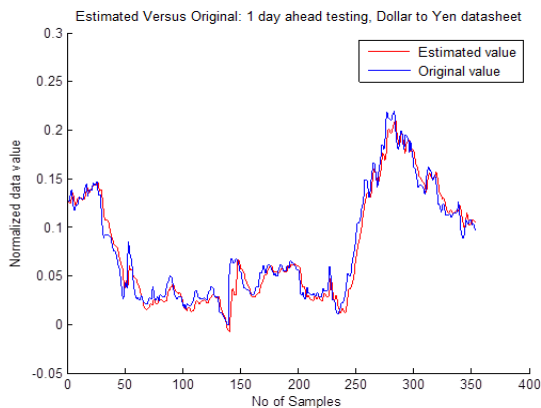


Fig 15: Original vs. predicted for 1day ahead testing of Neurofuzzy hybrid Model (Dollar to Yen)

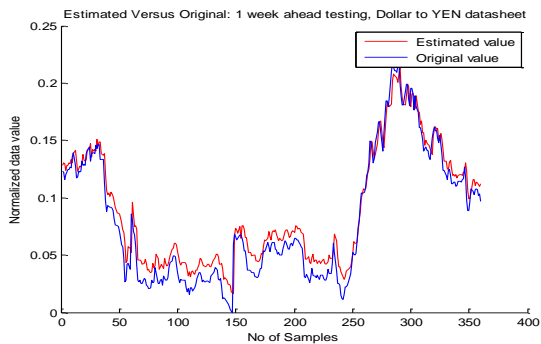


Fig 16: Original vs. predicted for 1 week ahead testing of Neurofuzzy Hybrid Model (Dollar to Yen)

Table 2. Performance of BP based LLWNN model (Training and Testing)

FOREX Data	Duration	RMSE (train)	MAPE (%) (train)	RMSE (test)	MAPE (%) (test)
DOLLAR to RUPEE	1-day	0.0235	3.2125	0.0685	4.284
	1-week	0.0387	4.9467	0.1315	7.86
	1-month	0.0242	3.7232	0.1735	9.8955
DOLLAR to YEN	1-day	0.0379	3.5469	0.1424	6.76
	1-week	0.0405	4.2679	0.1538	9.0047
	1-month	0.0419	4.7473	0.1983	12.586

Table 3. Performance of BP based NEUROFUZZY hybrid model (Training and Testing)

FOREX Data	Duration	RMSE (train)	MAPE (%) (train)	RMSE (test)	MAPE (%) (test)
DOLLAR to RUPEE	1-day	0.0181	2.9114	0.0596	3.758
	1-week	0.0231	3.6816	0.0879	5.6982
	1-month	0.0229	3.1458	0.1208	6.5267
DOLLAR to YEN	1-day	0.0187	2.0098	0.0544	4.1626
	1-week	0.0168	3.1542	0.0813	6.7549
	1-month	0.0134	2.6585	0.0922	8.9672

7. ANALYSIS OF RESULT

The experiment undertaken in this paper has taken into account two models, two data sets and three time horizons. The results are presented in terms of target vs. predicted values and error convergence speed. A comparative analysis of the performance of both the models for 1 day, 1 week and 1 month in advance is presented. The two normalized Forex indices i.e. Dollar to Rupee and Dollar to Yen are shown in Fig. 3. Fig. 4-10 and Fig. 11-16 deals with various plots of Dollar to Rupee and Dollar to Yen datasets respectively. One day ahead training prediction error, RMSE and MAPE plots for both the models for Dollar to Rupee dataset are presented in Fig 4-7 respectively. Fig. 8-10 represents 1 day, 1 week and 1 month ahead testing prediction of proposed Neurofuzzy hybrid model for Dollar to Rupee dataset. Similarly 1 day ahead training prediction error, RMSE and MAPE plots for both the models for Dollar to Yen dataset are presented in Fig 11-14 respectively. Fig. 15-16 represents 1 day and 1 week ahead testing prediction of proposed Neurofuzzy hybrid model for Dollar to Yen dataset. The performance measures RMSE and MAPE for Dollar to Rupee, Dollar to Yen during training and testing of BP based LLWNN model and BP based Neurofuzzy hybrid model during different time horizons are mentioned in Table 2 and 3 respectively. The performance of Neurofuzzy hybrid model is found to be better in comparison to simple LLWNN model.

8. CONCLUSION

Accurate Forex forecasting is always a very challenging task. The proposed Neurofuzzy hybrid model trained with back propagation is giving good result as per the recorded RMSE, and MAPE values during testing for one day, one week and one month ahead respectively in comparison to the simple LLWNN model trained with back propagation. The Neurofuzzy hybrid model is proved to be better in terms of prediction accuracy, error convergence speed and is more capable to handle more uncertainties in comparison to simple LLWNN. Further the prediction performance of LLWNN and Neurofuzzy hybrid model is to be verified with GA, PSO and DE based training algorithms.

9. REFERENCES

- [1] Neely C.J. and Weller P.A.2011.Technical Analysis in the Foreign Exchange Market, Research Division, Federal Reserve Bank of St. Louis, Working Paper Series.
- [2] Box G.E.P.1970.Time Series Analysis,Forecasting and Control.Holden Day,Sanfrancisco.
- [3] So M.K.P, Lam K and Li W.K.1999.Forecasting Exchange rate volatility using autoregressive random variance mode. *Appl Financial Econ* 9:583-591.
- [4] Ballini R., Luna I., Lima L.M.de and Silveria R.L.F. da. 2010. A Comparative analysis of Neurofuzzy, ANN and ARIMA models for Brazilian Stock Index Forecasting, Department of Economic theory, Institute of Economics, University of Campinas, Brazil.
- [5] Dhamija A.K. and Bhalla V.K.2010.Financial Timeseries forecasting: Comparison of neural networks and ARCH models,*International Research Journal of Finance and Economics*,ISSN 1450-2887 issue 49
- [6] Box G.E.P. and Jenkins G.M.1976.Time series analysis :Forecasting and control.Holden-Day.
- [7] Yager R.R.and Zadeh LA(eds).1994.Fuzzy sets,neural networks and soft computing.Van Nostrand Reinhold, New York.
- [8] Kyoung-jae Kim.April 2006.Artificial Neural Networks with evolutionary instance selection for financial forecasting. *Expert Systems with Applications*,30,pp .519-526,.
- [9] Yu Lean,Wang Shouyang and Lai Keung Kin.2005.Mining Stock Market tendency using GA based Support Vector Machine.Springer Verlag Berlin Heidelberg.
- [10] Gudise. V.G. and Venayagamoorthy. G.K.2003. Comparison of particle swarm optimization and BP as training Algorithms for Neural Networks.07803-7914-3/s10.00 IEEE.
- [11] Zhang X., Chen Y. and Yang J.Y.2007.Stock index forecasting using PSO based selective Neural Network Ensemble.*International Conference on Artificial Intelligence*, Vol.1, pp. 260-264.
- [12] Sermpinis G,Theofilatos K, Karathanasopoulos A and Dunis C.2013.Forecasting Foreign exchange rates with adaptive neural networks using radial basis functions and particle swarm optimization, *European Journal of Operational Research*.225 (30 pp.528-540. ISSN 0377-2217)
- [13] R. Storn and K. Price. March 1995.Differential Evolution-A Simple and efficient adaptive scheme for global optimization over continuous spaces, Technical Report TR-95-012,<http://www1.icsi.berkeley.edu/~storn/TR-95-012.pdf>
- [14] Das S and Suganthan P.N. Feb 2011.Differential Evolution- a survey of the State-of-the-art, *IEEE Transaction on Evolutionary Computing*, Vol. 15,No.1, pp-4-31.
- [15] Majhi Ritanjali, Panda G., Majhi B. and Sahoo G.Aug 2009.Efficient prediction of stock market indices using adaptive bacterial foraging optimization (ABFO) and BFO based techniques. *An International Journal Expert Systems with Applications* Volume 36 Issue 6.
- [16] Huang W, Lai K.K, Nakamori Y and Wang S.2004.Forecasting Foreign exchange rates with artificial neural networks:A Review,*International Journal of Information Technology and Decision Making*(3:1),pp 145-165.
- [17] Hornik K.1991.Approximation capabilities of multilayer feedforward networks, *Neural networks*(4:2), pp 251-257
- [18] Saad E.W., Prokhorov D.V. and Wunsch D.C.Nov 1998.Comparative study of stock Trend Prediction using Time delay, Recurrent and Probabilistic Neural Networks,*IEEE Transactions on Neural Networks*, vol.9, No.6.
- [19] Majhi B., Shalabi H. and Fathi Mowafak.2005. FLANN based forecasting of S&P 500 index. *Information Technology Journal*,4(3):289-292,Asian Network for Scientific Information.
- [20] Cao L,Hong Y,Fang H and He G.1995.Predicting Chaotic time series with wavelet Networks.,*Physica D*.85:225-238.
- [21] Yu Lixin and Zhang Y.Q.2005. Evolutionary Fuzzy Neural Networks for Hybrid Financial Prediction. *IEEE Transaction on systems, Man and Cybernetics-part-C. Applications and Reviews* ,vol. 35, No.2.
- [22] Mohapatra P, Raj Alok and Patra T.K.July 2012.Indian Stock Market Prediction using Differential Evolutionary Neural Network Model.,*IJECCT*,Volume 2,Issue 4..
- [23] Majhi R, Panda G and Sahoo G.2009.Development and Performance evaluation of FLANN based model for forecasting stock market. *Expert Systems with Applications*(36)6800-6808.
- [24] Dutta G,Jha P,Laha A.K. and Mohan N.2006.Artificial Neural network Models for forecasting stock price index in the Bombay Stock Exchange, *Journal of Emerging Market Finance* 5 (2006) 3.
- [25] Chakravarty S and Dash P.K.2012.A PSO based Integrated Functional link net and Interval type-2 Fuzzy logic system for predicting stock market indices. *Applied Soft Computing* 12 (2012) 931-941.
- [26] Chang P.-C.,Fan C.-Y and Chen S.-H.2007.Financial time series data forecasting by wavelet and TSK fuzzy rule based system.*Institute of Electrical and Electronics Engineers,Fourth International Conference on Fuzzy systems and Knowledge discovery*, IEEE,0-7695-2874.
- [27] Chokri S.2006.Neurofuzzy Network based on extended Kalman filtering for financial time series.*World Academy of Science, Engineering and Technology*, vol 15,ISSN 1307-6884.