Forecasting Foreign Exchange Rate Using an IT2 FCM based Type-2 Neuro-Fuzzy System

Emad Fallahzadeh[1], Mohammad Ali Montazeri [2]

[1]Dept. of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, Iran, e.fallahzadeh@ec.iut.ac.ir
[2]Dept. of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, Iran, montazeri@cc.iut.ac.ir

Abstract: A hybrid neuro-fuzzy model based on interval type-2 fuzzy c-means clustering, MLP neural network and interval type-2 fuzzy model is proposed for predicting the noisy forex market. To gain faster convergence for learning procedure, combination of back-resilient and back-propagation is used.

Two EURUSD and USDCHF exchange rates from forex market are used for experiments. The model is tested for convergence speed and one day ahead prediction. It is also compared with its fuzzy c-means based type-1 equivalent and a FLANN based neuro-fuzzy system. The performance of proposed model in convergence speed and prediction accuracy is proved by experimental results.

Key words: Neuro-fuzzy system; Interval type-2 fuzzy; IT2 fuzzy c-means

INTRODUCTION

Prediction of exchange rates could be categorized into fundamental and technical analysis methods. The first is done by considering fundamental factors like inflation rates or unemployment rates. But for the second, historical data patterns are used for prediction.

However, because various parameters are forming the market, a trader deals with complex patterns. It is said that the nature of such market is chaotic and noisy. Even in some literatures, a prediction model seems to be like random walk model [1].

Statistical approaches like auto regression integrated moving average (ARIMA) [2] and subsequently adaptive techniques proposed [2], [3]. These methods work in the non stationary and less data situations but using linear structure gives them less forecasting performance. By using soft computing, more hopes came to this area of research [4]. In the last two decades, multilayer artificial neural networks [5], Psi Sigma Neural Network [6], fuzzy logic [7], genetic algorithms [8] and support vector machines [9] have been applied for exchange rates prediction. Also, some comprehensive surveys are made [10], [11] and all are agree in the idea that soft computing methods could handle nonlinear structure of financial markets.

The nature of financial markets needs hybridization of various techniques. Until now, different hybrids like combination of computational intelligence with linear model [12], neuro-fuzzy with genetic algorithm [13], neuro-fuzzy with Kalman filtering [14], neuro-fuzzy with improved PSO [15], neuro-fuzzy with FLANN base network [16] are used. Artificial neural networks are good at learning patterns and adaptation to the variable environments and fuzzy models could handle uncertainty of the systems. So this hybrid is well for this environment.

In this paper, we've proposed a new hybrid model for predicting foreign exchange rates. IT2 fuzzy is used for the fuzzy part of the system. IT2 fuzzy c-means is also used for clustering the data and finding center ranges within each dimension of data and using them for membership functions. Neural part of the system is an MLP network which makes consequent part of rules. For optimizing the parameters of the system, integration of back-resilient and simulated annealing is used.

INTERVAL TYPE-2 FUZZY SYSTEM

As mentioned earlier, fuzzy systems are good at handling uncertainties. A type-1 fuzzy system, assigns a crisp membership for an input value through a type-1 fuzzy set. But a membership function is not always precise enough to giving us crisp values. Sometimes, there would be uncertainties in the membership functions and the type-1 fuzzy system is unable to solve this problem.

So, another kind of fuzzy system has been defined for handling membership function uncertainties and that is called Type-2 fuzzy system [17]. Although, this system solved the problem to high degree, but a newer problem has been raised with this system. The problem is computational complexity of such system would be very high. So, another model proposed to solve this earlier problem and that was interval Type-2 fuzzy system [18] which proposed lower complex model.

The inference system of a fuzzy model is composed of fuzzy rules which relate the inputs and outputs of the system. Following is a typical form of the rule n of a type-2 fuzzy system with one output:

$$\begin{aligned} &IF\ x_1\ is\ \tilde{X}_1^n\ and\ x_2\ is\ \tilde{X}_2^n\ and \cdots and\ x_m\ is\ \tilde{X}_m^n \\ &\quad THEN\ y\ is\ \tilde{Y}^n \end{aligned} \tag{1}$$

(

In the above rule, x_i (i = 1,…,m) represents the inputs to the fuzzy system and y is the output of it. For each input or output in a rule, a membership function is assigned to and here is represented by $\tilde{X}_i^n$ for inputs and by $\tilde{Y}^n$ for output.

We could use type-2 fuzzy sets in both antecedent and consequent sides of the rule or just using it in antecedent part. In [19], [20] some designing forms of type-2 fuzzy systems are mentioned.

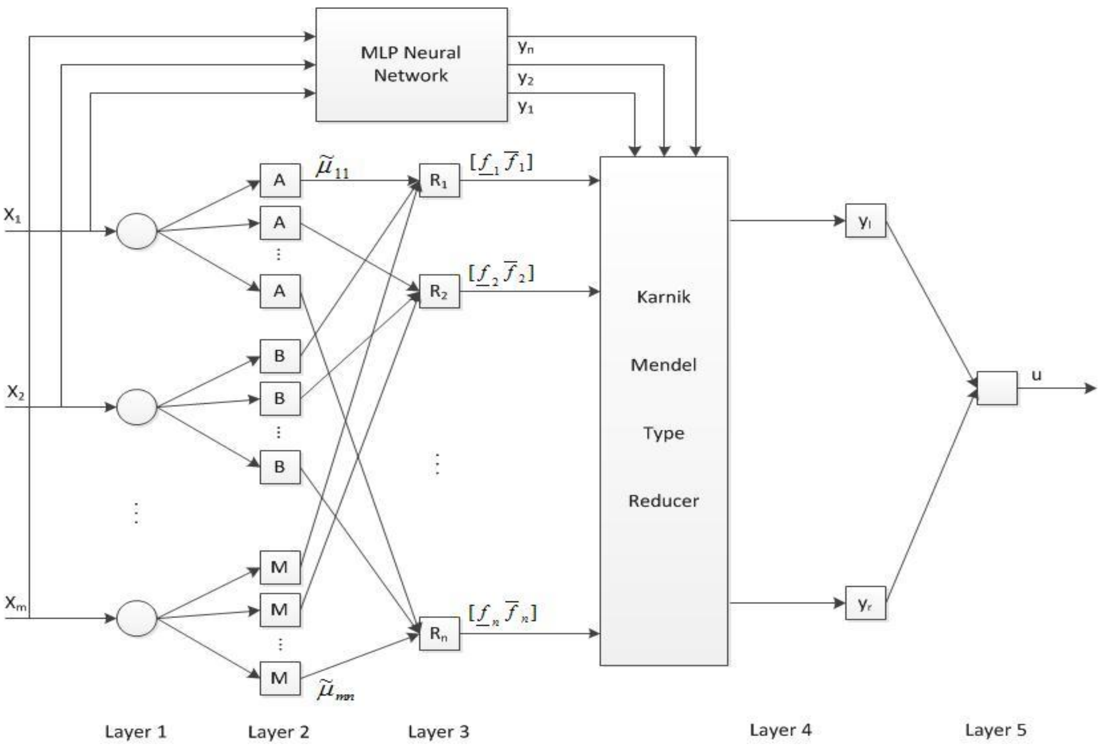


Figure 1. IT2FCM based neuro-fuzzy model

In this paper, interval type-2 fuzzy sets are used for antecedent parts and TSK type is used for consequent parts of rules. Also, parameters of the consequent parts are singletons.

DEVELOPMENT OF PROPOSED MODEL

Development of a neuro-fuzzy system consists of finding the structure of it and optimizing the parameters included in the antecedent and consequent parts of each fuzzy rule.
The structure of the proposed model has been shown in Figure 1 and is described as follows:

Layer 1

Each data point in the dataset consists of three indices. Closing price, %K and %D indices are used as inputs to this model. Layer 1 just distributes these inputs to the system.

Layer 2

For each input, membership degree should be determined. Because the model is IT2 fuzzy, type-2 Fuzzy sets are used and defined with Gaussian functions.

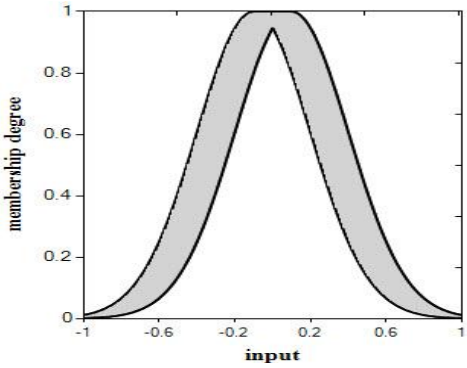


Figure 2. Interval type-2 Gaussian function with uncertain mean

Uncertainty in the Gaussian membership function could be defined with the uncertainty in the mean or in the STD of Gaussian function. Since considering uncertainty for both parameters could cause the parameter space very large, only mean uncertainty is assumed for this model. This kind of membership function could be described by two Gaussian functions and is shown in Figure 2.

An interval is used as the membership value. The lower ($\underline{\mu}_{ij}(x)$) and higher ($\overline{\mu}_{ij}(x)$) memberships from the input i=1,…,m to the rule j=1,…,n are calculated through the mentioned membership function in Figure 2 as follows:

$$\underline{\mu}_{ij}(x) = \begin{cases} Gaussian(c2_{ij}, \sigma_{ij}, x_i) & x_i \le \dfrac{c1_{ij} + c2_{ij}}{2} \\ Gaussian(c1_{ij}, \sigma_{ij}, x_i) & x_i > \dfrac{c1_{ij} + c2_{ij}}{2} \end{cases} \tag{2a}$$

$$\overline{\mu}_{ij}(x) = \begin{cases} Gaussian(c1_{ij}, \sigma_{ij}, x_i) & x_i < c1_{ij} \\ 1 & c1_{ij} \le x_i \le c2_{ij} \\ Gaussian(c2_{ij}, \sigma_{ij}, x_i) & x_i < c2_{ij} \end{cases} \tag{2b}$$

and the Gaussian function is defined as:

$$Gaussian(c_{ij}, \sigma_{ij}, x_i) = \exp\left(-\frac{1}{2}\frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}\right) \tag{3}$$

Where the x_i is the input to the Gaussian function and the c_{ij} and σ_{ij} are the mean and STD of this function. Because we are using Gaussian function with uncertain mean, two version of mean exist; One is the lower mean ($c1_{ij}$) and another is the higher mean ($c2_{ij}$).

Rather than initializing Gaussian function centers randomly, Interval Type-2 Fuzzy c-means is used for determining them in this paper.

IT2 Fuzzy c-means

The original FCM algorithm, is a clustering method which in it, each point in the data set could be assigned to multiple clusters with different degree of memberships [21], [22]. So, Clusters in this technique are not strictly separated.

Calculating the membership degree of each pattern to a cluster is done with considering the distance among each pattern and cluster prototypes. The formulas are as follows:

$$u_j(x_i) = \frac{1}{\sum_{k=1}^{c} \left(d_{ij} / d_{ik} \right)^{2/(m-1)}} \quad (4)$$

$$c_j = \frac{\sum_{i=1}^{n} x_i u_j(x_i)^m}{\sum_{i=1}^{n} u_j(x_i)^m} \quad (5)$$

In these formulas, $u_j(x_i)$ denotes the membership degree of x_i to cluster j. d_{ij} represents the distance of x_i to cluster j's prototype. k is the index for different clusters, c is number of clusters and m, m>1 is the fuzzifier of this method. The bigger value for m causes the more fuzziness for each cluster. c_j denotes the prototype of cluster j and n is number of inputs.

Initializing each cluster prototype is random and the steps of the algorithm is done iteratively in order to reach the point where the accumulate change in cluster prototypes would be below a certain threshold.

This algorithm could be used in type-1 fuzzy models [22]. But in many applications selecting a precise m value is a difficult decision. For solving this problem, another FCM algorithm proposed by Hwang and Rhee which is called IT2 Fuzzy c-means [23]. In this algorithm, IT2 fuzzy logic is used to handle the uncertainty caused by the value of m as a fuzzifier. So, instead of using just one m, an interval [m_1, m_2] is used for the fuzzifier value and the interval membership degree [$\underline{u}_j(x_i)$, $\bar{u}_j(x_i)$] of pattern x_i to the cluster c_j is calculated as follows:

$$\underline{u}_j(x_i) = \min\left(\frac{1}{\sum_{k=1}^{c} \left(d_{ij} / d_{ik} \right)^{2/(m_1-1)}}, \frac{1}{\sum_{k=1}^{c} \left(d_{ij} / d_{ik} \right)^{2/(m_2-1)}} \right) \quad (6)$$

$$\bar{u}_j(x_i) = \max\left(\frac{1}{\sum_{k=1}^{c} \left(d_{ij} / d_{ik} \right)^{2/(m_1-1)}}, \frac{1}{\sum_{k=1}^{c} \left(d_{ij} / d_{ik} \right)^{2/(m_2-1)}} \right) \quad (7)$$

In order to update the cluster prototypes, we should take into the account the intervals which are defined for membership degrees. The formula for calculating a cluster prototype could be defined as follows:

$$\tilde{c}_j = [c_j^L, c_j^R] = \sum_{u(x_1)\in J_{x_1}} \cdots \sum_{u(x_1)\in J_{x_n}} 1 \Big/ \frac{\sum_{i=1}^{n} x_i u_j(x_i)^m}{\sum_{i=1}^{n} u_j(x_i)^m} \quad (8)$$

This is a general formula and the m which is used in it will be switched based on corresponding membership that is used as following: Calculation of each dimension of the prototype will be done separately. First of all we've sorted data points in a dimension with relocating corresponding lower and higher membership degrees beside them. Then we've applied the Karnik-Mendel algorithm [24] to find the lower and higher center for that cluster in the calculated dimension.

The c_j^L and c_j^R centers which have been computed for a cluster are then used as the lower and higher thresholds for the means of corresponding membership function in neuro-fuzzy model. That is, the Gaussian function centers could fluctuate between these values.

Layer 3

In this layer the firing strength for each rule is calculated based on membership values coming from layer 2. There are two common t-norm functions used for this purpose. Min or prod are the choices and in this paper the prod t-norm is selected. It is shown by * operator and for the rule n is computed like the following:

$$\underline{f}_n = \underline{\mu}_{1n} * \underline{\mu}_{2n} * \cdots * \underline{\mu}_{mn} \quad (9)$$

$$\bar{f}_n = \bar{\mu}_{1n} * \bar{\mu}_{2n} * \cdots * \bar{\mu}_{mn} \quad (10)$$

Layer 4

Until now all layers made the antecedent parts of fuzzy rules. The next layers are going to make the consequent part of rules. In the layer 4, the computed output from the neural part of system (y_j) is participated into the fuzzy part. This neural network is a MLP network with tangent hyperbolic as the activation function. Then y_l and y_r for a fuzzy system with N rules are calculated as follows:

$$y_l = \frac{\sum_{j=1}^{L} \bar{f}_j y_j + \sum_{j=L+1}^{N} \underline{f}_j y_j}{\sum_{j=1}^{L} \bar{f}_j + \sum_{j=L+1}^{N} \underline{f}_j} \quad (11)$$

$$y_r = \frac{\sum_{j=1}^{R} \underline{f}_j y_j + \sum_{j=R+1}^{N} \bar{f}_j y_j}{\sum_{j=1}^{R} \underline{f}_j + \sum_{j=R+1}^{N} \bar{f}_j} \quad (12)$$

The algorithm for this purpose is also the Karnik-Mendel [24] which was used earlier for calculating each cluster center. The parameters L and R are computed by this algorithm. This procedure is called type reduction.

Layer 5

The final step of this model is defuzzification, which is computed as follows in order to giving a crisp output:

$$u = \lambda y_l + (1-\lambda) y_r \quad (13)$$

The parameter λ which is used in this formula has values between 0 and 1. We've initialized this value with 0.5 and then submitted it to the learning procedure of the system for optimization.

Optimizing the Parameters

After building the neuro-fuzzy model, it's time to define the optimization method which is used in this network as the learning. First of all, the error function should be defined as target function to be optimized:

$$E=\frac{1}{2}(u_d-u)^2 \tag{14}$$

Where the u_d is the ideal output and u is the actual output of the system.

In order to optimize the parameters, Back-Propagation (BP) algorithm is used for the fuzzy parameters and Back-Resilient (BR) algorithm [25] is used for learning the neural parameters. That's because fuzzy parameters have more restricts and neural parameters have more freedom for changing. Using back-resilient, causes the learning of the network to a faster convergence.

For this purpose, the gradient of the error function should be calculated with respect to each parameter:

$$\begin{aligned}\frac{\partial E}{\partial y_j}&=\frac{\partial E}{\partial u}\frac{\partial u}{\partial y_j}\\&=(u-u_d)\left[\frac{\lambda f_{jL}}{\sum_{l=1}^{n} f_l}+\frac{(1-\lambda)f_{jR}}{\sum_{r=1}^{n} f_r}\right]\end{aligned} \tag{15}$$

This derivation is computed for different outputs (y_j) from the neural part of the system. The parameter f_{jL} is coefficient for the y_j and f_l is coefficient for different outputs of neural network used for calculating y_l. Also, f_{jR} and f_l are coefficients used for computing y_r.

$$\frac{\partial E}{\partial \lambda}=\frac{\partial E}{\partial u}\frac{\partial u}{\partial \lambda}=(u-u_d)(y_l-y_r) \tag{16}$$

$$\frac{\partial E}{\partial \sigma_{ij}}=\sum_j \frac{\partial E}{\partial u}\left(\frac{\partial u}{\partial \underline{f}_j}\frac{\partial \underline{f}_j}{\partial \underline{\mu}_{ij}}\frac{\partial \underline{\mu}_{ij}}{\partial \sigma_{ij}}+\frac{\partial u}{\partial \bar{f}_j}\frac{\partial \bar{f}_j}{\partial \bar{\mu}_{ij}}\frac{\partial \bar{\mu}_{ij}}{\partial \sigma_{ij}}\right) \tag{17}$$

$$\frac{\partial E}{\partial c1_{ij}}=\sum_j \frac{\partial E}{\partial u}\left(\frac{\partial u}{\partial \underline{f}_j}\frac{\partial \underline{f}_j}{\partial \underline{\mu}_{ij}}\frac{\partial \underline{\mu}_{ij}}{\partial c1_{ij}}+\frac{\partial u}{\partial \bar{f}_j}\frac{\partial \bar{f}_j}{\partial \bar{\mu}_{ij}}\frac{\partial \bar{\mu}_{ij}}{\partial c1_{ij}}\right) \tag{18}$$

$$\frac{\partial E}{\partial c2_{ij}}=\sum_j \frac{\partial E}{\partial u}\left(\frac{\partial u}{\partial \underline{f}_j}\frac{\partial \underline{f}_j}{\partial \underline{\mu}_{ij}}\frac{\partial \underline{\mu}_{ij}}{\partial c2_{ij}}+\frac{\partial u}{\partial \bar{f}_j}\frac{\partial \bar{f}_j}{\partial \bar{\mu}_{ij}}\frac{\partial \bar{\mu}_{ij}}{\partial c2_{ij}}\right) \tag{19}$$

$$\frac{\partial E}{\partial u}=u-u_d \tag{20}$$

$$\frac{\partial u}{\partial \underline{f}_j}=\lambda\left[\frac{y_jL_f-L_{fy}}{(L_f)^2}\right]+(1-\lambda)\left[\frac{y_jR_f-R_{fy}}{(R_f)^2}\right] \tag{21}$$

$$L_f=\sum_{j=1}^{L}\bar{f}_j+\sum_{j=L+1}^{N}\underline{f}_j \tag{22}$$

$$L_{fy}=\sum_{j=1}^{L}\bar{f}_jy_j+\sum_{j=L+1}^{N}\underline{f}_jy_j \tag{23}$$

$$R_f=\sum_{j=1}^{R}\underline{f}_j+\sum_{j=R+1}^{N}\bar{f}_j \tag{24}$$

$$R_{fy}=\sum_{j=1}^{R}\underline{f}_jy_j+\sum_{j=R+1}^{N}\bar{f}_jy_j \tag{25}$$

It is important to note that left side of the sum in (21) is only involved when $\underline{f}_j$ is participated in the calculation of y_l. Also, right side of the sum involved when $\underline{f}_j$ is participated in y_r. The $\frac{\partial u}{\partial \underline{f}_j}$ derivation is also calculated like (21) and the only difference is the value of y_j.

$$\frac{\partial \underline{f}_j}{\partial \underline{\mu}_{ij}}=\prod_{\substack{k=1\\k\neq i}}^{m}\underline{\mu}_{kj} \tag{26}$$

$$\frac{\partial \bar{f}_j}{\partial \bar{\mu}_{ij}}=\prod_{\substack{k=1\\k\neq i}}^{m}\bar{\mu}_{kj} \tag{27}$$

The parameters i and m are considered as different inputs into the system. Based on functions defined in (2a) and (2b), other derivations calculated as follows:

$$\frac{\partial \bar{\mu}_{ij}}{\partial c1_{ij}}=\begin{cases}Gaussian(c1_{ij},\sigma_{ij},x_i)\frac{(x_i-c1_{ij})}{\sigma^2_{ij}} & x_i<c1_{ij}\\0 & c1_{ij}\le x_i\le c2_{ij}\\0 & x_i>c2_{ij}\end{cases} \tag{28}$$

$$\frac{\partial \underline{\mu}_{ij}}{\partial c1_{ij}}=\begin{cases}0 & x_i\le\frac{c1_{ij}+c2_{ij}}{2}\\Gaussian(c1_{ij},\sigma_{ij},x_i)\frac{(x_i-c1_{ij})}{\sigma^2_{ij}} & x_i>\frac{c1_{ij}+c2_{ij}}{2}\end{cases} \tag{29}$$

$$\frac{\partial \bar{\mu}_{ij}}{\partial c2_{ij}}=\begin{cases}0 & x_i<c1_{ij}\\0 & c1_{ij}\le x_i\le c2_{ij}\\Gaussian(c2_{ij},\sigma_{ij},x_i)\frac{(x_i-c2_{ij})}{\sigma^2_{ij}} & x_i>c2_{ij}\end{cases} \tag{30}$$

$$\frac{\partial \underline{\mu}_{ij}}{\partial c2_{ij}}=\begin{cases}Gaussian(c1_{ij},\sigma_{ij},x_i)\frac{(x_i-c1_{ij})}{\sigma^2_{ij}} & x_i<\frac{c1_{ij}+c2_{ij}}{2}\\0 & x_i\ge\frac{c1_{ij}+c2_{ij}}{2}\end{cases} \tag{31}$$

$$\frac{\partial \bar{\mu}_{ij}}{\partial \sigma_{ij}}=\begin{cases}Gaussian(c1_{ij},\sigma_{ij},x_i)\frac{(x_i-c1_{ij})^2}{\sigma^3_{ij}} & x_i<c1_{ij}\\0 & c1_{ij}\le x_i\le c2_{ij}\\Gaussian(c2_{ij},\sigma_{ij},x_i)\frac{(x_i-c2_{ij})^2}{\sigma^3_{ij}} & x_i>c2_{ij}\end{cases} \tag{32}$$

$$\frac{\partial \mu_{ij}}{\partial \sigma_{ij}} = \begin{cases} Gaussian(c2_{ij}, \sigma_{ij}, x_i) \frac{(x_i - c2_{ij})^2}{\sigma^3{}_{ij}} & x_i \le \frac{c1_{ij} + c2_{ij}}{2} \\ Gaussian(c1_{ij}, \sigma_{ij}, x_i) \frac{(x_i - c1_{ij})^2}{\sigma^3{}_{ij}} & x_i > \frac{c1_{ij} + c2_{ij}}{2} \end{cases} \quad (33)$$

After calculating the gradients, updating are made to parameters as follows:

$$\lambda(t+1) = \lambda(t) - s\eta \frac{\partial E}{\partial \lambda} \quad (34)$$

$$c1_{ij}(t+1) = c1_{ij}(t) - s\eta \frac{\partial E}{\partial c1_{ij}} \quad (35)$$

$$c2_{ij}(t+1) = c2_{ij}(t) - s\eta \frac{\partial E}{\partial c2_{ij}} \quad (36)$$

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) - s\eta \frac{\partial E}{\partial \sigma_{ij}} \quad (37)$$

Where $s\eta$ is the learning parameter used in back-propagation algorithm. These gradients are related to the fuzzy part of system. Because of constraints exit for updating them, slower learning is used for them.
But as mentioned earlier, to gain faster convergence in the learning of the system, back-resilient technique is used for the neural part. For this purpose, gradients of the neural outputs (y_j) are calculated and then these gradients are submitted to neural network for learning the weights with back-resilient.

EXPRIMENTAL RESULTS

Two forex datasets are used for training and testing experiments and all of them are obtained from Meta trader 5 software. One of the datasets that is used is EURUSD exchange rate and another is USDCHF. %K and %D from Stochastic oscillator with Closing price are used as the inputs to the model. Definition for the Stochastic oscillator is a follows [26]:

$$\%\text{K} = 100\left(\frac{\text{C - L14}}{\text{H14 - L14}}\right) \quad (38)$$

%D = 3-period moving average of %K (39)

C is the most recent closing price; L14 and H14 are the lowest and highest prices traded during previous 14 days period.

The datasets are included with 2000 records, each for one separate day. The data is gathered from 1/4/2005 to 10/12/2012. Eighty percent of these records are used for training and the rest are used as the testing data.

In order to find relatively better number of clusters that could be used for clustering and as the number of rules, a test is made for different number of clusters. The MLP part of neuro-fuzzy system is made with one hidden layer contains 20 neurons. 50 epochs are used for training and the results are shown in Table 1. For comparing the results, Mean Squared Error (MSE) is used.

As can be seen in Table 1, the best result for EURUSD dataset is obtained when 7 rules is used and for USDCHF dataset, best result is achieved for 6 rules. So, for the next experiments these rule numbers are used.

Table 1: Error for different number of rules for datasets

Number of rules	EURUSD MSE	USDCHF MSE
2	0.0142	0.0165
3	0.0358	0.0297
4	0.0032	0.0067
5	0.0705	0.0031
6	0.0061	0.0024
7	0.0021	0.0107
8	0.0039	0.0116

Also, we've compared our proposed system with two other neuro-fuzzy systems. One is the type-1 implementation of made system (T1FCMFNS) which is a combination of fuzzy c-means based type-1 fuzzy system with MLP neural network. Another model is FLIT2FNS [16] and it is an IT2 fuzzy system without clustering combined with FLANN neural network

In order to survey how fast these algorithms could be converging, we've compared them in a convergence test. Only 20 epochs is used for training.

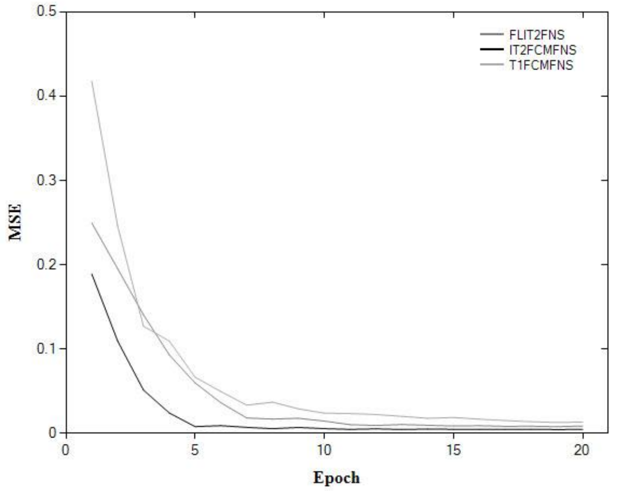
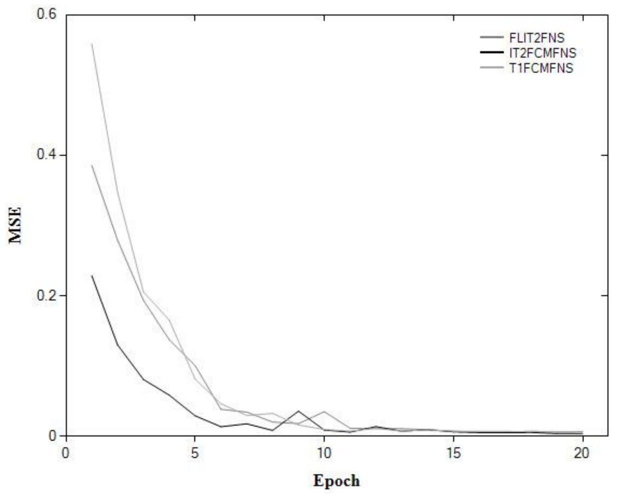


Figure 3. Convergence for EURUSD dataset.



As it is shown in Figure 3 and Figure 4, even with few epochs used for training, the convergence bellow the 0.1 MSE will be occurring under 5 epochs. Also, in compare with other models, better results are obtained from proposed model (IT2FCMFNS).

In the next test, we've surveyed how close would be the results of three algorithms to the target values. For this test 300 epochs are used for both datasets. The results from proposed model are more close to target price than others for both datasets. With IT2FCMFNS, fluctuations and uncertainties are better handled than its type-1 equivalent. MSE in Table 2 is also showing the point.

Figure 4. Convergence for USDCHF dataset.

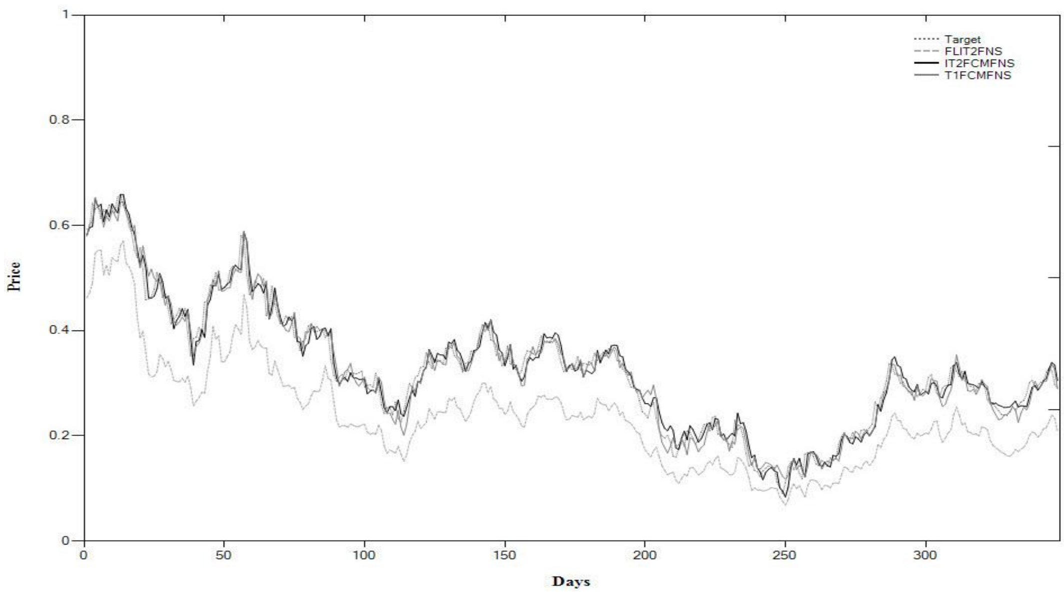


Figure 5. EURUSD one day ahead prediction

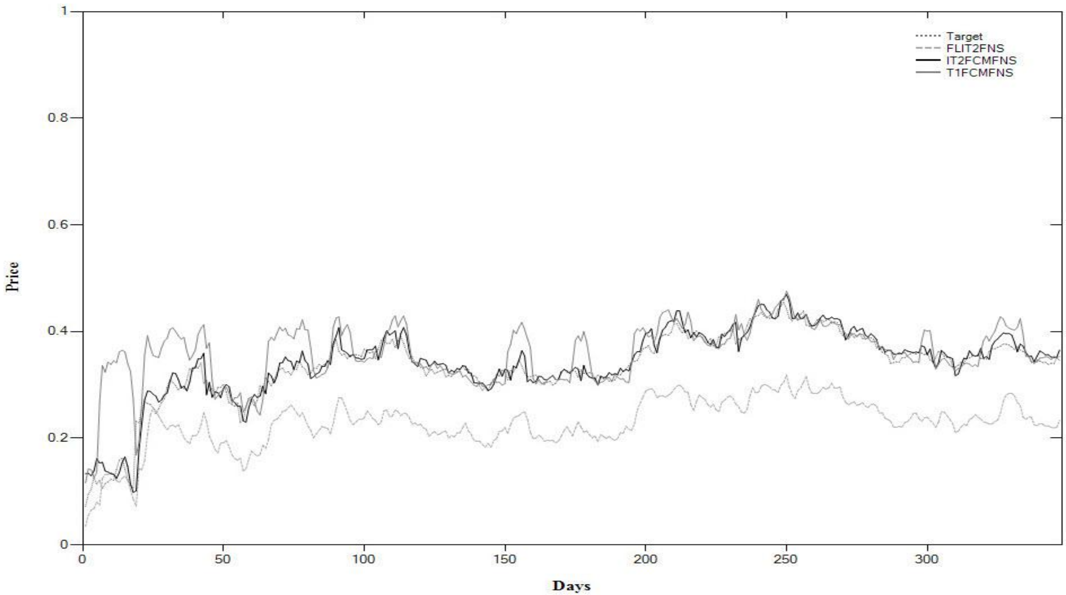


Figure 6. USDCHF one day ahead prediction

Table 2. MSE error for EURUSD and USDCHF

Model / Dataset	FLIT2FNS	IT2FCMFNS	T1FCMFNS
EURUSD	0.03526	0.00167	0.0019
USDCHF	0.04963	0.00121	0.01198

CONCLUSION

A neuro-fuzzy system which combined an IT2FCM based type-2 fuzzy model with a MLP neural network proposed to predict forex market exchange rates. Although this market has a noisy nature, experimental results have shown that the proposed combination could handle the fluctuations to a good degree of accuracy.

REFERENCES

[1] **R. A. Meese and K. Rogoff, "Empirical exchange rate models of the seventies: Do they fit out of sample?,"** ***Journal of international economics,*** **vol. 14, pp. 3-24, 1983.**

[2] **P. GE, "Box., and G. M, Jenkins.,"Time series analysis, forecasting and control"," ed: San Francisco, CA: Holden Day, 1970.**

[3] D. C. Montgomery, L. A. Johnson, and J. S. Gardiner, *Forecasting and time series analysis*: McGraw-Hill New York etc., 1990.

[4] M. D. D. Evans, *Exchange-rate dynamics*: Princeton University Press, 2011.

[5] J. Kingdon, *Intelligent systems and financial forecasting*: Springer Verlag, 1997.

[6] G. Sermpinis, C. Dunis, J. Laws, and C. Stasinakis, "Forecasting and Trading the EUR/USD Exchange Rate with Stochastic Neural Network Combination and Time-Varying Leverage," *Decision Support Systems,* 2012.

[7] Y. Ju, C. Kim, and J. Shim, "Genetic-based fuzzy models: interest rate forecasting problem," *Computers & industrial engineering,* vol. 33, pp. 561-564, 1997.

[8] S. Bhattacharya and K. Meheta, "Staged learning of trading rules using genetic algorithm," in *Proceedings of 3rd INFORMS, conference on info. sys. and tech*, 1998, pp. 62-66.

[9] W. Huang, Y. Nakamori, and S. Y. Wang, "Forecasting stock market movement direction with support vector machine," *Computers & Operations Research,* vol. 32, pp. 2513-2522, 2005.

[10] M. Li and S. Lin, "Several Problems of Exchange Rate Forecasting Using Neural Network," in *International Symposium on Intelligent Information Systems and Applications*, 2009.

[11] Y. Li and W. Ma, "Applications of Artificial Neural Networks in Financial Economics: A Survey," in *Computational Intelligence and Design (ISCID), 2010 International Symposium on*, 2010, pp. 211-214.

[12] T. Imam, K. Tickle, A. Ahmed, and W. Guo, "Linear Relationship Between The AUD/USD Exchange Rate And The Respective Stock Market Indices: A Computational Finance Perspective," *Intelligent Systems in Accounting, Finance and Management,* vol. 19, pp. 19-42, 2012.

[13] L. Yu and Y. Q. Zhang, "Evolutionary fuzzy neural networks for hybrid financial prediction," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on,* vol. 35, pp. 244-249, 2005.

[14] C. Slim, "Neuro-fuzzy network based on extended Kalman filtering for financial time series," *World Academy of Science, Engineering and Technology,* vol. 22, pp. 134-139, 2006.

[15] H. Fu-yuan, "Integration of an Improved Particle Swarm Algorithm and Fuzzy Neural Network for Shanghai Stock Market Prediction," in *Power Electronics and Intelligent Transportation System, 2008. PEITS'08. Workshop on*, 2008, pp. 242-247.

[16] S. Chakravarty and P. K. Dash, "A PSO based integrated functional link net and interval type-2 fuzzy logic system for predicting stock market indices," *Applied Soft Computing,* 2011.

[17] O. Castillo, P. Melin, J. Kacprzyk, and W. Pedrycz, "Type-2 Fuzzy Logic: Theory and Applications," in *Granular Computing, 2007. GRC 2007. IEEE International Conference on*, 2007, pp. 145-145.

[18] J. M. Mendel, R. I. John, and F. Liu, "Interval Type-2 Fuzzy Logic Systems Made Simple," *Fuzzy Systems, IEEE Transactions on,* vol. 14, pp. 808-821, 2006.

[19] N. N. Karnik, J. M. Mendel, and Q. Liang, "Type-2 fuzzy logic systems," *Fuzzy Systems, IEEE Transactions on,* vol. 7, pp. 643-658, 1999.

[20] J. M. Mendel, "Computing derivatives in interval type-2 fuzzy logic systems," *Fuzzy Systems, IEEE Transactions on,* vol. 12, pp. 84-98, 2004.

[21] J. Bezdek, "Fuzzy mathematics in pattern recognition," Ph. D. Thesis, Applied Mathematics Center, Cornell University, 1973.

[22] J. C. Bezdek, *Pattern recognition with fuzzy objective function algorithms*: Kluwer Academic Publishers, 1981.

[23] C. Hwang and F. C. H. Rhee, "Uncertain fuzzy clustering: interval type-2 fuzzy approach to c-means," *Fuzzy Systems, IEEE Transactions on,* vol. 15, pp. 107-120, 2007.

[24] N. N. Karnik and J. M. Mendel, "Centroid of a type-2 fuzzy set," *Information Sciences,* vol. 132, pp. 195-220, 2001.

[25] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The RPROP algorithm," in *Neural Networks, 1993., IEEE International Conference on*, 1993, pp. 586-591.

[26] J. J. Murphy, *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications*: Prentice Hall Press, 1999.