

Times Series Forecasting using Chebyshev Functions based Locally Recurrent neuro-Fuzzy Information System

A.K. Parida¹, R. Bisoi², P.K. Dash^{2*}, S. Mishra³

¹*Orissa Engineering College, Bhubaneswar, Odisha, India*
Email: erakparida@gmail.com

²*Multidisciplinary Research Cell, SOA University, Bhubaneswar, Odisha, India,*
²*E-mail: ranjeeta.bisoi@gmail.com*
^{2*}*E-mail: dashpk13@yahoo.com*

Indira Gandhi Institute of Technology, Sarang, Odisha, India
³*E-mail: sasmita.mishra.csea@gmail.com*

Received 10 May 2016

Accepted 30 October 2016

Abstract

The model proposed in this paper, is a hybridization of fuzzy neural network (FNN) and a functional link neural system for time series data prediction. The TSK-type feedforward fuzzy neural network does not take the full advantage of the use of the fuzzy rule base in accurate input-output mapping and hence a hybrid model is developed using the Chebyshev polynomial functions to construct the consequent part of the fuzzy rules. The model to be known as locally recurrent neuro fuzzy information system (LRNFIS) is used to provide an expanded nonlinear transformation to the input space thereby increasing its dimension which will be adequate to capture the nonlinearities and chaotic variations in the time series. The locally recurrent nodes will provide feedback connections between outputs and inputs allowing signal flow in both forward and backward directions, giving the network a dynamic memory useful to mimic dynamic systems. For training the proposed LRNFIS, an improved firefly-harmony search (IFFHS) learning algorithm is used to estimate the parameters of the consequent part and feedback loop parameters. Three real world time series databases like the electricity price of PJM electricity market, the widely studied currency exchange rates between US Dollar (USD) and other four currencies i.e. Australian Dollar (AUD), Swiss Franc (CHF), Mexican Peso (MXN), Brazilian Real (BRL), along with S&P 500 and Nikkei 225 stock market data are used for performance validation of the newly proposed LRNFIS.

Keywords: Recurrent neuro-fuzzy network, Chebyshev polynomials, TSK fuzzy rules, Firefly-Harmony search algorithm, electricity price forecasting, currency exchange rate prediction, stock indices prediction

1. Introduction

List of important symbols

μ	membership function
c, σ	centre and spread of the fuzzy set
λ	Locally recurrent feedback parameter

ω, ϕ	weight and chebyshev polynomial functions, respectively
r_{ij}	Cartesian distance between i th and j th fireflies
β	attractiveness value between two fireflies
α	random movement factor
k, K	iteration count, maximum number of

	iterations, respectively
HM	harmony memory
HMCR	harmony memory consideration rate
PAR	pitch adjustment rate
BW	pitch adjustment bandwidth
U,L	upper and lower bound of the variables
rand	random number between 0 and 1

Time series forecasting has a wide range of applications in finance, signal processing, electric load forecasting, weather forecast, hydrological prediction, sunspot prediction, condition monitoring of equipments for alarm processing, and electricity price prediction in energy markets, etc. Most of the time series data that occur in financial and engineering fields are chaotic in nature and exhibit nonstationary behavior amounting to chaotic fluctuations. Significant research efforts have been undertaken to mine these highly chaotic time series databases in the way of either forecasting or classifying patterns in the database.

Many techniques have been employed over the years for time series forecasting purpose, including statistical methods like Autoregressive Moving Average (ARMA), Autoregressive Integrated Moving Average (ARIMA), and Generalized Autoregressive Conditional Heteroskedasticity (GARCH) models [1-5], intelligence techniques like artificial neural networks (ANNs) [6-11], fuzzy inference system (FIS) [12-15], and support vector machines (SVMs) [16-20], etc. Moreover, time series models like ARIMA, and GARCH, have also been proven to be effective in the stock and electricity price forecasting/modeling. However, for short-term forecasts, Artificial Neural Networks (ANNs) are flexible computing frameworks and universal approximators that can be applied to a wide range of time series forecasting problems with a high degree of accuracy. According to a recent study, the machine learning methods (ANNs, SVMs) outperform other time series methods (ARMA, GARCH) due to their superior handling capacity of the chaotic behavior of the time series databases. Due to the generalization capabilities and robustness, the ANN is extensively used in various time series forecasting.

Among other neural network models; Radial Basis Function Neural Network (RBFNN) [21-23], Local Linear Radial Basis Function Neural Network (LLRBFNN) [24], Wavelet Neural Network (WNN)

[25], Local Linear Wavelet Neural Network (LLWNN) [26-27], Recurrent Neural Network (RNN) [28-29], are widely used in time series prediction. Further to meet the increasing needs for better forecasting models; several nonlinear models have been developed by researchers. Fuzzy Logic theory [30-34] has found to be suitable in many applications as these methods are suitable due to their superior abilities in the handling of non probabilistic uncertainties that occur in financial and energy markets. Adaptive Neural Network based Fuzzy Information Systems (ANFIS) [35-39] and Fuzzy Neural Networks (FNN) work successfully in many time series forecasting problems.

Time series forecasting accuracy is an important yet often difficult task faced by the researchers in many real world problems. Numerous theoretical and empirical studies have confirmed that hybridized models can be an effective way for improving the accuracy of the forecast. Each model can capture certain characteristics hidden in the data and contribute effectively in improving the overall forecasting performance. A single model may not be able to capture all the uncertainties in the time series and therefore may not be able to predict the futuristic behavior of the time series accurately. Thus the hybridization of statistical models like ARMA, ARIMA or GARCH with feedforward neural networks [40-41] is used widely for time series forecasting especially in financial and energy price time series databases. In recent years, more hybrid forecasting models have been proposed using feed-forward neural networks and applied in many areas with good prediction performance. Among the various hybrid models, like the Fuzzy information systems with ARIMA [42-43], Fuzzy wavelet neural networks with GARCH [40], etc. have been successfully applied in time series prediction.

Although there are several studies that involve fuzzy inference systems, and neuro-fuzzy inference systems for financial time series prediction, there is hardly any substantial research in the application of recurrent neuro-fuzzy inference systems for predicting highly chaotic nonlinear times series like financial or electricity market data. Normally the recurrent structure in the network is introduced either in the firing strength layer or in output and input layers and so on. Thus in this research, a new locally recurrent neuro-fuzzy inference System (LRNFIS) is presented that includes a feedback loop in the firing strength layer to memorize

the temporal nature of the fuzzy rule base and provide a dynamic nature to the time series forecasting problem.

In the widely used neuro-fuzzy information systems, the consequent part of the fuzzy rule base is of the TSK-type that has more free parameters to adjust providing a better input-output mapping. However, the TSK-type feedforward recurrent fuzzy neural network does not take the full advantage of the use of the fuzzy rule base in approximating functions and therefore a hybridization procedure is adopted by using Chebyshev polynomial functions to construct the consequent part. The Chebyshev polynomial function model will provide an expanded nonlinear transformation to the input space thereby increasing its dimension which will be adequate to capture the nonlinearities and chaotic variations in time series prediction problems. Further hybridizing it with recurrent fuzzy neural information system will result in a significant convergence and accuracy in time series prediction. It is well known that the weight parameters of the fuzzy neural information systems are normally trained by gradient descent, recursive least squares, extended and unscented Kalman filters and most of these algorithms suffer from slow convergence and high computational overhead. To circumvent these problems metaheuristic evolutionary algorithms like Firefly [44-46] and Harmony search [47-50] are hybridized to estimate the parameters of the weights associated with TSK Fuzzy rule base and locally recurrent feedback loops. The hybridization is done primarily to overcome the local search problems associated with the conventional Firefly algorithm and the improved hybrid version (IFFHS) provides better diversity, speed of convergence, and forecasting accuracy. The proposed LRNFIS model is compared with the well known RBFNN trained by the same IFFHS metaheuristic evolutionary algorithm for the prediction of PJM electricity market prices time series. Further, financial time series databases like currency exchange rate time series and stock time series are taken for prediction over a time frame varying from one day to several days ahead to prove the validity of the proposed LRNFIS model.

The paper is organized in 7 sections. Section 2 presents the TSK fuzzy rule based LRNFIS model and its training algorithm comprising the IFFHS algorithm is presented in section 3. Section 4 provides performance matrix whereas section 5 presents applications to predict the PJM electricity market,

AUD/USD, CHF/USD, MXN/USD, and BRL/USD currency exchange rate, and S&P 500 and Nikkei 225 stock market time series databases. Finally conclusion is presented in section-6.

2. TSK fuzzy rule based LRNFIS model used for Time Series prediction

This section presents the structure of the LRNFIS where the firing strength of each rule is locally fed back to itself creating a locally recurrent fuzzy system. The locally recurrent structure of the LRNFIS provides memory to the learning paradigm thereby capturing the dynamic behavior of the highly fluctuating time series databases. The consequent part of each recurrent fuzzy rule has a first order TSK-type built around Chebyshev polynomial basis functions. The LRNFIS comprises seven layers and is depicted in Fig.1. Different layers of the LRNFIS perform different mathematical functions which are described below:

Layer 1 (Input Layer):

This layer contains the input data comprising the past values of the time series are directly transmitted to the next layer which is a fuzzification layer.

Layer 2 (Fuzzification Layer):

In this layer the input data is fuzzified using a suitable Gaussian membership function for the fuzzy set A_{ij} as:

$$\mu_{ij}(x_j, c_{ij}, \sigma_{ij}) = \left\{ \exp(-.5(x_j - c_{ij})^2 / \sigma_{ij}^2) \right\} \quad (1)$$

$i = 1, 2, \dots, n$, and $j = 1, 2, \dots, m$

where c_{ij} and σ_{ij} are the centre and spread of the j th membership function with regards to i th input, respectively.

Layer 3 (Spatial Firing Layer):

Here each node functions as a spatial rule node corresponding to each TSK type fuzzy rule, which is described in the following form:

$$\begin{aligned} &\text{If } x_1 \text{ is } A_1, \wedge \dots \wedge x_n \text{ is } A_n \\ \text{Rule: } &\text{Then } y = f(x_1, x_2, \dots, x_n) \end{aligned} \quad (2)$$

where x_1, x_2, \dots, x_n are the input variables and A_1, A_2, \dots, A_n are the fuzzy sets represented by the membership functions $\mu A_j(x_i)$ and y is output from the consequent part. The function $f(\cdot)$ is chosen to comprise a nonlinear functional expansion of the input variables and therefore, the fuzzy rule base takes the form

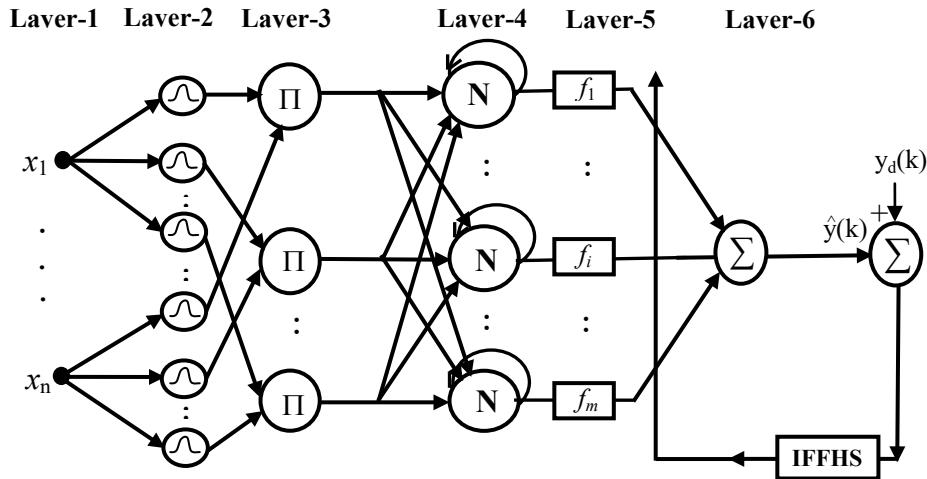


Fig.1. Locally recurrent neuro-fuzzy information system (LRNFIS)

$$R_j : \text{If } x_1 \text{ is } A_{j1}, \wedge x_2 \text{ is } A_{j2}, \dots, \\ x_i \text{ is } A_{ji}, \dots, x_n \text{ is } A_{jn} \quad (3)$$

$$\text{Then } y_j = \beta_j f_j(x_1, x_2, \dots, x_n)$$

where the A_{ji} is the j th fuzzy set associated with the i th input.

Thus, the firing strength of the j th fuzzy rule is obtained with the help of product T-norm of the antecedent fuzzy sets as

$$\mu_j(x_i) = \prod_{i=1}^n \exp\left\{-.5(x_i - c_{ij})^2 / \sigma_{ij}^2\right\} \quad (4)$$

Layer 4 (Normalization and locally recurrent layer)

Each node output in this layer is a normalized temporal firing strength, which is obtained from the ratio of the j th rule's temporal firing strength to the sum of all rules' temporal firing strengths as follows:

$$u_j = \mu_j / \sum_{i=1}^m \mu_j \quad (5)$$

where m is the total number of fuzzy rules used in the rule base. This layer contains the locally recurrent rule node providing dynamic nature to the network structure and its output is a temporal firing strength O_j which depends not only on the current spatial firing strength u_j , but also on the one step delayed temporal firing strength O_j^{t-1} . At any instant ' t ' the temporal firing strength of the recurrent rule is described as a linear combination of the current spatial firing strength and the one step delayed temporal firing strength in the following way:

$$O_j^t = \lambda_j \times u_j + (1 - \lambda_j) \times O_j^{t-1} \quad (6)$$

where $0 \leq \lambda_j \leq 1$ represents the internal recurrent feedback parameters. These parameters control the contribution ratio of current spatial and delayed temporal firing strength to the final output from this layer towards the network outputs.

Layer 5 (Consequent Layer):

The consequent layer is a layer of adaptive nodes which comprises the TSK rule base using Chebyshev polynomial basis function and its output is obtained as

$$f_j = \tanh(\omega_{j0}\phi_0 + \omega_{j1}\phi_1 + \dots + \omega_{jM}\phi_M) \quad (7)$$

and $M = 3n$. Further the lower and higher order Chebyshev Polynomials are given in equations (8) and (9) as

$$c_1(x_i) = x, \quad c_2(x_i) = 2x^2 - 1 \quad (8)$$

$$c_3(x_i) = 4x^3 - 3x, \quad c_4(x_i) = 8x^4 - 8x^2 + 1$$

The higher order Chebyshev Polynomials can be generated using the generalized recursive formula given in equation (9)

$$c_{r+1} = 2 * x * c_r(x) - c_{r-1}(x_i) \quad (9)$$

Thus comparing equations (8) and (9)

$$\phi_0 = 1, \quad \phi_1 = x_1, \quad \phi_2 = 2x_1^2 - 1, \quad \phi_3 = 4x_1^3 - 3x_1$$

$$\phi_{M-2} = x_n, \quad \phi_{M-1} = 2x_n^2 - 1, \quad \phi_M = 4x_n^3 - 3x_n \quad (10)$$

Layer 6 (Output Layer):

In this layer, the overall output of the LRNFIS is calculated as the summation of the rule consequences as follows:

$$\hat{y} = \sum_{j=1}^m O_j^t f_j \quad (11)$$

The performance of the proposed LRNFIS model mainly depends on fine tuning of the antecedent, consequence, internal feedback weights for highly fluctuating time series databases.

3. Training Algorithm for the LRNFIS

In this paper, a hybrid learning paradigm is used for LRNFIS to obtain the parameters of the fuzzy sets. A clustering approach is used to initialize the centre and spread (c_{ij} and σ_{ij}) of the fuzzy membership values after they are updated using a gradient descent algorithm. However, for updating the weights associated with the local recurrent outputs and the consequent parts of the TSK fuzzy rules; the first step is known as the structure learning and the second step is designated as the weight parameter learning.

Step 1: Structure Learning

To obtain the updated fuzzy membership parameters, an error cost function $E(k)$ is formulated as

$$E(k) = \frac{1}{2} (y_d(k) - \hat{y}(k))^2 \quad (12)$$

Using the above error cost function and gradient descent rule the parameter adjustment is done as follows:

$$c_{ij}(k+1) = c_{ij}(k) - \eta e(k) \frac{\partial E}{\partial c_{ij}} \quad (13)$$

$$\sigma_{ij}(k+1) = \sigma_{ij}(k) - \eta e(k) \frac{\partial E}{\partial \sigma_{ij}} \quad (14)$$

On simplification we obtain

$$\frac{\partial E}{\partial c_{ij}} = (\hat{y} - y_d) \frac{(y_j - \hat{y})}{\sum_j \mu_j} \mu_j(x_i) \frac{2(x_i - c_{ij})}{\sigma_{ij}^2} \quad (15)$$

$$\frac{\partial E}{\partial \sigma_{ij}} = (\hat{y} - y_d) \frac{(y_j - \hat{y})}{\sum_j \mu_j} \mu_j(x_i) \frac{2(x_i - c_{ij})^2}{\sigma_{ij}^3} \quad (16)$$

Step 2: TSK and locally recurrent feedback loop weight parameter learning

It is well known that the evolutionary algorithms yield global optimized solutions for many multimodal real

world problems and hence this paper presents a hybrid Firefly and differential Harmony search algorithm for parameter tuning of the TSK type fuzzy rule base. Firefly algorithm (FA) originally proposed by Xin-She Yang and others [44-46] is a metaheuristic algorithm which is based on the flashing behavior of the fireflies. The flashing light is produced by a chemical reaction known as the bioluminescence, and has a short and rhythmic characteristic. The flashing light of the fireflies is considered as a signaling system to attract other fireflies and warn the predators. As fireflies are unisex one firefly can be attracted to the other one depending on its brightness, which is measure of higher value of the objective function. Despite being an effective optimization tool, FA proposed by Yang, [45] has difficulty of randomization between global solution and local search and thus an improved Harmony search (HS) [47-50] is adopted with conventional FA to overcome the drawback of randomness by increasing diversity in firefly population. The HS algorithm is a relatively new metaheuristic search algorithm originally conceptualized from musical instruments which depends on the operators like the harmony memory (HM) and its size (HMS), the harmony memory consideration rate (HMCR), the pitch adjustment rate (PAR), and the pitch adjustment bandwidth (BW). Further, the value of each decision variable in HS optimization algorithm is determined by the three parameters HMCR, PAR, and BW. In this paper, the error cost function in equation (17) is chosen as the objective function for initiating the search process.

The parameters of the model have been estimated by minimizing the root mean square error (RMSE) criterion. Therefore, the objective function for optimization is defined as:

$$E(x) = \sqrt{\sum_{i=1}^N \left(y_i - \hat{y}_i \right)^2} \quad (17)$$

where x is the parameter set that includes the weights of the feedback loops used in the network. y_i is the desired time series value and \hat{y}_i is its predicted value; N is the no. of data points. The objective function is a highly nonlinear function of x .

Step 3: Firefly Algorithm (FF) for Weight parameter estimation of the LRNFIS

Initially ' N ' no. of fireflies are declared for ' D ' dimensions randomly. The number of dimensional field

is the number of decision variables, which is again the number of TSK Legendre basis functions associated parameters to be estimated along with the recurrent path gain parameters. The randomized range of these decision variables is assumed and the final expression of solution vector (X) is:

$$[X]_{N \times D} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1D} \\ x_{21} & x_{22} & \cdots & x_{2D} \\ \vdots & \vdots & & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{ND} \end{bmatrix} \quad (18)$$

The attractiveness value β between any two fireflies i and j varies with the Cartesian distance r_{ij} between them and is obtained as

$$\beta = \beta_0 e^{-\gamma r_{ij}^2} \quad (19)$$

$$\text{where } r_{ij} = \sqrt{\sum_{n=1}^D (x_{ik} - x_{jk})^2} \quad (20)$$

and β_0 is the attractiveness at distance between any two fireflies $r_{ij} = 0$. The range of ' β ' is considered from 0 to 1. The γ is the absorption co-efficient and it varies within a range of 0 to 1; k is the iteration count. Here γ is set to a value 1. For conventional FF algorithm the next iteration gains are modified based on one firefly ' i ' moving towards another firefly ' j ' by using the expression:

$$x_{i,k} = x_{i,k} + \beta(x_{j,k} - x_{i,k}) + \alpha(\text{rand}_{i,k} - 0.5) \quad (21)$$

where α is a random movement factor and its higher value points to global search, whereas local search is performed for lower value of α . Usually α is chosen between 0 and 1. However, in this paper α is varied in each iteration as:

$$\alpha^{k+1} = \left(\frac{1}{2k_{\max}} \right)^{1/k_{\max}} \alpha^k \quad (22)$$

According to the concept of global optima, the fireflies with minimum light intensity are opted as best fireflies (x_{gbest}) for that particular generation. Instead of comparing each firefly with other fireflies, it is more convenient to compare it with best firefly to reduce computational complexity. Thus the Eq.(17) is modified by using the global optimization concept as:

$$x_{i,k} = x_{i,k} + \beta_{best} * (x_{gbest,k} - x_{i,k}) + \alpha * (\text{rand}_{i,k} - 0.5) \quad (23)$$

where best attractiveness (β_{best}) is expressed as:

$$\beta_{best} = (\beta_{\max} - \beta_{\min}) e^{-\gamma r_{igbest}^2} + \beta_{\min} \quad (24)$$

Here β_{\max} and β_{\min} are maximum and minimum attractiveness for that generation. The best distance (r_{igbest}) is obtained from equation (25) as

$$r_{igbest} = \sqrt{\sum_{k=1}^D (x_{ik} - x_{gbest,k})^2} \quad (25)$$

Step 4: Improved Harmony Search algorithm (IFFHS) for better exploration

In the Firefly algorithm, the fireflies having low brightness resulting from local search operation cannot compete with best fireflies which move towards global solutions and therefore are discarded. Thus, all fireflies that do not achieve exploitability with best fireflies are not considered while updating solution vector (X) for the next iteration. Due to this property conventional FA may give erroneous local solutions, and hence an improved harmony search algorithm is considered for those firefly populations to improve their performance by using mutation process. The harmony search approach uses the philosophy of music improvisation, where musician endeavor his musical tune improvisation to achieve optimal harmony. To hybridize the conventional FA, HM vector is the initially declared random position of the fireflies.

$$HM = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1D} \\ x_{21} & x_{22} & \cdots & x_{2D} \\ \vdots & \vdots & & \vdots \\ x_{HMS1} & x_{HMS2} & \cdots & x_{HMSD} \end{bmatrix} \quad (26)$$

where harmony memory size (HMS) is the number of fireflies declared in the firefly algorithm. The rate at which one value is chosen from the HM is known as harmony memory consideration rate ($HMCR$) and varies within $0 < HMCR < 1$. If $HMCR$ is too small the selection of a few best harmonies results in a slow convergence of the process, however, if it is too high then the remaining harmonies will not be explored well. Pitch adjustment band width (BW) and pitch adjustment rate (PAR) play a

vital role in deciding the convergence speed of algorithm. The PAR expresses the probability of moving from the existing harmony to a new harmony and a high value of PAR results in a solution to scatter around an optimum solution. Typical values of PAR lie in range of 0.4-0.75. The proposed improved harmony search has exponentially decreased ‘BW’ and linearly increased ‘PAR’ whereas in conventional harmony search algorithm ‘BW’ and ‘PAR’ are considered as fixed valued. The following steps are covered to implement improved harmony search to the existing FA:

To find the improved solution globally as well as locally, *HMCR* and *PAR* are the two responsible parameters. For conventional HS, more number of iterations are required to find optimal solution, due to consideration of fixed valued *PAR* and its *BW*. For this improved HS, during improvisation of new harmony, both *PAR* and *HMCR* are expressed as linear incremental function and *BW* is obtained as exponentially decreasing function.

The *PAR* is linearly varied by using the equation:

$$PAR(k) = PAR_{\min} + (PAR_{\max} - PAR_{\min}) \times (1 - k / K) \quad (27)$$

In a similar way the *HMCR* is varied as

$$HMCR(k) = HMCR_{\min} (HMCR_{\max} - HMCR_{\min}) \times (k / K) \quad (28)$$

and the band width is exponentially decreased by using the equation:

$$BW(k) = BW_{\max} \times \exp \left(\frac{\log \left(\frac{BW_{\min}}{BW_{\max}} \right)}{K} \right) \times k \quad (29)$$

where k is the current iteration number and K is the maximum value of iterations; PAR_{\min} and PAR_{\max} are the minimum and maximum pitch adjustment rate, and BW_{\min} and BW_{\max} are the minimum and maximum pitch adjustment bandwidth, respectively. The reason behind such improvisation is existing HS is because, larger *BW* value at initial generations ensures considerable diversity in solution vector, where during final generations small *BW* value provides fine tuning of solution vector.

The pseudo code for proposed improved Hybrid Firefly and Harmony search Algorithm (IFFHS) is presented below:

BEGIN

- (1) The objective of the optimization problem is to minimize the root mean square error $E(x)$ given in eq. (17);

$$X = (x_1, x_2, x_3, \dots, x_M),$$

$$x_i \in [U_i, L_i], i = 1, 2, \dots, M$$

where X is a solution vector comprising the decision variables x_i and U_i and L_i are the upper and lower bounds for the decision variables, respectively; $K = 1, 2, \dots, D$; $D = \text{number of dimensions} = M$.

- (2) Initialize different algorithm parameters like, $\alpha = \text{randomization parameter}$, β_{\max} , β_{\min} , PAR_{\max} , PAR_{\min} , BW_{\max} , BW_{\min} , γ , NI ; $NI = \text{number of iterations or generations}$.
- (3) Generate initial population of n number of fireflies. Declare random gain matrix X_{ND} having value of decision variables within the range.
- (4) Light intensity I_i is calculated for each X_i which is equal to the fitness function value at X_i by using the root mean square error between the actual and predicted output value.

- (5) Find the current best solution called as global best solution (X_{gbest}) and the corresponding function value ($I_{gbest} = f_{gbest}$).

While $k < NI$

- Evaluate pitch adjustment rate (PAR), *HMCR*, and pitch adjustment bandwidth (BW) using equations (27), (28), (29), respectively.
- **for** $i=1$ to no. of fireflies
 - If** $I_{gbest} < I_i$

Calculate attractiveness of the fireflies β_{best} and distance between two fireflies r_{igbest} by using the eq. (24) and eq. (25), respectively.

Move all fireflies (i) towards the least bright firefly ($gbest$) by using eq. (23).

Else

If $rand(0,1) < HMCR$

Take a variable $a = \text{any random integer } a \in (1, N)$. We can write it as

Then $x_{i,k} = x_{a,k}$

If $rand(0,1) < PAR$

$x_{i,k} = x_{i,k} \pm rand(0,1) \times BW(k)$

if $f(x_{i,k}) \leq f(x_{a,k})$ **then**

$x_{a,k} = x_{i,k}$

```

End if
End if
Else
     $x_{i,k} = L_{i,k} + \text{rand}(0,1) \times (U_{i,k} - L_{i,k})$ 
    End if
End if
End for

```

- (6) Check whether the values of the decision variables are within the range or not. If the value of the decision variables is not within the range then the lower limit of that decision variable will be assigned to it.
- (7) Evaluate the light intensity values of the firefly which will be used for next generation (iteration) using updated parameters.
- (8) Rank the fireflies according to the light intensity and find the current best solution based on the rank given to the fireflies. Update the global best solution (X_{gbest}) and the corresponding functional value ($I_{gbest} = f_{gbest}$).

End of while. (Stopping criteria)

END

4. Performance Metrics

4.1. Various Error Metrics for Prediction Accuracy Evaluation

The Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE) and Mean Absolute Error (MAE) are used to compare the performance of the models for predicting the time series datasets using LRNFIS and IFFHS learning algorithm. To evaluate the performance of the proposed recurrent fuzzy neural system the following error metrics are defined as:

$$\begin{aligned}
 RMSE &= \sqrt{\frac{1}{N} \sum_{k=1}^N (y_k - \hat{y}_k)^2} \\
 MAPE &= \frac{1}{N} \sum_{k=1}^N \left| \frac{y_k - \hat{y}_k}{y_k} \right| \times 100 \\
 MAE &= \frac{1}{N} \sum_{k=1}^N |y_k - \hat{y}_k|
 \end{aligned} \tag{30}$$

where y_k = actual closing price on k th day

\hat{y}_k = predicted closing price on k th day

N = number of test data.

4.2. Model Performance using Superior Predictive Ability (SPA) test

The forecasting performance of the several models can be assessed in a better way using the Superior Predictive Ability (SPA) test than the error metrics values like MAPE, etc. This test helps to identify whether any of the competing models ($m=1, 2, \dots, M$) is better than the benchmark model in terms of expected loss. For this purpose, the null hypothesis is specified as none of the models is better than the benchmark model against the alternative that among the competing models at least one is significantly better than the benchmark. A supremum over the standardized performance values has been applied by Hansen to test the null hypothesis as:

$$H_0 : \max_{m=1, \dots, M} \frac{\mu_m}{\sqrt{\text{var}\left(n^{1/2} \bar{d}_m\right)}} \leq 0 \tag{31}$$

Using the statistic:

$$T_n^{SPA} = \max \left[\max_{m=1, \dots, M} \frac{n^{1/2} \bar{d}_m}{\sqrt{\hat{\text{var}}\left(n^{1/2} \bar{d}_m\right)}}, 0 \right] \tag{32}$$

where $\hat{\text{var}}\left(n^{1/2} \bar{d}_m\right)$ is an estimate of the variance of $\left(n^{1/2} \bar{d}_m\right)$ obtained by using the bootstrap technique;

μ_m is the expected loss of the model. The bootstrap technique which is an approach for estimating the distribution of an estimator or statistic uses out of samples observations by resampling the data. The average performance of model m relative to benchmark model is represented as:

$$\bar{d}_m = \frac{1}{n} \sum_{t=1}^n d_{m,t} \tag{33}$$

where $d_{m,t} = L_{o,t} - L_{m,t}$

The loss function at time t for the benchmark model is defined by $L_{o,t}$ and $L_{m,t}$, respectively that indicates the value of the corresponding loss function for another competing model m ; n is the no. of out of sample data. The distribution of the test statistics under null hypothesis is approximated by the empirical distribution derived from bootstrap resample based on stationary bootstrap. This is defined as:

$$Z_{m,b,t}^* = d_{m,b,t}^* - g_i(\bar{d}_m) \quad (34)$$

for $b=1, \dots, B$, $t=1, \dots, n$, $i=l, c, u$

where $d_{k,b,t}^*$ is the bootstrap performance of model m relative to benchmark model at time t and $g(x)$ is the estimator function defined as follows:

$$\begin{aligned} g_l(x) &= \max(0, x) \\ g_c(x) &= x \cdot 1 \\ g_u(x) &= x \end{aligned} \quad (35)$$

$$\text{and } \left\{ x \geq - \sqrt{\left(\frac{w_k}{n} \right)^2 2 \log \log n} \right\} \quad (36)$$

Using the three $g(x)$ values, there are three corresponding SPA test values i.e. SPA_l , SPA_c and SPA_u . SPA_l represents the lower bound, SPA_c represents the consistent value and SPA_u represents the upper bound of SPA test statistics. Before calculating P value of SPA test first the SPA statistics for each bootstrap is evaluated in the following manner:

$$T_{b,n}^{SPA*} = \max \left\{ 0, \max_{m=1, \dots, M} \left\{ \frac{n^{1/2} \bar{z}_{m,b}^*}{\sqrt{\text{var}(n^{1/2} \bar{d}_m)}} \right\} \right\} \quad (37)$$

for each $b=1, 2, \dots, B$

Using the comparison of SPA statistics of each bootstrap with original SPA statistics, the P value of SPA test is obtained as follows:

$$P^{SPA} = \sum_{b=1}^B \frac{\{ T_{b,n}^{SPA*} > T_n^{SPA} \}}{B} \quad (38)$$

where B = total number of bootstraps.

A high p^{SPA} value indicates that the Null hypothesis cannot be rejected, which means that the bench mark model is not outperformed by the competing models. On the other hand a lower p^{SPA} value signifies the rejection of the Null hypothesis clearly indicating the bench mark has been outperformed by other competing models. The upper bound of p -value (SPA_u) assumes that all the competing models are precisely as good as the benchmark in terms of the loss function

μ_m ($m=1, 2, \dots, M$). On the contrary the lower bound of p -value (SPA_l) that the models with worse performance than the benchmark are in fact poor models in the limit. To avoid the data snooping problem, the upper and lower bound of the SPA test over MAPE error has been calculated for the PJM electricity market dataset.

5. Applications of LRNFIS to Real World Time Series Databases

For implementing the proposed LRNFIS to forecast the future values of the time series, a Flow chart is shown in Fig.2. Further to evaluate the performance of the proposed LRNFIS model, three real world time series databases are used. Also for comparison the widely used RBFNN using IFFHS evolutionary algorithm is analyzed to provide the validation of improved performance of the proposed LRNFIS model.

Dataset-1:

The first time series considered for analysis is the electricity price, which is a key issue for all the power market stake holders keeping in view of accurate bidding strategy in the day-ahead power pools. With a proper day-ahead forecasting of electricity prices, the end users or the electricity consumers can minimize their utilization cost and at the same time the generation companies can maximize their daily revenues. The market participants face lots of difficulties to develop their proper business planning with regard to the market clearing prices because of many uncertainties like high spikes in prices, temperature effects, seasonal variations and stochastic nature of demands etc. Because of all these uncertain factors, the price time series in almost all markets exhibits non-stationary means and variances. Several methods that involve both statistical methods like ARMA, ARIMA, and GARCH [3,40,51] models and intelligent system techniques like ANN, FIS, SVM [11,52,53,14,16], etc. have been proposed by researchers for day-ahead electricity price forecasting incorporating different uncertain effects in the deregulated electricity markets.

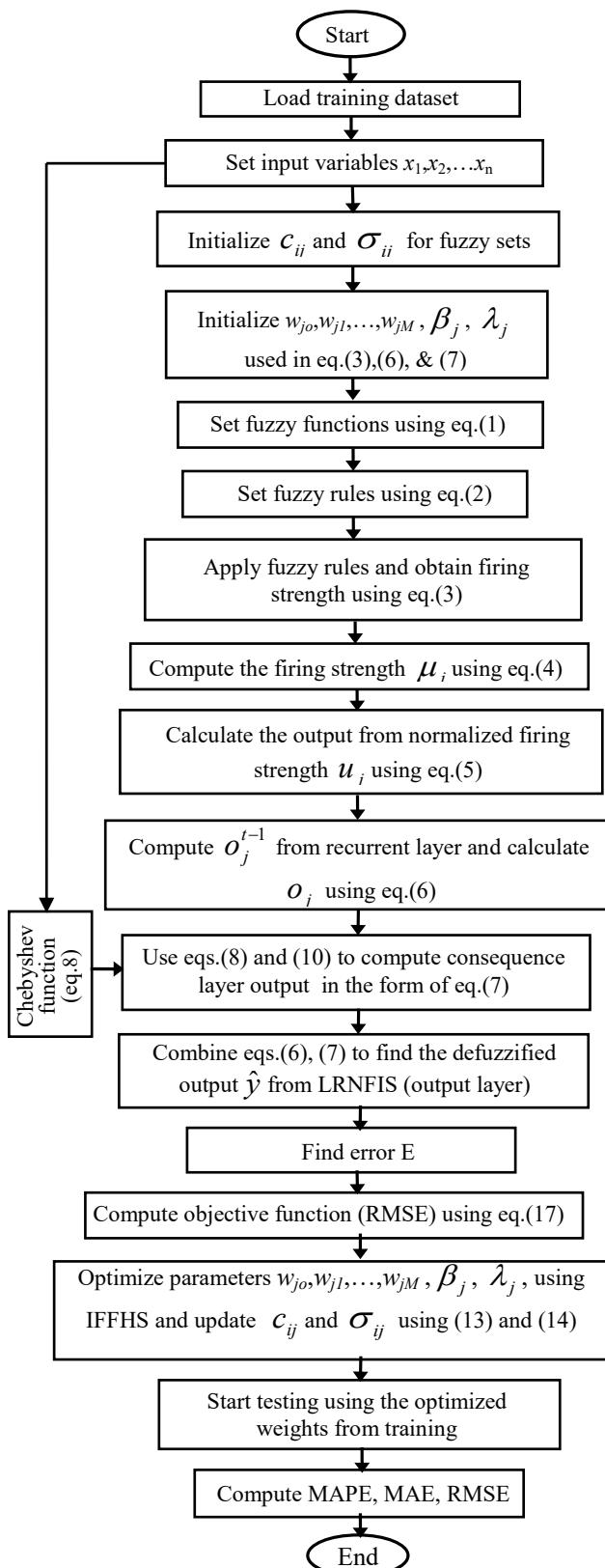


Fig. 2. Flow Chart for LRNFIS Implementation

Before incorporating the electricity price time series data into the proposed model, it is highly essential for the data series to be pre-processed. For this dataset, the single-period continuously compounded return series, known as log-return series is considered for day ahead prediction. The return series has efficient statistical properties and it is easy to handle the highly fluctuating price signals. The single-period continuously compounded return series is defined as:

$$R_t = \ln(P_t) - \ln(P_{t-1}) = \ln\left(\frac{P_t}{P_{t-1}}\right) \quad (39)$$

where R_t is the single-period log return at time t , P_t is the electricity price at time t . For prediction, the electricity price time series from Jan 1, 2014 to December 31, 2014 of the PJM market is shown in Fig.3 (a), and its return series during the same period is shown in Fig.3 (b). It can be observed that the mean value of both demand and price return series is nearly equal to zero and the variance over the period indicates the return series to be more homogeneous. Thus, for predicting 24-hour ahead electricity price, the considered return series is given by

$$P_t = f(R_{t-1}, R_{t-2}, R_{t-23}, R_{t-24}, R_{t-47}, R_{t-48}, \dots, R_{t-167}, R_{t-168}) \quad (40)$$

where f represents a nonlinear function of the price return series inputs.

For highly fluctuating time series the number of lag terms in equation (40) is determined by taking the autocorrelation and partial autocorrelation of lag k for the return price series (d signifies the day of forecast):

$$ACF_k = \frac{\sum_{t=1}^{T-k} (P_{t,d} - \bar{P}_d)(P_{t+d,k} - \bar{P}_d)}{\sum_{t=1}^{T-k} (P_{t,d} - \bar{P}_d)^2} \quad (41)$$

Using the autocorrelation function (ACF) and partial autocorrelation function (PACF) the order of the return price series is thus identified. It is observed that the log return series has good statistical properties in comparison to the original series. In the return series the ACF is observed to die out more quickly with 95% confidence interval. With single period compounded log return series, it is seen that both the ACF and PACF die out after lag 2 with a damping sine-cosine curve. Therefore, keeping in view of stationary relationship, the price time series is processed with inputs from return series. The ACF and PACF plots for the 2014

PJM return price series are depicted in Fig. 4(a) and 4(b), respectively.

The proposed methodology for predicting electricity price time series is implemented using PJM electricity market data from January 1, 2014 to December 31, 2014. In all the considered weeks of forecast, for a particular day's forecast, the training dataset is used for the prior 20 days. The inputs to all the proposed models comprise the return price time series given by the lags of $(t-23)$, $(t-24)$, $(t-47)$, $(t-48)$, $(t-71)$, $(t-72)$, $(t-95)$, and $(t-96)$ hours, respectively. The training data comprises the actual price data up to the day before the day of forecast. Keeping in view of seasonal variations, four different periods have been chosen to evaluate the performance of the proposed models. The four one-week periods include February 1-7, May 1-7, August 1-7, and November 1-7 of the year 2014. Originally the electricity data set has been divided into training and testing sets.

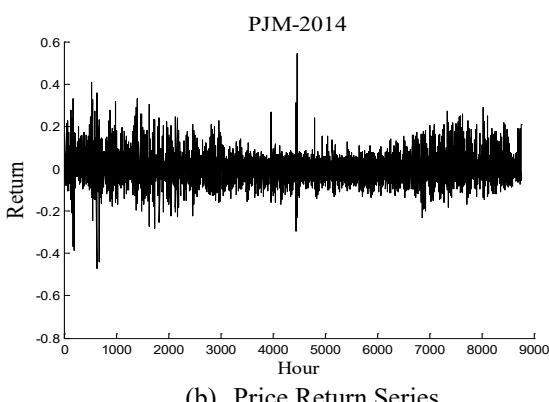
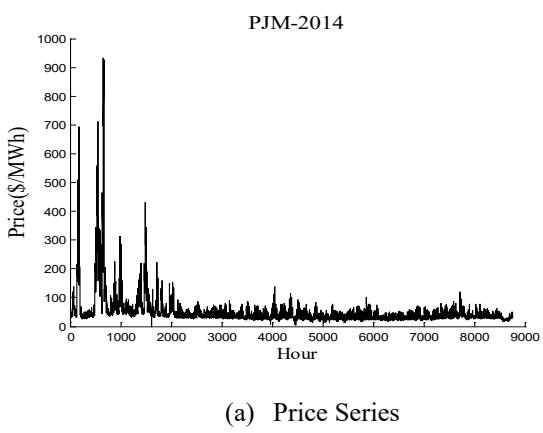


Fig.3 PJM Electricity price and its Return datasets

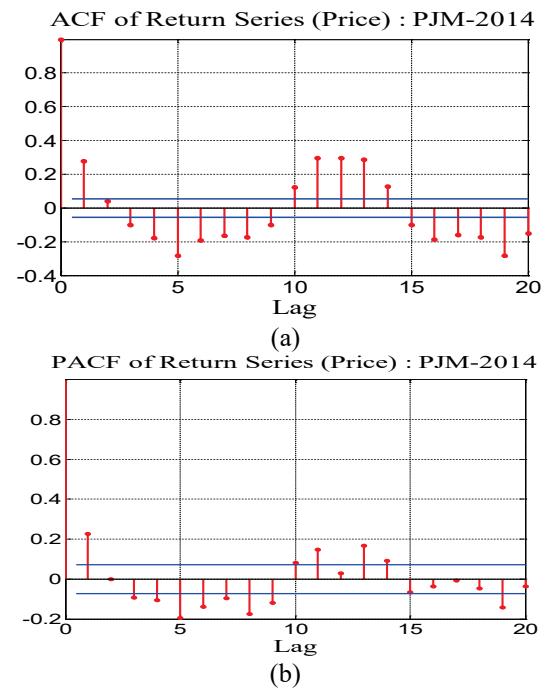
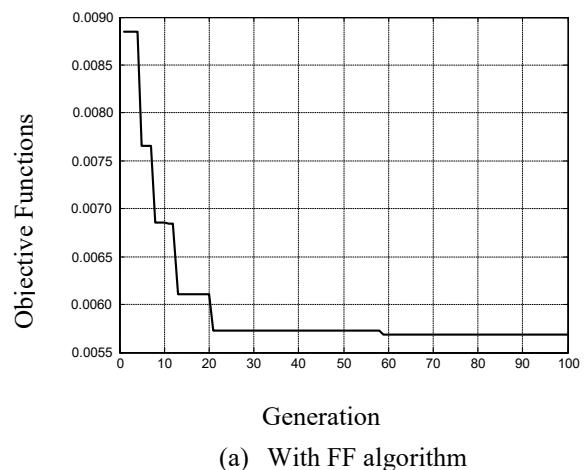


Fig.4 ACF and PACF plots for the 2014 PJM Return Price Data

As the evolutionary algorithms are based on randomness, the performance metrics are observed from ten independent executions of the algorithm with same training and testing sets. The stability of prediction is ascertained from the convergence characteristics shown in Fig.5 for both the FF and IFFHS algorithms. The LRNFIS trained by conventional FF algorithm takes nearly 60 iterations to converge, whereas the IFFHS algorithm takes nearly 40 iterations with less error in the objective function. Table-1 provides the details of the various settings of the hybrid IFFHS algorithm.



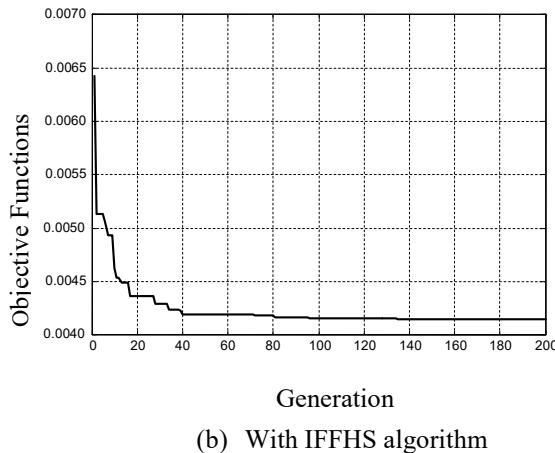


Fig. 5 Convergence characteristics of the FF and IFFHS algorithms

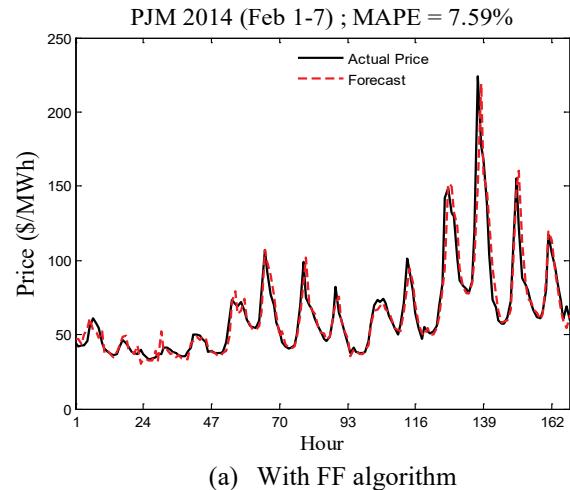
Table 1: Details of proposed IFFHS algorithm

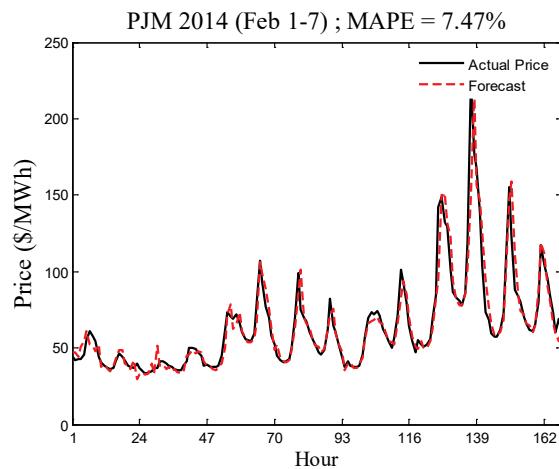
Parameters	Values
Maximum number of generation (T_{max})	100
Total number of fireflies (N_{ff})	20
Initial randomization parameter (γ)	0.5
Absorption co-efficient (β)	0.8
Maximum and minimum attractiveness (β_{max} and β_{min})	0.95 & 0.3
Maximum and minimum PAR (PAR_{max} & PAR_{min})	0.9 & 0.3
Maximum and minimum bw (BW_{max} & BW_{min})	0.5 & 0.2
Harmonic consideration rate ($HMCR$)	0.8

Figs. 6(a) to 6(h) show the weekly forecasts for the first weeks of February, May, August, and November, respectively using both the FF and IFFHS algorithm, and the MAPE values are also given in Table-2. From the results it is clearly seen that the actual values of electricity prices in the considered week of November, 2014 are captured more accurately in comparison to the weekly forecast during Feb.1-7, 2014. This discrepancy indicates the presence of low spikes in electricity prices in November as compared to those in the month of February 2014. The MAPE is found to be 6.30 % during Nov.1-7 using FF based LRNFIS in comparison to 5.27% for the IFFHS based LRNFIS. Similarly the period of August 1-7 exhibits MAPE value of 4.23% for the FF-LRNFIS compared to 4.02% for the IFFHS-LRNFIS. Table-2 provides the weekly comparison of

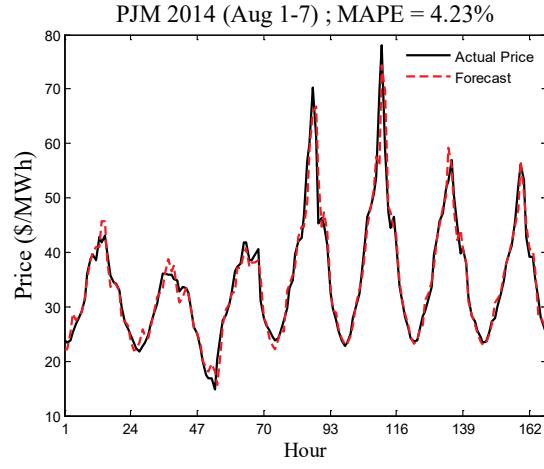
MAPE values for different typical months of the year. Further a comparison with the well known radial basis function neural network (RBFNN) and functional link neural network using IFFHS algorithm for weight parameter training is given in Table-2, showing clearly the superiority of prediction performance of the LRNFIS with IFFHS algorithm over RBFNN and FLANN models.

The three models namely the LRNFIS with FF, LRNFIS with IFFHS, and RBFNN with IFFHS algorithm are subjected to Superior ability performance test to provide a comparison of their relative performance except the FLANN, which gives the worst performance. To avoid the data snooping problem, the upper and lower bound of the SPA test over MAPE errors have been calculated for the PJM weekly electricity market data used for prediction as shown in Table-3. Table-3 presents the P-values of upper and lower bound of SPA test with 3 different size of bootstrap re-samples namely 500, 800, and 1000 chosen from the 2014 year. In this test each model is considered individually as the benchmark model to evaluate whether a particular model is significantly outperformed by other competing models. For the considered data sets, the P-value of SPA test over MAPE error with different bootstrap size is showing higher value indicating that the other competing models are not producing better performance compared to the proposed model. From Table-3 it is clearly observed that the LRNFIS model with IFFHS model completely outperforms the LRNFIS model with FF and RBFNN model trained by IFFHS algorithms in predicting the electricity price time series.

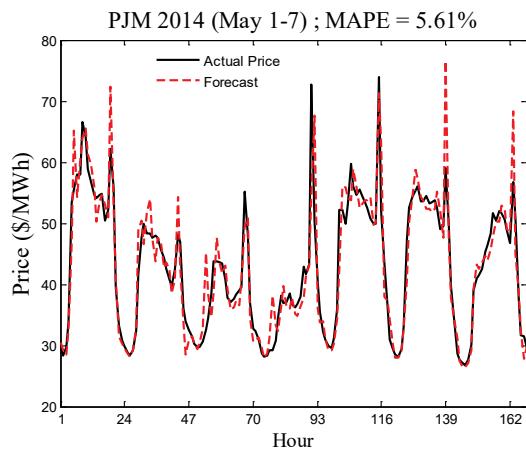




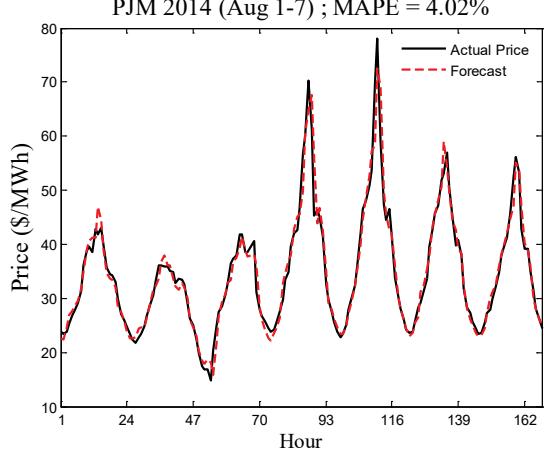
(b) With IFFHS algorithm



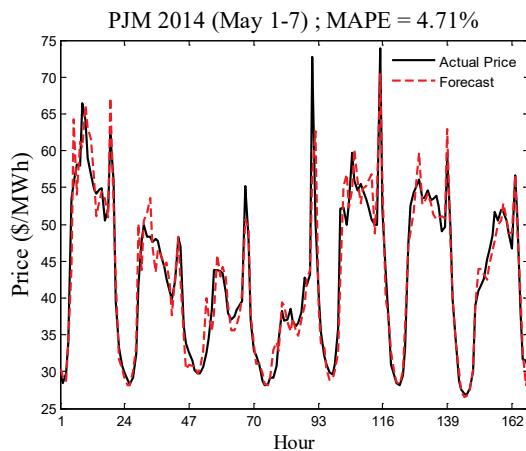
(e) With FF algorithm



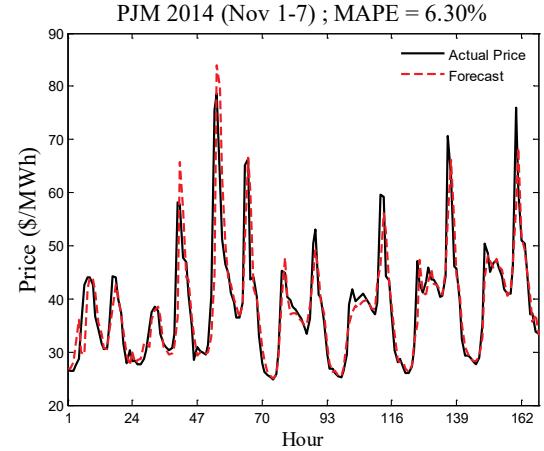
(c) With FF algorithm



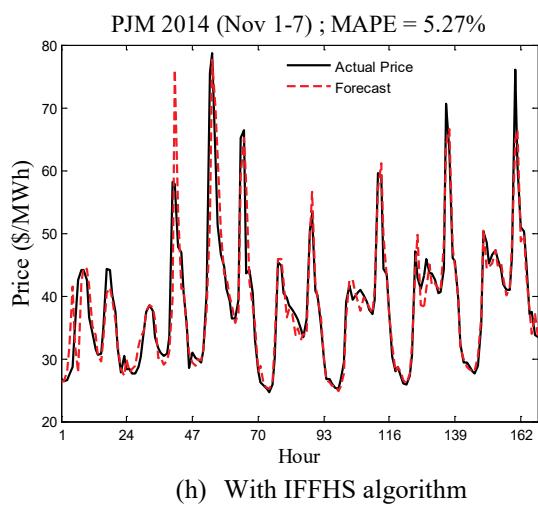
(f) With IFFHS algorithm



(d) With IFFHS algorithm



(g) With FF algorithm



(h) With IFFHS algorithm

Fig.6 Seasonal MAPE variations for the PJM Electricity market during the year 2014 using LRNFIS

Table-2 RMSE and MAPE (%) of PJM Electricity market

Period	LRNFIS with FF		LRNFIS with IFFHS		RBFNN with IFFHS		FLANN with IFFHS	
	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
Feb 1-7	0.00 93	7.5 9	0.00 7	7.47 7	0.00 98	8.7 3	0.01 6	9.2 6
May 1-7	0.00 56	4.9 4	0.00 41	4.71 74	0.00 3	5.8 97	0.00 3	6.1 3
Aug 1-7	0.00 47	4.2 3	0.00 40	4.02 62	0.00 6	5.7 89	0.00 6	6.0 6
Nov 1-7	0.00 72	6.3 0	0.00 53	5.27 84	0.00 5	7.3 2	0.01 5	7.9 5

Table-3 P-value of Superior Predictive Ability test over MAPE error for PJM Electricity market

Benchmark model	No. of Bootstraps					
	500		800		1000	
	SPA _u	SPA _l	SPA _u	SPA _l	SPA _u	SPA _l
LRNFIS with IFFHS	0.984	0.760	0.969	0.678	0.975	0.547
RBFNN with IFFHS	0.835	0.692	0.817	0.562	0.834	0.532
LRNFIS with FF	0.717	0.583	0.754	0.435	0.713	0.482

Dataset-2:

After establishing the superior performance of the LRNFIS model trained by the IFFHS evolutionary algorithm in electricity price time series prediction, its performance is evaluated for one day ahead prediction of daily currency exchange rates for dataset-2. Four major currencies such as Australian Dollar (AUD), Swiss Francs (CHF), Mexican Pesos (MXN), and Brazilian Dollar (BRL) against US Dollar (USD) are chosen for experimental analysis. Each currency exchange dataset covers time span from January 2010 to March 2013 and contains 800 data out of which initial 500 data are taken for training and the following 300 data are taken for testing. The details are depicted in Table-4. Before starting the training process, the data (actual prices) are normalized between 0 and 1 using the min-max formula using eq.(42). In this case five lagged scaled prices are used as inputs for current day price prediction i.e. the price on current day ‘d’ depends on the prices of days ‘d-1’, ‘d-2’, ‘d-3’, ‘d-4’, ‘d-5’ and this concept is followed for each day ahead price prediction for the training and testing datasets. Once the training process of the LRNFIS model completes then the testing is started with the optimized weights from the training.

$$y = \frac{pr - pr_{\min}}{pr_{\max} - pr_{\min}} \quad (42)$$

where y = normalized value.

pr = value to be normalized

pr_{\min} = minimum value of the series to be normalized

pr_{\max} = maximum value of the series to be normalized

Table-4 Datasets chosen for experimental study

Datasets-2	Currency Exchange rates against US Dollar	Training Period	Testing Period
	AUD/USD	January	January
	CHF/USD	2010 to December	2012 to March
	MXN/USD		
	BRL/USD	2011	2013

The testing performances of the currency exchange rates prediction are measured in terms of MAPE, MSE, RMSE and are shown in Table-5. From Table-5 it is clear that the four currency exchange rates prediction show an average MAPE of 0.5 and the MSE and RMSE are very small. The one-day ahead testing results (Target vs Predicted) of AUD/USD, CHF/USD,

MXN/USD, BRL/USD are depicted in Figs. 7, 8, 9, and 10, respectively.

The model has again been verified by predicting the 5-days, 7-days, and 10-days ahead currency exchange rates and findings from these experiments are given in Table-5 and graphical prediction results are shown in Figs. 11-14. Fig.11 shows the 5-days, 7-days, and 10-days ahead prediction of AUD/USD where as Fig.12 depicts the 5-days, 7-days, and 10-days ahead prediction of CHF/USD, respectively. Similarly the MXN/USD and BRL/USD prediction results are shown in Figs.13 and 14, respectively.

From Table-5, it is clear that the average MAPE for the four chosen currency exchange rates for one day ahead prediction is 0.5. The 5-days, 7-days, and 10-days ahead predictions MAPE of AUD/USD are 0.7305, 0.8816, and 1.0188, respectively. The 10-days ahead prediction MAPE is higher than 5-days and 7-days ahead prediction MAPE. The MAPEs for AUD/USD prediction result are lower as compared to the CHF/USD, MXN/USD, and BRL/USD which indicates that the AUD/USD prediction performance is better than the other three currency exchange rates.

Table-5 Performance Evaluation using Datasets 2

1-day ahead prediction			
Currency Exchange rates against US Dollar	MAPE	MSE	RMSE
AUD/USD	0.5391	4.2665e-05	0.0065
CHF/USD	0.5724	4.5064e-05	0.0067
MXN/USD	0.5692	0.0100	0.1001
BRL/USD	0.5872	2.5305e-04	0.0159
5-days ahead prediction			
AUD/USD	0.7305	7.9477e-05	0.0089
CHF/USD	0.8419	8.8983e-05	0.0094
MXN/USD	0.6234	0.0127	0.1125
BRL/USD	0.8441	4.9434e-04	0.0222
7-days ahead prediction			
AUD/USD	0.8816	1.0355e-04	0.0102
CHF/USD	0.9026	1.1235e-04	0.0106
MXN/USD	0.9123	0.0249	0.1578
BRL/USD	0.9044	7.1107e-04	0.0267
10-days ahead prediction			
AUD/USD	1.0188	1.4035e-04	0.0118
CHF/USD	1.1884	1.6995e-04	0.0130
MXN/USD	1.2098	0.0417	0.2042
BRL/USD	1.1218	9.7488e-04	0.0312

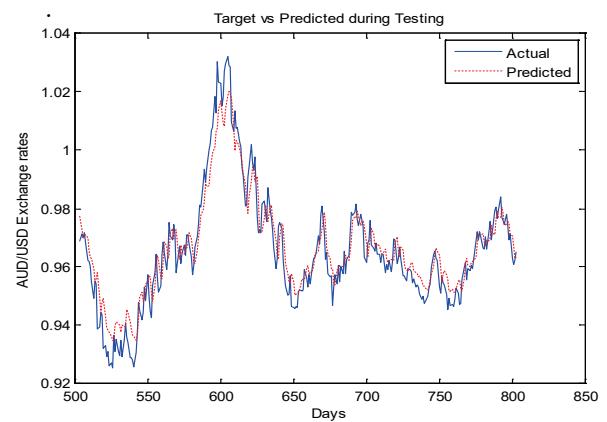


Fig.7 AUD/USD predicted Testing results

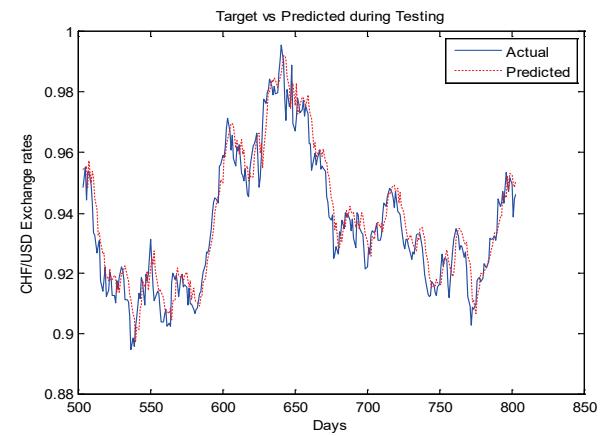


Fig.8 CHF/USD predicted Testing results

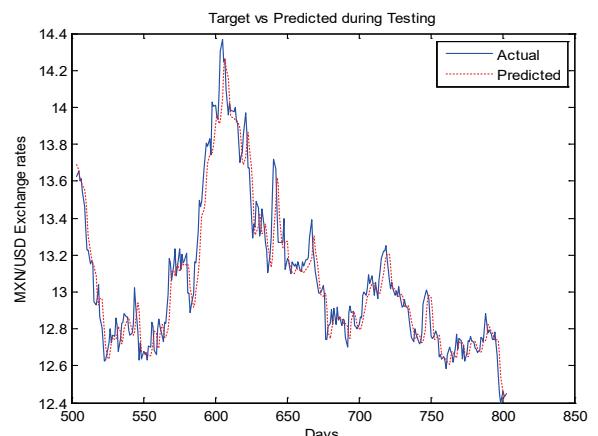


Fig.9 MXN/USD predicted Testing results

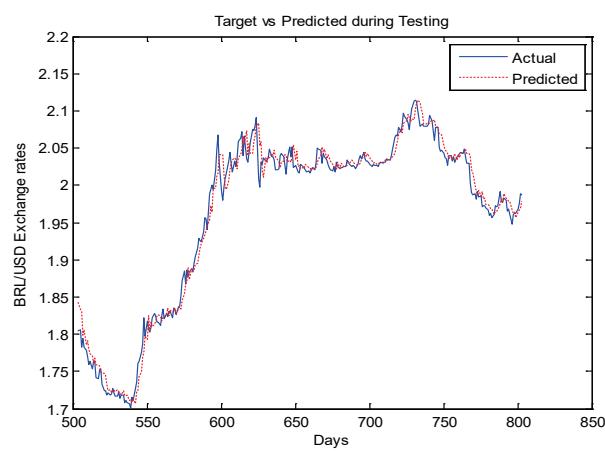


Fig.10 BRL/USD predicted Testing results

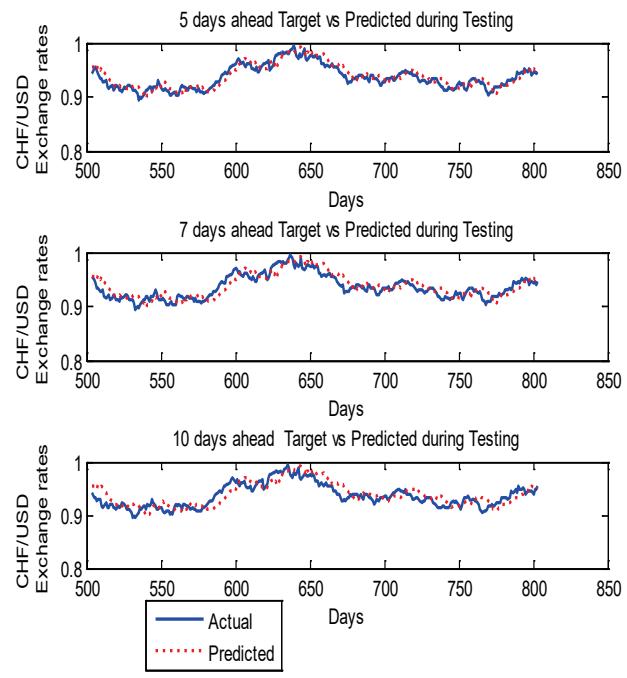


Fig.12. 5, 7, and 10 -days ahead prediction results of CHF/USD Currency exchange rates

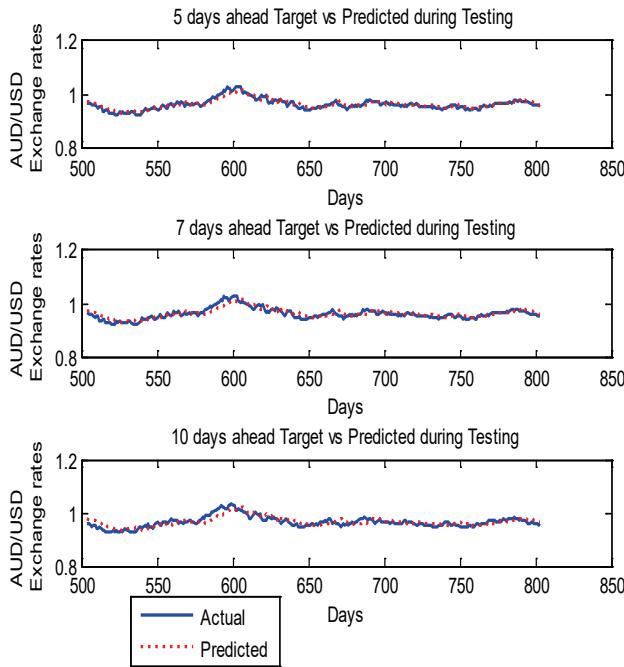


Fig.11. 5, 7, and 10 -days ahead prediction results of AUD/USD Currency exchange rates

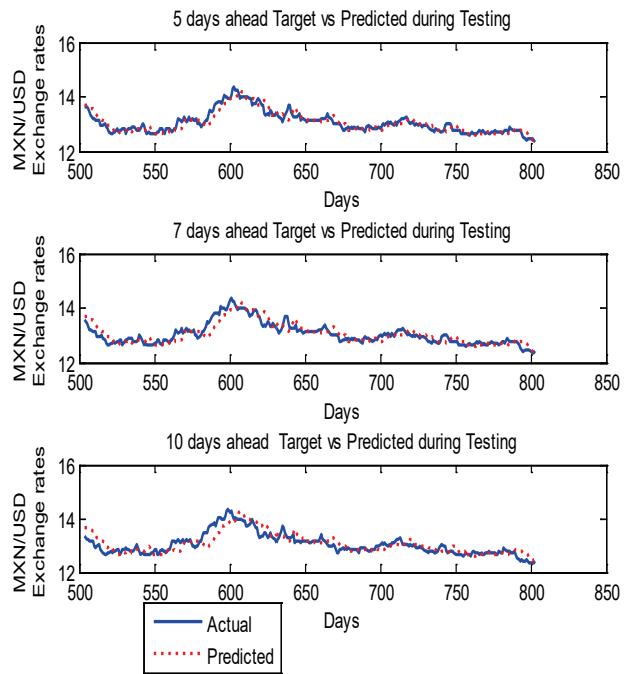


Fig.13. 5, 7 and 10 -days ahead prediction results of MXN/USD Currency exchange rates

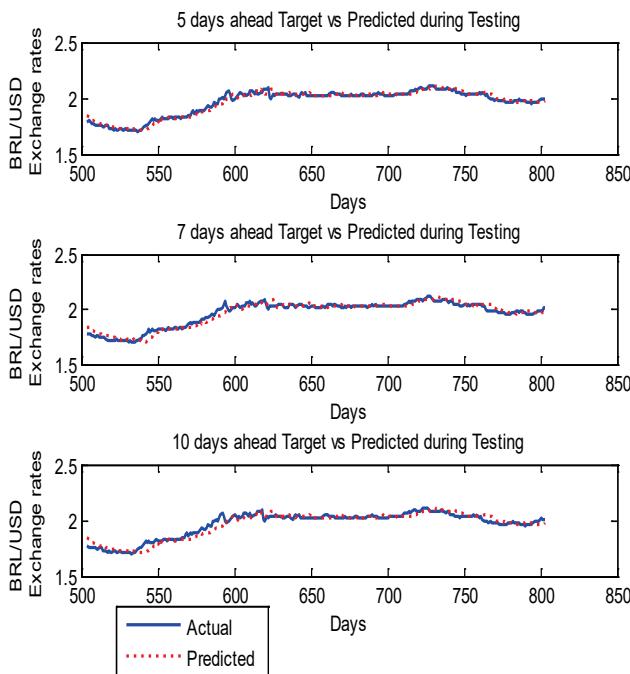


Fig.14. 5,7 and 10 -days ahead prediction results of BRL/USD Currency exchange rates

Dataset-3:

The LRNFIS model further proves its efficiency using two other stock market time series datasets: Standard & Poor's 500 stock market abbreviated as S&P 500, an American stock market and Nikkei 225 commonly known as Nikkei Index from Tokyo stock exchange. The daily closing prices of S&P 500 and Nikkei 225 stock market are collected for the purpose of day ahead price prediction. Each of the two considered datasets contains 800 data covering the time span from 04 January 2010 to 04 March 2013 out of which the first 500 data from 04 January 2010 to 23 December 2011 are chosen for training the model and the rest 300 data from 27 December 2011 to 04 March 2013 are taken for testing to prove the efficiency of the model. Here the data processing is similar to the procedure used for Dataset-2. The testing outcomes of the two stock markets are presented in Table-6 and the graphical representation of the testing results shown in Figs. 15 and 16 which validate the performance of the testing results. The S&P 500 testing performance shows a MAPE of 0.6288 and Nikkei 225 shows a MAPE of 0.7105 which proves the efficiency of the LRNFIS model.

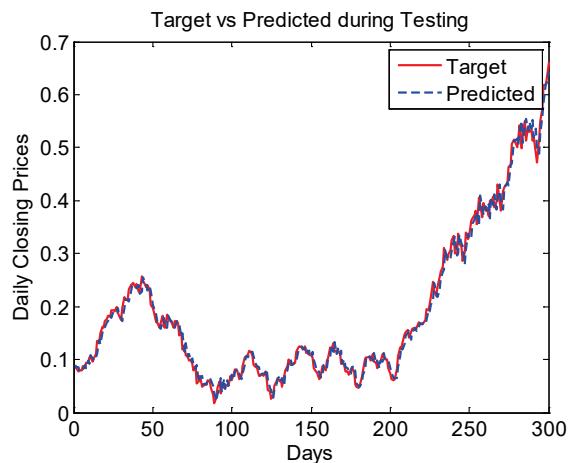


Fig.15 S&P 500 Testing Results

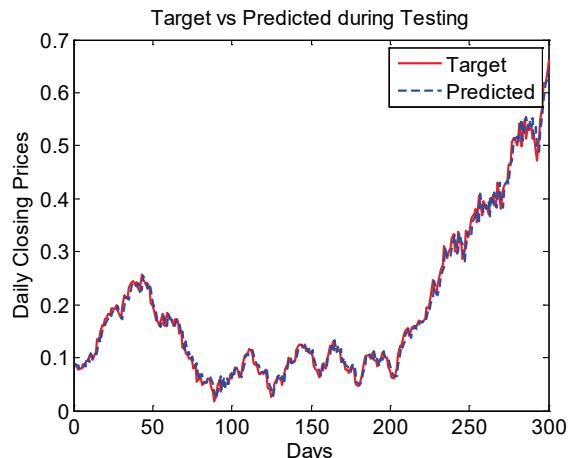


Fig.16 Nikkei 225 Testing Results

Table-6 Performance Evaluation using Datasets 3

Stock market	1-day ahead prediction		
	MAPE	MSE	RMSE
S&P 500	0.6288	2.5745e-04	0.0160
Nikkei 225	0.7105	2.6089e-05	0.0051

6. Conclusion

A new locally recurrent neuro-fuzzy information system (LRNFIS) is presented in this paper for time series prediction. The LRNFIS uses TSK type fuzzy rule base where the consequent part comprises Chebyshev polynomial functions providing an expanded nonlinear transformation of the inputs suitable for forecasting chaotic time series. Further dynamic memory is provided by using locally recurrent nodes after the fuzzy output normalization layer allowing past and present

inputs to be used for a robust prediction. The weights of the LRFNFS are trained by an improved firefly harmony search algorithm whereas the membership functions of the fuzzy system are learnt by a gradient descent algorithm. The performance of the proposed LRFNFS is validated by using three real world time series data sets namely the PJM electricity market, currency exchange market (AUD/USD, CHF/USD, MXN/USD, BRL/USD), and S&P 500 and Nikkei 225 stock markets providing robust prediction results. A comparison with widely used RBFNN is presented, which also reveals the same conclusion. Further a superior predictive ability test is used to choose the correct forecasting model.

References

1. M. Khashei and M. Bijari, A novel hybridization of artificial neural networks and ARIMA models for time series forecasting, *Appl. Soft Comput.* 11(2) (2011) 2664–2675.
2. J. L. Torres, A. García, M. De Blas and A. De Francisco, Forecast of hourly average wind speed with ARMA models in Navarre (Spain), *Solar Energy*. 79(1) (2005) 65–77.
3. R. C. Garcia, J. Contreras, M. van Akkeren, J. B. C. Garcia, A GARCH forecasting model to predict day-ahead electricity prices, *IEEE Trans. Power Syst.* 20(2) (2005) 867–874.
4. Y. H Wang, Nonlinear neural network forecasting model for stock index option price: Hybrid GJR-GARCH approach, *Expert Syst. Appl.* 36(1) (2009) 564–570.
5. M. A. Basel and V. C. Awartani, Predicting the volatility of the S&P-500 stock index via GARCH models: the role of asymmetries, *Int. J. Forecast.* 21(1) (2005) 167–183.
6. J. Z Wang, J. J. Wang, Z.G. Zhang and S.P. Guo, Forecasting stock indices with back propagation neural network, *Expert Syst. Appl.* 38(11) (2011) 14346–14355.
7. E. Guresen, G. Kayakutlu and T. U. Daim, Using artificial neural network models in stock market index prediction, *Expert Syst. Appl.* 38(8) (2011) 10389–10397.
8. K. Y. Lee, Y. T. Cha and J. H. Park, Short-term load forecasting using an artificial neural network, *IEEE Trans. Power Syst.* 7(1) (1992) 124–132.
9. D. C. Park, M. A. El-Sharkawi, R. J. Marks and L. E. Atlas, Electric load forecasting using an artificial neural network, *IEEE Trans. Power Syst.* 6(2) (1991) 442–449.
10. T. Senju, H. Takara, K. Uezato and T. Funabashi, One-hour-ahead load forecasting using neural network, *IEEE Trans. Power Syst.* 17(1) (2002) 113–118.
11. P. Mandal, T. Senju, N. Urasaki, T. Funabashi and A. K. Srivastava, A novel approach to forecast electricity price for PJM using neural network and similar days method, *IEEE Trans. Power Syst.* 22(4) (2007) 2058–2065.
12. P. C. Chang, D. Wang and C. Zhou, A novel model by evolving partially connected neural network for stock price trend forecasting, *Expert Syst. Appl.* 39(1) (2012) 611–620.
13. C. F. Liu, C.Y. Yeh and S. J. Lee, Application of type-2 neuro-fuzzy modeling in stock price prediction, Application of type-2 neuro-fuzzy modeling in stock price prediction, *Appl. Soft Comput.* 12(4) (2012) 1348–1358.
14. N. Amjadi, Day-ahead price forecasting of electricity markets by a new fuzzy neural network, *IEEE Trans. Power Syst.* 21(2) (2006) 887–896.
15. C.F. Juang, A TSK-type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithms, *IEEE Trans. Fuzzy Syst.* 10(2) (2002) 155–170.
16. S. Fan, C. Mao and L. Chen, Next-day electricity-price forecasting using a hybrid network, *IET Gener. Transm. Distrib.* 1(1) (2007) 176–182.
17. W. Huang, Y.N. Nakamori and S.Y. Wang, Forecasting stock market movement direction with support vector machine, *Comput. Oper. Res.* 32(10) (2005) 2513–2522.
18. D. Niu, Y. Wang and D. D. Wu, Power load forecasting using support vector machine and ant colony optimization, *Expert Syst. Appl.* 37(3) (2010) 2531–2539.
19. S. Fan and L. Chen, Short-term load forecasting based on an adaptive hybrid method, *IEEE Trans. Power Syst.* 21(1) (2006) 392–401.
20. K. Kim, Financial time series forecasting using support vector machines, *Neurocomputing*. 55(1–2) (2003) 307–319.
21. W. Shen, X. Guo, C. Wu, D. Wu, Forecasting stock indices using radial basis function neural networks optimized by artificial fish swarm algorithm, *Knowl-Based Syst.* 24(3) (2011) 378–385.
22. Y. F. Sun, Y. C. Liang, W. L. Zhang, H. P. Lee and W. Z. Lin, Optimal partition algorithm for the RBF neural network for financial time series forecasting, *Neural Comput. Appl.* 14(1) (2005) 36–44.
23. Z. Yun, Z. Quan, S. Caixin and L. Shaolan, RBF neural network and ANFIS-based short-term load forecasting approach in real-time price environment, *IEEE Trans. Power Syst.* 23(3) (2008) 853–858.
24. V. Nekoukar and M. T. H. Beheshti, A local linear radial basis function neural network for financial time series forecasting, *Springer, Appl. Intell.* 33(3) (2010) 352–356.
25. Z. Bashir and M. E. El-Hawary, Short term load forecasting by using wavelet neural networks, in proc. 2000 Canadian Conference on Electrical and Computer Engineering (Halifax, NS, 07 Mar 2000–10 Mar 2000, 2000), Vol. 1, pp. 163–166.
26. Y. Chen, B. Yang and J. Dong, Time series prediction using a local linear wavelet neural network, *Neurocomputing*, 69(4–6) (2006) 449–465.
27. Y. Chen, X. Dong and Y. Zhao, Stock index modeling using EDA based local linear wavelet neural network, in proc. 2015 IEEE International Conference on Neural Networks and Brain (Beijing, 13–15 Oct. 2005), vol. 3, pp. 1646–1650.

28. C. L. Giles, S. Lawrence and A.C. Tsoi, Noisy time Series prediction using recurrent neural networks and grammatical inference, *Mach. Learn.* 44(1) (2001) 161–183.
29. T. G. Barbounis, J. B. Theocharis, M. C. Alexiadis and P. S. Dokopoulos, Long-term wind speed and power forecasting using local recurrent neural network models, *IEEE Trans. Energy Convers.* 21(1) (2006) 273–284.
30. S. Coupland and R. John, Geometric Type-1 and Type-2 fuzzy logic systems, *IEEE Trans. Fuzzy Syst.* 15(1) (2007) 3–15.
31. C. T. Lin and C. S. G. Lee, Neural-network-based fuzzy logic control and decision system, *IEEE Trans. Comput.* 40(12) (1991) 1320–1336.
32. G. S. Atsalakis and K. P. Valavanis, Forecasting stock market short-term trends using a neuro-fuzzy based methodology, *Expert Syst. Appl.* 36(7) (2009) 10696–10707.
33. Y. Leu, C. P. Lee and Y. Z. Jou, A distance-based fuzzy time series model for exchange rates forecasting, *Expert Syst. Appl.* 36(4) (2009) 8107–8114.
34. V. Kodogiannis and A. Lolis, Forecasting financial time series using neural network and fuzzy system-based techniques, *Neural Comput. Appl.* 11(2) (2002) 90–102.
35. J. S. R. Jang, ANFIS: Adaptive-network-based fuzzy inference system, *IEEE Trans. Syst., Man, and Cybern.* 23(3) (1993) 665–685.
36. L. C. Ying and M. C. Pan, Using adaptive network based fuzzy inference system to forecast regional electricity loads, *Energ. Convers. Manage.* 49(2) (2008) 205–211.
37. A. Esfahanipour and W. Aghamiri, Adapted neuro-fuzzy inference system on indirect approach TSK fuzzy rule base for stock market analysis, *Expert Syst. Appl.* 37(7) (2010) 4742–4748.
38. P. Melin, J. Soto, O. Castillo and J. Soria, A new approach for time series prediction using ensembles of ANFIS models, *Expert Syst. Appl.* 39(3) (2012) 3494–3506.
39. A. Bagheri, H. M. Peyhani and M. Akbari, Financial forecasting using ANFIS networks with quantum-behaved particle swarm optimization, *Expert Syst. Appl.* 41(14) (2014) 6235–6250.
40. Z. Tan, J. Zhang, J. Wang and J. Xu, Day-ahead electricity price forecasting using wavelet transform combined with ARIMA and GARCH models, *Appl. Energy.* 87(11) (2010) 3606–3610.
41. H. Mohammadi and L. Su, International evidence on crude oil price dynamics: Applications of ARIMA-GARCH models, *Energ. Econ.* 32(5) (2010) 1001–1008.
42. F. M. Tseng and G.H. Tzeng, A fuzzy seasonal ARIMA model for forecasting, *Fuzzy Sets Syst.* 126(3) (2002) 367–376.
43. F.M. Tseng, G. H. Tzeng, H. C. Yu and B. J. C. Yuan, Fuzzy ARIMA model for forecasting the foreign exchange market, *Fuzzy Sets Syst.* 118(1) (2001) 9–19.
44. X.S. Yang, *Nature-Inspired Metaheuristic Algorithms*. (Frome: Luniver Press, U.K.) ISBN 1-905986-10-6, 2008.
45. X.S. Yang, Firefly algorithm, stochastic test functions and design optimisation, *Int. J. Bio-Inspir. Com.* 2(2) (2010) 78–84.
46. X. Tao, Y. Bao and Z. Hu, Multiple-output support vector regression with a firefly algorithm for interval-valued stock price index forecasting, *Knowl-Based Syst.* 55 (2014) 87–100.
47. Z.W. Geem, J. H. Kim and G. V. Loganathan, A new heuristic optimization algorithm: harmony search, *Simulation.* 76(2) (2001) 60–68.
48. K.S. Lee and Z. W. Geem, A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice, *Comput. Methods in Appl. Mech. Eng.* 194(36–38) (2005) 3902–3933.
49. W.K. Wong and Z. X. Guo, A hybrid intelligent model for medium-term sales forecasting in fashion retail supply chains using extreme learning machine and harmony search algorithm, *Int. J. Prod. Econ.* 128(2) (2010), 614–624.
50. M. Mahdavi, M. Fesanghary and E. Damangir, An improved harmony search algorithm for solving optimization problems, *Appl. Math. Comput.* 188(2) (2007) 1567–1579.
51. J. Contreras, R. Espinola, F. Nogales and A. Conejo, ARIMA models to predict next-day electricity prices, *IEEE Trans. Power Syst.* 18(3) (2003) 1014–1020.
52. V. Vahidinasab, S. Javid, A. Kazemi, Day-ahead price forecasting in restructured power systems using artificial neural networks, *Electr. Pow. Syst. Res.* 78(8) (2008) 1332–1342.
53. Y.Y. Hong, C.Y. Hsiao, Locational marginal price forecasting in deregulated electric markets using a artificial intelligence, *IEE Proc. Gener. Transm. Distrib.* 149(5) (2002) 621–626.