2015 IEEE Power, Communication and Information Technology Conference (PCITC)
Siksha 'O' Anusandhan University, Bhubaneswar, India.

A Low Complexity Evolutionary Computationally Efficient Recurrent Functional Link Neural Network for Time Series Forecasting

Ajit Kumar Rout
GMR Institute of Technology, Rajam,
Andhra Pradesh
ajitkumar.rout@gmrit.org

R. Bisoi[1], P.K. Dash[2]
Multidisciplinary Research Cell
Siksha O Anusandhan University, Bhubaneswar, India.
[1]ranjeeta.bisoi@gmail.com, [2]pkdash.india@gmail.com

Abstract—**The paper presents a low complexity recurrent Functional link Artificial Neural Network for predicting the time series data like the stock market indices over a time frame varying from one day ahead to one month ahead. Further an adaptive bioinspired Firefly algorithm is adopted here to find the optimal weights for the recurrent computationally efficient functional link neural network (RCEFLANN) using a combination of Linear and hyperbolic tangent basis functions. The performance of the Recurrent computationally efficient FLANN model is applied for the prediction stock prices of Standard & Poor's 500 (S&P500) and NIKKEI 225 data sets providing significant accuracy. Also another time series data like the electricity price of Pennsylvania–New Jersey–Maryland (PJM) energy market has been considered for weekly electricity price prediction using the proposed approach with satisfactory results.**

***Keywords*: Recurrent FLANN, stock time series, Electricity price time series,**

I. INTRODUCTION

Time series data are more complicated than other statistical data due to the long term trends, cyclical variations, seasonal variations and irregular movements. Predicting such highly fluctuating and irregular data is usually subject to large errors. So developing more realistic models for predicting financial time series data is a great interest of research in financial time series data mining. The traditional statistical models used for financial prediction were simple, but suffered from several shortcomings due to the nonlinearity of data. Hence researchers have developed more efficient and accurate soft computing methods like ANN [1]; Fuzzy set theory [2], Support Vector Machine [3], Rough Set theory [4],etc. for financial prediction. Various ANN based methods like Multi Layer Perception Network (MLP) [5], Radial Basis Function Neural Network (RBF) [6], Wavelet Neural Network (LLWNN) [7], Recurrent Neural Network (RNN) [8], and Functional Link Artificial Neural Network (FLANN) [9] are extensively used for stock market prediction due to their inherent capabilities to identify complex nonlinear relationship present in the time series data based on historical data and to approximate any nonlinear function to a high degree of accuracy. The use of ANN to predict the behavior and tendencies of stocks has demonstrated itself to be a viable alternative to existing conventional techniques [10]. The FLANN originally proposed by Pao in1992 is single layer single neuron architecture, having two components: Functional expansion component and Learning component. The functional block helps to introduce nonlinearity by expanding the input space to higher dimensional space through a basis function without using any hidden layers like MLP structure. The mathematical expression and computational calculation of a FLANN structure is same as MLP. But it possesses higher rate of convergence and lesser computational cost than those of a MLP structure. A wider application of FLANN models for solving non linear problems like channel equalization, non linear dynamic system identification, electric load forecasting, prediction of earthquake, financial forecasting have demonstrated its viability, robustness and ease of computation.

It is well known that the recurrent neural networks (RNNs) usually provide a smaller architecture than most of the nonrecursive neural networks like MLP, RBFNN, etc. Also their feedback properties make them dynamic and more efficient to model nonlinear systems accurately which are imperative for nonlinear prediction and time series forecasting. Many of the Autoregressive Moving Average (ARMA) processes have been accurately modeled by RNNs for nonlinear dynamic system identification. In this paper, therefore, a computationally efficient and robust recurrent FLANN is described for financial time series forecasting.

One of the familiar approaches of training the RNNs is the Real-Time Recurrent Learning (RTRL) [11] algorithm, which has problems of stability and slow convergence. In nonlinear time series forecasting problems it gets trapped in local minima and cannot guarantee to find global minima. On the other hand, evolutionary learning techniques such as differential evolution, particle swarm optimization, genetic algorithm, bacteria foraging, firefly algorithm, etc. have been applied to time series forecasting successively. In recent year a new swarm intelligence technique introduced by Yang [12] in 2008 and it is a metaheuristic bioinspired technique. It is used for optimization problems [13]. It has been used in many time series applications for optimizing errors and to converge quickly. In this paper the simple metaheuristic firefly algorithm is used to train the weights of the RCEFLANN. Two application examples: one in

978-1-4799-7455-9/15/$31.00 ©2015 IEEE

financial market and the other one energy market are considered for prediction using the proposed RCEFLANN and adaptive firefly technique.

II. COMPUTATIONALLY EFFICIENT RECURRENT FLANN (RCEFLANN)

In conventional FLANN models since no output lagging terms are used there is no correlation between the training samples. However, when lagged output samples are used as inputs, a correlation exists between the training samples and hence an incremental learning procedure is required for the adjustment of weights. A recurrent version of this FLANN is shown in Fig.1, where one step delayed output samples are fed to the input to provide a more accurate forecasting scenario.

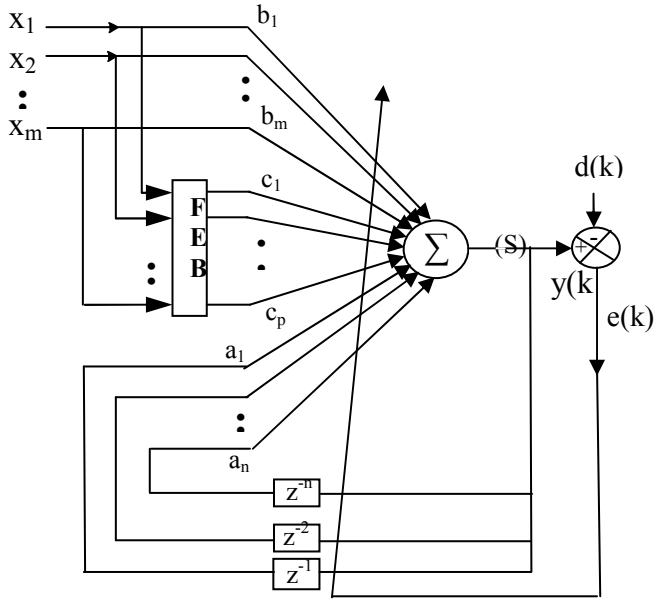

Fig. 1. The Proposed RCEFLANN model

As shown in Fig.1 the input vector contains a total number $m+n+p$ inputs comprising n inputs from the delayed outputs, m inputs from the stock closing price indices, p inputs from the functional expansion. Thus the input vector is obtained as

$$U(k)=[RC^T(k),X^T(k),FE^T(k)]$$

which is written in an expanded form as

$$U(k)=[y_1(k-1),y(k-2),.....y_1(k-n),x_1,x_2.....x_m, \phi_1.\phi_2.....\phi_p] \quad (1)$$

and for a low complexity expansion ϕ_i takes the form

$$\phi_i=\tanh(w_{i0}+w_{i1}x_1+w_{i2}x_2+.....+w_{im}),i=1,2,.....,p$$

Thus the output is obtained as

$$y(k)=\gamma S(u(k)) = \gamma S(W_1(k)RC(k))+W_2(k)X(k)+W_3(k).FE(k)) \quad (2)$$

where

$$W_1=[a_1a_2a_3.....a_n]^T, W_2=[b_1b_2a_3.....b_m]^T,$$
$$W_3=[c_1c_2c_3.....c_p]^T$$

The error $$e(k)=d(k)-y(k) \quad (3)$$

The most common gradient based algorithms used for on-line training of recurrent neural networks are BP algorithms and real-time recurrent learning (RTRL) [25]. The RTRL algorithm is shown in the following steps:

Using the same cost function for the recurrent FLANN model, for a particular weight $w(k)$, the change of weight is obtained as

$$\Delta w(k)=-\eta\frac{\partial e(k)}{\partial w(k)}=\eta e(k)S'(u(k))\frac{\partial y(k)}{\partial w(k)} \quad (4)$$

where η is the learning rate

The partial derivative of the above equation (4) is obtained in a modified form for the RTRL algorithm as

$$\frac{\partial y(i)}{\partial a_k}=y(i-k)+\sum_{j=1}^{n}a_j\frac{\partial y(i-j)}{\partial a_k},k=1,2,.....,n$$
$$\frac{\partial y(i)}{\partial b_k}=x_k+\sum_{j=1}^{n}a_j\frac{\partial y(i-j)}{\partial b_k},k=1,2,.....,m \quad (5)$$
$$\frac{\partial y(i)}{\partial c_k}=\phi_k+\sum_{j=1}^{n}a_j\frac{\partial y(i-j)}{\partial c_k},k=1,2,.....,p$$
$$\frac{\partial y(i)}{\partial w_{k0}}=c_k\sec^2\phi_k+\sum_{j=1}^{n}a_j\frac{\partial y(i-j)}{\partial w_{k0}},k=1,2,.....,p$$

$$\frac{\partial y(i)}{\partial w_{kr}}=c_k x_k\ sec^2\phi_k+\sum_{j=1}^{n}a_j\frac{\partial y(i-j)}{\partial w_{kr}},$$
$$k=1,2,....,\ p,\ r=1,2,....,\ p$$

The weight adjustment formulas are, therefore, obtained as (with k taking values n,m, and p for the weights of the recurrent, input, functional expansion parts, respectively):

$$a_k(i)=a_k(i-1)+\eta e(k)S'(u(k))\frac{\partial y(i)}{\partial a_k}$$
$$b_k(i)=b_k(i-1)+\eta e(k)S'(u(k))\frac{\partial y(i)}{\partial b_k} \quad (6)$$
$$c_k(i)=c_k(i-1)+\eta e(k)S'(u(k))\frac{\partial y(i)}{\partial c_k}$$
$$w_{kr}(i)=w_{kr}(i-1)+\eta e(k)S'(u(k))\frac{\partial y(i)}{\partial w_{kr}}$$

Further if the learning rate is kept small, the weights do not change rapidly and hence

$$\frac{\partial y(i-1)}{\partial a_k} \approx \frac{\partial y(i-1)}{\partial a_k(i-1)} \quad (7)$$

Denoting the gradient

$$\pi_k(i) = \frac{\partial y(i)}{\partial w_k} \quad (8)$$

and the gradient values in successive iterations are generated assuming

$$\pi_k(0) = 0 \quad (9)$$

In the above algorithm the gradient descent based on first order derivatives is used to update the synaptic weights of the network. However, this approach, exhibits slow convergence rates because of the small learning rates required, and most often they become trapped in local minima.. To avoid the common drawbacks of back propagation algorithm and to increase the accuracy, a bioinspired Firefly algorithm is used to train the weight parameters of the RCEFLANN model.

III. SELF ADAPTIVE FIREFLY (FF) ALGORITHM FOR RCEFLANN WEIGHT ADJUSTMENT

The Firefly is a metaheuristic, nature-inspired, population based optimization algorithm that depends on the flashing behavioral patterns of fireflies emitting light. The movement of the individual fireflies towards a global optimal solution is similar to swarms in the particle swarm optimization algorithm. The fireflies are unisex and the less brighter fireflies are attracted to the brighter ones irrespective of their gender. Using the FF algorithm the fireflies are scattered randomly in the search space and their brightness increases with the decreased Euclidean distance between them. The fitness function chosen for optimization determines the brightness of an individual firefly, and for an error minimization problem the firefly with a lower value of the objective function has more brightness or light intensity. Assuming an initial population of fireflies $[x_1, x_2, ..., x_i, ..., x_n], i = 1,2,....., n$, the Euclidean distance between any two fireflies i and j in the search space is obtained as

$$r_{i,j} = \left\| x_i - x_j \right\| = \sqrt{\sum_{k=1}^{d} (x_i - x_j)^2} \quad (10)$$

Since the attractiveness of the fireflies *(β)* depends on the light intensity seen by the nearby fireflies and it varies with the distance between them, it is obtained from equation ( ):

$$\beta = \beta_0 e^{-\gamma r_{ij}^2} \quad (11)$$

where β_0 is the attractiveness at $r = 0$, and γ represents the light absorption coefficient at the source. Further as the fireflies move in the search space the movement of the *i*th firefly to the proximity of a more attractive *j*th firefly is formulated as

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2}(x_j - x_i) + \alpha(rand_i - 0.5) \quad (12)$$

In the above equation the second term signifies attraction whereas the third term involves a random parameter α, which is normally selected in the range [0,1]. For most applications β_0 =1, and γ signifies the variation of the attractiveness that determines the speed of convergence of the FF algorithm. Normally it varies from 0.01 to 100.

However, a modification of the attractiveness parameter β results in faster convergence of the FF algorithm which ultimately results in the avoidance of local optima and this is done as follows [ ]:

$$\beta = (\beta_{\max} - \beta_{\min}) e^{-\gamma (r_{ij})^2} + \beta_{\min} \quad (13)$$

where $\beta_{\max}$ and $\beta_{\min}$ are set to 1.0 and 0.2, respectively; The random movement factor α controls the searching ability of the fireflies and a large value of α facilitates global search similar to PSO, while a small α is responsible for local search. Thus for improved performance of the FF algorithm α is varied from a $\alpha_{\max}$ to $\alpha_{\min}$ as

$$\alpha(k) = \alpha_{\max} - (k / k_{\max})(\alpha_{\max} - \alpha_{\min}) \quad (14)$$

where k is the iteration count $k_{\max}$ is the maximum number of iterations. Also to increase the attractiveness of the fireflies the value of γ is varied as

$$\gamma = \gamma_{\max} \exp((k / k_{\max}) \ln(\gamma_{\min} / \gamma_{\max}) \quad (15)$$

where $\gamma_{\max}$ =maximum attractiveness,

$\gamma_{\min}$ =minimum attractiveness.

Here $\alpha_{\max}$ =0.5, $\alpha_{\min}$ =0.01, $\gamma_{\max}$ =10, $\gamma_{\min}$ =0.1.

A. Pseudo Code for FF Algorithm

Initialize FF algorithm parameters:
$G_{\max}$ = the maximum number of generations
Generate initial population of n fireflies or
$x_i, i = 1,2,....., n$
Obtain light intensity of I_i at x_i from the Objective function $F(x_i)$ from equation (3)
While (generation g <$G_{\max}$)
for i = 1 to n (all n fireflies);
for j=1 to n (all n fireflies)
If ($I_i > I_j$), move firefly i towards j;
end if
Obtain the value of light absorption coefficient γ from equation (15)
Vary the attractiveness β from equation (13) after calculating the distance $r_{i,j}$
Compute the random term α
Update the new value of x_i using equation (12)

update light intensity;
end for *j*;
end for *i*;
Rank the fireflies and find the current best;
end while;

The weights of the recurrent FLANN are obtained from x_i values.

IV. Experimental Result Analysis

In this paper, financial market and energy market are considered for experimental study. In financial market, the S&P 500 and NIKKEI 225 have been selected for day ahead stock price prediction and in energy market (EM), the PJM has been chosen for hour ahead electricity price prediction. The details of the both the markets chosen for experimental study are described below. In the firefly algorithm the total number of inputs for the recurrent FLANN including 3 expansions and 3 recurrent inputs from the output and five original historical data inputs is 5+18+3=26. For each weight 5 fireflies are used and the initial parameters are given in the algorithm description itself. The total number of generations is limited to 100. The Mean Square Error (MSE) is taken as the objective function.

A. Stock PriceDdata

In this study the sample data sets from Standard's & Poor's 500 (S&P500) and from NIKKEI 225 comprising the daily closing prices have been selected for experimenting and measuring the performances of different FLANN models trained using evolutionary methods. The total no. of samples for S&P500 is 1066 from 2 January 2009 to 15 March 2013. The dataset is divided into two sets: one for training consisting of 500 data from 2 January 2009 to 13 December 2010 and another set for testing consisting of 400 data from 14 December 2010 to 16 July 2012). The NIKKEI 225 dataset contains total 1000 data out of which the initial 500 data covering the period from 5 January 2009 to 17 December 2010 are taken for training and the following 400 data covering the period from 20 December 2010 to 03 August 2012 are chosen for testing.

B. Electricity Price Data

The day ahead electricity price prediction has been done using the Pennsylvania–New Jersey–Maryland (PJM) energy market. The PJM market consists of 8760 hourly electricity prices data (365 × 24) covering the period from 1 January to 31 December 2013. The hourly behaviour of the dataset for the year 2013 shown in Fig.4(a) clearly exhibits the complex behaviour of the electricity price fluctuation with significant outliers. Three different weeks in the month of May, August, and December have been chosen for day ahead electricity price prediction of PJM market. The model is first trained using 264 samples covering the period from 20th April to 30th April 2013 and and testing is started using the next 168 samples covering the 1st week of May (1-7 May 2013).For predicting the price for 1st week of August (1-7 August 2013), the training samples include data from 21st July to 31st July 2013 and training samples from 10th to 20th December are used for predicting the December end week (21-27 December 2013) price variations.

The data are normalized between 0 and 1 using the following formula:

$$y = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{16}$$

where y = normalized value.
x = value to be normalized
x_{min} = minimum value of the series to be normalized
x_{max} = maximum value of the series to be normalized

The Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) are used to compare the performance of different evolutionary FLANNs for predicting the closing price of the S&P500 and NIKKEI 225 index in one day, 7 days, 15 days, 30 days, 60 days in advance with different learning algorithms. The RMSE, MSE, and MAPE are defined as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{n}\left(y_k - \hat{y}_k\right)^2} \tag{17}$$

$$MSE = \frac{1}{N}\sum_{k=1}^{n}\left(y_k - \hat{y}_k\right)^2 \tag{18}$$

$$MAPE = \frac{1}{N}\sum_{k=1}^{n}\left|\frac{y_k - \hat{y}_k}{y_k}\right| \times 100 \tag{19}$$

where y_k = actual closing price on kth day
$\hat{y}_k$ = predicted closing price on *k*th day
N = number of test data.
3- days Moving Average(MA)

Moving average (eq.(19)) for *m* days is used to emphasize the direction of a trend and smooth out price

$$MA(m) = \sum_{i=t-m}^{t} y_i / m \text{ , } y_i \text{=closed price} \tag{20}$$

Standard Deviation (SD)

$$SD(\sigma) = \sqrt{\sum_{i=1}^{n}(y_i - \mu)^2 / m} \text{ , } \mu = \sum_{i=1}^{n} y_i / m \tag{21}$$

The stock price prediction results for S&P500 and NIKKEI 225 are shown in Figs. 2 and 3, respectively. Fig.5 (a) shows the chaotic behavior of the S&P500 stock market. Fig.2(b) shows the prediction results of S&P500 for the selected testing samples and Fig.2(c) shows the corresponding error found in testing. Similarly the NIKKEI 225 prediction result is shown in Fig.3. The performances measured in terms of MAPE and RMSE are presented in Table-1.

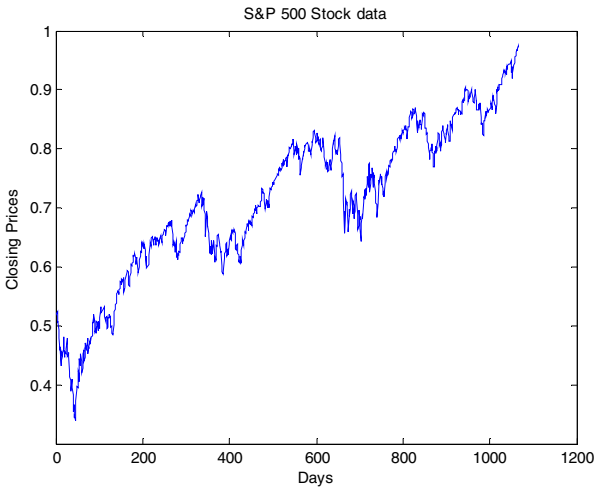

Fig.2(a) S&P 500 stock market data from 2nd Jan 2009 to 15th March 2013

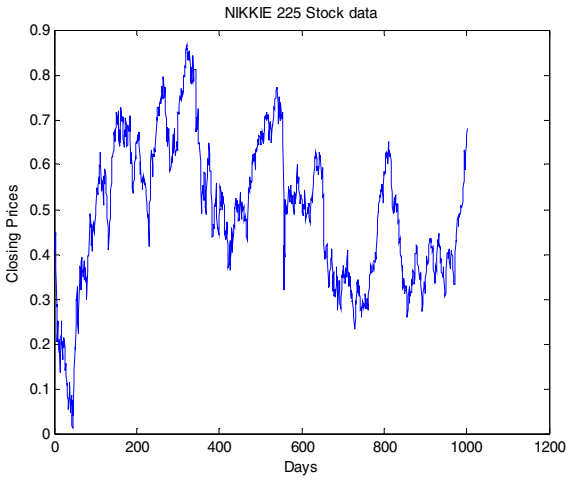

Fig.3(a) NIKKIE 225 stock market data from 2nd Jan 2009 to 15th March 2013

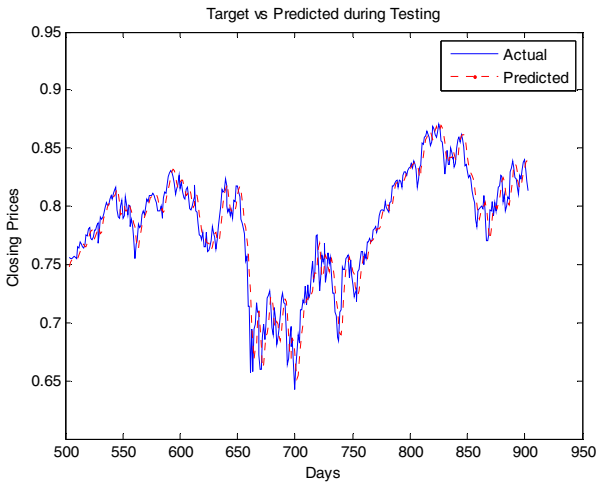

Fig.2(b) S&P 500 stock price prediction results during Testing

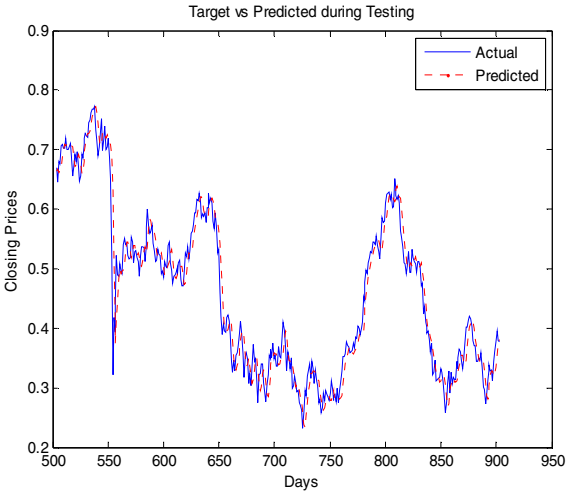

Fig.3(b) NIKKIE 225 stock price prediction results during Testing

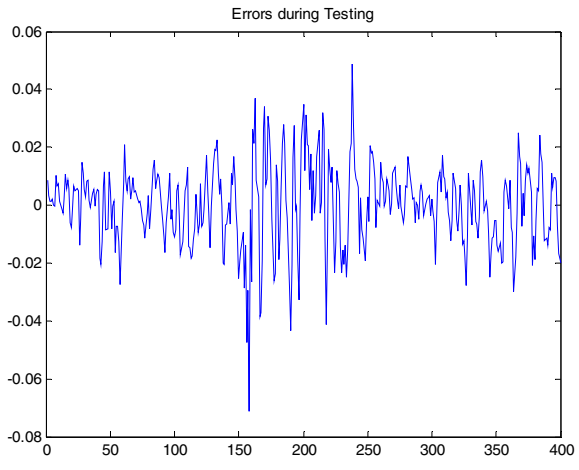

Fig.2(c) Testing errors of S&P 500 stock

Fig.2 S&P 500 Prediction Results

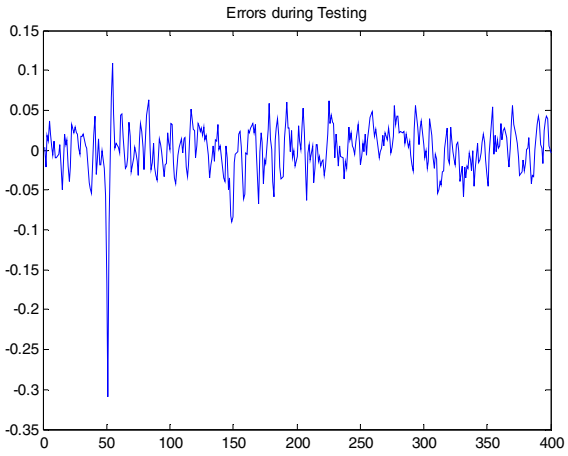

Fig.3(c) Testing errors of NIKKIE 225stock

Fig.3 NIKKIE 225 Prediction Results

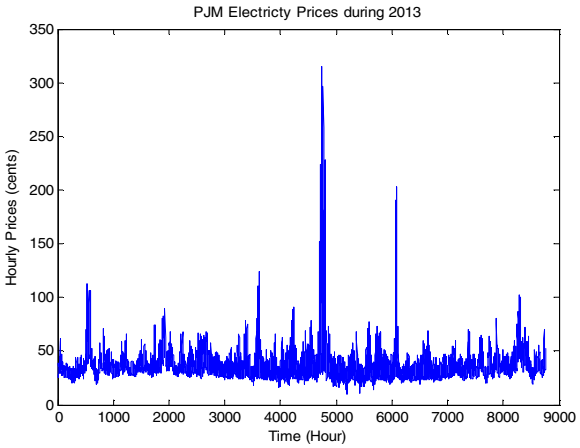

Fig.4(a) PJM Electricity prices during the year 2013

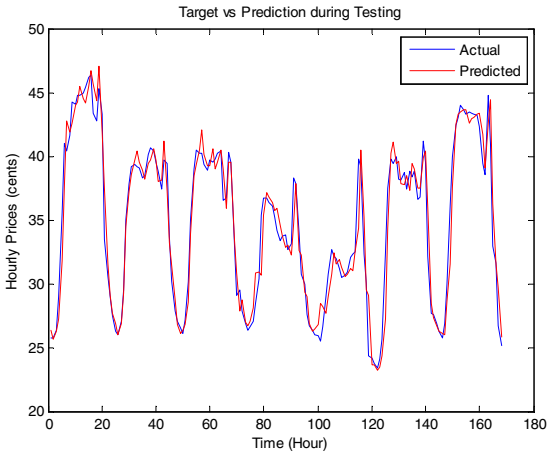

Fig.4(b) Testing results of PJM from 1-7 May 2013

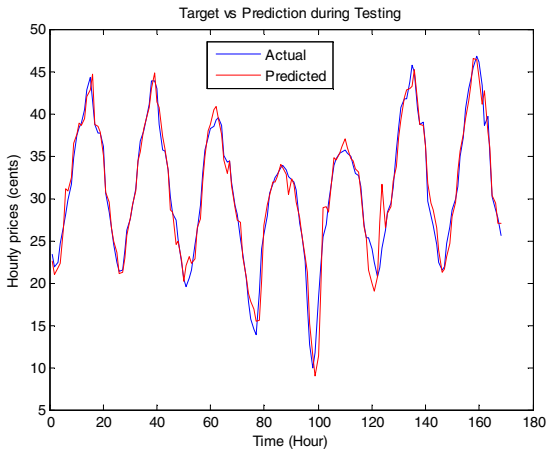

Fig.4(c) Testing results of PJM from 1-7 Aug. 2013

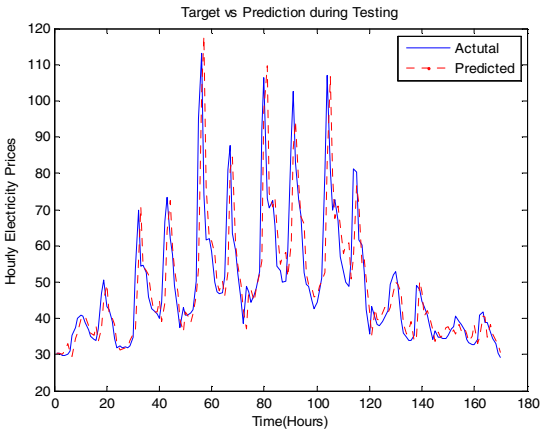

Fig.4(d) Testing results of PJM from 21-27 Dec. 2013

Fig.4. PJM prediction results

TABLE-I PERFORMANCE MEASUREMENTS FOR STOCK PRICE PREDICTION

Performance Metrics	S&P 500		NIKKEI 225	
	Firefly	RTRL	Firefly	RTRL
MAPE	1.4142	2.683	2.4482	3.152
MSE	0.00019	0.0067	0.0011	0.0145
RMSE	0.0141	0.082	0.0325	0.122

From Table-I, it is revealed that the proposed adaptive Firefly learning algorithm produces more accurate prediction metrics in comparison to the RTRL algorithm using gradient descent.

The same RCEFLANN model is taken to predict the day ahead electricity price for PJM energy market using both the RTRL and adaptive Firefly algorithms . In this case, the inputs to the system comprise the lagged return prices and the series is defined as $(R_{t-23}, R_{t-24}, R_{t-47}, R_{t-48}, R_{t-71}, R_{t-72}, R_{t-95}, R_{t-96})$ for predicting the price at time t (hour). Once the training is completed, the testing for different weeks are started and accomplished. The performances are measured in terms of MAPE (18), AMPAE, and error variance (21) and tabulated in Table-2 for PJM.

The error variance in a time span T is defined as:

$$\sigma_e^2 = \frac{1}{T}\sum_{i=1}^{T}\left[\left\{\frac{y_i - \hat{y}_i}{\left(\frac{1}{T}\sum_{i=1}^{T} y_i\right)}\right\} - MAPE\right]^2 \qquad (22)$$

Fig. 5(a) shows the PJM historical electricity prices during the year 2013. The actual vs predicted results for the three weeks in May, August, and December are shown in Figs. 5(b), 5(c), and 5(d), respectively. The MAPE, AMAPE, and error variance are comparatively smaller than the December week prediction results as the prices

fluctuation is very significant in December. To verify the model in highly fluctuated data; in December we have chosen the end week for prediction. Tables II and III exhibit the prediction results using the Firefly and RTRL algorithms, respectively and they clearly reveal the superiority of the adaptive firefly algorithm in producing more accurate weekly price forecast.

TABLE-II PERFORMANCE USING FIREFLY (ELECTRICITY PRICE)

EM	Predicted weeks	WMAPE	AMAPE	Error Variance
PJM	May 1 to 7, 2013	3.0141	3.0644	0.0011
	Aug 1 to 7 2013	3.4198	3.3766	0.0016
	Dec 21 to 27, 2013	7.1170	6.3639	0.0038

TABLE-III PERFORMANCE USING RTRL LEARNING (ELECTRICITY PRICE)

EM	Predicted weeks	WMAPE	AMAPE	Error Variance
PJM	May 1 to 7, 2013	4.6141	4.873	0.006
	Aug 1 to 7 2013	5.317	5.296	0.0052
	Dec 21 to 27, 2013	9.26	8.582	0.0079

V. CONCLUSION

In this his paper, we have developed a recurrent FLANN based model with a combination of linear and hyperbolic tangent function for time series prediction. This model is simpler and shown to be effective in predicting the day ahead stock price of S&P500 and NIKKEI 225 as well as day ahead electricity price prediction of PJM market. Two learning paradigms namely the RTRL and Firefly are used for training the weights of the recurrent FLANN model. Further several parameters of the bioinspired metaheuristic Firefly algorithm are tuned adaptively to produce better forecasting performance metrics. In comparison to many standard neural architectures the proposed approach is simple and takes less computational overhead to produce satisfactory prediction results. Day ahead stock price and electricity price prediction are implemented using the proposed approach validating its satisfactory performance.

REFERENCES

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

[2] S. Chakravarty, 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, Vol.12, no.2, February 2012, pp.931–941.

[3] Lijuan Cao and Francis E.H. Tay, Financial Forecasting Using Support Vector Machines, Neural Computing and Applications, Vol.10, pp.184–192, 2001.

[4] Yi-Fan Wang, Mining stock price using fuzzy rough set system, Expert Systems with Applications, Vol.24, no.1, January 2003, pp. 13–23.

[5] Michał Paluch, Lidia Jackowska-Strumiłło, The influence of using fractal analysis in hybrid MLP model for short-term forecast of close prices on Warsaw Stock Exchange, Proceedings of the 2014 Federated Conference on Computer Science and Information Systems pp. 111–118.

[6] Wei Shen, Xiaopen Guo, Chao Wub, Desheng Wuc, Forecasting stock indices using radial basis function neural networks optimized by artificial fish swarm algorithm, Knowledge-Based Systems, Vol.24, 2011, pp.378–385.

[7] Tsung-Jung Hsieh, Hsiao-Fen Hsiao, Wei-Chang Yeh, Forecasting stock markets using wavelet transforms and recurrent neural networks: An integrated system based on artificial bee colony algorithm, Applied Soft Computing, Vol.11, no.2, March 2011, pp.2510–2525.

[8] Akhter Mohiuddin Rather, Arun Agarwal, V.N. Sastry, Recurrent neural network and a hybrid model for prediction of stock returns, Expert Systems with Applications, Vol.42, no. 6, April 2015, pp.3234–3241.

[9] B. Majhi, S. Hasan, F. Mowafak, FLANN based forecasting of S&P 500 index, Information Technology Journal, Vol.4, no.3, 2005, pp.289–292.

[10] Erkam Guresen, Gulgun Kayakutlu, Tugrul U. Daim, Using artificial neural network models in stock market index prediction, Expert Systems with Applications, Vol.38, no.8, August 2011, pp.10389–10397.

[11] Campolucci, P., Uncini, A., Piazza, F. and Rao, B.D., On-line learning algorithms for locally recurrent neural networks, IEEE Transactions on Neural Networks, Vol.10, no. 2, 1999, pp.253–271.

[12] Yang, X. S. Firefly algorithms for multimodal optimization in stochastic algorithms: Foundations and applications. In SAGA. Lecture Notes in Computer Sciences, Vol. 5792, pp. 169–178, 2009.

[13] Ahmad Kazem, Ebrahim Sharifi, Farookh Khadeer Hussain, Morteza Saberi, Omar Khadeer Hussain, Support vector regression with chaos-based firefly algorithm for stock market price forecasting, Applied Soft Computing, Vol.13, no.2, February 2013, pp. 947–958.