Chaotic Time Series Forecasting Using Higher Order Neural Networks

Waddah Waheeb[#][*], Rozaida Ghazali[#]

[#] *Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Batu Pahat, Johor, Malaysia*

[*] *Computer Science Department, Hodeidah University, P.O. Box 3114 Alduraihimi, Hodeidah, Yemen*
E-mail: waddah.waheeb@gmail.com; rozaida@uthm.edu.my

Abstract—**This study presents a novel application and comparison of higher order neural networks (HONNs) to forecast benchmark chaotic time series. Two models of HONNs were implemented, namely functional link neural network (FLNN) and pi-sigma neural network (PSNN). These models were tested on two benchmark time series; the monthly smoothed sunspot numbers and the Mackey-Glass time-delay differential equation time series. The forecasting performance of the HONNs is compared against the performance of different models previously used in the literature such as fuzzy and neural networks models. Simulation results showed that FLNN and PSNN offer good performance compared to many previously used hybrid models.**

Keywords— **Chaotic time series; Sunspot time series; Mackey-Glass time series; higher order neural network; pi-sigma neural network; functional link neural network.**

I. INTRODUCTION

Time series forecasting is very important in many applications such as financial forecasting, weather forecasting, traffic forecasting, etc. Time series forecasting aims to build a model that take advantage of past observations to forecast the future. Time series in nature is usually non-linear or chaotic [1]. According to [2], chaotic system has four fundamental characteristics: aperiodic, bounded, sensitivity to initial conditions and deterministic. Aperiodic means that the same state will not be repeated, bounded indicates that neighbour states keep within a finite range and does not approach infinity, sensitivity to initial conditions meaning that small changes in initial conditions will cause divergence between two close points as the state of system progress, and deterministic means there is a rule with no random term to forecast the future state of the system. Chaotic time series forecasting has been observed in many areas such as power load [3], marketing system [4], exchange rate [5], etc.

A number of methods have been used to forecast chaotic time series in the literature such as support vector echo-state machines [1], self-organizing map [6], and fuzzy and neuro-fuzzy [7]–[10].

Artificial neural networks (ANN) have been also used to forecast chaotic time series. ANN is an intelligent-based approach which is inspired by biological nervous systems; it can learn from historical data and adjust its weight matrices to build model that can predict the future.

Different types of ANN have been utilized for chaotic time series with varying degrees of success including Elman–Nonlinear Autoregressive with eXogenous input neural networks [2], beta basis function neural networks [11], orthogonal function neural network [12], radial basis function [13], multilayer perceptron network [14].

Multi-layered ANN structure needs a large number of units to deal with complex mapping problems, which results in low learning rate and poor generalization [15]. To overcome the drawbacks of multilayered networks, different higher order neural networks (HONNs) with a single layer were introduced.

HONNs utilize higher order terms (i.e., product units) which allow them to transform the input space into a higher dimensional space in which linear separability is possible, thus reducing the complexity of the network [16]. Unlike multi-layered ANN structure, HONNs have only one single layer of hidden nodes which helps to accelerate the training.

Different types of HONNs have been used for time series forecasting [16]–[20] but no much attention has been paid to apply HONNs to forecast benchmark chaotic time series and compare their performance with other existing models.

In summary, the contribution of this work as follows:

- Application of two HONNs namely, Functional Link Neural Network (FLNN) and Pi-Sigma Neural Network to forecast two benchmark chaotic time series: the monthly sunspot number and the Mackey-Glass time-delay differential equation time series.

- Comparison of the forecasting performance of these models with other existing models.

The remainder of this paper is organized as follows. Section 2 describes a brief about HONNs, FLNN and PSSN. The experimental design used in this work is also discussed in this section. In section 3 we present the results and discussion. Finally, the conclusion is given in section 4.

II. PROPOSED MODELS AND METHOD

This section gives a brief about HONNs, FLNN and PSNN. Furthermore, it shows experimental design steps that we used in this works including the used time series, data preprocessing, network topology and training, and the evaluation metrics.

A. Higher Order Neural Networks (HONNs):

HONNs are feedforward neural network with a combination of summing and product units. They can expand the non-linear input space into higher dimensional space where linear separability is possible [16].

Using product units can increase the information capacity of the network thus helping to deal with complex problems with smaller network structure. As a results of the simple architectures of HONNs, they reduce the number of free parameters thus they can learn faster [16]. However, some HONNs suffer from the combinatorial explosion of the higher order terms and demonstrate slow learning [16].

This paper uses two HONNs models namely, the Functional Link Neural Network and Pi-Sigma Neural Network. With different strength and capabilities, the characteristic and structure of these networks are presented below.

B. Functional Link Neural Networks (FLNN):

FLNN was introduced by Giles and Maxwell [21]. FLNN extends the structure of feedforward network by introducing supplementary inputs to the network. Therefore, the hyperplane generated by the FLNN provides greater discrimination capability in the input pattern space [22]. FLNN has been used for different problems such as classification [23], system identification [24] and time series forecasting [18].

There are two common models of the FLNN: tensor product and functional expansion [22]. In the former model as shown in Fig. 1, each component of the input vector is multiplied by other components of this input vector. In other words, instead of describing input patterns in terms of a set of components $\{x_i\}$, it is described as $\{x_i, x_{ij}\}$, where $j \geq i$, or as $\{x_i, x_{ij}, x_{ijk}\}$, where $k \geq j \geq i$, and so on. Therefore, no new information has been added, but joint activations have been made available to the network. The latter model, which is shown in Fig. 2, expands the dimension of the inputs by choosing an appropriate set of functions to deal with the problem at hand. The problem with this model is that choosing a good set of functions to expand input dimensions is difficult [25]. Therefore, in this work we only consider the FLNN with tensor product.

The FLNN in Fig. 1 is an example of a third order FLNN. It consists of three external inputs and four high order inputs. The learning algorithm for FLNN using the incremental backpropagation algorithm is as follows:
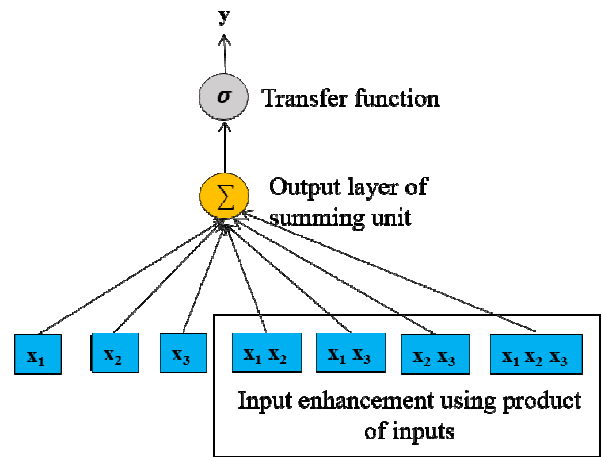

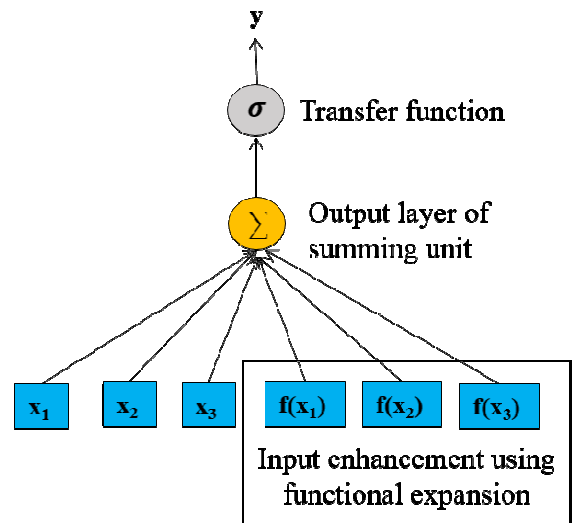Fig. 1 Functional link neural network of type tensor product model.

Fig. 2 Functional link neural network of type functional expansion model.

For a given input,
- Calculate the output as follows:

$$y = \sigma(W_0 + \sum_i W_i x_i + \sum_i \sum_j W_{ij} x_i x_j + \sum_i \sum_j \sum_k W_{ijk} x_i x_j x_k)$$
(1)

where σ is an activation function, W_0 are the biases, W_i, W_{ij} and W_{ijk} are weights that link input nodes with the output node, x is a component of input vector X.
- Compute the weight changes:

$$\Delta W_i = \eta(d_i - y_i) y_i (1 - y_i) X_k$$
(2)

where η is the learning rate and d is the desired output.
- Update the weights:

$$W_i = W_i + \Delta W_i$$
(3)

- Continue until termination condition is satisfied.

C. Pi-Sigma Neural Networks (PSNN):

PSNN is a feedforward neural network with one layer of trainable weights. PSNN calculates its output as product of sum of the input components [26]. The motivation to develop PSNN was to develop a model which maintains fast learning property and powerful mapping capability whilst

avoiding the combinatorial explosion in the number of free parameters that occurs in FLNN.

As shown in Fig. 3, PSNN consists of two layers; the product layer and the summing layer. The trainable weights are found only between the inputs and the summing units. The structure of PSNN is highly regular due the fact that the summing units can be added incrementally until a specified goal is achieved.
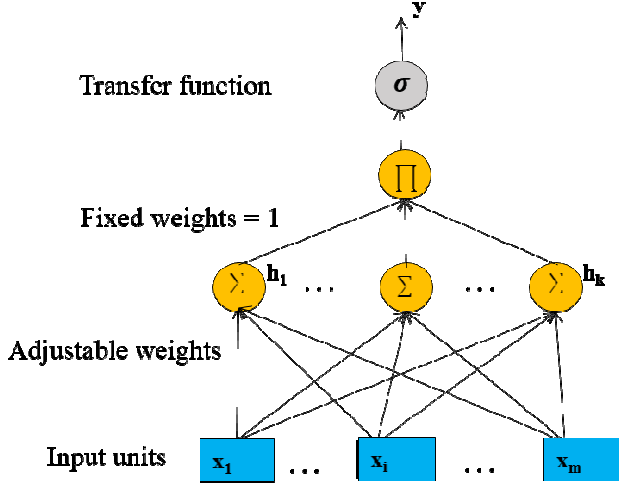

Fig. 3 Pi-Sigma neural network.

Despite the fact that PSNN is not a universal approximator [27], it demonstrated competent ability to deal with many problems such as classification [28], time series forecasting [19], image coding [29] and visual cryptography [30].

The learning algorithm for PSNN using the incremental backpropagation algorithm is as follows:
For a given input,
• Calculate the output as follows:

$$y = \sigma(\prod_{j=1}^{k} h_j) \qquad (4)$$

$$h_j = \sum_{i=1}^{m} W_{ij}X_i + W_{0j} \qquad (5)$$

where σ is an activation function, W_{0j} are the biases, W_{ij} are weights that link input nodes with the summing nodes, x is a component of input vector X.
• Compute the weight changes:

$$\Delta W_i = \eta(d_i - y_i)y_i(1 - y_i)\prod_{j \neq i}^{M} h_j X_k \qquad (6)$$

where η is the learning rate and d is the desired output.
• Update the weights:

$$W_i = W_i + \Delta W_i \qquad (7)$$

• Continue until termination condition is satisfied.

D. Experimental Design:

1) *Time Series benchmark data:* we used two benchmark chaotic time series, namely the monthly smoothed sunspot numbers and the Mackey-Glass time-delay differential equation time series.

Sunspot time series is a good indication of solar activity for solar cycles [31]. It is very important to forecast Sunspot time series due to the observed impact of solar activity on earth, climate, weather, satellites and space missions [31].

In this paper, we downloaded the monthly smoothed sunspot time series from [32]. To compare the performance of FLNN and PSNN with other models in [31], two thousands points from November 1834 to June 2001 were selected.

Mackey-Glass time series is a benchmark problem that has been used by many researchers [11]–[14]. This time series is given by the following delay differential equation:

$$\frac{dx}{dt} = \beta x(t) + \frac{\alpha x(t-\tau)}{1 + x^{10}(t-\tau)} \qquad (8)$$

where α=0.2, β=−0.1, $x(0)$=1.2, and τ=17. With this setting the series produce chaotic behaviour and we can compare the forecasting performance of FLNN and PSNN with other models in the literature. This time series can be found in mgdata.dat in MATLAB [33].

The input-output data pairs and the number of training and testing samples that we used in this paper for these two time series are shown in Table I. Fig. 4 and Fig. 5 show the used interval for training and testing samples for both time series.

TABLE I
TIME SERIES INFORMATION

Time series	Input-output data pairs	Training samples#	Out-of-sample Samples
Sunspot	[$x(t$ - 4), $x(t$ - 3), $x(t$ - 2), $x(t$ - 1), $x(t)$; $x(t$ + 1)]	1000	1000
Mackey-Glass	[$x(t$ - 18), $x(t$ - 12), $x(t$ - 6), $x(t)$; $x(t$ + 6)]	500	500

2) *Data Preprocessing:* We scaled the points to the range [0.2, 0.8] because we used the sigmoid activation function. We used the minimum and maximum normalization method which is given by:
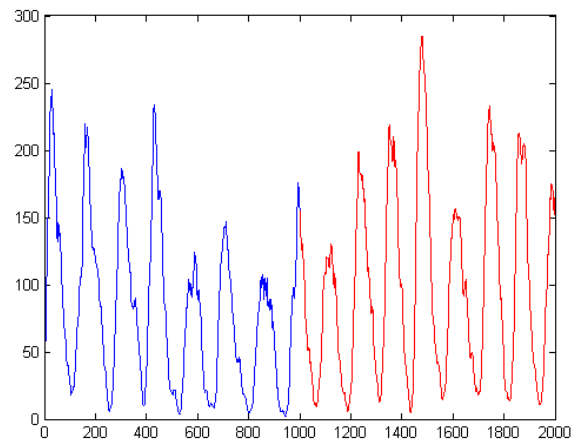

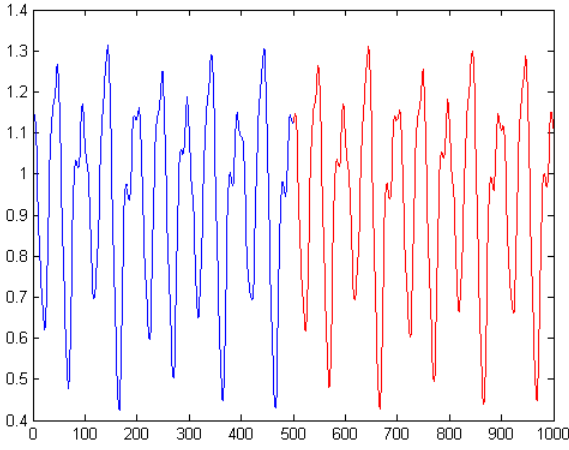Fig. 4 Sunspot time series. Blue points for training while red points for testing.

Fig. 5 Mackey-Glass time series. Blue points for training while red points for testing.

$$\hat{x} = (\max_2 - \min_2) * \left(\frac{x - \min_1}{\max_1 - \min_1} \right) + \min_2 \qquad (9)$$

where $\hat{x}$ is the normalized value of x, min_1 and max_1 are the minimum and maximum values of all observations, and min_2 and max_2 refer to the minimum and maximum values of the new range.

3) Network Topology and Training: The topology of the FLNN and PSNN that we used is shown in Table 2. Most of the settings are selected empirically.

TABLE II
NETWORK TOPOLOGY

Setting	Value
Activation function	Sigmoid function
PSNN order	Empirically selected from 2 to 15
FLNN order	Empirically selected between 2-5 for Sunspot and between 2-4 for Mackey-Glass
Stopping criteria	Maximum number of epochs =3000
Initial weights	[-0.5,0.5]
Learning rate	[0.01-1]
Momentum	[0.4-0.8]

4) *Performance Metrics:* Due we aim to compare the forecasting performance of FLNN and PSNN with other models in the literature, we used the Normalized Mean Squared Error (NMSE) and the Root Mean Squared Error (RMSE) metrics. NMSE and RMSE are given by:

$$NMSE = \frac{1}{N\sigma^2} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

$$\sigma^2 = \frac{1}{N-1} \sum_{i=1}^{N} (y_i - \bar{y})^2 \qquad (10)$$

$$\bar{y} = \frac{1}{N-1} \sum_{i=1}^{N} y_i$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2} \qquad (11)$$

where N, y and $\hat{y}$ represent the number of out-of-sample data, actual output and network output, respectively.

III. RESULTS AND DISCUSSION

The forecasting models for FLNN and PSNN of the two time series are built via the experimental design settings. In order to obtain fair comparison between FLNN and PSNN and avoid weight initialization influence, the average performance of 30 simulations are reported as shown in Table III and Table IV. Note that, the results that are shown in these two tables are the de-normalized results. That means, we de-normalized the forecasted value and compared it with the original desired value before calculating the used metrics. Best average results are in boldface.

As can be seen from Table III and Table IV, FLNN outperform PSNN on Sunspot time series, while PSNN is better than FLNN on Mackey-Glass. Therefore, each one has its ability based on the time series properties.

TABLE III
AVERAGE RESULTS FOR SUNSPOT TIME SERIES

HONN model	Network order	Number of parameters	RMSE	NMSE
FLNN	5	32	**2.7398**	**0.0015**
PSNN	2	12	4.6745	0.0045

TABLE IV
AVERAGE RESULTS FOR MACKEY-GLASS TIME SERIES

HONN model	Network order	Number of parameters	RMSE	NMSE
FLNN	3	15	0.0369	0.0267
PSNN	6	30	**0.0129**	**0.0033**

During the simulations, we noticed that increasing network order of PSNN results in decreasing forecasting performance on Sunspot time series but it helps PSNN on Mackey-Glass time series. Learning curves for the best simulations are shown in Fig. 6 and Fig. 7. It can be seen that there is no much improvement in learning after 500 epochs. Note that, we do not de-normalize the RMSE in Fig. 6 and Fig. 7.

The best performance with FLNN and PSNN using the out-of-sample data for Sunspot and Mackey-Glass time series are shown in Fig. 8 to Fig. 11. As it can be noticed from these figures that FLNN and PSNN to some extent can follow the dynamics behaviour of the time series.
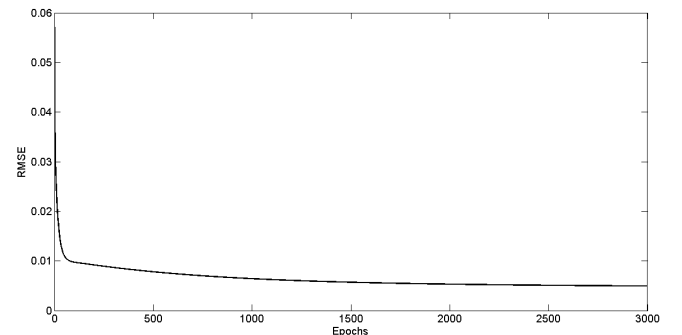

Fig. 6 Learning curve for best FLNN simulation on Sunspot time series.
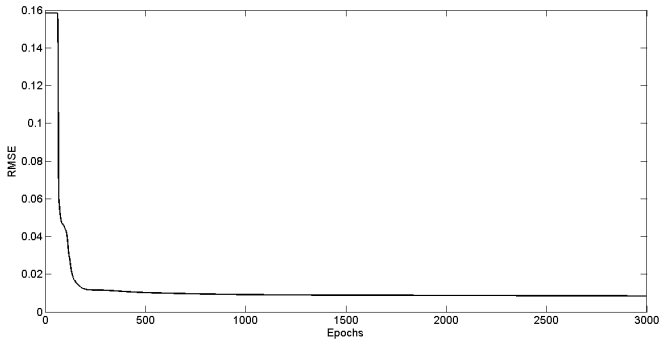
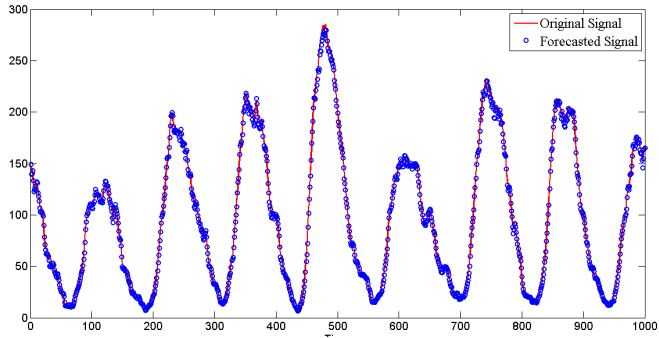Fig. 7 Learning curve for best PSNN simulation on Mackey-Glass time series.

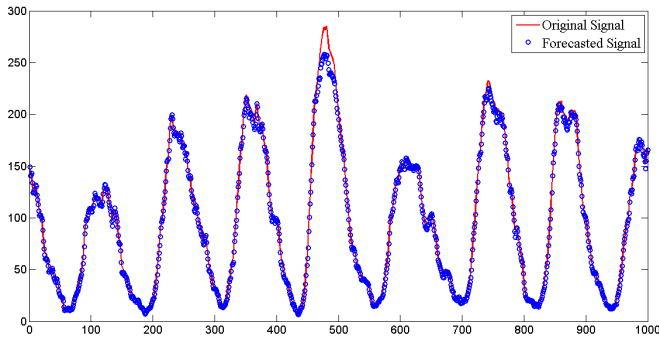Fig. 8 Out-of-sample forecasting for best FLNN simulation on Sunspot time series.

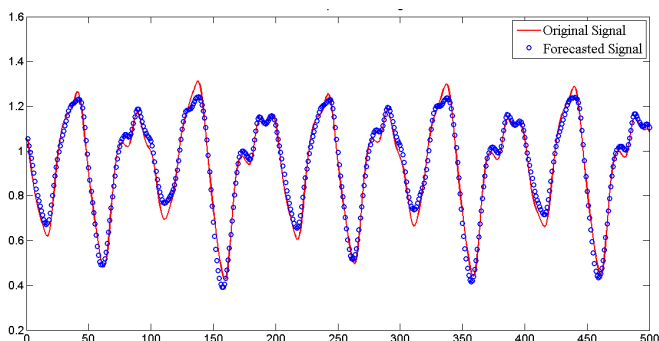Fig. 9 Out-of-sample forecasting for best PSNN simulation on Sunspot time series.

Fig. 10 Out-of-sample forecasting for best FLNN simulation on Mackey-Glass time series.

Finally, a comparison among different models in the literature with FLNN and PSNN is shown in Table V and Table VI. It should be noted that based on our search we could not find studies that used the normalization range that we used in this work. For that, we used the de-normalized results for best FLNN and PSNN simulations and compared them with the de-normalized results in the literature or with

studies that did not use any normalization method. The results show that FLNN and PSNN offer good performance compared to other hybrid models in the literature. Therefore, hybridizing other models with FLNN or PSNN could enhance the forecasting performance.
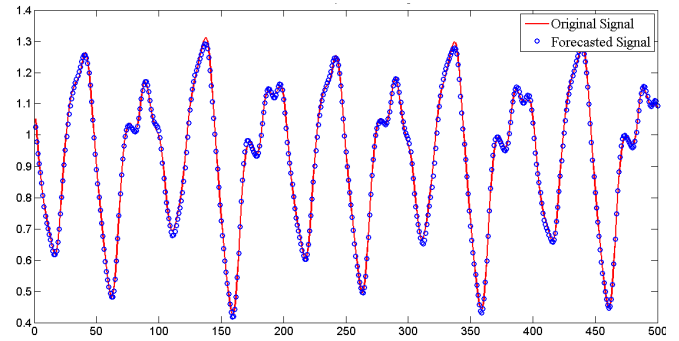

Fig. 11 Out-of-sample forecasting for best PSNN simulation on Mackey-Glass time series.

TABLE V
COMPARISON OF THE PERFORMANCE OF VARIOUS EXISTING MODELS ON SUNSPOT TIME SERIES

Model	NMSE
Fuzzy neural networks (FNN) [31]	0.0174
AdaBoost.regression and threshold-FNN [31]	0.0160
Modified-AdaBoost.regression and threshold-FNN [31]	0.0135
PSNN -Order 2 (this work)	**0.0044**
FLNN -Order 5 (this work)	**0.0015**

TABLE VI
COMPARISON OF THE PERFORMANCE OF VARIOUS EXISTING MODELS ON MACKEY-GLASS TIME SERIES

Model	RMSE
Fuzzy modelling method with Singular Value Decomposition (SVD) [7]	0.0894
Gustafson-Kessel fuzzy clustering method + Kalman Filtering Algorithm (KFA) with SVD [8]	0.0748
Orthogonal function neural network + recursive KFA based on SVD [34]	0.05099
Adaptive fuzzy inference system with local search for learning algorithm [9]	0.045465
FLNN -Order 4 (this work)	**0.03656**
Beta basis function neural networks + Differential evolution algorithm [11]	0.030
Dynamic evolving computation system [35]	0.0289
Backpropagation network + hybrid K-means-greedy [14]	0.015
Modified differential evolution + radial basis function [13]	0.013
PSNN -Order 6 (this work)	**0.0118**
Takagi-Sugeno fuzzy system-singleton + simulated annealing [10]	0.00898

IV. CONCLUSIONS AND FUTURE WORKS

This paper presents the application of two higher order neural networks, namely functional link neural network (FLNN) and pi-sigma neural network (PSNN) to forecast two benchmarks chaotic time series; the monthly smoothed sunspot numbers and the Mackey-Glass time series. Results showed that FLNN outperforms PSNN on Sunspot time

series while PSNN is better than FLNN on Mackey-Glass time series. Furthermore, a comparison with different hybrid models in the literature showed that FLNN and PSNN offer good performance compared to these hybrid models. Future works could be hybridizing swarm intelligence techniques with FLNN or PSNN and applying them to forecast chaotic time series.

REFERENCES

[1] Z. Shi and H. Min, "Support vector echo-state machine for chaotic time-series prediction," *IEEE Transactions on Neural Networks,* vol. 18, pp. 359-372, 2007.

[2] M. Ardalani-Farsa and S. Zolfaghari, "Chaotic time series prediction with residual analysis method using hybrid Elman–NARX neural networks," *Neurocomputing,* vol. 73, pp. 2540-2553, 2010.

[3] P. Qingle and Z. Min, "Very short-term load forecasting based on neural network and rough set," in *Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on*, vol. 3, pp. 1132-1135. IEEE, 2010.

[4] B. Qian and K. Rasheed, "Stock market prediction with multiple classifiers," *Applied Intelligence, vol.* 26, pp. 25-33, 2007.

[5] A. Das and P. Das, "Chaotic analysis of the foreign exchange rates," *Applied mathematics and computation, vol.* 185, pp. 388-396, 2007.

[6] G. A. Barreto, "*Time series prediction with the self-organizing map: A review*," In Perspectives of neural-symbolic integration, pp. 135-158, Springer Berlin Heidelberg, 2007.

[7] Y. Wen and H. Wang, "Fuzzy prediction of time series based on Kalman filter with SVD decomposition," In *Fuzzy Systems and Knowledge Discovery, 2009. FSKD'09. Sixth International Conference on*, vol. 4, pp. 458-462, IEEE, 2009.

[8] H. Wangand J. Lian, "Fuzzy prediction of chaotic time series based on fuzzy clustering," *Asian Journal of Control,* vol. 13, pp. 576-581, 2011.

[9] H. Zhang and X. N. Liu, "Local search for learning algorithm in adaptive fuzzy inference system," In *Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on*, pp. 93-96, IEEE, 2012.

[10] M. Almaraashi and R. John, "Tuning of type-2 fuzzy systems by simulated annealing to predict time series," In *Proceedings of the World Congress on Engineering*, vol. 2, pp. 976-980, 2011.

[11] H. Dhahri and A. Alimi, "Automatic selection for the beta basis function neural networks," In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2007),* pp. 461-474, Springer Berlin Heidelberg, 2008.

[12] H. Wang and H. Gu, "Prediction of chaotic time series based on neural network with Legendre polynomials," In *International Symposium on Neural Networks*, pp. 836-843, Springer Berlin Heidelberg, 2009.

[13] H. Dhahri and A. M. Alimi. "The modified differential evolution and the RBF (MDE-RBF) neural network for time series prediction," In *the 2006 IEEE International Joint Conference on Neural Network Proceedings*, pp. 2938-2943, IEEE, 2006.

[14] J. Y. B. Tan, D. B. L. Bong, and A. R. H. Rigit, "Time Series Prediction using Backpropagation Network Optimized by Hybrid K-means-Greedy Algorithm," *Engineering Letters,* vol. 20, pp. 203-210, 2012.

[15] X. Yu, L. Tang, Q. Chen, and C. Xu, "Monotonicity and convergence of asynchronous update gradient method for ridge polynomial neural network," *Neurocomputing*, vol. 129, pp. 437-444, 2014.

[16] R. Ghazali, A. Hussain, and W. El-Deredy, "Application of ridge polynomial neural networks to financial time series prediction," In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pp. 913-920, IEEE, 2006.

[17] R. Ghazali, A. J. Hussain, and P. Liatsis. "Dynamic Ridge Polynomial Neural Network: Forecasting the univariate non-stationary and stationary trading signals," *Expert Systems with Applications,* vol. 38, pp. 3765-3776, 2011.

[18] Y. M. M. Hassim and R. Ghazali, "Functional link neural network–artificial bee colony for time series temperature prediction." In *International Conference on Computational Science and Its Applications*, pp. 427-437, Springer Berlin Heidelberg, 2013.

[19] N. A. Husaini, R. Ghazali, L. H. Ismail, and T. Herawan, "A Jordan Pi-Sigma Neural Network for Temperature Forecasting in Batu Pahat Region," In *Recent Advances on Soft Computing and Data Mining*, pp. 11-24, Springer International Publishing, 2014.

[20] W. Waheeb, R. Ghazali, and T. Herawan, "Time series forecasting using ridge polynomial neural network with error feedback," In *The Second International Conference on Soft Computing and Data Mining (SCDM-2016),* in press.

[21] C. L. Giles and T. Maxwell, "Learning, invariance, and generalization in high-order neural networks," *Applied optics,* vol. 26, pp. 4972-4978, 1987.

[22] Y. Pao, "*Adaptive pattern recognition and neural networks,*" 1989.

[23] Y. M. M. Hassim and Rozaida Ghazali, "Improving Functional Link Neural Network Learning Scheme for Mammographic Classification," In *Advances in Neural Networks*, pp. 213-221, Springer International Publishing, 2016.

[24] R. Y. Redi and R. T. Bambang, "System Identification using Combined FIR and Functional Link Neural Networks: Application to ANC Secondary Path," *International Journal of Artificial Intelligence*, vol. 7, pp. 360-367, 2011.

[25] J. C. Patra, and A. Van den Bos, "Modeling of an intelligent pressure sensor using functional link artificial neural networks," *ISA transactions,* vol. 39, pp. 15-27, 2000.

[26] Y. Shin and J. Ghosh, "The pi-sigma network: An efficient higher-order neural network for pattern classification and function approximation," In *Neural Networks, 1991., IJCNN-91-Seattle International Joint Conference on*, vol. 1, pp. 13-18. IEEE, 1991.

[27] Y. Shin and J. Ghosh, "Ridge polynomial networks," *IEEE Transactions on neural networks,* vol. 6, pp. 610-622, 1995.

[28] D. P. Kanungo, J. Nayak, B. Naik, and H. S. Behera, "Non-linear classification using higher order pi-sigma neural network and improved particle swarm optimization: an experimental analysis," In *Computational Intelligence in Data Mining—Volume 2*, pp. 507-518. Springer India, 2016.

[29] A. J. Hussain and P. Liatsis, "Recurrent pi-sigma networks for DPCM image coding," *Neurocomputing,* vol. 55, pp. 363-382, 2003.

[30] S. Ge, P. Changgen, and M. Xuelan, "Visual cryptography scheme using pi-sigma neural networks." In *2008 International symposium on information science and engineering*, vol. 2, pp. 679-682. IEEE, 2008.

[31] Y. Dong and J. Zhang, "An improved boosting scheme based ensemble of Fuzzy Neural Networks for nonlinear time series prediction," In *2014 International Joint Conference on Neural Networks (IJCNN)*, pp. 157-164. IEEE, 2014.

[32] (2016) SIDC (World Data Center for the Sunspot Index). [Online] Available: http://www.sidc.be/silso/datafiles

[33] (2016) MATLAB. [Online] Available: http://www.mathworks.com/examples/fuzzy-logic/mw/fuzzy-ex38166291-predict-chaotic-time-series.

[34] H. Wang, "Modeling of nonlinear systems based on orthogonal neural network with matrix value decomposition," In *Intelligent Control and Information Processing (ICICIP), 2012 Third International Conference on*, pp. 298-301, IEEE, 2012.

[35] Y. M. Chen and C. T. Lin, "Dynamic parameter optimization of evolutionary computation for on-line prediction of time series with changing dynamics," *Applied Soft Computing,* vol. 7, pp. 1170-1176, 2007.