2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

The Performance of a Recurrent HONN for Temperature Time Series Prediction

R. Ghazali[1], N. A. Husaini[1], L. H. Ismail[1], T. Herawan[2] and Y. M. M. Hassim[1]
[1]Universiti Tun Hussein Onn Malaysia
Batu Pahat, Johor
[2]Universiti of Malaya
Kuala Lumpur

***Abstract*—This paper presents a novel application of Recurrent HONN to forecast the future index of temperature time series data. The prediction capability of Recurrent HONN, namely the Recurrent Pi-Sigma Neural Network was tested on a five-year temperature data taken from Batu Pahat, Malaysia. The performance of the network is benchmarked against the performance of Multilayer Perceptron, and the standard Pi-Sigma Neural Network. The predictions demonstrated that Recurrent Pi-Sigma Neural Network is capable in predicting the future index of temperature series in comparison to other models. It is observed that the network is able to find an appropriate input output mapping of the chaotic temperature signals with a good performance in learning speed and generalization capability.**

Keywords—Multilayer Perceptron, Recurrent Pi-Sigma Neural Network, temperature forecasting

I. INTRODUCTION

Multilayer Perceptrons (MLP) is a feedforward network which is formed by a collection of summing units that are connected by their associated weights. The network has a hierarchical structure of several perceptrons, and has the ability to overcome the shortcomings of single-layer networks. Due to its capability in learning a rich variety of nonlinear decision surfaces, the highly popularized MLP models have been successfully applied in meteorological forecasting [1]-[3]. However, MLP utilizes computationally intensive training algorithms such as the error back-propagation and can get stuck in local minima. In addition, the network has problems in dealing with large amounts of training data, while demonstrating poor interpolation properties, when using reduced training sets. On the other hand, High Order Neural Networks (HONNs) are type of feedforward neural networks which have the ability to transform the nonlinear input space into higher dimensional space where linear separability is possible [4].

HONNs have been used in various applications such as image compression [5], time series prediction [6]-[8], system identification [9], function approximation [10], and pattern recognition [10]-[12]. HONNs are simple in their architectures and require fewer numbers of weights to learn the underlying equation. This potentially reduces the number of required training parameters. As a result, they can learn faster since each iteration of the training procedure takes less time. This makes them suitable for complex problem solving where the ability to adapt to new data in real time is critical [13]-[15]. However, they suffer from the combinatorial explosion of the higher-order terms and demonstrate slow learning, when the order of the network becomes excessively high.

Considering the limitations of standard HONNs, Pi-Sigma Neural Network (PSNN), a class of HONN, has been introduced [16]. PSNN is able to perform high learning capabilities that require less memory in terms of weights and nodes, and at least two orders of magnitude less number of computations when compared to the MLP for similar performance levels, and over a broad class of problems. In conjunction with the benefits of PSNN, a new model called Recurrent Pi-Sigma Neural Network (RPSNN) which posses a Jordan Neural Network architecture [17] is proposed to perform the non-linear mapping of input-output data.

On another side, a great interest in developing methods for more accurate predictions for temperature forecasting has led to the development of several methods which employ the use of physical methods, statistical-empirical methods and numerical-statistical methods [18-19]. These methods, however, constitutionally complex, limited and restricted to that of numerical weather prediction products. Considering the downside of those methods, in this paper, we propose the extended version of PSNN, namely the Recurrent PSNN, for the application of temperature time series prediction. The Recurrent PSNN (RPSNN) has a feedback link from the output layer back to the input layer, therefore giving the network the capability of storing previous memory of the network's output that can be used for current input processing. RPSNN maintains the fast learning property and powerful mapping of single layer HONNs whilst avoiding the explosion of weights and processing units required as the number of input increases.

II. THE NETWORKS

A. *Pi-Sigma Neural Network (PSNN)*

PSNN was first introduced by Shin and Ghosh [16]. It is a feedforward network with a single 'hidden' layer and product units in the output layer. PSNN calculates the product of summing units at the output layer and pass it to a nonlinear

The work of R. Ghazali is supported by University Tun Hussein Onn Malaysia and Ministry oh Higher Education Malaysia.

function. The network is able to learn in a stable manner even with fairly large learning rates.

Previous research found that PSNN is a good model for various applications. Shin and Ghosh [16] investigated the applicability of the network for shift, scale and rotation invariant pattern recognition. Results for both function approximation and classification were extremely encouraging when compared to the MLP for achieving similar quality of solution. Ghosh and Shin [20] argued that PSNN requires less memory (weights and nodes), and at least two orders of magnitude less number of computations when compared to MLP for similar performance level, and over a broad class of problems.

The output of the PSNN is computed as follows:

$$Y = \sigma\left(\prod_{j=1}^{k}\sum_{k=1}^{N}\left(w_{kj}x_k + w_{jo}\right)\right) \qquad (1)$$

where w_{kj} is the adjustable weight, x_k is the input vector, K is the number of summing unit, N is number of input nodes, and σ is a suitable nonlinear transfer function. PSNN demonstrated competent ability to solve many scientific and engineering problems.

A. *The Proposed Recurrent Pi-Sigma Neural Network (RPSNN)*

The structure of RPSNN is quite similar to the ordinary PSNN. The main difference is the architecture of RPSNN is constructed by having a recurrent link from output layer back to the input layer. This structure gives the temporal dynamics of the time-series process that allows the network to compute in a more parsimonious way [21]. The architecture of the proposed RPSNN is shown in Figure 1 below.

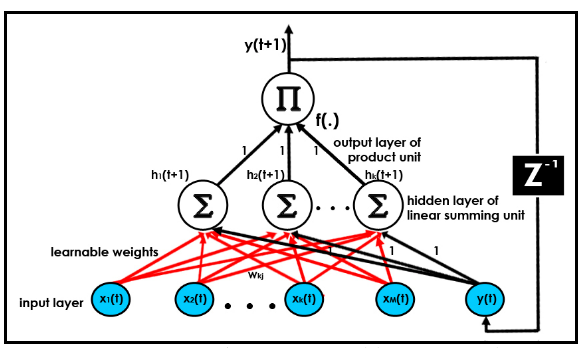

Fig. 1. RPSNN with 1 output node

where

$x(t)$	-	the input nodes at *t-th* time
w_{kj}	-	the trainable weights
$h_k(t+1)$	-	the summing unit
$y(t+1)$	-	the output at time $t+1$
$y(t)$		the output at time t
$f(\cdot)$		the activation function

Weights from the input layers $x(t)$ to the summing units' layer are tunable, while weights between the summing unit layers and the output layer are fixed to 1. The tuned weights are used for network testing to see how well the network model generalizes on unseen data. Z^{-1} denotes time delay operation.

Let the number of external inputs to the network be M and the number of the output be 1. Let $x_m(t)$ be the m *-th* external input to the network at time *t*. The overall input at time t is the concatenation of $y(t)$ and $x_k(t)$, *where* $(k = 1,\ldots, M)$, and is referred to $z(t)$ where:

$$z_k(t) = \begin{cases} x_k(t) & if \quad 1 \le k \le M \\ y_k(t) & if \quad k = M+1 \end{cases} \qquad (2)$$

Meanwhile, weights from $z(t)$ to the summing unit are set to 1 in order to reduce the complexity of the network.

The proposed RPSNN combines the properties of both PSNN and Recurrent Neural Network (RNN) so that better performance can be achieved. When utilizing the newly proposed RPSNN as predictor for one-step-ahead prediction, the previous input values are used to predict the next elements in the data. Since network with recurrent connection holds several advantages over ordinary feedforward MLP especially in dealing with time-series problems, therefore, by adding the dynamic properties to the PSNN, this network may outperform the ordinary feedforward MLP and also the ordinary PSNN. Additionally, the unique architecture of RPSNN may also avoid from the combinatorial explosion of higher-order terms as the network order increases.

III. Traditional Approach to Time Series Prediction

The standard method for predicting financial time series is the statistical linear approach. In this approach, the signal S_n is considered as the output of a system with unknown input u_n and its value is determined by the linear combinations of previous outputs and inputs according to the following equation [22]:

$$S_n = \sum_{k=1}^{P} a_k S_{n-k} + G\sum_{m=0}^{q} b_m u_{n-m}, \qquad b_0 = 1 \qquad (3)$$

where a_k, b_m and G are the model parameters. Usually the input u_n is modelled by a zero mean Gaussian noise source. The above equation can be specified in the frequency domain by taking the Z transform of both sides of the equation. Let $H(Z)$ represent the transfer function of the system in the Z domain, then:

$$H(Z) = \frac{S(Z)}{U(Z)} = G\,\frac{1 + \sum_{m=1}^{q} b_m z^{m}}{1 + \sum_{k=1}^{p} a_k z^{k}} \qquad (4)$$

and the Z transform of the signal is:

$$S(Z) = \sum_{n=-\infty}^{\infty} s_n z^{-n} \tag{5}$$

In this case, the roots of the numerator and the denominator of the transfer function $H(Z)$ are the zeros and the poles of the model, respectively. When $a_k = 0$, the model is considered as all poles and called the Moving Average (MA) model, when $b_m = 0$, the model is considered as all poles and known as Autoregressive (AR) model, while a model that has poles and zeros values is referred to as an autoregressive moving average (ARMA) model.

For the nonlinear model, we have:

$$g(\mathrm{S}_{\mathrm{n}}, \mathrm{S}_{\mathrm{n-1}}, \mathrm{S}_{\mathrm{n-2}}, ...) = u_n \tag{6}$$

In this case, u_n is a zero mean white noise. The function g is a highly nonlinear and very complicated. Nonlinear prediction can be determined using either the Volterra or the bilinear models, where the process is assumed to be inevitable, i.e., u_n can be approximated using a finite number of terms and in which:

$$S_n = \sum_i a_i u_n + \sum_i \sum_j a_{ij} u_{n-i} u_{n-j} + \sum_i \sum_j \sum_k a_{ijk} u_{n-i} u_{n-j} u_{n-k} + ... \tag{7}$$

using the discrete Volterra series expansion. Where $\{u_i\}$, $\{u_{ij}\}$, $\{u_{ijk}\}$ are Gaussian random variables and $\{a_i\}$, $\{a_{ij}\}$, $\{a_{ijk}\}$ are sets of constant coefficients.

Using the bilinear model, we can determine S_n as follows:

$$S_n = \sum_{i=1}^{P} a_i S_{n-i} + \sum_{j=1}^{q} a_j u_{n-j} + \sum_{l=1}^{P} \sum_{m=1}^{q} b_{lm} S_{n-l} u_{n-m} \tag{8}$$

where $c_0 = 0$, and u_n is a white noise process.

To solve the nonlinear model, it is required to determine the unknown parameters, which are usually very difficult to determine using traditional methods. Neural networks can be used to solve this problem in which the parameters (weights and biases) are determined implicitly using suitable training algorithms.

Feedforward Neural Networks are Nonlinear Autoregressive (NAR) models, on the other hand Recurrent Neural Networks are nonlinear ARMA models (NARMA). This means that Recurrent Neural Network have advantages over Feedforward Neural Network, similar to the advantages in which ARMA model posses over AR model [23].

IV. TEMPERTAURE FORECASTING

In this work, temperature forecasting takes an existing series of data $x_{t-n}, ..., x_{t-2}, x_{t-1}, x_t$ and forecasts the next incoming values of the series $x_{t+1}, x_{t+2}, ...$. Forecasting the behavior of the meteorological data is a nontrivial task, as it is often masked by noise and has nonlinear and non-stationary behavior.

Temperature is a kind of atmospheric time-series data where the time index takes on a predetermined or unlimited set of values. The temperature can have a greater influence in daily life than any other single element on a routine basis. Therefore, some great observations are needed to obtain accuracies for the temperature measurement [24]. Existing methods for estimating temperature can work efficiently; however, it is inadequate to represent the efficiency of temperature forecasting due to the relatively primitive output post-processing of the current techniques which is competitively superior to subjective prediction. Therefore, because temperature parameter itself can be nonlinear and complex, a powerful method is needed to deal with it. With the advancement of computer technology and system theory, there have been more meteorological models conducted for temperature forecasting [1], including soft computing approaches [2-3], [25-27].

On the other hand, the development of HONN has captured researchers' attention. PSNN which lies within this area, has the ability to converge faster and maintain the high learning capabilities of HONN. Yet, this paper focuses on developing a new alternative network model, namely the RPSNN, to overcome such drawbacks in MLP and taking the advantages of PSNN with the recurrent term added for temporal sequences of input-output mappings.

V. EXPERIMENTAL SETTING

A univariate data of a 5-years daily temperature measurement in Batu Pahat Malaysia, ranging from 2005 to 2009 was used for the simulation. The data was obtained from the Central Forecast Office, Malaysian Meteorological Department (MMD).

The data is segregated in time order and is divided into 3 sets; training, validation and the out-of-sample data, giving a distribution of 914, 456, and 456, respectively. To avoid computational problems, the data is normalized between the upper and lower bounds of Sigmoid transfer function.

Temperature time series are highly nonlinear. They exhibit high complexity and contain lot of noise. To purify the data for further processing, it is needed to identify and remove the contaminating effects of the outlying objects on the data. Therefore, the temperature time series were scaled using standard minimum and maximum normalization method which then produces a new bounded dataset. One of the reasons for using data scaling is to process outliers, which consist of sample values that occur outside normal range. The advantage of this transformation is that the distribution of the transformed data will become more symmetrical and will follow more closely to normal distribution.

As there is no rule of thumb for identifying the number of input, a trial-and-error procedure was determined. All networks were built considering 5 different numbers of input nodes ranging from 4 to 8. A single neuron was considered for the output layer. The number of hidden nodes (for MLP), and the higher order terms (for PSNN and RPSNN) were initially started with 2, and increased by one until a maximum of 5.

The prediction performance of all networks was evaluated using two performance metrics. The normalized mean squared error (NMSE) is used to measure the deviation between the actual and the predicted signals. The smaller the value of NMSE, the closer is the predicted signals to the actual signals. The signal to noise ratio (SNR) provides the relative amount of useful information in a signal as compared to the noise it carries.

The RPSNN is benchmarked against the MLP and PSNN, which were trained with the incremental backpropagation learning algorithm [28]. Early stopping with maximum number of 3000 epochs was utilized. For all networks, an average performance of 20 trials was used. By considering all in-sample dataset that have been trained, the best value for the momentum term $\alpha = 0.2$ and the learning rate $\eta = 0.1$, were chosen based on extensive simulations done by trial-and-error procedure.

VI. SIMULATION RESULTS

In this section, the simulation results for the prediction of Batu Pahat temperature time series is presented using various neural networks. The simulation results of the RPSNN are benchmarked against the MLP and standard PSNN. Notice that out of all measuring criteria, network model with the lowest NMSE is selected. This is due to the fact that NMSE is much more significant as an estimator of the overall deviations between predicted and measured values.

TABLE I. AVERAGE RESULTS OF ALL NETWORKS ON TEMPERATURE PREDICTION

	MLP	PSNN	RPSNN
NMSE	0.7815	0.7791	0.7710
MAE	0.0636	0.0635	0.0635
SNR	18.6971	18.7104	18.7557
Epochs	2850	1212	1461

In order to compare the predictive performance of the three models, Table I presents the best simulation results obtained on out-of-sample data from all neural network models. Over all the training process, it is verified that RPSNN exhibited the lowest prediction error, in terms of NMSE on the out-of-sample dataset. This indicates that the network is capable of representing nonlinear function better than the two benchmarked models. Apart from the NMSE, RPSNN and PSNN obtained a lower MAE, which is 0.0635; while the MAE for MLP is 0.0636. By considering the MAE, it shows that RPSNN is able to make a very close forecast to the actual output in analysing the temperature signals. Moreover, it can be seen that RPSNN reached the highest value of SNR. Therefore, it can be said that the network can track the signal better than PSNN and MLP. In the case of learning speed, particularly on the number of epoch utilized, PSNN converged much faster than the RPSNN and MLP. However, RPSNN reached a smaller number of epochs when compared to the MLP. On the whole, the performance of RPSNN gives a good comparison when compared to the two benchmarked models.

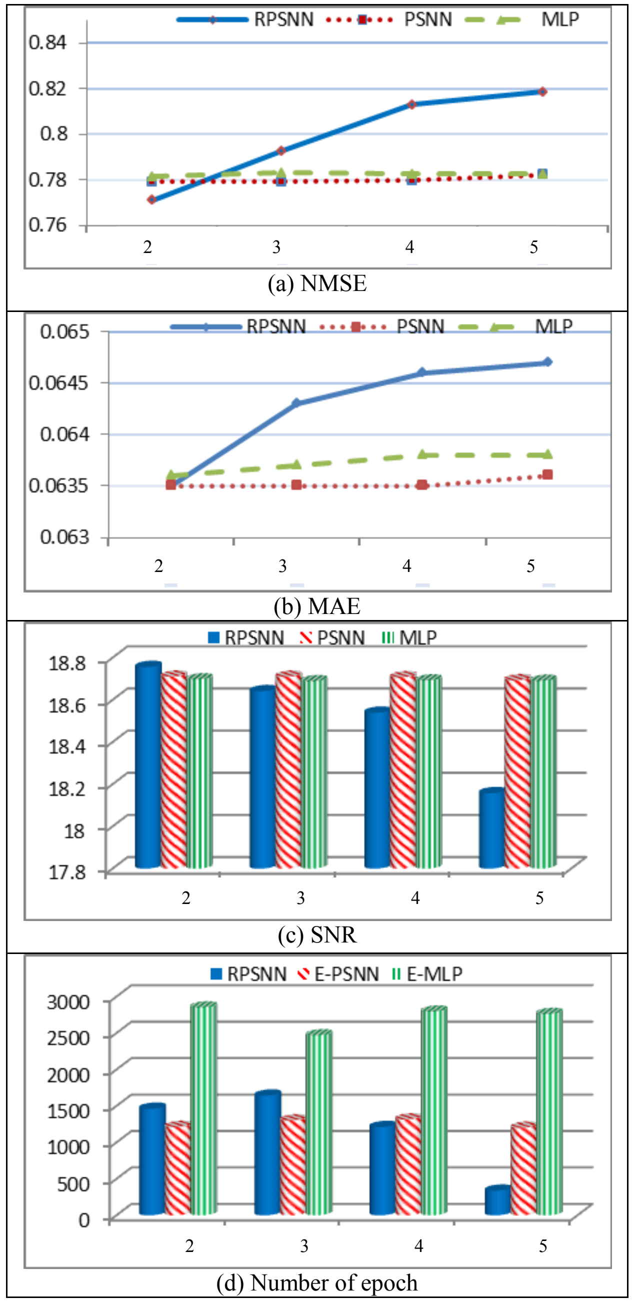

Fig. 2. The effects of increasing order or hidden nodes to the network performances

In order to test the modeling capabilities of all network models, Figure 2 shows the average result of MAE, NMSE, and SNR tested on out-of-sample data, and maximum epochs reached during the training of the network, where the number of higher order terms (for RPSNN and PSNN) or number of hidden nodes (for MLP) are tested between 2 to 5. From the plots in Figures 2 (a) and (b), it can be noticed that RPSNN has shown obvious increments in NMSE and MAE as the order increases. Using the same measuring criteria, the performance of both PSNN and MLP in NMSE and MAE were slightly increased along the growth of the network size. Meanwhile, the SNR for RPSNN and PSNN, as shown in

Figure 2 (c) were decreasing when the order increases. On the other hand, when referring to Figure 2 (d), there is no significant indicator in the maximum number of epochs reached by all network models when the network size is increased, as there are up and down scores in the epoch size. In the same plot, MLP however have obtained the largest epoch in all network structures. In most cases, the plots in Figure 2 indicate that RPSNN can steadily learn and generalize the temperature data when the number of higher order is small.

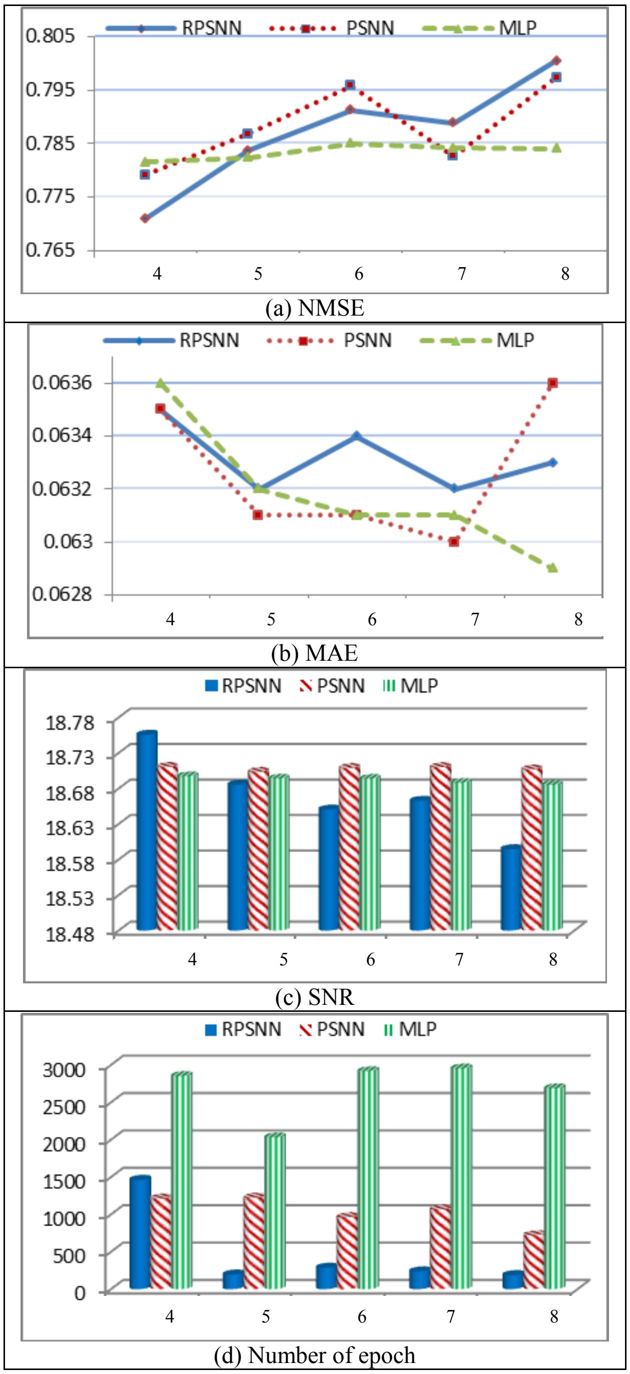

Fig. 3. The effects of increasing input nodes to the network performances

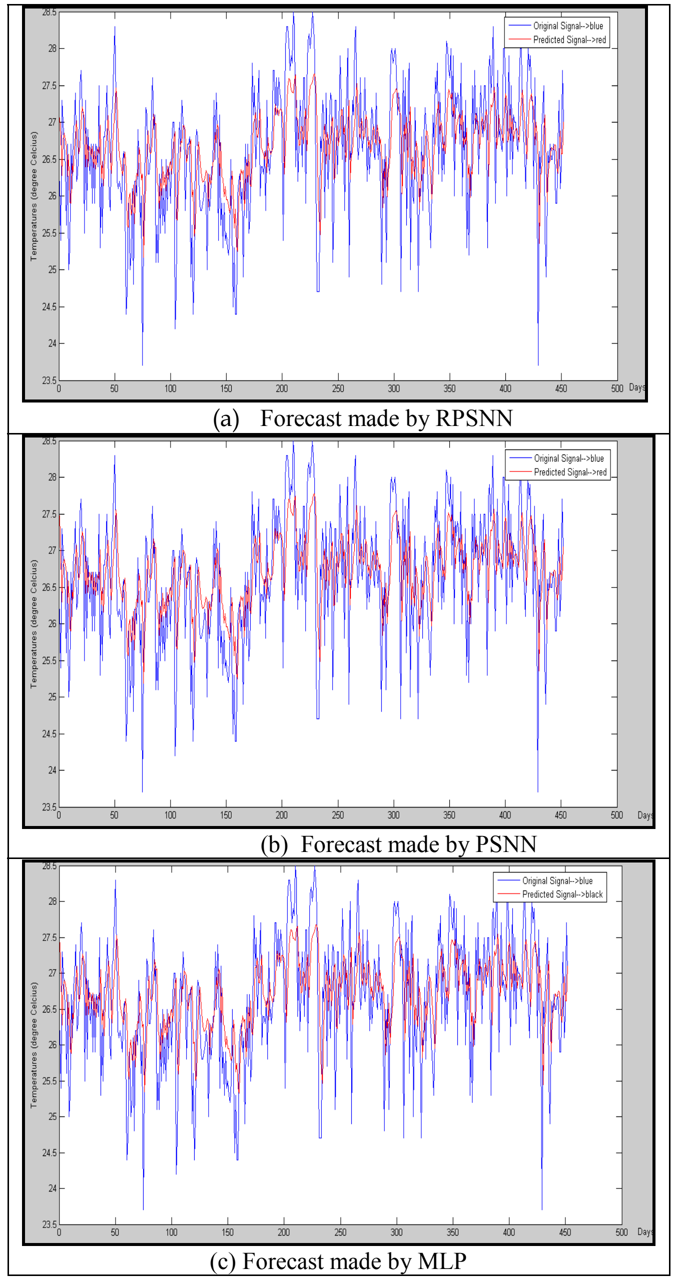

Fig. 4. Temperature forecast made by all networks on unseen data

Besides, the input-output mapping capabilities of all network models were also tested on varies number of input nodes, which is from 4 to 8. The performance of the network specifically when tested on out-of-sample temperature data on the NMSE, MAE and SNR are depicted in Figure 3 (a, b, and c), respectively. On the other hand, Figure 3 (d) depicts the number of epoch reached during the training of the networks. From the plots in Figures 3 (a) and (b), it can be noticed that both NMSE and MAE have shown an up and down values, regardless of the number of input nodes used. On the SNR as shown in Figure 3 (c), both RPSNN and MLP have a decreased SNR along with the increasing number of input nodes, while at the same time PSNN has up and down scores. On the other hands, as shown in Figure 3 (d), all networks in most cases have utilized larger number of epoch during the training, when equipped with least input nodes. In the same plot, MLP however have obtained the largest epoch in all network structures. In most cases, the plots in Figure 3 show that there is no clear indicator whether the higher or smaller number of input nodes could contribute to better performance of the networks.

Meanwhile, Figure 4 (a-c) depicts the best forecasts made by RPSNN, PSNN, and the MLP on temperature time series. As shown in the plots, the blue line represents the trend of the actual values, while the red line represents the predicted values. The predicted values of daily temperature measurement made by all network models almost fit the actual values with minimum error forecast.

VII. DISCUSSIONS

It is verified that RPSNN has the ability to perform an input-output mapping of temperature data as well as good performance when compared to the benchmarked MLP and PSNN. Besides, the evaluations on MAE, NMSE, and SNR over the temperature data demonstrated that RPSNN acceptably improved the performance level compared to the two benchmarked network models, PSNN and MLP. The better performance of temperature forecasting is allocated based on the vigour properties it contains. Hence, it can be seen that the thrifty representation of higher order terms in RPSNN assists the network to model effectively.

RPSNN generalized well and achieved the lowest NMSE and highest SNR on the unseen data, in which this is a desirable property in nonlinear time series prediction. The presence of a single layer of adaptive weights in the RPSNN allowed fast and rapid training. The good performance in the prediction of temperature time series is due to the well regulated structure which led to network robustness. The parsimonious representation of high order terms in the network provided the network with the ability to accurately forecast the future index of the series. The network learned the underlying mapping steadily as the order of the network increased. There is saturation in performance for RPSNN with larger network's order due to overfitting. This indicates that the interaction between the input signals of RPSNN of smaller number of higher order terms contains significant information for the prediction task.

VIII. CONCLUSION AND FUTURE WORKS

In this paper, an improved version of ordinary Pi-Sigma Neural Network, namely the Recurrent Pi-Sigma Neural Network is proposed for the prediction of temperature time series. The network preserves all the advantages of ordinary feedforward PSNN whilst having a temporal dynamic induced by the recurrent connection. The utilization of product units in the output layer indirectly incorporates the capabilities of RPSNN while using a small number of weights and processing units. Results obtained from the experimental simulations showed that RPSNN outperformed Multilayer Perceptron and Pi-Sigma Neural Network. The powerful learning capabilities of RPSNN allow the network to produce superior performance in terms of reduced NMSE and MAE, and also a higher SNR in comparison to other networks. Future work will involve the use of swarm intelligent approach, such as bat and cuckoo search algorithm, for finding suitable initial tunable weights for the network.

REFERENCES

[1] Paras, et al., "A Feature Based Neural Network Model for Weather Forecasting," Proceedings of World Academy of Science, Engineering and Technology, 2007. 34: pp. 66-74.

[2] N. R. Pal et. al, "SOFM-MLP: a hybrid neural network for atmospheric temperature prediction," *Geoscience and Remote Sensing, IEEE Transactions on*, 41(12): p. 2783-2791, 2003.

[3] B. A. Smith, G. Hoogenboom, and R.W. McClendon, "Artificial neural networks for automated year-round temperature prediction," *Comput. Electron. Agric*., 68(1): p. 52-61, 2009.

[4] Y. Pao, "Adaptive pattern recognition and neural networks," Addison Wesley, 1989.

[5] P. Liatsis and A. J Hussain, "Nonlinear one-dimensional DPCM image prediction using polynomial neural networks," *Proc. SPIE, Applications of Artificial Neural Networks in Image Processing IV*, San Jose, California, 28-29 January, Vol. 3647, pp. 58-68, 1999.

[6] H. Tawfik and P. Liatsis, "Prediction of non-linear time-series using Higher-Order Neural Networks," *Proceeding IWSSIP'97 Conference*, Poznan, Poland, 1997.

[7] R. Ghazali, A. J. Hussain, P. Liatsis and H. Tawfik, "The application of ridge polynomial neural network to multi-step ahead financial time series prediction," *Neural Computing & Applications*, 17(3) 311–323, 2008.

[8] Rozaida Ghazali, Abir Hussain, Wael El-Deredy, "Application of Ridge Polynomial Neural Networks to Financial Time Series Prediction," International Joint Conference on Neural Networks, IJCNN 2006, Vancouver, BC, Canada, 16-21, pp 913-920, 2006.

[9] L. Mirea and T. Marcu, "System identification using Functional-Link Neural Networks with dynamic structure," *15th Triennial World Congress*, Barcelona, Spain, 2002.

[10] Y. Shin and J. Ghosh, "The Pi-Sigma Networks: An efficient Higher-order Neural Network for pattern classification and function approximation," *Proceedings of International Joint Conference on Neural Networks,* Vol.1, pp.13-18, Seattle, Washington, July 1991.

[11] Y. Shin, J. Ghosh and D. Samani, "Computationally Efficient Invariant Pattern Classification with Higher-order Pi-sigma Networks," In Burke and Shin, (Ed), *Intelligent Engineering Systems through Artificial Neural Networks-II, Dagli*, pp. 379-384. ASME Press. 1992.

[12] T. Kaita, S. Tomita and J. Yamanaka, "On a Higher-order Neural Network for distortion invariant pattern recognition," Pattern Recognition Letters 23, pp 977-984, 2002.

[13] R. Cass and B. Radl, "Adaptive process optimization using Functional-Link Networks and Evolutionary Algorithm," *Control Eng. Practice*, Vol. 4, No. 11, pp. 1579-1584, 1996.

[14] Y. H. Pau and S. M. Phillips, "The Functional Link Net and learning optimal control," *Neurocomputing* 9, pp. 149-164, 1995.

[15] E. Artyomov and O. Y. Pecht, "Modified High-order neural Network for pattern recognition," *Pattern Recognition Letters*, 2004

[16] Y. Shin and J. Ghosh, "The Pi-Sigma Networks: An efficient Higher-order Neural Network for pattern classification and function approximation," *Proceedings of International Joint Conference on Neural Networks,* Vol.1, pp.13-18, Seattle, Washington, July 1991.

[17] M. I., Jordan, "Attractor Dynamics and Parallelism in a Connectionist Sequential Machine," Proceedings of the Eighth Conference of the Cognitive Science Society, New Jersey, USA, 1986.

[18] R. Barry & R. Chorley. *Atmosphere, Weather, and Climate*: Methuen, 1982.

[19] A. C., Lorenc, "Analysis Methods for Numerical Weather Prediction," *Quarterly Journal of the Royal Meteorological Society, 112*(474), pp. 1177-1194, 1986.

[20] J. Ghosh and Y. Shin, "Efficient Higher-order Neural Networks for function approximation and classification" *Int. J. Neural Systems*, vol. 3, no. 4, pp. 323-350, 1992.

[21] A. J. Hussain and P. Liatsis, "Recurrent Pi-Sigma Networks for DPCM Image Coding," *Neurocomputing, 55*, pp. 363-382, 2002.

[22] J. Makhoul, "Linear prediction: A tutorial review," *Proceedings of the IEEE*, 63(4), pp 561-580, 1975.

[23] J. Connor and L. Atlas, "Recurrent Neural Networks and Time Series Prediction," *IEEE International Joint conference on Neural networks,* New York, USA, pp I 301- I 306, 1991.

[24] D. Ibrahim, "Temperature and its Measurement," In *Microcontroller Based Temperature Monitoring & Control* (pp. 55-61). Oxford: Newnes, 2002.

[25] R. Bhardwaj et. al, "Bias-free rainfall forecast and temperature trend-based temperature forecast using T-170 model output during the monsoon season," 14: p. 351-360, 2007.

[26] Lee, L.-W., L.-H. Wang, and S.-M. Chen, "Temperature prediction and TAIFEX forecasting based on high-order fuzzy logical relationships and genetic simulated annealing techniques," Expert Systems with Applications, 34(1): p. 328-336, 2008.

[27] Y. Radhika and M. Shashi, "Atmospheric Temperature Prediction using Support Vector Machines," International Journal of Computer Theory and Engineering, 1(1): p. 55-58, 2009.

[28] S. Haykin, "Neural Networks. A comprehensive foundation," Second Edition, Prentice-Hall, Inc., New Jersey, 1999.