ISSN 2278-3091
Volume 8, No.1.3, 2019
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse5381.32019.pdf
https://doi.org/10.30534/ijatcse/2019/5381.32019

An Improved Pi-Sigma Neural Network with Error Feedback for Physical Time Series Prediction

Urooj Akram[1]*, Rozaida Ghazali, Lokman Hakim Ismail, Muhammad Zulqarnain, Noor Aida Husaini, Muhammad Faheem Mushtaq

[1]Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia (UTHM), 86400, Parit Raja, Batu Pahat, Johor, Malaysia, uroojakram.cs@gmail.com

ABSTRACT

Forecasting natural occurring phenomena has been addressed and analyzed in many domains of science and gets more attention because of its vast range of applications. Traditional time series forecasting tools have some limitations like slow training process, less efficient training methods that effect on performance. Higher Order Neural Network (HONN) using recurrent feedback appear as a powerful technique and it has the ability to expand the input space, make them more efficient for solving complex problems and perform high learning abilities in the time series forecasting. Recurrent networks commonly used the network output as the feedback terms. This study proposed a model called improved Pi-Sigma Neural Network with Error Feedback (PSNN-EF) which combines the properties of Pi-Sigma Neural Network (PSNN), recurrence and error feedback. PSNN-EF uses back propagation gradient descent algorithm for training purpose and tested with time series signals namely; humidity, evaporation and wind direction datasets. The benefit of using recurrence and error feedback is that it generates more accurate results of prediction and provide more promising results. Based on the obtained results, our proposed method shows better performance and can be an alternative solution to Jordan Pi-Sigma Neural Network (JPSN) and PSNN for one step ahead prediction of those three datasets.

Key words: Error feedback, Higher order neural networks, Jordan pi-sigma neural network, Pi-Sigma neural network, Time series forecasting

1. INTRODUCTION

During the last few years, time series forecasting becomes a dynamic research area that grabbed the attention of researchers and developers. The data which is obtained during a specific time period like hourly, weekly, daily, monthly and annually is called time series data. Time series are exploited in statistics, weather forecasting, earthquake prediction, econometrics and many other applications [1]. The purpose of time series analysis is to identify the nature of phenomena in the data that can be used to predicting future outcomes of the time series variable [2]. Moreover, forecasting of time series is the most difficult to be determined and predicted among the many time series. That's why there was a need for such algorithms that could perform the nonlinearity mapping of input-output [3].

To solve the prediction problem of time series various techniques have been appeared over many years to enhance the accuracy of forecasting [4]. Nowadays, artificial neural networks (ANNs) have utilized effectively for time series forecasting [5]. There are some advantages of ANN that are more appropriate to solve the certain problems [6]. First of all, ANNs have the capability to learn [7] and have a non-linear relationship between input and output to approximate any continuous function and it is beneficial for complex data. Secondly, after the completion of the learning process from initial inputs, ANN can provide a robust alternative, given its ability to model and extract unseen features and relationships that make the model more generalize. The output is compared with the result of the neural network model that was mathematically calculated in which the value of weights and bias are adjusted to reduce the error.

After analysis, it is proved that the ANN provides better forecasting performance as compared to other traditional methods [8]. Furthermore, Artificial Neural Networks (ANNs) are the intelligent based models of the biological neurons [9] and it is also used effectively for time series prediction [10]. Based on the NN architecture it is categorized into two types: Recurrent Neural Networks (RNN) and Feed forward Neural Networks (FNN). In recurrent networks, there is a feedback connection between the units to get the output that can form a cycle [11]. On the other hand, a feed forward network is a non-recurrent network which contains three layers of inputs, outputs, and hidden layers. In this network, the signals can pass in one direction only.

The most common type of feed forward ANN is Multilayer Perception (MLP) [12]. The MLP is capable to learn from input-output signals [13]. Furthermore, to get the required input-output relationship of the network, the weights of linear combination are adjusted during the process of training in MLP by minimizing the error function [14]. Although the structure of MLP is multilayered that's why to solve complex nonlinear mapping problems, it needs not only greater number

of units that consequently shows low learning rate and poor generalization [15] but also takes a long time for training and reach at local minima [16]. Hence, to overcome the problem regarding MLP, there is a need of NN with single layer trainable weights that is called Higher Order Neural Networks (HONNs). This structure has fewer units having nonlinear mapping ability and reducing the network's complexity [15-12]. HONN appeared as a famous network for time series prediction and has widely used in many scientific and engineering problems [17]. There are several types of HONNs that were used to forecasting of time series. Pi-Sigma Neural Network (PSNN) and Jordan Pi-Sigma Neural Network (JPSN) are the types of HONN. PSNN has a simpler structure, it used less time for training and use a smaller number of weights [18]. It has the ability of fast learning that reduce the complexity of the network after utilized efficient polynomials for different input layer variables. This type of network structure is more effective and efficient rather than MLP because it encounters the problem of over fitting that found in MLP. However, this network model requires more effective and efficient tools in order to enhance the accuracy of forecasting [19].

Part of this research is due to some beneficial features of the PSNN, another type of HONN was developed for the temperature forecasting that is JPSN. JPSN is utilized with less memory during the training process [20]. Although there are many benefits of using JPSN, on the other hand, this network has also some issues like fixed weights are found in between recurrent node and the hidden nodes it decreases the network performance [21]. Moreover, the training process of JPSN is very slow because the weights are initialized with small values.

To conquer these problems, recently many types of neural network algorithms have been used also new techniques are developed. To solve the slow speed of learning several techniques have been presented to accelerate it. Considering the limitations of JPSN and using the properties of PSNN, the goal of this research is to propose an improved network model that used for increasing the training efficiency, enhance the accuracy and out of sample observation for prediction. This model is called an improved Pi-Sigma Neural Network using Error Feedback (PSNN-EF) that is a combination of the properties of PSNN, error feedback and recurrence [22]. In this regard, the backpropagation algorithm is used in this network for the purpose of learning from error. Hence, using the proposed PSNN-EF, this research leads to improving the accuracy of prediction for physical time series. In order to test the efficiency of the proposed model, different experiments are done using three different physical time series datasets and benchmark with two existing models.

The remaining paper is arranged as follows: Section II explains some discussion on the implementation of improved Pi-Sigma Neural Network using Error Feedback (PSNN-EF) for time series prediction. This section explains the architecture of the proposed model also the training algorithm for this model. The data and experimental settings, and performance metrics are also explained in this section. Section III presented analysis of the result related to prediction is shown in graphs. The comparison of the simulation results with different models like JPSN and PSNN is presented the results and discussion. Section IV presents the conclusion and useful suggestions for expanding the proposed model in the future.

2. THE PROPOSED NETWORK MODEL

The structure of PSNN was firstly proposed by [23] and retains the capabilities of HONN that employ product cells as output units and used less number of weights. This study uses the similar structure of PSNN but having error feedback and recurrent connection to train the weights in order to overcome the drawbacks of JPSN network and minimize the prediction error. The purpose of this research is to improve the performance of prediction, making them more accurate and slightly lower prediction error.

2.1. The Architecture of PSNN-EF

The structure of PSNN-EF is same as the ordinary PSNN. The architecture of PSNN-EF is constructed using error feedback link from the output layer towards the input layer to train weights. The structure of recurrent network models combines with a feedback connection that allows input-output mapping dynamically, used for storing information for later use and attractor dynamics [24]. Moreover, network error feedback used for input which helps to minimize the whole network error and improve the forecasting as compared to the network output feedback and also enhances the performance of forecasting for the network [25-26]. The structure of the network start from a small network and it becomes larger as during the training process until reached to the desired level of error.

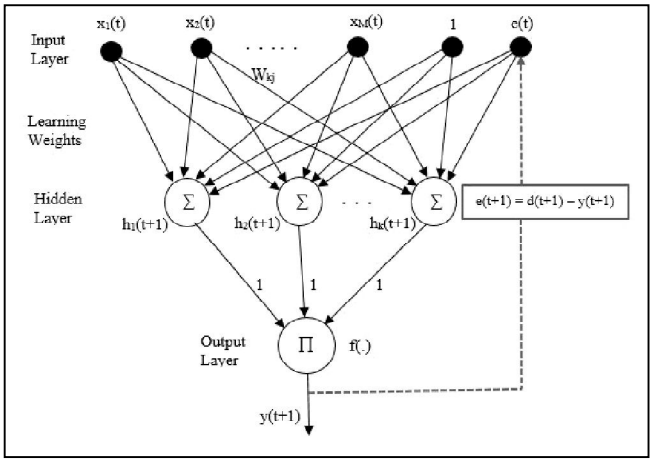

Figure 1: Pi-Sigma Neural Networks with Error Feedback (PSNN-EF)

Let M is the number of external inputs $x(t)$ in the network, and that $e(t)$ is an error of the network at time t and $x_m(t)$ to

be the m^{th} external input to the network at the time *t*. The input to the network at time *t* is the concatenation of $x_j(t)$, $j = 1,....., M$ and $e(t)$ is referred to as $z(t)$ where we have

$$z_j(t) = \begin{cases} x_j(t) & if \quad 1 \le i \le M \\ 1 & if \quad j = M+1 \\ e(t) & if \quad j = M+2 \end{cases} \tag{1}$$

An additional input is added for the bias which has value 1. Weights that are found from input layers $x(t)$ to the summing unit layer are trainable and the weights in between the summing layers and output layer are fixed to 1. The trainable weights are used to test the network that how this network performs well. Figure 1 shows the proposed network's architecture.

2.2. Training algorithm of PSNN-EF

In this regard, PSNN-EF uses Backpropagation Gradient Descent Algorithm that is basically utilized to network training and it learns from the error [27]. The error is calculated by taking the difference of the network output with the targeted output. The calculated error is used to alter the link between the layers that become the network input for the next time training cycle for getting better output. Hence, the intention of this training process is to make the error lower among the actual and predicted output by the adjustment of weights between the subsequent layers. The output is symbolized by $y(t+1)$, is computed as follows:

The processing equation of pi-sigma neural network using error feedback are given as follows:

$$y(t+1) = f\left(\prod_{L=1}^{k} h_L(t+1)\right) \tag{2}$$

$$h_L(t+1) = \sum_{M=1}^{M+2} w_{LM}\, z_m(t) \tag{3}$$

where $y(t+1)$ is the output of the network, $h_L(t+1)$ is the activation of *L* unit at a time $t+1$, f is the transfer function.

Network error is calculated using the sum squared error as follows:

$$e(t) = \frac{1}{2}\sum e(t)^2 \tag{4}$$

where $e(t) = d(t) - y(t)$

where *d* is desired output and *y* is predicted output.
By using the backpropagation algorithm

$$\Delta w_{kj} = -\eta * \frac{\partial E(t+1)}{\partial w_{kj}} \tag{5}$$

Where η is the learning rate. The term $\frac{\partial E(t+1)}{\partial w_{kj}}$ is defined as:

$$\frac{\partial E(t+1)}{\partial w_{kj}} = e(t+1) * \frac{\partial e(t+1)}{\partial w_{kj}} \tag{6}$$

The term $\frac{\partial e(t+1)}{\partial w_{kj}}$ is found by the chain rule, where

$$\frac{\partial e(t+1)}{\partial w_{kj}} = \frac{\partial e(t+1)}{\partial y_i(t+1)} * \frac{\partial y_i(t+1)}{\partial w_{kj}} \tag{7}$$

$$\frac{\partial e(t+1)}{\partial w_{kj}} = -1 * \frac{\partial y_i(t+1)}{\partial w_{kj}}$$

$$\frac{\partial y(t+1)}{\partial w_{kj}} = \frac{\partial y(t+1)}{\partial h_L(t+1)} * \frac{\partial h_L(t+1)}{\partial w_{kj}} \tag{8}$$

The term $\frac{\partial y(t+1)}{\partial h_L(t+1)}$ is obtained by differentiating the network's processing equations

$$\frac{\partial y(t+1)}{\partial h_L(t+1)} = \frac{\partial}{\partial h(t+1)} * \left(y(t+1)\right)$$

$$= y(t+1)' * \left(\prod_{\substack{L=1 \\ L \ne i}}^{k} h_L(t+1) \right) \tag{9}$$

The term $\frac{\partial h_L(t+1)}{\partial w_{kj}}$ is determined as follows:

$$\frac{\partial h_L(t+1)}{\partial w_{kj}} = \begin{cases} x_j(t) + w_i(M+2)\frac{\partial e(t)}{\partial w_{kj}} & for \quad j \le M \\ 1 + w_i(M+2)\frac{\partial e(t)}{\partial w_{kj}} & for \quad j = M+1 \\ e(t) + w_i(M+2)\frac{\partial e(t)}{\partial w_{kj}} & for \quad j = M+2 \end{cases} \tag{10}$$

Therefore, from equation (1) and (10)

$$\frac{\partial h_L(t+1)}{\partial w_{kj}} = z_j(t) + w_i(M+2)\frac{\partial e(t)}{\partial w_{kj}} \tag{11}$$

Let the sensitivity $P_{ij}(t+1)$ of the output at time $t+1$

$$P_{ij}(t+1) = \frac{\partial e(t+1)}{\partial w_{kj}} \tag{12}$$

Weight updating rule.

$$\Delta W_{kj}(t+1) = -\eta e(t+1) P_{ij}(t+1) + \alpha \Delta W_{kj}(t) \tag{13}$$

where

$$P_{ij}(t+1) = -1 * y(t+1)' * \left(\prod_{\substack{L=1 \\ L \neq i}}^{k} h_L(t+1) \right)$$

$$[z_j(t) + w_i(M+2)P_{ij}(t)] \qquad (14)$$

The aim of this proposed architecture is to adjust the network weights, to produce network output that is more near to the desired output. Moreover, recurrent link structure provides the dynamics of the network to compute the error in more efficient way that could be utilized for the input in next time step. Due to this feedback connection, the PSNN-EF has capabilities not found in the ordinary PSNN like store the information for later use and attractor dynamic. Therefore, it can be implemented to highly nonlinear dynamic system identification.

2.3. Data and Experimental Setting for PSNN-EF

The process of proposed methodology presents in this section that includes the structure of neural networks to the prediction of physical time series.

A univariate time series is a combination of all that observations that are related to the same variables with different times at uniform intervals. The purpose of a univariate time series is to predict the future values based on a given variable by observing its behavior in past [28]. So, the certainty of the univariate time series is that for the explanation it can utilize only for one variable, but multivariate time series can use many variables for an explanation. Therefore, by deliberate those configurations, our research considers using input data that is univariate: the 5-years daily measurement of evaporation, wind direction and humidity in Batu Pahat region, from 2005 to 2009 for our model. The datasets were collected from the Malaysian Meteorological Department (MMD) [29].

Input layer and nodes get the inputs and send it to the hidden layer along with their weights for summation. After computing the hidden layer this weighted sum passed to the output layer and do backpropagation and error signal generated here. At the same time, the error at the previous time step passed to the input layer in order to adjust the weights. After calculating the error for the first cycle, this error feedbacks to the input layer to complete the next cycle of the training until decrease the error. Nevertheless, it should be kept in mind that too less and too many nodes of input can impact on the network's learning and prediction ability. In this research, after conducting a pilot study and by performing eight number of trials, the networks of experimental design were constructed using five number of input nodes after 15 simulations and the single neuron was employed for the output layer. Since, the purpose of this study is an effort to predict the evaporation, humidity and wind direction event for one-day ahead, it will emphasis most of its effort on reducing the forecast by minimizing the error. The transfer function is used for the purpose of preventing the output to reach at very large values which can paralyse the network. Thus, this study considers of using the sigmoid activation function. The reason why activation function is used in the network because without activation function the neural network cannot learn non-linear relationships. Therefore, every network has only one activation function that is utilized for all neurons in that network. The sigmoid function is mostly used for the network models [30]. The sigmoid function which often to be a smooth step function can be expressed by $f(x) = 1/1+e^{-x}$ where x is the value of the output neuron. Consequently, this study used a Min-Max normalization technique which scales the data values into the range 0 and 1 based on the following equation [31- 32].

$$v' = (new\_\max A - new\_\min A) \times \frac{v - \min A}{\max A - \min A} + new\_\min A \qquad (15)$$

where *A* is the data and *min A, max A* denotes the minimum and maximum values of the data *A, v* indicates the normalized value. So, after normalization the data become relatively uniform and very close to normal distribution, hence it can be utilized in terms of the network inputs.

To simulate the proposed model each time series dataset is segregated and further distributed into three different sets; training 50%, validation 25% and testing 25% (out-of-sample). The training datasets served the model to complete the training process and this set should be big to the comparison of the other two sets. The training includes the adjustment of the weights by testing the initial set of weights against each input vector. The purpose of the training is to change the weights that determine the minimum error function. In this training process, the weights are updated by using equation 13. Therefore, every time the data is passed until the last row it changes the weights, every time it improves the weights in order to have a minimum error because they are comparing the actual results and target values. Training is affected by many parameters such as momentum, learning rate, epochs values and the stopping criteria. The validation set is very compulsory in this network. It has different functions like to implement the early stopping criteria that prevent the data from overfitting and select best prediction results from the network simulations. On the other hand, to measure the network performances, the testing set is used.

For the purpose of comparison, the performances on PSNN-EF for the prediction of evaporation, humidity and wind is benchmarked with JPSN and ordinary PSNN using four performance metrics.

2.4. Performance Metrices

This study used the performance metrics to evaluate the proposed model using Signal to Noise Ratio (SNR) [33], Mean Squared Error (MSE) [34], Mean Absolute Error (MAE) and Normalised Mean Squared Error (NMSE) [35]. The equations for these metrics are as follows:

Mean Analysis of SNR – to measure how much a signal has been corrupted by noise. The higher the ratio, the less obtrusive the background noise is. If the value of SNR is higher it means the noise level is lower than signal levels. The higher value of SNR shows that the performance of the model is good.

$$SNR = 10 * \log\left(\frac{m^2 * n}{SSE}\right) \quad (16)$$

$$SSE = \sum_{i=1}^{n}\left(P_i - P_i^*\right) \quad (17)$$

$$m = \max\left(P\right)$$

where n is the total number of data patterns, SSE is the sum of squared error, P and Pi* represent the actual and predicted output value.

Mean Analysis of MSE Testing – to calculate the difference between the desired and the predicted values of testing data by measuring the average of the squared error. The lower value of MSE shows the accurate result in terms of lower error.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(P_i - P_i^*\right)^2 \quad (18)$$

Mean Analysis of MSE Training – to measure the difference between the actual and predicted values of training data by measuring the average of the squared error.

Mean Analysis of MAE – to measure the forecasts or predictions are how close in time-series analysis. For the better performance of the network the value of MAE should be less than other network models.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|P_i - P^*{}_i\right| \quad (19)$$

Mean Analysis of NMSE – for measuring bias and scatter between the actual and the predicted values. If the value of NMSE is lesser then it means the network shows lower prediction error.

$$NMSE = \frac{1}{\sigma^2 n}\sum_{i=1}^{n}\left(P_i - P_i^*\right)^2 \quad (20)$$

$$\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(P_i - P_i^*\right)^2$$

$$P_i^* = \sum_{i=1}^{n} P_i$$

The performance metrics can include the quality of service and productivity measurements and help to capture internal performance. Using the above parameters, we are able to calculate the performance of our proposed model over the benchmarked model.

3. RESULTS AND DISCUSSION

This section provides the discussion about the simulation results using three different network models such as an improved Pi-Sigma Neural Network using Error Feedback (PSNN-EF), Jordan Pi-Sigma Neural Network (JPSN) and Pi-Sigma Neural Network (PSNN). The entire network models were trained and tested with evaporation, humidity and wind direction data collected from MMD [29]. The training of the network is affected by its initial internal state that includes different learning rate and momentum with random weights. To obtain more accurate and robust evaluations 15 simulations are performed for three network models. To stop the training, there are three stopping criteria. For every network training, there was a point at which training should be stopped. In this study, three stopping criteria were used: (1) training error that is set to 0.0001, (2) maximum epoch which is 3000 and (3) early stopping.

Error Feedback is used in the proposed model PSNN-EF that is capable to handle complex structures output spaces naturally. It predicts the physical time series after calculating and minimizing the error. Network error feedback is used as an input through iteration in which the error signal is used as a training signal. It helps to decrease the whole network error and increase the forecasting accuracy. Using error feedback, the proposed model is able to get the prediction error more accurate and slightly lower than other network architectures. Using error feedback, we are able to get the prediction error more accurate and slightly lower than other network architectures. According to the Figure 2, it could be observed that the proposed method PSNN-EF shows the lowest Mean Squared Error (MSE) in training and testing part on evaporation as compared to the rest of the techniques. By using error feedback and the properties of HONN like reduce the network complexity and use less memory the proposed model is able to get better performance and make our method more optimize.

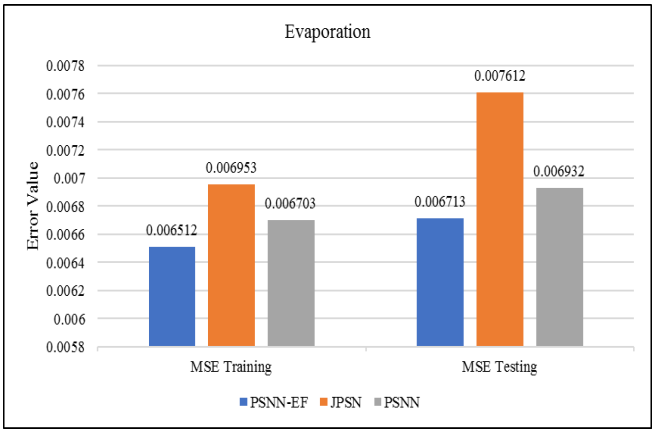

Figure 2: MSE on three different models for evaporation dataset

The performances of the models were also evaluated on humidity dataset using the same training method. Figure 3 shows the graphical representation of the result on MSE training and testing for the three network models based on

humidity dataset. Using the Higher Order Neural Network (HONN) properties, error feedback and recurrence in this proposed model, we are able to get better performance in terms of all performance metrics.

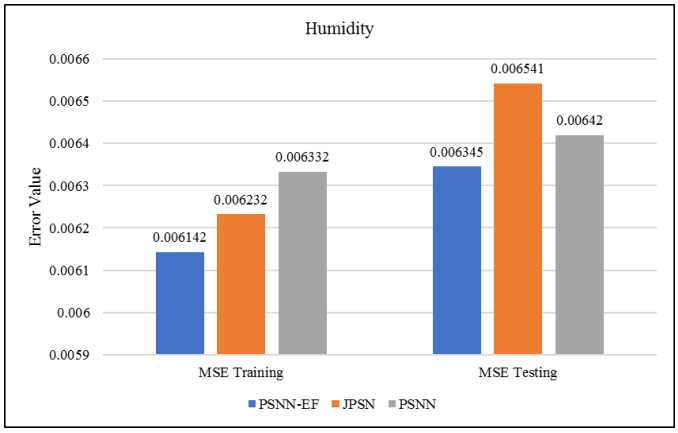

Figure 3: MSE on three different models for humidity dataset

Figure 4 shows the comparison of the error result based on benchmark models. Results show that the Mean Squared Error we got from our proposed model has lesser value. If a model has a low MSE, then it is performing well. As shown in Figure 2, 3 and 4 the least number of MSE is collected by PSNN-EF as compared with other models. The results demonstrate that how close the predictions are to the eventual outcomes in time-series analysis. It is to be proved that PSNN-EF model gives us slightly lower Mean squared error than other two benchmark models.

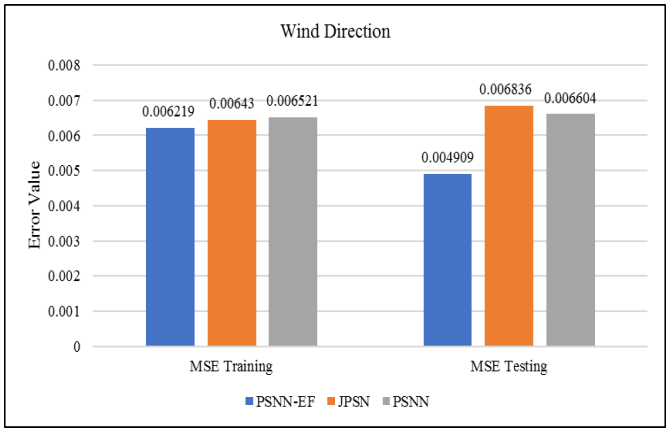

Figure 4: MSE on three different models for wind direction dataset

It is very important to demonstrate the sturdiness of PSNN-EF by comparing its performance on the prediction of physical time series with the ordinary JPSN and PSNN. This study has determined that several rates of learning rate gives the optimum results towards the selected data in the network. Each dataset has different optimum learning rates. For evaporation data the learning parameters are $\eta = 0.05$, $\alpha = 0.5$, and for the humidity data $\eta = 0.05$, $\alpha = 0.3$ and for wind direction $\eta = 0.05$ and $\alpha = 0.6$.

Figure 5 shows the graphical demonstration of the SNR results on all three datasets. However, from this figure we can analyze that proposed model have the higher SNR value for wind dataset while the other network models show lower value of SNR. If the value of SNR is higher it means the noise level is lower than signal levels. The higher value of SNR shows that the performance of the model is good. Its mean that the higher the SNR value get; the higher performance can achieve. So, this is the best simulation results for this model. It has three datasets that work well with this particular algorithm.

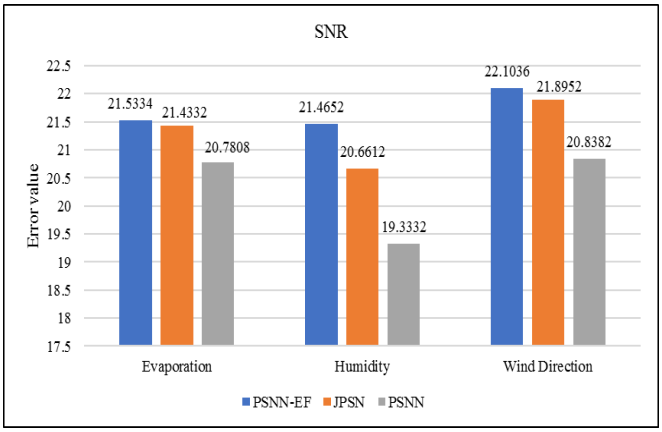

Figure 5: SNR for PSNN-EF, JPSN and PSNN

The graphical representation of NMSE results on all data sets is shown in Figure 6. It shows the best result of its experiment simulations which highlight that JPSN and PSNN models generate results on three different datasets have a higher value for NMSE but still this result is less accurate than our improved model. Hence, PSNN-EF produces lower NMSE as compared to other existing models. So, we can say that the lower value of NMSE shows best simulation results.

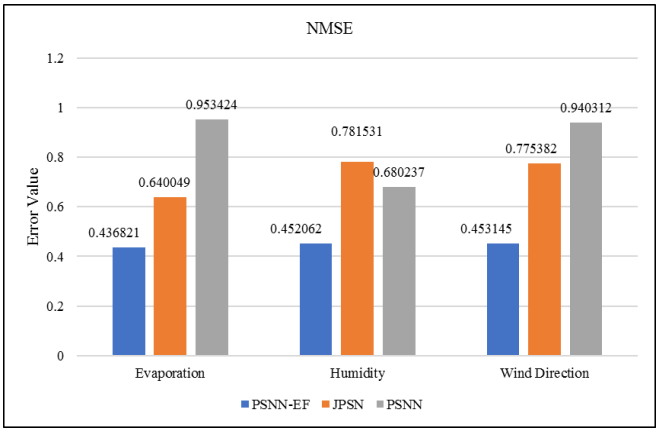

Figure 6: NMSE for PSNN-EF, JPSN and PSNN

Using the properties of HONN that were found in ordinary PSNN and error feedback this proposed model is capable to

minimize the error in terms of Mean Absolute Error (MAE) for all-time series. With the chosen training parameters, the network model manages to get the lower prediction with other comparing models. PSNN-EF shows the lowest Mean Absolute Error (MAE) for evaporation, humidity and wind direction which is 0.06362, 0.065318 and 0.06481 respectively. While the MAE for JPSN is 0.064130, 0.065427, 0.065313 with the respect of evaporation, humidity and wind direction datasets. Moreover, PSNN shows the result of MAE 0.065012, 0.066313 and 0.065785. In the regard of MAE, it shows the lower forecasting error that has been provided by PSNN-EF (refer to Figure 7).

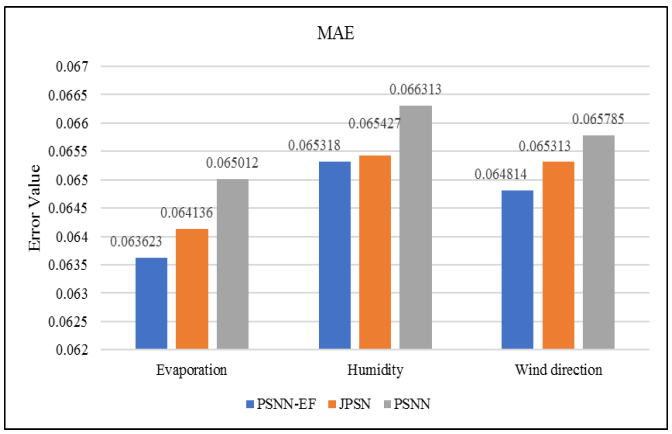

Figure 7: MAE for PSNN-EF, JPSN and PSNN

On the whole, improvement in the prediction accuracy of all datasets can be viewed where PSNN-EF shows higher accuracy especially on humidity dataset. The equation to find the accuracy improvement is given in Equation 21.

$$\text{Improvement (\%)} = \left(\frac{PSNN\_EF - JPSN}{JPSN}\right) \times 100\,\% \qquad (21)$$

The overall improvement of PSNN-EF on all three datasets with rest of the network models can be obtained by taking an average of each comparison model improvement value. The accuracy improvement of PSNN-EF is well explained. On the whole, it can be noted that PSNN-EF is capable to predict time series with the highest improvement. The highest improvement can be viewed in humidity dataset, where PSNN-EF shows 7.45% accuracy. The accuracy improvement of PSNN-EF over JPSN and PSNN for time series prediction is given in Table 1.

Table 1: Improvement of PSNN-EF in percentage (%)

Datasets	Accuracy Improvement of PSNN-EF (%)
Evaporation	2.06 %
Humidity	7.45 %
Wind	3.51 %

Furthermore, Figure 8, 9 and 10 shows the plot for the forecasting of evaporation, humidity and wind direction respectively for the PSNN-EF network model.

As mentions in the plots, the red line denotes the actual values and the green line shows the predicted values. The predicted values of daily measurements show the minimum forecast error. It is certified that the improved PSNN-EF is capable of nonlinear mapping also gives better performance when compared to other two network models.

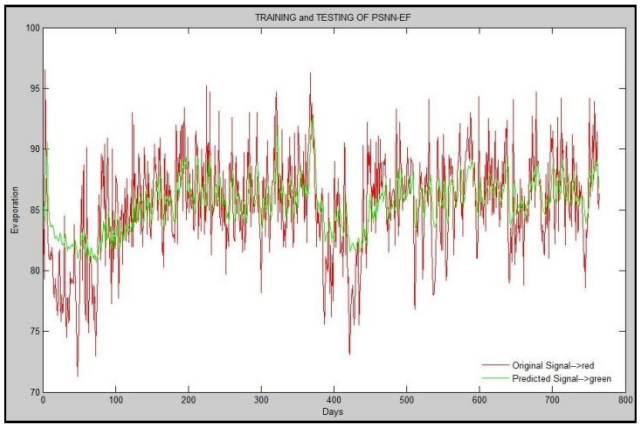

Figure 8: Evaporation forecast generated by PSNN-EF

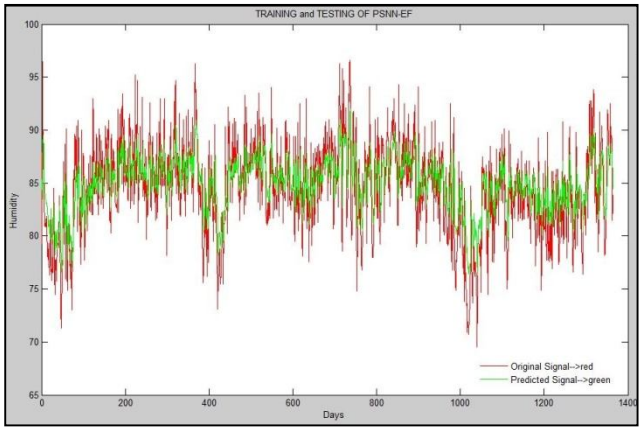

Figure 9: Humidity forecast generated by PSNN-EF

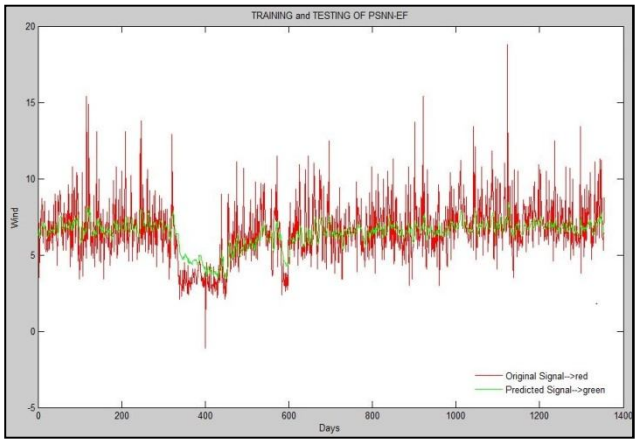

Figure 10: Wind forecast generated by PSNN-EF

Overall, the evaluations based on SNR, MAE, NMSE, MSE Training and MSE Testing over the evaporation, humidity and wind data verified that PSNN-EF merely increased the performance level rather than the two other existing models, JPSN and PSNN.

4. CONCLUSION

The paper presented an improved model for the prediction of physical time series. To address the gaps in JPSN, this study proposed a model called an improved Pi-Sigma Neural Network using Error Feedback (PSNN-EF) to examine the forecasting capability. The purpose of this study is to contribute an improved network structure of the PSNN that takes the benefits of higher order neural network, error feedback and recurrence. Due to the dynamic properties of the time series recurrent networks are more considered as best networks as compared to feedforward networks. Error feedback of the network is employed as an input that helps to minimize the network error instead of using network output feedback and it enhances the overall performance of prediction. PSNN-EF can improve the performance of prediction, generate more accurate and slightly lower prediction error than other network models. Due to better performance level, we found that the proposed model is easy to use. This research can be further examined in future by testing the proposed model with more time series signals of different lengths and compare with other HONN models to ensure the significant forecasting performance. The system may not be stable, therefore Lyapunov functions can be used with this proposed model to address the stability issue in the network. To increase the reliability of forecasting the proposed model can be hybridizing with other models.

ACKNOWLEDGEMENT

The authors would like to thank the Universiti Tun Hussein Onn Malaysia (UTHM) and Ministry of Higher Education Malaysia for financial supporting this research under the Fundamental Research Grant Scheme (FRGS) No. 1641.

REFERENCES

[1] "**Time Series**," Available: https://en.wikipedia.org/wiki/Time _series. Accessed January 15, 2017.

[2] V. Cross. **The Advantages of the Time Series Method of Forecasting**. [Online]. Available: https://bizfluent.com/info-8610535-advantages-time-series-method-forecasting.html. Accessed September 26, 2017.

[3] Shah, H., Ghazali, R. & Nawi, N. M., "**Hybrid Ant Bee Colony Algorithm for Volcano Temperature Prediction**", *International Multi Topic Conference*, (2012), 453–465. https://doi.org/10.1007/978-3-642-28962-0_43

[4] Shah, H., & Ghazali, R., "**Prediction of Earthquake Magnitude by an Improved ABC-MLP**", *International Conference on developments in eSystems Engineering*, (2011), 312–317. https://doi.org/10.1109/DeSE.2011.37

[5] J. Kamruzzaman, R. K. Begg, and R. A. Sarker, **Artificial Neural Networks in Finance and Manufacturing**. *Idea Group Publishing*, (2006), 80-108. https://doi.org/10.4018/978-1-59140-670-9

[6] W. K. Wong, M. Xia, and W. C. Chu, **Adaptive neural network model for time-series forecasting**. *European Journal of Operational Research*, 207, (2010), 807–816. https://doi.org/10.1016/j.ejor.2010.05.022

[7] Mushtaq, M. F., Akram, U., Tariq, A., Khan, I., Zulqarnain, M., & Iqbal, U., **An Innovative Cognitive Architecture for Humanoid Robot**. *International Journal of Advanced Computer Science and Applications (IJACSA*), 8, (2017), 60–67. https://doi.org/10.14569/IJACSA.2017.080808

[8] C. A. Mitrea, C. K. M. Lee, and Z. Wu, **A Comparison between Neural Networks and Traditional Forecasting Methods: A Case Study**. *International Journal of Engineering Business Management*, 1, (2009), 19–24. https://doi.org/10.5772/6777

[9] Mushtaq, M. F., Khan, D. M., Akram, U., Ullah, S., & Tariq, A.. **A Cognitive Architecture for Self Learning in Humanoid Robots**, *International Journal of Computer Science and Network Security*, 17(5), (2017), 26–36.

[10] Malik, N. "**Artificial Neural Networks and their applications**", *National Conference on Unearthing Technological Developments & Their Transfer for Serving Masses*, (2005).

[11] A. Hussain, "**Physical Time-Series Prediction Using Second- Order Pipelined Recurrent Neural Network**," *IEEE International Conference on Artificial Intelligence Systems*, (2002).

[12] 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 and Applications*, 17, (2008), 311–323. https://doi.org/10.1007/s00521-007-0132-8

[13] V. Guldal and H. Tongal, **Comparison of recurrent neural network, adaptive neuro-fuzzy inference system and stochastic models in egirdir lake level forecasting**. *Water Resources Management*, 24, (2010), 105–128. https://doi.org/10.1007/s11269-009-9439-9

[14] M. Rumbayan and K. Nagasaka, "**Estimation of Daily Global Solar Irradiation in Indonesia with Artificial Neural Network (ANN) Method**", *Proceeding of the International Conference on Advanced Science, Engineering and Information Technology*, (2011), 190–193. https://doi.org/10.18517/ijaseit.1.2.40

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

https://doi.org/10.1016/j.neucom.2013.09.015

[16] R. Ghazali and D. Al-Jumeily, **Application of Pi-Sigma Neural Networks and Ridge Polynomial Neural Networks to Financial Time Series Prediction**. *Artificial Higher Order Neural Networks for Economics and Business IGI Global*, (2009), 271–273. https://doi.org/10.4018/978-1-59904-897-0.ch012

[17] Y. M. M. Hassim and R. Ghazali, **Using Artificial Bee Colony to improve Functional Link Neural Network training**. *Applied Mechanics and Materials*, 266, (2013), 2102–2108. https://doi.org/10.4028/www.scientific.net/AMM.263-266.2102

[18] N. A. Husaini, R. Ghazali, N. M. Nawi, L. H. Ismail, M. M. Deris, and T. Herawan, **Pi-Sigma Neural Network for a One-Step-Ahead Temperature Forecasting**. *International Journal of Computational Intelligence and Applications*, 13, (2014), 1450023-1-1450023–16. https://doi.org/10.1142/S1469026814500230

[19] U. Akram, R. Ghazali, and M. F. Mushtaq, **A Comprehensive Survey on Pi-Sigma Neural Network for Time Series Prediction**. *Journal of Telecommunication, Electronic and Computer Engineering*, 9, (2017), 57–62.

[20] N. A. Husaini, R. Ghazali, N. M. Nawi, and L. H. Ismail, **The Jordan Pi-Sigma Neural Network for Temperature Prediction**. *Springer-Verlag Berlin Heidelberg*, (2011), 547–558. https://doi.org/10.1007/978-3-642-20998-7_61

[21] R. Ghazali, N. A. Husaini, L. H. Ismail, and N. A. Samsuddin, **An Application of Jordan Pi-Sigma Neural Network for the Prediction of Temperature Time Series Signal**. *INTECH Open Access Publisher*, (2012), 275–290. https://doi.org/10.5772/36026

[22] D. Wan, Y. Hu, and X. Ren, "**BP neural network with error feedback input research and application**", *Second International Conference on Intelligent Computation Technology and Automation*, (2009), 2, 63–66. https://doi.org/10.1109/ICICTA.2009.24

[23] **Y. Shin and J. Ghosh, "The pi-sigma network: an efficient higher-order neural network for pattern classification and function approximation**", *International Joint Conference on Neural Networks*, (1991), 1–18. https://doi.org/10.1371/journal.pone.0167248

[24] A. J. Hussain and P. Liatsis, **Recurrent pi-sigma networks for DPCM image coding**. *Neurocomputing*, 55, (2002), 363–382.

[25] W. Waheeb, R. Ghazali, and T. Herawan, **Ridge Polynomial Neural Network with Error Feedback for Time Series Forecasting**. *PLoS One*, 458, (2016), 1–34.

[26] R. Ghazali, A. Jaafar Hussain, N. Mohd Nawi, and B. Mohamad, **Non-stationary and stationary prediction of financial time series using dynamic ridge polynomial neural network**. *Neurocomputing* 72, (2009), 2359–2367. https://doi.org/10.1016/j.neucom.2008.12.005

[27] David E. Rumelhart, G. E. Hinton, and R. J. Williams, **Learning Representations By Backprogating Errors**. *Nature*, 323, (1986), 533–536. https://doi.org/10.1038/323533a0

[28] I. A. Gheyas and L. S. Smith, "**A Neural Network Approach to Time Series Forecasting**", *Proceedings of the World Congress on Engineering*, 2, (2009), 1–3.

[29] Malaysian Meteorological Department, "**Malaysian Meteorological Department**," *MET Malaysia*, 2010. [Online]. Available: http://www.met.gov.my/.

[30] H. Yonaba and V. F. F. Anctil, **Comparing Sigmoid Transfer Functions for Neural Network Multistep Ahead Streamflow Forecasting Comparing**. *Journal of Hydrologic Engineering*, (2010), 275–283. https://doi.org/10.1061/(ASCE)HE.1943-5584.0000188

[31] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, **Handling imbalanced datasets : A review**. *GESTS International Transactions on Computer Science and Engineering*, 30, (2006), 25–36.

[32] Y. K. Jain and S. K. Bhandare, **Min max normalization based data perturbation method for privacy protection**. *International Journal of Computer and communication Technology*, 2, (2011), 45–50.

[33] R. Ghazali, A. Hussain, and W. El-Deredy, "**Application of Ridge Polynomial Neural Networks to Financial Time Series Prediction**", *Proceedings of the IEEE International Joint Conference on Neural Network*, (2006), 913–920.

[34] Z. Wang and A. C. Bovik, **Mean squared error: Lot it or leave it? A new look at signal fidelity measures**. *IEEE Signal Processing Magazine*, 26, (2009), 98–117.

[35] M. C. V. Ramırez, H. F. de C. Velho and N. J. Ferreira, **Artificial neural network technique for rainfall forecasting applied to the Sao Paulo region**. *Journal of Hydrology*, 301, (2005), 146–162.