

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/274376282>

Forecasting Seasonal Time Series with Functional Link Artificial Neural Network

Conference Paper · February 2015

DOI: 10.1109/SPIN.2015.7095387

CITATIONS

3

READS

200

3 authors:



Ina Khandelwal

The LNM Institute of Information Technology

6 PUBLICATIONS 29 CITATIONS

[SEE PROFILE](#)



Udit Satija

Shiv Nadar University

28 PUBLICATIONS 77 CITATIONS

[SEE PROFILE](#)



Ratnadip Adhikari

Fractal Analytics, Bengaluru, India

21 PUBLICATIONS 262 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



IT Growth Driver Analysis using Bayesian Belief Network (BBN) [View project](#)



Market Mix Modeling (MMM) using BBN [View project](#)

Forecasting Seasonal Time Series with Functional Link Artificial Neural Network

Ina Khandelwal*, Udit Satija†, Ratnadip Adhikari*

*Computer Science and Engineering

The LNM Institute of Information Technology Jaipur, India

Email: {ina, adhikari.ratan}@lnmiit.ac.in

†School of Electrical Sciences

Indian Institute of Technology Bhubaneswar, India

Email: us11@iitbbs.ac.in

Abstract—Many economic and business time series exhibit trend and seasonal variations. In this paper, we deal with efficient modeling of time series having seasonality and definitive trends. The traditional statistical models eliminate the effect of trend and seasonality from a time series before making future forecasts. This kind of preprocessing increases the computational cost and may even degrade the forecasting accuracy. Here, we present the effectiveness of Functional Link Artificial Neural Network (FLANN) model for seasonal time series forecasting, using unprocessed raw data. The forecasting results of FLANN for four seasonal time series are compared with those of the widely popular random walk model as well as the common feedforward neural network. The comparison clearly shows that FLANN produces considerably better forecasting accuracy than all other models for each of the four seasonal time series.

Keywords—seasonal time series, forecasting accuracy, functional link artificial neural network, random walk model.

I. INTRODUCTION

Time series forecasting (TSF) has crucial importance in various science and engineering domains [1]. The primary objective of time series analysis is to construct an appropriate mathematical model to estimate the underlying data generation process of the series and then to forecast the desired number of future observations through this model. Many economic and business time series contain inherent trend and seasonal patterns [2], which considerably complicate adequate modeling and forecasting. In time series analysis, it is often required to preprocess the data through a suitable transformation technique prior to fitting the model. Such a transformation is mainly applied for stabilizing the variance, removing trend or seasonal effects, smoothing out the outliers and normalizing the data. *Seasonal differencing* and *deseasonalization* are two most common preprocessing techniques to remove the inherent seasonal effect from a time series [2]. Seasonal differencing is carried out as follows:

$$\mathbf{T}_{data} = \mathbf{O}_{data}(s+1) - \mathbf{O}_{data}(1) \quad (1)$$

Here, \mathbf{T}_{data} represents the transformed data after deseasonalization and \mathbf{O}_{data} is the raw unprocessed data. The period of seasonality is denoted by “ s ”. Traditional approaches in seasonal time series forecasting require the series to be differenced and deseasonalized in order to make it stationary and free from the seasonal effect. However, these transformations reasonably increase the computational costs and also sometimes degrade the forecasting precision [2]. Also, it has been observed that deseasonalization often leads to overestimation of the seasonal

components whenever there is a presence of trend and as such produces reasonably bad forecasts [3].

Till now, a large amount of research has been carried out on adequate modeling and forecasting of seasonal time series. The outcome of this extensive research is the development of various seasonal time series forecasting models in literature. *Artificial Neural Networks* (ANNs) have been extensively used for time series forecasting since late 1980s. Some recognized works presented comprehensive study of ANNs in forecasting applications and their related modeling issues [4]. ANNs have also gained notable popularity in recognizing and predicting the seasonal pattern in a time series [5]. Hamzaebi [5], proposed a new ANN structure, known as the *Seasonal ANN* (SANN) for seasonal time series forecasting that determines the numbers of input and output neurons through the associated seasonal period of the series. It was showed that SANN outperformed both the traditional statistical models and the feedforward ANN in forecasting time series having strong seasonal patterns. Zhang [6] proposed a hybrid methodology by combining *Autoregressive Integrated Moving Average* (ARIMA) and ANN in order to separately model the linear and nonlinear components of a time series. Some studies claimed that ANNs are effective in directly forecasting the seasonal raw data [7–11]. But, some others claimed that in order to obtain effective ANN forecasts, seasonal patterns should be removed from the raw data [12–14]. Existing literature reveals varieties of ANN structures, such as SANN, *Generalized Regression Neural Network* (GRNN), *Elman ANN* (EANN) etc. to forecast seasonal patterns.

ANN’s forecasting performance solely depends on the appropriate network architecture, the hidden and output layer activation functions, the appropriate training algorithm etc. Theoretically, there is no rigorous method to determine these parameters [4, 6]. Usually, the suitable network parameters are selected through in-sample experiments. A single neuron based architecture, viz. Functional Link ANN (FLANN) is proposed by Pao [15]. FLANN has less computational cost and can be easily implemented in hardware applications due to the absence of hidden layer. The set of polynomials in FLANN captures the nonlinear relationship between the inputs and output [16]. FLANN has been used in stock market and exchange rate prediction [17, 18]. However, till now, it found little applications in forecasting seasonal time series. It seems that FLANN will be capable of modeling the seasonal and trend effects without applying any preprocessing techniques

on raw data. The major benefit of FLANN is that it does not require hidden layer, hidden nodes and activation functions. Because of its lower computational cost than ANN, FLANN can be a potentially good choice for seasonal time series forecasting. This paper addresses the performance of FLANN over the random walk and traditional feedforward ANN in seasonal time series forecasting.

The rest of the paper is organized as follows. Section II briefly describes about three different time series forecasting models, viz. random walk, ANN and FLANN. Experimental part is discussed in section III. In section IV, we present forecasting results for four real seasonal time series datasets and finally Section V concludes the paper.

II. TIME SERIES FORECASTING MODELS

In literature, different models have been developed for TSF. In this section, we present a brief discussion about the forecasting models, those are used in the present work.

A. Random walk Model

This model can be used, at least as a first approximation for many time series. The predictions through random walk are obtained as follows:

$$y_t = y_{t-1} + \varepsilon_t \quad (2)$$

Here, ε_t denotes a purely random process. In this model, the value of \mathbf{Y} at time t is equal to its value at time $(t-1)$ plus a random shock. The predicted series does not form a stationary process as it is easy to show that the variance increases through time. However, the first differences of the series, viz. $y_t - y_{t-1}$ do form a stationary process.

B. Neural Network Model

ANNs are universal approximators, which are able to approximate various nonlinear patterns in the data upto any desired level of accuracy [4]. ANN is a very important tool for modeling and prediction of nonlinear time series because of its nonlinear mapping from input nodes to output node. A single hidden layer feedforward ANN with one output node is most widely used in forecasting applications. The relationship between the inputs $\{z_{t-1}, z_{t-2}, \dots, z_{t-p}\}$ and the output z_t with p input nodes and q hidden nodes can be represented by following mathematical representation [6]:

$$z_t = \beta_o + \sum_{j=1}^q \beta_j h(\gamma_{oj} + \sum_{i=1}^p \gamma_{ij} z_{t-i}) + \varepsilon_t \quad (3)$$

Here, β_j, γ_{ij} ($i = 0, 1, 2, \dots, p; j = 0, 1, 2, \dots, q$) are the weights; p, q are respectively the number of input and hidden nodes and ε_t is white noise. Here, logistic function is used as the hidden layer activation function h , i.e.

$$h(x) = \frac{1}{1 + \exp(-x)} \quad (4)$$

The major drawback of ANN is that it largely depends on the number of nodes in each layer and the associated activation function. Due to this, ANN is often computationally expensive compared to other traditional statistical forecasting models.

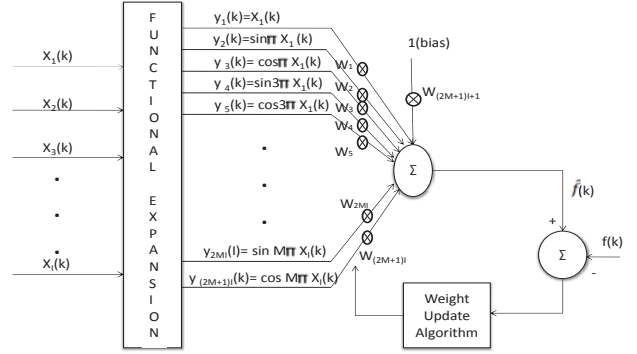


Fig. 1: FLANN Model

C. FLANN Model

Multi-layer neural network can be replaced by single layer neural network to overcome the computational complexities. But, such an architecture sometimes fails to adequately model the nonlinear relationships due to the linear nature of single layer network. In order to resolve the problem of high computational complexity and linearity in the single layer neural network, FLANN model is suggested [19]. It uses a single layer feed forward neural network and functionally expands the input vector to overcome the issue of linearity [20].

Let, \mathbf{x} be the dataset available at the input of the model as shown in Fig. 1, which can be written as $\mathbf{x}_i(k) = [\mathbf{x}_1(k), \mathbf{x}_2(k) \dots \mathbf{x}_I(k)]^T$, where $k = 1, 2, \dots, N$ is the number of samples in the datasets; I is the set of features in the dataset and T is the transpose operator. Then, each \mathbf{x}_i is functionally expanded (here, trigonometric expansion is used), as shown in Fig. 1 [17], which can be written as:

$$\mathbf{y} = [\mathbf{y}_1(k), \mathbf{y}_2(k) \dots \mathbf{y}_I(k)]^T \quad (5)$$

where each $\mathbf{y}_i(k)$ is the expansion of each $\mathbf{x}_i(k)$, which is,

$$\mathbf{y}_i(k) = [x_i(k), \sin(2n-1)\pi x_i(k), \cos(2n-1)\pi x_i(k)]^T \quad (6)$$

Here, $n = 1, 2, \dots, M$. The value of M will depend on the number of expansions. It is to be noted that the total number of expansions of each $\mathbf{y}_i(k)$ is $(2M+1)$. Bias weight is added to help in successful learning by shifting the activation function. It can be written as:

$$\mathbf{v} = [\mathbf{y}; 1] \quad (7)$$

Then, random weights are initialized in the range $[0, 1]$ and then multiplied with the expanded values along with bias weight, and then finally summed up to produce the estimated output of the model [20]. The final estimated output can thus be written as follows:

$$\hat{\mathbf{f}}(j) = \Phi(\mathbf{w}^T \mathbf{v}) = \sum_{t=1}^{(2M+1)I+1} \mathbf{v}_t(k) \cdot \mathbf{w}_t(j) \quad (8)$$

Here, $\hat{\mathbf{f}}(j)$ is the estimated output for j^{th} iteration; Φ is the non linear activation function and $\mathbf{w}(j) = [w_1(j), w_2(j), \dots, w_{(2M+1)I}(j), w_{(2M+1)I+1}(j)]$, where $j = 1, 2, \dots, L$. The error term is computed as [20]:

$$\mathbf{e}(j) = \mathbf{f}(j) - \hat{\mathbf{f}}(j) \quad (9)$$

TABLE I: Description of the four time series datasets

Time series	Description
Temperature	Average monthly temperature in New Delhi, India (January, 1986–May, 2011) (Source: www.datamarket.com) Total size: 302, Training size: 242, Testing size: 60
Turnover	Monthly turnover in services in Spain (January, 2000–August, 2014) (Source: www.datamarket.com) Total size: 176, Training size: 140, Testing size: 36
Australian Wine	Monthly Australian wine sales (January, 1980–July, 1995) (Source: www.datamarket.com) Total size: 187, Training size: 151, Testing size: 36
Indian Mining	Monthly mining data of India (April, 1981–March, 1988) (Source: www.data.gov.in) Total size: 204, Training size: 174, Testing size: 30

Weight update algorithm: Suppose, N patterns contribute to one experiment. Weights are updated using the gradient decent algorithm [17] at the end of each experiment by computing the average change in the m^{th} weight, which is given by:

$$w_m(j+1) = w_m(j) + \Delta w_m(j) \quad (10)$$

Here, $j = 1, 2, \dots, L$. The average change of the m^{th} weight at the j^{th} experiment is computed as:

$$\Delta w_m(j) = \frac{1}{N} \sum_{n=1}^N 2\mu \mathbf{v}(k) \mathbf{e}(j) \quad (11)$$

III. EXPERIMENTAL SETUP

A. Data Collection

The performance of the FLANN model is evaluated using four real seasonal time series datasets. Three of these datasets are collected from the Data Market repository (www.datamarket.com) and the fourth one is collected from the Open Government data platform, India (www.data.gov.in). All the four dataset are depicted in Fig. 2. In addition, their description is presented in Table I.

B. Training and Testing Processes

During the training process, weights are initialized to some random values in the interval $[0, 1]$. Weights are updated once after each experiment in the training process and respective errors are stored. In this study, Mean Square Error (MSE) is considered as the cost function for the training process. The weights for which the MSE value is minimum are stored for testing the network. In this paper, we have evaluated the performance of different forecasting models through three error functions, viz. Mean Absolute Error (MAE), MSE and Root Mean Square Error (RMSE). These three error measures are defined below:

$$\text{MAE} = \frac{1}{N} \sum_{t=1}^N |e_t| \quad (12)$$

$$\text{MSE} = \frac{1}{N} \sum_{t=1}^N e_t^2 \quad (13)$$

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{N} \sum_{t=1}^N e_t^2} \quad (14)$$

Here, $e_t = y_t - \hat{y}_t$ is the prediction error, y_t , \hat{y}_t being the actual and predicted observation at time t , respectively and N is the size of the testing dataset.

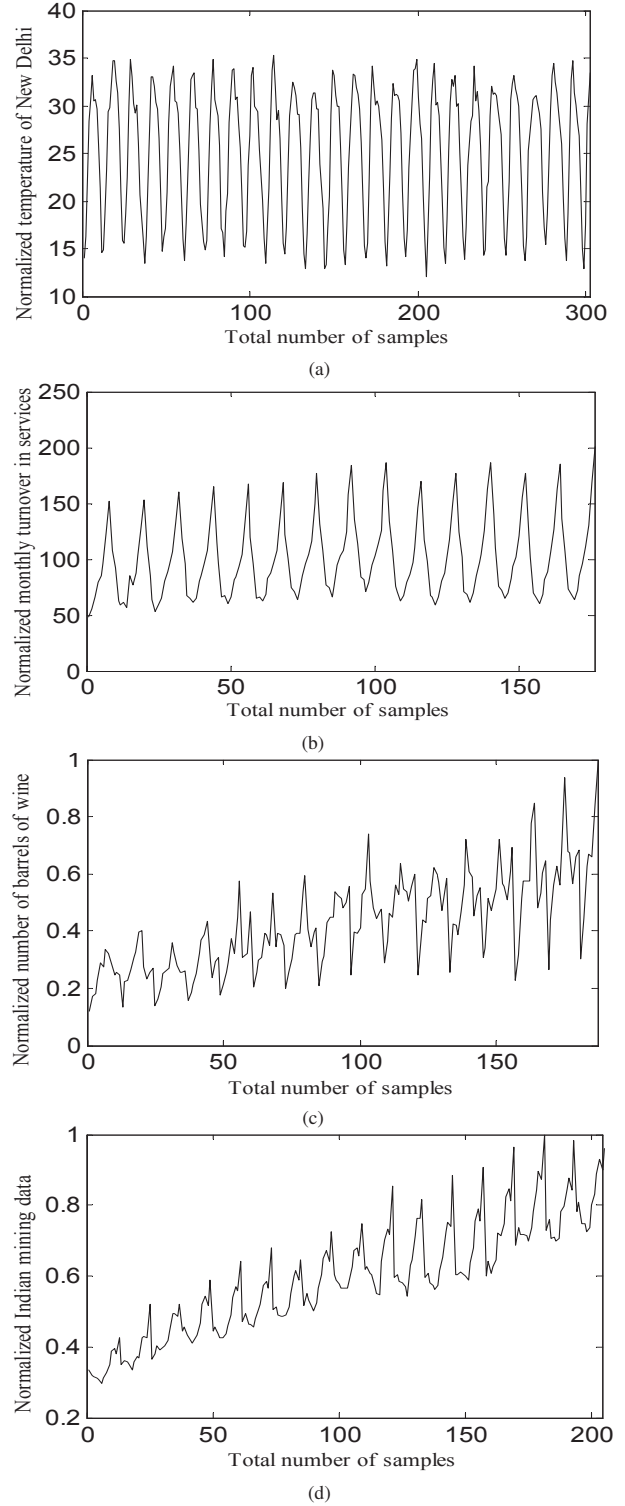


Fig. 2: Plots of the time series: (a) Temperature, (b) Turnover, (c) Australian wine, (d) Indian mining

IV. RESULTS AND DISCUSSIONS

In this study, two forecasting models, viz. random walk and ANN are used for comparison with FLANN. For each seasonal time series, the forecasting accuracy of FLANN is tested and compared with the other two models through the three error measures, viz. MAE, MSE and RMSE.

Table II gives the forecasting results for the Temperature time series. Random walk is performed. A $(10 \times 8 \times 1)$ ANN, having 10 input nodes and 8 hidden nodes has been used for this series. FLANN with 4 inputs and 9 trigonometric expansions has been found to be the most appropriate model than the other two models in terms of forecasting accuracy. Results of turnover in services is shown in Table III. A $(12 \times 6 \times 1)$ ANN structure has been found to be appropriate for this series. FLANN model with 4 inputs and 11 expansions give much better results than the random walk and ANN models. Table IV gives the forecasting results for the Australian wine series. For this series, random walk and a $(10 \times 7 \times 1)$ ANN is used. FLANN with 4 inputs and 9 expansions is fitted. The results of FLANN are quite better than random walk as well as ANN. Finally, the forecasting results of the Indian mining dataset is presented in Table V. For this dataset, FLANN with 4 inputs and 11 expansions has been fitted and the results of FLANN are better than those of both random walk and $(12 \times 9 \times 1)$ ANN models.

The plot of the actual testing datasets and its forecast through FLANN for each of the four seasonal time series is depicted in Fig. 3. From Tables II–V, it can be clearly seen that FLANN model has achieved lower forecasting errors than both random walk and ANN. This important finding demonstrates the capability of FLANN in forecasting seasonal time series, without eliminating the effect of seasonality.

TABLE II: Results for Temperature

Forecasting Model	MSE	MAE	RMSE
Random Walk	0.06251	0.19980	0.25002
ANN $(10 \times 8 \times 1)$	0.01039	0.08284	0.10196
FLANN	0.00511	0.06257	0.07153

TABLE III: Results for Turnover

Forecasting Model	MSE	MAE	RMSE
Random Walk	0.03640	0.15862	0.19079
ANN $(12 \times 6 \times 1)$	0.00923	0.08114	0.09610
FLANN	0.00735	0.06424	0.08575

TABLE IV: Results for Australian Wine

Forecasting Model	MSE	MAE	RMSE
Random Walk	0.03514	0.15190	0.18748
ANN $(10 \times 7 \times 1)$	0.01944	0.11885	0.13944
FLANN	0.01672	0.10080	0.12931

TABLE V: Results for Indian Mining

Forecasting Model	MSE	MAE	RMSE
Random Walk	0.00687	0.06978	0.08290
ANN $(12 \times 9 \times 1)$	0.00401	0.05551	0.06338
FLANN	0.00346	0.04548	0.05888

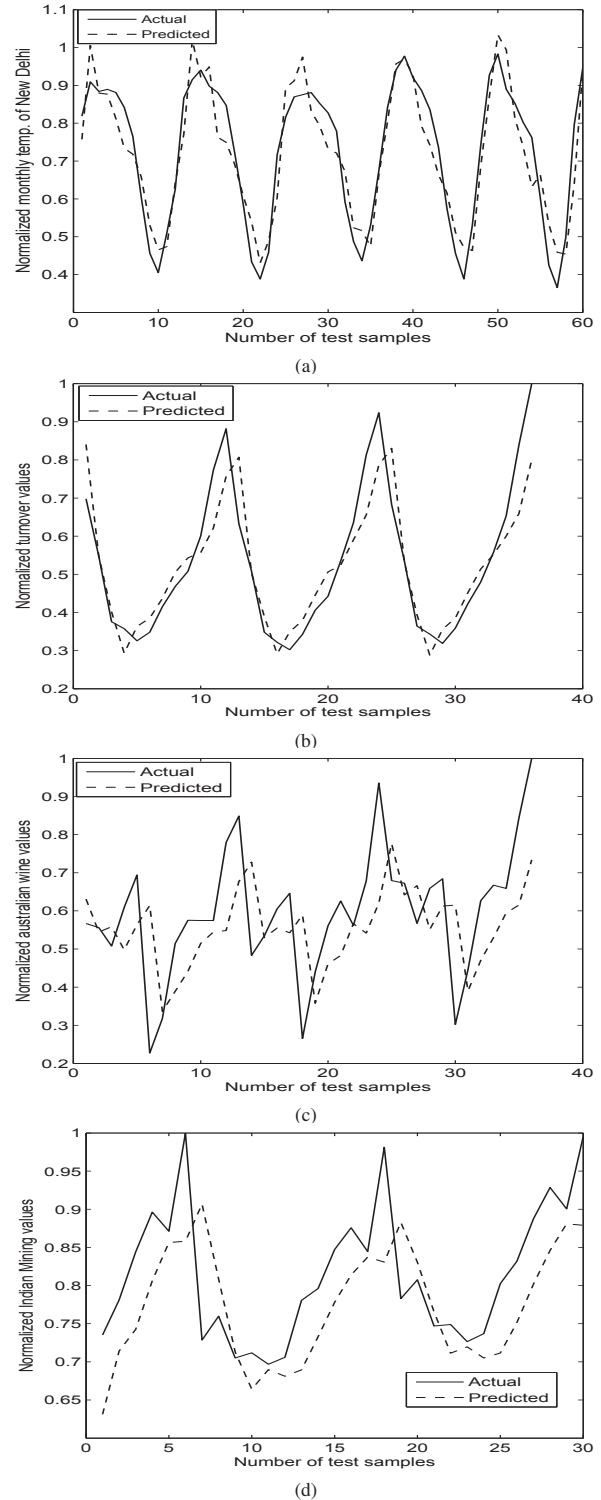


Fig. 3: Testing dataset and its FLANN forecast for: (a) Temperature, (b) Turnover, (c) Australian wine, (d) Indian mining

V. CONCLUSION

In this paper, the FLANN based model is introduced for seasonal time series forecasting. FLANN model is structurally simple, computationally economical and less complex than the iterative feedforward ANN. FLANN can model the seasonal and trend patterns without applying any preprocessing technique on raw data which helps in reducing the associated computational time. This model basically depends on two parameters: the number of inputs and number of expansions. So, it is necessary to choose the suitable combination of these parameters to get better results. The experimental work in this study shows that FLANN achieved reasonably better forecasting results than the random walk as well as traditional feedforward ANN for each of the four seasonal time series. Thus, from this study, it can be said that FLANN can be a computationally economical and effective model for seasonal time series forecasting.

REFERENCES

- [1] J. G. De Gooijer, R. J. Hyndman, "25 years of time series forecasting," *Int. J. Forecasting*, vol. 22, no. 3, pp. 443-473, 2006.
- [2] R. J. Hyndman, "The interaction between trend and seasonality," *Int. J. Forecasting*, vol. 20, no. 4, pp. 561-563, 2004.
- [3] D. M. Miller, "Damping seasonal factors : Shrinkage estimators for the X-12-ARIMA program," *Int. J. Forecasting*, vol. 20, no.4, pp. 529-549, 2004.
- [4] G. Zhang, B. Eddy Patuwo and M. Y. Hu, "Forecasting with artificial neural networks - The state of art," *Int. J. Forecasting*, vol. 14, pp. 35-62, 1998.
- [5] C. Hamzaebi, "Improving artificial neural networks performance in seasonal time series forecasting," *Inform. Sci.*, vol. 178, no. 23, pp. 4550-4559, 2008.
- [6] G. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159-175, 2003.
- [7] R. Sharda and R. B. Patil, "Connectionist approach to time series prediction : an emperical test," *J. intell. Manufacturing*, no. 3, pp. 317-323, 1992.
- [8] Z. Tang, C. Almedia and P. A. Fishwick, "Time series forecasting using neural networks vs Box-Jenkins methodology," *Simulation*, vol. 57, no. 5, pp. 303-310, 1991.
- [9] K. Nam and T. Schaefer, "Forecasting international airline passenger traffic using neural networks," *Logistic and transportation*, vol. 31, no. 3, pp. 239-251, 1995.
- [10] P. H. Franses, G. Draisma, "Recognizing changing seasonal patterns using artificial neural networks," *J. Econometrics*, vol. 81, pp. 273-280, 1997.
- [11] I. Alon, M. Qi and R. J. Sadowski, "Forecasting aggregate retail sales: a comparison of artificial neural networks and traditional methods," *J. Retailing Consumer Services*, vol. 8, no. 3, pp. 147-156, 2001.
- [12] T. Kolarik and G. Roderfer, "Time series forecasting using neural networks," *APL Quote Quad*, vol. 25, no. 1, pp. 86-94, 1994.
- [13] M. Nelson, T. Hill, W. Remus and M. O. connor, "Can neural networks applied to time series forecasting learn seasonal patterns : an emperical investigation," *Proc. 27th Annu. Hawaii Int. conf. Syst. Sci.*, vol. 3, pp. 649-655, 1994.
- [14] G. Zhang and M. Qi, "Neural network forecasting for seasonal and trend time series," *European J. Operational Research*, vol. 160, pp. 501-514, 2005.
- [15] Y. H. Pao, *Adaptive pattern recognition and neural networks*, Reading, MA: Addison-Wesley, 1989.
- [16] C. J. Lin and C. F. Wu, "An efficient symbolic particle swarm optimization for recurrent functional neural network design," *Iranian J. Fuzzy Syst.*, vol. 11, no. 4, pp. 262-271, 2009.
- [17] R. Majhi, G. Panda and G. Sahoo, "Development and performance evaluation of FLANN based model for forecasting of stock markets," *Expert Syst. Applicat.*, 2009.
- [18] B. Majhi, M. Rout, R. Majhi, G. Panda, P. J. Fleming, "New robust forecasting models for exchange rates prediction," *Expert Syst. Applicat.*, vol. 39, no. 16, pp. 12658-12670, 2012.
- [19] R. Agrawal, T. Imielinski and A. Swami, "Database mining: A performance perspective," *IEEE Trans. Knowledge Data Eng.*, no. 5, pp. 914-925, 1993.
- [20] B. B. Misra and S. Dehuri, "Functional Link Artificial Neural Network for Classification Task in Data Mining," *J. Comput. Sci.*, vol. 12, no. 3, pp. 948-955, 2007.
- [21] Y. H. Pao, S. M. Phillips and D. J. Sobajic, "Neural net computing and intelligent control systems," *Int. J. Control*, vol. 56, no. 2, pp. 263-289, 1992.