Indian Journal of Science and Technology, Vol 10(16), DOI: 10.17485/ijst/2017/v10i16/104994, April 2017
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

# A Novel Fuzzy Time Series Model for Stock Market Index Analysis using Neural Network with Tracking Signal Approach

**D. Ashok Kumar[1], S. Murugan[2*]**

[1]Department of Computer Science, Government Arts College, Tiruchirappalli – 620022, Tamil Nadu, India; Drashoktrichy22@gmail.com

[2]Department of Computer Science, Alagappa Government Arts College, Karaikudi – 630003, Tamil Nadu, India; muruganjit@rediffmail.com

## Abstract

**Objectives:** To present a novel Fuzzy Time Series Neural Network (FTSNN) with Tracking Signal (TS) approach for forecasting the closing index of the stock market. **Methods/Statistical Analysis:** A novel approach strives to adjust the number of hidden neurons of a Multi-Layer Feed Forward Neural Network (MLFFNN) model. It uses the Tracking Signal (TS) and rejects all models which result in values outside the interval of [-4, 4]. **Findings:** The effectiveness of the proposed approach is verified with one step ahead of Bombay Stock Exchange (BSE100) closing stock index of Indian stock market and Taiwan Stock Exchange Stock Index (TAIEX). This novel approach reduces the over-fitting problem, reduces the neural network structure and improves forecasting accuracy. In addition, the presented approach has been tested on standard NN3 (Neural Network 3) forecasting competition time series dataset and this approach out performs the various models tested with the NN3 forecasting competition. **Applications/Improvements:** The proposed approach can be applied to different types of neural network for forecasting closing stock index/price of stock market data.

**Keywords:** Forecasting, Fuzzy Time Series Data, Neural Network, Stock Index, Tracking Signal

## 1. Introduction

Forecasting stock market return has gained more attention in recent days. If the future of a stock market is successfully predicted then the investors are better guided. Though various prediction models are available, no model predicts consistently. These ambiguous, inconsistent predictions have motivated the researcher to explore a new model to forecast the stock market effectively. If a system can developed with consistency in predicting the trends of the dynamic market, then it would take developer to cloud nine.

Time series forecasting is used to predict the future according to the historical observations. Traditional methods includes time-series regression, Auto Regressive Integrated Moving Average (ARIMA) and exponential smoothing are based on linear models. All these methods assume that linear relationship among the past values of the forecast variable and therefore non-linear patterns cannot be captured by these models[1]. A number of Neural Network (NN) models [2-7] and hybrid models [8,9] have been proposed during the last few years for obtaining accurate forecasting results. These were attempts to improve the conventional linear and nonlinear approaches.NNs are non-linear in nature, So, NN are preferred over the traditional models. Application of NN[10] on credit ratings[2], Dow Jones Forecasting[3], customer satisfaction analysis[4], stock ranking[5], and Foreign exchange rate forecasting[6] and tourism demand[7] was varied and effective. The reason is that the NN is a universal function approximation which can map any linear or non-linear functions.

Although NNs have the advantages of accurate forecasting, (i) there is no systematic rule to identify neuron

*Author for correspondence

numbers in the hidden layer[9]. (ii) NN model suffers due to under-fittingor over-fitting problems.

To solve the problem of neuron numbers in the hidden layer issue, a geometric pyramid rule[12], for a three layer NN with m output and n input neurons, the hidden layer may have square root (n∗m) neurons. A NN with 2N + 1 hidden neuron and one hidden layer is sufficient for N inputs, and observedthat the optimum number of hidden layers and hidden neurons are highly problem dependent[13]. As the accuracy of NN model depends on the careful NN model design, adetailed Neural Network (NN) designing methodology and training process is reported in the literature[14-16]. The performance of various types of training algorithms[17,18] analyzed and the Levenberg-Marquardt training algorithm has better performance than all other training algorithms and also its error rate is very low when compared to all other training algorithms. Greg Heath[19] suggests that design of ten neural networks with different types of random initial weights to mitigate the occasional bad random start. Training a great number of ANN with different configurations and selecting the optimum model will improve forecasting accuracy[20].

In many applications[9,10,21] the data set is divided into two sets: training and testing set. This data partition leads to over-fitting or under-fitting in NN performance. To avoid over-fitting or under-fitting problem and increase the robustness of the NN performance, the original dataset is divided into three different parts training set; validation set (a small portion of training set) and test set[22]. The published research articles[8,23,24] reported that the optimum NN model selection is based on minimum forecasting error in validation set of some performance measure (SMAPE, NMSE, RMSE, etc) and reports its corresponding results in test set to avoid over-fit problem.

This study claimed that, after selecting the optimum neural network model, still, there exists over-forecast or under-forecast in training, validation and test set. The performance of NN model degrades if over-forecast or under-forecast occurs. To solve the e mentioned problem, this paper recommends a novel fuzzy time series model using neural network with Tracking Signal (FTSNN with TS) approach. TS are used to identify the presence of over-forecast or under-forecast in the NN model. The proposed FTSNN with TS approach systematically constructs different fuzzy-neural network model from simple architecture to complex architecture; and the optimum fuzzy-neural network model selection is based on the TS interval value [-4, 4] in the training set and validation set which contains minimum forecasting performance error in SMAPE (instead of SMAPE, some other performance can be used) of validation set for solving the problem of identifying best neural network model which reduces over-fitting or under-fitting problem.

In[25] reported that, the TS are a statistical measure which is used to assess the presence of bias in the forecast model; and also it warns when there are unexpected outcomes from the forecast. In[22] proposed that adaptive smoothing approach is used to adjust the NN learning parameters automatically by TS under dynamic varying environments. In their study TS is used during the NN training. In the present study, the TS are used to analyze and select the best NN model after the NN training to improve forecasting accuracy.

The contribution of this study is, first, different fuzzy-neural network architecture created for forecasting the closing stock index of the BSE100 and TAIEX stock market. Second, the performance measure Tracking Signal (TS) is introduced to select the optimum fuzzy-neural network model which reduces the network complexity; faster in convergence; improves better forecast accuracy; and avoids over-forecast and under-forecast. Third, the in-sample (train set and validation set) and the out-of-sample (test set) forecasting performance analyzed using the different performance measure such as SMAPE, RMSE, POCID and TS using FTSNN with TS approach and FTSNN without TS approach. Fourth, the neuron numbers in the hidden layer is identified for BSE100 and TAIEX stock market. Fifth, the proposed approach has been tested on standard NN3 forecasting competition time series dataset and it outperforms the various models tested with the NN3 forecasting competition. Sixth, the performance of the proposed approach is compared with the neural network based fuzzy time series (NNFTS) model proposed by[10] and it outperformed. Seventh, unlike the report of[11] the investigation of this study proves that the in-sample (training and validation set) model selection criteria can be provide a reliable guide to out-of-sample (test set) performance and there can be an apparent connection between in-sample (training and validation set) model fit and out-of-sample (test set) forecasting performance.

Rest of this study is organized as follows: Section 2 describes the essential part of MLFFNN model, fuzzy time series model, TS and performance measures which are used to assess the performance of the proposed approach;

Section 3 describes the details of proposed FTSNN with TS approach and FTSNN without TS approach; Section 4 reports the experimental results attained by the FTSNN with TS approach and FTSNN without TS approach using real world financial time series such as BSE100, TAIEX stock market and NN3 time series forecasting dataset. Finally this study is concluded in section 5.

## 1.1 Multi-Layer Feed Forward Neural Network Model

MLFFNN model comprises of an input layer, an output layer and one or more hidden layers. The hidden layer collects weight from input layer. Each subsequent layer collects weight from the previous layer. The neurons present in the hidden and output layers have biases, which are the connection from the units and its activation is always shown in Figure 1.

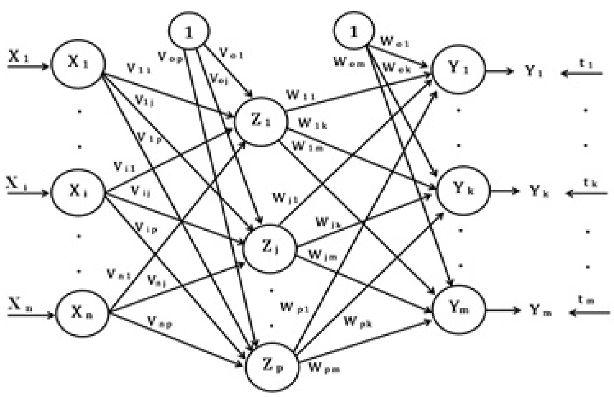

**Figure 1.** Multi-Layer Feed Forward Neural Network Architecture.

The bias term also acts as weights and it shows the architecture of Back Propagation Neural Network, illustrating only the direction of information flow for the feed forward phase. During the back propagation phase of learning, signals are sent in reverse direction. The inputs are sent to the back propagation network and the output obtained from the net could be either binary 0, 1 or bipolar -1, +1 activation function. The error back propagation training algorithm is purely based on the gradient descent method[26].

## 1.2 Fuzzy Time Series Model

Fuzzy time series models, a complement of traditional time series models, have become more increasingly popular in recent years. Some successful application of fuzzy time series models such as high-order models, first-order models, bivariate models, multivariate models seasonal models and hybrid models[10].

Fuzzy time series data are structured by fuzzy sets. Let U be the universe of discourse, such that $U = \{u_1, u_2, \ldots. u_n\}$. Let us defined a fuzzy set A of U by

$$A = \frac{f_A(u_1)}{u_1} + \frac{f_A(u_2)}{u_2} + \ldots \frac{f_A(u_n)}{u_n}$$

where $f_A$ is the membership function of A, and $f_A: U \rightarrow [0, 1]$. $f_A(u_i)$ is the membership value of $u_i$ in A, where $f_A(u_i) \in [0, 1]$ and 1 ≤ i ≤ n. Tiffany Hui-Kuang Yu and Kun-Huang Huarng[10] proposed a sequence of steps to design Neural Network based Fuzzy Time Series (NNFTS) model.

## 1.3 Tracking Signal

The calculation of the TS[25] is represented in the equation (4). If the forecast value is lower than the actual value then the model is in under forecasting and TS will be positive. If the forecast value is higher than the actual value then the model is in over forecasting and TS will be negative. If the TS limit is between the interval [-4, +4] then the forecast model is working correctly. The threshold of 4 is really a threshold of 3.75 (3SD). This 3.75 number comes from the statistical control limit theory which establishes the relationship between Mean Absolute Error or Deviation and Standard Deviation. The relationship between the Standard deviation and MAD in a normally distributed population is built as 1.25 MAD = 1 SD (standard deviation of the distribution).

## 1.4 Forecasting Performance Measure

The forecasting performance is evaluated using the statistical measures, namely, Symmetric MeanAbsolute Percentage Error[23] (SMAPE), Percentage of Change in Direction[8] (POCID), Root Mean Square Error[10] (RMSE) and Tracking Signal[25] (TS).

In the following measure $f_t$ represents forecasted value and $y_t$ represents actual value, $e_t = y_t - f_t$ represents forecast error and n represents size of test set.

The global performance of a forecasting model is evaluated by the SMAPE[23] which is used in NN3 (monthly time series), NN5 (daily time series) and NNGC1 (Neural

Network Grand Competition) forecasting competition. A smaller SMAPE value suggests the better forecasting accuracy. It can be expressed as

$$SMAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{|e_t|}{\frac{y_t + f_t}{2}}X100 \qquad (1)$$

The RMSE[10] is the square root of calculated MSE. All the properties of MSE hold for RMSE as well. RMSE can be expressed as

$$RMSE = \frac{\sqrt{\sum_{t=1}^{n} e_t^{\,2}}}{n} \quad (2)$$

POCID (Percentage of Change in Direction)[8] maps the accuracy in the forecasting of the future direction of the time series. A larger POCID value indicates better forecasting accuracy. It leads to 100 % means, the model is considered as a perfect model. It can be represented as

$$POCID = 100\frac{\sum_{t=1}^{n} D_t}{n} \qquad (3)$$

$$where\ D_t = \begin{cases} 1\ if\ (y_t - y_{t-1})(f_t - f_{t-1}) \geq 0 \\ 0\ otherwise \end{cases}$$

Cecil bozrath[25] reported that the Tracking Signal (TS) is used to pinpoint forecasting models that need adjustment. As long as the TS are between -4 and +4, assume the model is working correctly. It can be represented as,

$$TS = \frac{\sum_{t=1}^{n} e_t}{MAD} \qquad (4)$$

The Mean Absolute Deviation (MAD) measures the average absolute deviation of forecasted values from original ones.

$$MAD = \frac{\sum_{t=1}^{n} |e_t|}{n} \qquad (5)$$

## 2. Proposed Methodology

Over fitting is the main issue in neural network modeling. In order to reduce the over fitting problem, this study proposes a novel approach Fuzzy Time Series Neural Network with Tracking Signal (FTSNN with TS) which is used to forecast the closing index of the stock market. Multi-Layer Feed Forward Neural Network (MLFFNN) receives fuzzified data and trains different network by using different random initial weight and different neurons. Tracking Signalmeasure is used to reject all FTSNN model which results in values outside the interval of [-4, +4] in training set and validation set of different neural networks.

Training parameter and the weight play an important role in neural network modeling to increase the forecasting accuracy. The proposed FTSNN with TS approach is tried to find optimal parameter, particularly, neuron numbers in the hidden layer and optimal weight for the forecasting problem in time series.

Forecasting strategies are taken a step ahead of prediction in this study. Let $y_1$, $y_2$, $y_3$ ….. $y_t$ be a time series. As time t for t>=1, the next value $y_{t+1}$ is predicted based on the observed realizations of $y_t$, $y_{t-1}$, $y_{t-2}$ ….. $y_1$. The resultoutNN can be used for multi-step prediction by feeding the prediction back to the input of NN recursively. The FTSNN with TS approach is represented in Figure 2.

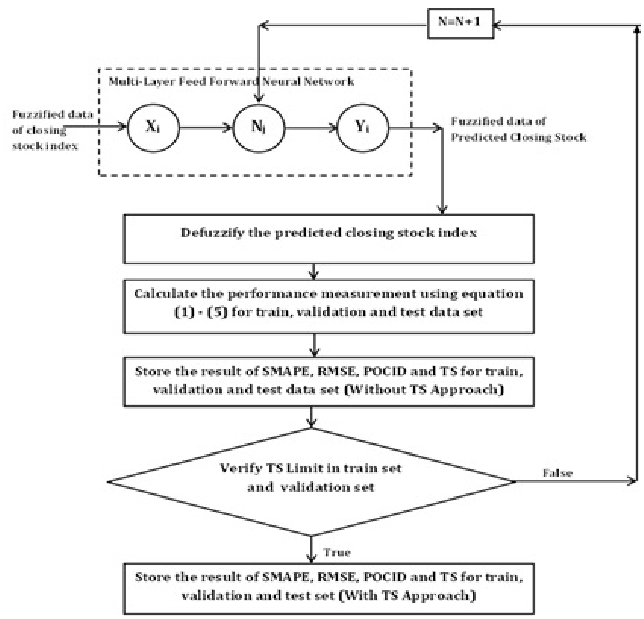


**Figure 2.** Fuzzy Time Series model using Neural Network with Tracking Signal Approach.

In Figure 2, $X_i$ is the fuzzified data of closing stock index vector, $Y_i$ is the fuzzified data of predicted closing stock index from neural network model and $N_j$ is neurons size in hidden layer. For every NN model, verify the presence of tracking signal interval [-4, +4] in training set and validation set. If it is present, the model is considered as feasible model otherwise the model is rejected. This process is repeated until the specified trial number (random initial weight) and maximum neuron size is reached.

The implementation procedure of FTSNN with TS approach is represented in Algorithm 1, and explained further as follows. Neural network training process is an iterative process. Before training the NN, the input data and target data should be converted into fuzzy data using the step 1 to 4 in Algorithm 1.

**Algorithm 1.A Novel Fuzzy Time Series Model Using Neural Network with Tracking Signal (FTSNN with TS Approach).**

**Input:** Fuzzy time series data for the closing stock index vector.

**Output:**Fuzzy Time Series data for Predicted closing stock index vector.

1. **Difference:**Obtain the differences between every two subsequent observations at t and t-1, $d(t-1,t) = obs(t) - obs(t-1)$ where obs(t) and obs(t−1) are two subsequent observation at t and t−1, d (t−1) is their difference.
2. **Adjustment:** The differences may be negative. To make all the Universes of discourse are positive, add various positive constants to the differences for various years $d'(t-1,t) = d(t-1,t) + const$

For each year, find the maximum and minimum of all the differences, $D_{\min}$ and $D_{max}$.

$D_{max} = \max\{d'(t-1,t)\}$, EMBED Equation.3 $t$.

3. **Universe of discourse:** The Universe of discourse U is defined as $[[D]_{min} - D_1, D_{max} + D_2]$, where $D_2$ and $D_1$ are two proper positive numbers. The length of the interval is fix to l, then divide U into equal intervals and let it be $u_1, u_2, u_3, \ldots$

Where $u_1 [= [D]_{min} - D_1, D_{min} - D_1 + l]$,

$u_2 [= [D]_{min} - D_1 + 2l, \quad D_{min} - D_1 + 3l]$,

……

$u_k [= [D]_{min} - D_1 + (k-1)l, \quad D_{min} - D_1 + kl]$.

Their corresponding midpoints are

$$m' = \frac{D_{min} - D_1 + D_{min} - D_1 + l}{2} = D_{min} - D_1 + \frac{l}{2}$$

$$m^2 = \frac{D_{min} - D_1 + l + D_{min} - D_1 + 2l}{2} = D_{min} - D_1 + \frac{3l}{2}$$

…..

$$m^k = D_{min} - D_1 + \frac{2\,x\,(k-1)x\,l}{2}$$

Define the linguistic values of the fuzzy sets. Suppose $A_1, A_2, A_3, \ldots$ are linguistic values. Label all the fuzzy sets by all possible linguistic values $u_1, u_2, u_3, \ldots$

4. **Fuzzification:** $d'(t-1,t)$ can be fuzzified into a set of degrees of membership, V(t-1,t), where
$$V(t-1,t) = \mu^1_{t-1,t}, \mu^2_{t-1,t}, \ldots.$$
5. **Neural Network Creation and Training:** Before training the neural network, Set the maximum number of neuron size MAX_NEURON in hidden layer, maximum number of trial MAX_TRIAL (random initial weight) for random weight generation and SD (Standard Deviation) value for assigning TS limit.
5.1 FOR NEURON = 1 TO MAX_NEURON
5.2 FOR TRIAL = 1 TO MAX_TRIAL

5.2.1. Create neural network architecture; specify the input and target vector from step 1, NEURON, TRIAL, training function, transfer function used in the hidden and output layer.

5.2.2. Select the data division ratio using divide function and divide the dataset into training dataset, validation dataset and test dataset using divideparam function. Training dataset and validation dataset are referred to as in-sample observation. Test dataset is referred to as out-of-sample observation.

5.2.3. Train the NN using train function.

6. **Neural Network Forecasting:** With $V(t-1,t)$ can proceed to forecast $V(t,t+1)$ by means of the trained NN. In-sample observations are divided into two sets namely training dataset and validation dataset. In-sample observations are referred to as training dataset and Out-of-sample observations are referred to as test dataset.
7. **Defuzzification:** Defuzzify the degrees of membership:
$$fd(t-1,t) = \frac{\sum_{k=1} \mu^k_{t-1,t} \; X \; m^k}{\sum_{k=1} \mu^k_{t-1,t}}$$

Where $fd(t-1,t)$, the forecasted difference between t-1 and t. is $\mu^k_{t-1,t}$ denotes the forecasted degrees of membership and $m^k$ represents the corresponding midpoints of the interval $\mu^k_{t-1,t}$.

8. **Forecasting:** After obtain the forecasted difference between t-1 and t, find the forecast for t:
$$forecast(t) = fd'(t-1,t) + obs_{t-1}$$
9. **Performance Evaluation:** Calculate the performance measure SMAPE, POCID, RMSE and TS for train, validation and test set using equation (1) - (5).
10. **Record** the result of neuron size, trial number, epoch (convergence speed), training time and performance

measure specified in step 9. It contains the performance of different **FTSNN without TS** approach.

**11.** Verify the interval $[-\theta, +\theta]$ of Tracking Signal in training set (TStrain) and validation set (TSvalidation) from step 10, where $\theta$ =round (SD * 1.25).

If (TStrain ≥ $-\theta$ &&TStrain ≤ $+\theta$ ) and (TSvalidation ≥$-\theta$ && TSvalidation ≤ $+\theta$ ) then go to step 12. Otherwise, go to step 5.2.

**12.** Record the result of neuron size, trial number, epoch (convergence speed), training time and performance measure specified in step 9. It contains the performance of different **FTSNN with TS** approach.

**13.** END// MAX_TRIAL

**14.** END // MAX_NEURON

**15.** From the step 10, select the optimum fuzzy time series neural network model, which provides less error in SMAPE for FTSNN without TS approach.

**16.** From the step 12, select the optimum fuzzy time series neural network model, which provides less error in SMAPE for FTSNN with TS approach.

The fuzzified data can be divided into three parts: a training, validation and test dataset. Training dataset can be used to fit the models, validation dataset can be used to evaluate the forecasting error for model selection; test dataset can be used to assess the generalization error in the final model. Divide block method is used to distribute the dataset into train, validation and test data set. After the division of data chosen, FTSNN model with tan sigmoidal function in the hidden layer and linear function in the output layer is used. The tan sigmoidal function and linear function is defined in equation (6) and (7).

$$tansig(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{6}$$

$$purelin(x) = x \tag{7}$$

Levenberg Marquardt is used as a training algorithm. After training the NN, simulate the NN and defuzzify the simulated output using the step 7 and 8 in Algorithm 1. Finally analyze the performance of NN using performance measure equation (1) - (5).

The FTSNN training process is represented in step 1 to step 10 of Algorithm2 is known as FTSNN without TS approach and the remaining steps are known as FTSNN with TS approach.

In FTSNN without TS approach, after defuzzifying the simulated data, stores the results of performance measure SMAPE, RMSE, POCID and TS of training set, validation set and test set for different FTSNN model. The optimum FTSNN model selection is based on minimum forecasting error in validation set of SMAPE.

After selecting the optimum model using FTSNN without TS approach, still, there exists over-forecast or under-forecast in training dataset, validation dataset and test dataset. For example, the level of over-forecast and under-forecast in training dataset and validation dataset of BSE100 stock market with fifteen test cases (trial) of FTSNN model with neuron 7 which is represented in Figure 3.

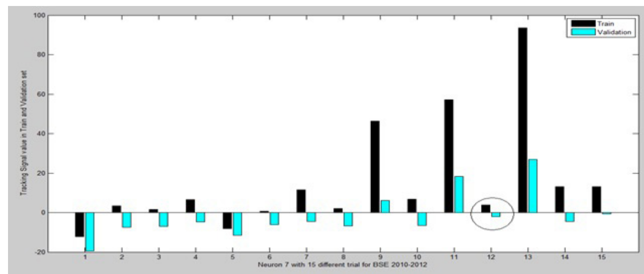


**Figure 3.** Tracking Signal (TS) value in Training set and validation set for BSE 2010-2012.

Test case 12 is identified as the optimum FTSNN model by the TS measure marked with the circle in the Figure 3, which contain TS interval value [-4, +4] in training and validation set. Remaining test cases are rejected which contain beyond the TS interval value [-4, +4] in training and validation set.

FTSNN with TS approach is used to assess the over-forecast or under-forecast in training dataset, validation dataset and test dataset. For every FTSNN model, check the TS interval $[-\theta, \quad +\theta]$ in the training dataset and validation dataset, where $\theta = 4$ and SD=3. It rejects all FTSNN model which results in values outside the interval of [-4, +4]; it accepts the FTSNN model which results in values inside the interval of [-4, +4]. If the TS interval value [-4, +4] does not exist, modify the value of SD. Finally, the optimum FTSNN model selection is based on the interval value [-4, +4] in the training dataset and validation dataset which contains minimum forecasting performance error in SMAPE (Instead of SMAPE any other performance measure can be used) of validation set.

## 4. Experimental Results

In this section, there are two main issues: first, to verify the effectiveness of the proposed FTSNN without TS approach and FTSNN with TS approach for closing stock index forecasting; second, to demonstrate the superiority of the proposed FTSNN without TS approach and FTSNN with TS approach by comparing it with existing time series forecasting methods. The results were carried out in MATLAB 8.1.0.604 (R2013a) - 32 Bit with INTEL i3 processor @ 2.20 GHz and 4 GB RAM.

### 4.1 BSE100 Index

The effectiveness of the proposed FTSNN with TS approach is tested on BSE100 index. The dataset consists of BSE100 closing stock index for the period from January 1, 2010 to December 31, 2012 from the BSE Website[28]. For each NN created with different random initial weight for neuron 1 to neuron 18. The choice of random initial weight (trial) size and maximum neuron size is selected by user. In this study, random initial weight size is 15 and maximum neuron size is 18 for BSE100 stock market index. The data division ratio is 50/25/25.

The results of performance measure of 18 different models from 9-1-9 to 9-18-9 were generated. Every FTSNN model contains fifteen different random initial weight generations. From the eighteen architectures of different trial, some models are selected by the FTSNN with TS approach which contain the interval [-4, +4] in the tracking signal of training dataset and validation dataset; and some models are rejected by the FTSNN with TS approach which does not contains the interval [-4, +4] in the training dataset and validation dataset of tracking signal. Rejection of model and selection of model using FTSNN with TS approach is represented in Table 1.

The performance measure of SMAPE, RMSE, POCID and TS of training set, validation set and test set using FTSNN with TS approach and FTSNN without TS approach for the BSE100 index in the year 2010 to 2012 with 50/25/25 data division ratio and the result of optimum model is reported in Table 2 and best forecasting result are highlighted by bold face.

**Table 1.** Model rejection and selection in FTSNN with TS approach

| Ratio | Model Rejection | Model Selection |
|---|---|---|
| 50/25/25 | 9 -1-9, 9-3-9, 9-4-9, 9-5-9, 9-6-9, 9-8-9, 9-9-9, 9-10-9, 9 -11-9, 9-12-9, 9-13-9, 9-14-9, 9 -16-9, 9-17-9 | 9-2-9, 9-7-9, 9-15-9, 9-18-9 |

From Table 2, the results of performance measure in train, validation and test set is reported in four aspects. (i), whether the forecasting error is high or low?; (ii) whether the NN is suffered due to over-fitting or under-fitting problem? (iii) Correctness of the predicted direction in the test set; (iv) and the effectiveness of the tracking signal.

First, the performance measure SMAPE and RMSE of test set in FTSNN with TS approach is 0.46 and 36.90; the performance measure SMAPE and RMSE of test set in FTSNN without TS approach is 0.47 and 37.40. It indicates that the forecasting error is minimum in the FTSNN with TS approach when compared to FTSNN without TS approach. In addition, it is observed that the forecasting error of SMAPE and RMSE in validation set is high in FTSNN with TS approach when compared to FTSNN without TS approach; FTSNN with TS approach produce lowest forecasting error in SMAPE and RMSE of the test set even it produce highest forecasting error value in validation set.

Second, the difference between the performance measure SMAPE and RMSE of training dataset and test dataset in FTSNN with TS approach is very close to each other when compared to the performance measure SMAPE and RMSE of training dataset and test dataset

**Table 2.** Performance measures of train, validation and test set for the year 2010 – 2012 of BSE 100 Index

| Measure | FTSNN With TS | | | FTSNN Without TS | | |
|---|---|---|---|---|---|---|
| | Train | Val | Test | Train | Val | Test |
| SMAPE | 0.60 | 0.96 | 0.46 | 0.61 | 0.82 | 0.47 |
| RMSE | 46.10 | 62.40 | 36.90 | 50.70 | 58.90 | 37.40 |
| TS | 3.81 | -2.04 | 15.10 | -18.50 | -10.40 | 18.90 |
| POCID | 92.20 | 95.20 | 92.00 | 85.00 | 82.40 | 81.80 |

**Table 3.** Performance measures of train, validation and test for the year 2000 of TAIEX using FTSNN with TS and FTSNN without TS approach compared with NNFTS[10] Model

| **Measure** | **FTSNNWith TS** | | | **FTSNN Without TS** | | | **NNFTS[10]** | | |
|---|---|---|---|---|---|---|---|---|---|
| | **Train** | **Val** | **Test** | **Train** | **Val** | **Test** | **Train** | **Val** | **Test** |
| SMAPE | 1.50 | 1.52 | 1.48 | 1.16 | 0.91 | 0.91 | - | - | - |
| RMSE | 155.00 | 147.00 | **149.00** | 145.00 | 117.00 | **121.00** | - | - | 149.60 |
| TS | -2.68 | 3.95 | 7.00 | -6.79 | -11.80 | 7.16 | - | - | - |
| POCID | 95.60 | 94.00 | 97.00 | 68.90 | 67.20 | 61.20 | - | - | - |

Note: "-" data not available

**Table 4.** Performance measures of train, validation and test for the year 2001of TAIEX using FTSNN with TS and FTSNN without TS approach compared with NNFTS[10] Model

| Measure | FTSNN With TS | | | FTSNN Without TS | | | NNFTS[10] | | |
|---|---|---|---|---|---|---|---|---|---|
| | Train | Val | Test | Train | Val | Test | Train | Val | Test |
| SMAPE | 1.34 | 1.63 | 1.64 | 1.21 | 0.86 | 1.65 | - | - | - |
| RMSE | 97.20 | 91.90 | **97.20** | 87.00 | 88.80 | **98.10** | - | - | 98.91 |
| TS | 8.60 | -9.31 | 24.50 | 11.60 | -17.00 | 31.90 | - | - | - |
| POCID | 86.00 | 85.00 | 83.30 | 95.10 | 85.10 | 75.00 | - | - | - |

in FTSNN without TS approach. This is the main purpose of tracking signal used in this study. This closeness of training and testing performance measure of SMAPE and RMSE indicates that the in-sample (training dataset) model selection criteria can be provide a reliable guide to out-of-sample (testing dataset) performance and can be an apparent connection between in-sample model fit and out-of-sample model forecasting performance. It happens due to the model selection is based on tracking signal.

Third, the performance measure POCID of test set in FTSNN with TS approach is 92; the performance measure POCID of test set in FTSNN without TS approach is 81.80. It indicates the correctness of the forecasting direction is very high in FTSNN with TS approach when compared to FTSNN without TS approach. Higher in POCID value indicates better forecasting model.

Fourth, the performance measure TS of train and test set is 3.81 and -2.04 in FTSNN with TS approach; the performance measure TS of train and test set tracking signal value is -18.50 and 10.40 in FTSNN without TS approach. It indicates the level of over-forecasting and the level of under-forecasting which are identified by tracking signal measure. The value of tracking signal in the test dataset of FTSNN with approach is very low when compared to the value of tracking signal in the test dataset of FTSNN without TS approach. The value of TS is within the interval [-4, +4] in the train and validation set indicates better forecasting model.

After the analysis of train, validation and test set of various models using FTSNN without TS approach is identified the neuron numbers in the hidden layer is 13 with the computational time of training is 1.9 seconds, the training process is completed in 18 epoch with affecting the over fitting problem; and FTSNN with TS approach is identified the neuron number in the hidden layer is 7 with the computational time of training is 1.5 seconds, the training process completed in 7 epoch without affecting the over fitting problem. It is observed that the neural network complexity is reduced; training time is reduced and faster convergence in FTSNN with TS approach when compared to FTSNN without TS approach.

## 4.2 TAIEX Index

To demonstrate the excellence of the proposed FTSNN with TS and FTSNN without TS approach, this study compares the out-of-sample RMSE's for different years in the NNFTS[10]. The data set consists of TAIEX closing stock index for the period from January 1, 2000 to December 31, 2004.

The proposed FTSNN with TS approach differs in two ways when compared to NNFTS[10] model. First, NNFTS[10] model used 12-24-12 (12 input node- 24 neuron in the hidden layer – 12 output node) architecture for the year 2000. The neuron number in the hidden layer is set to the sum of the number of input and output nodes, which are 24 as shown in Figure 4.

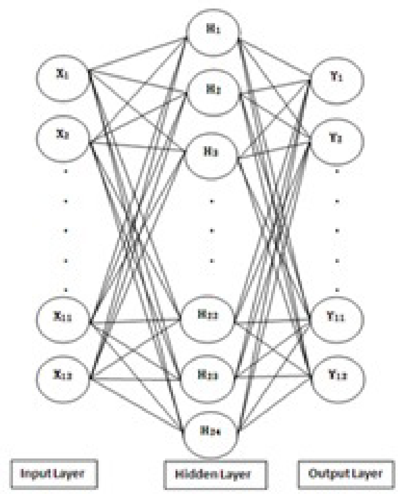


**Figure 4.** 12-24-12 Architecture proposed by Huarng and Yu for the Year 2000.

In the proposed approach, neuron number in the hidden layer is different for every year and it is identified by creating different NN (12 input nodes – neuron in the hidden layer starts from 1 to MAX_NEURON – 12 output nodes) architecture. FTSNN with TS approach reduces the complexity of NN architecture as shown in Figure 5 and Figure 6.

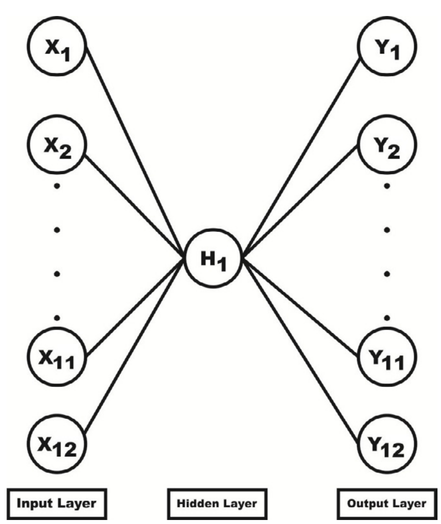


**Figure 5.** 12-1-12 Architecture proposed by FTSNN with TS approach for the Year 2004.

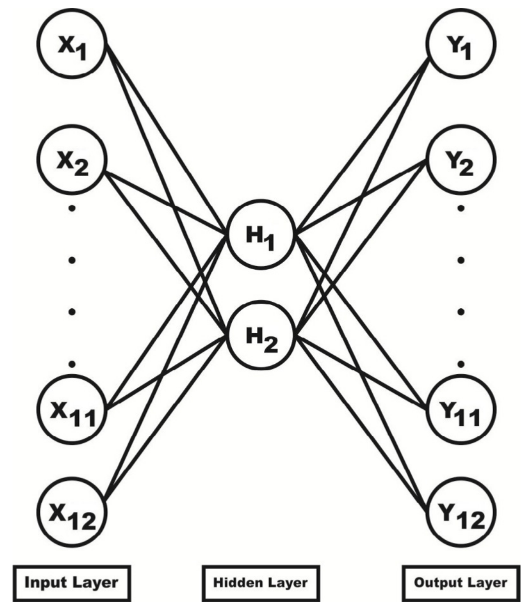


**Figure 6.** 12-2-12 Architecture proposed by FTSNN with TS approach for the year 2000 and 2003.

Second, the performance measure of RMSE of test data only reported in NNFTS[10] model. In the proposed approach, performance measure SMAPE, RMSE, TS and POCID of train, validation and test data analyzed and reported to analyze the close relationship between in-sample (training dataset) forecasting and out-of-sample (testing dataset) forecasting performance.

The performance measurement of SMAPE, RMSE, POCID and TS of training set, validation set and test set using FTSNN with TS approach and FTSNN without TS approach for the TAIEX in the year 2000 to 2004 are represented in Table 3 to Table 7. The results of train and validation set are not available in NNFTS[10], which is represented by "-"symbol in Table 3 to Table 7. Best forecasting results are represented by bold face.

From Table 3, in the year 2000, the value of test set of RMSE in FTSNN with TS and FTSNN without TS approach is 149 and 121. The value of test set of RMSE in NNFTS[10] model is 149.6. The FTSNN with TS and FTSNN without TS approach are outperformed than NNFTS[10] model with respect to RMSE.

The value of test set of SMAPE and RMSE in FTSNN with TS approach is high when compared to FTSNN without TS approach. It is also observed that the difference between training set and test set in FTSNN without TS approach is high when compared to FTSNN with TS approach. If a close relationship between model fit (train set) and out of sample forecasts (test set) does not exist,

**Table 5.** Performance measures of train, validation and test for the year 2002 of TAIEX using FTSNN with TS and FTSNN without TS approach compared with NNFTS[10] Model

| Measure | FTSNN With TS | | | FTSNN Without TS | | | NNFTS [10] | | |
|---|---|---|---|---|---|---|---|---|---|
| | Train | Val | Test | Train | Val | Test | Train | Val | Test |
| SMAPE | 1.13 | 1.21 | 1.11 | 1.11 | 1.09 | 1.07 | - | - | - |
| RMSE | 85.60 | 80.00 | **69.30** | 86.00 | 78.50 | **71.10** | - | - | 78.71 |
| TS | 14.30 | -10.30 | 18.90 | 42.00 | 9.04 | 39.00 | - | - | - |
| POCID | 91.90 | 91.80 | 93.40 | 87.90 | 91.80 | 92.40 | - | - | - |

**Table 6.** Performance measures of train, validation and test for the year 2003of TAIEX using FTSNN with TS and FTSNN without TS approach compared with NNFTS[10] Model

| Measure | FTSNN With TS | | | FTSNN Without TS | | | NNFTS[10] | | |
|---|---|---|---|---|---|---|---|---|---|
| | Train | Val | Test | Train | Val | Test | Train | Val | Test |
| SMAPE | 1.19 | 0.92 | 0.72 | 1.23 | 0.84 | 0.78 | - | - | - |
| RMSE | 72.60 | 63.90 | **56.20** | 75.20 | 60.50 | **58.60** | - | - | 58.78 |
| TS | -0.46 | -2.32 | -6.94 | 23.20 | 6.70 | 15.89 | - | - | - |
| POCID | 90.30 | 88.50 | 90.20 | 93.50 | 88.50 | 87.80 | - | - | - |

**Table 7.** Performance measures of train, validation and test for the year 2004 of TAIEX using FTSNN with TS and FTSNN without TS approach compared with NNFTS[10] Model

| Measure | FTSNNwith TS | | | FTSNNwithout TS | | | NNFTS[10] | | |
|---|---|---|---|---|---|---|---|---|---|
| | Train | Val | Test | Train | Val | Test | Train | Val | Test |
| SMAPE | 1.31 | 0.82 | 0.68 | 1.32 | 0.79 | 0.71 | - | - | - |
| RMSE | 109.00 | 60.20 | **55.60** | 111.00 | 60.50 | **55.80** | - | - | 55.91 |
| TS | -3.69 | -1.87 | 1.42 | 11.60 | 12.60 | 14.30 | - | - | - |
| POCID | 91.90 | 80.30 | 90.20 | 95.10 | 90.20 | 89.20 | - | - | - |

then it is hard to argue that selection of NN model should be based on minimum model fitting errors[27]. This study observed that the FTSNN with TS approach has close relationship between training set and test set. FTSNN with TS approach outperformed than FTSNN without TS approach with respect to POCID.

The results of performance measure SMAPE, RMSE, TS and POCID in train, validation and test set is reported in Table 3 to Table 7. Like BSE100 index, the results are interpreted in four aspects and it can be represented in Table 8. The best model is identified by lower prediction error in SMAPE and RMSE of test set; TS value should be within the interval in test set; and higher value in POCID of test set.

In the year 2000, 2003 and 2004, the TS interval [-4, +4] exist in the training dataset and validation dataset. In the year 2001 and 2002, TS interval [-4, +4] does not exist in the training dataset and validation dataset. In this situation, the value of SD can be changed. The value of SD is

**Table 8.** Performance measures using FTSNN with TS, FTSNN without TS approach and NNFTS[10]

| Approach | SMAPE | RMSE | TS | POCID |
|---|---|---|---|---|
| FTSNNwith TS | Low | Low | within the interval | High |
| FTNN without TS | High | High | Interval limit exceed | Low |
| NNFTS[10] | - | High | - | - |

Note: Lower value in SMAPE, RMSE and higher value in POCID represents best predictive model. "-" indicates data not available

3 for the year 2000, 2003 and 2004; the value of SD is 8 for the year 2001 and the value of SD is 12 for the year 2002.

After the analysis of train, validation and test set of every year, the performance measure of optimum model reported in table (3) to (7) and their corresponding neuron number in the hidden layer, training time and convergence speed using FTSNN with TS approach and FTSNN without TS approach is reported in Table 9 and best results are represented by boldface.

**Table 9.** Optimum model selection using FTSNN with TS approach and FTSNN without TS approach for the year 2000 to 2004.

| Year | With TS | | | Without TS | | |
|---|---|---|---|---|---|---|
| | Neuron | Time (sec.) | epoch | Neuron | Time (sec.) | epoch |
| 2000 | **2** | **1.71** | 5 | 16 | 2.77 | **2** |
| 2001 | **6** | **1.02** | **4** | 9 | 3.89 | 6 |
| 2002 | **6** | **1.26** | **5** | **6** | 1.31 | 7 |
| 2003 | **2** | **0.79** | **1** | 9 | 0.96 | 3 |
| 2004 | **1** | **0.92** | **1** | 5 | 1.17 | 4 |

**Table 10.** Proposed approach compared with NNFTS[10] model by RMSE measure

| Models | 2000 | 2001 | 2002 | 2003 | 2004 |
|---|---|---|---|---|---|
| NNFTS Model [10] | 149.60 | 98.91 | 78.71 | 58.78 | 55.91 |
| FTSNN with TS Approach | 149.00 | **97.20** | **69.30** | **56.20** | **55.60** |
| FTSNN without TS Approach | **121.00** | 98.10 | 71.10 | 58.60 | 55.80 |

In FTSNN with TS approach, the neuron number in the hidden layer is one for the year 2004; neuron number in the hidden layer is 2 for the year 2000 and 2003; and the neuron number in the hidden layer is 6 for the year 2001 and 2005. Similarly, neuron number in the hidden layer using FTSNN without TS approach is represented in Table 9. It is observed that the neural network complexity is reduced; training time is reduced and fast convergence in FTSNN with TS approach when compared to FTSNN without TS approach in every year.

Table 10 shows the performance measure RMSE of every year in the proposed approach of this study is compared with the NNFTS model[10].Best results are represented by boldface. In the year 2000 to 2004, The RMSE of the FTSNN with TS approach and FTSNN without TS approach is much smaller than their corresponding values in NNFTS[10] model. It indicates that FTSNN with TS approach is outperformed than NNFTS[10]. It is also noted that the RMSE of the FTSNN with TS approach is smaller than FTSNN without TS approach in the year 2001 to 2004.

## 4.3 NN3 Forecasting Competition Time Series Data

The excellence of the proposed FTSNN with TS approach is compared with NN3 forecasting competition time series dataset. The dataset A contains 111 monthly business time series and dataset B contains 11 monthly business time series. In every time series of NN3 dataset, last 18 points are reserved for test dataset. Remaining data points are divided into two parts, 74% of the total data points used for training and 13% of the total data points are used for validation. The experimental parameter is same as mentioned in BSE100 stock market data.

**Table 11.** Rank on SMAPE of FTSNN with TS and FTSNN without TS approach compared with NN3 111 time series data

| Participant | SMAPE |
|---|---|
| FTSNN without TS approach | 12.80 |
| FTSNN with TS approach | 14.60 |
| Stat. Contender – Wildi | 14.84 |
| Stat. Benchmark - Theta Method (Nikolopoulos) | 14.89 |
| Illies, Jäger, Kosuchinas, Rincon, Sakenas, Vaskevcius | 15.18 |
| Stat. Benchmark - ForecastPro (Stellwagen) | 15.44 |
| CI Benchmark - Theta AI (Nikolopoulos) | 15.66 |
| Stat. Benchmark - Autobox (Reilly) | 15.95 |
| Adeodato, Vasconcelos, Arnaud, Chunha, Monteiro | 16.17 |
| Flores, Anaya, Ramirez, Morales | 16.31 |
| Chen, Yao | 16.55 |
| D'yakonov | 16.57 |

The performance measure SMAPE of this study is also compared with the NN3 forecasting result[29]. From the NN3 forecasting result, this study select top five statistical methods and computational intelligence based methods for the purpose of comparison. The average SMAPE of this study and various methods in forecasting competition using 111 time series (NN3 dataset A) and 11 time series (NN3 dataset B) are shown in Table 11 and Table 12 is arranged by least error in SMAPE. The best results are highlighted in boldface. This study observed that this FTSNN with TS approach and FTSNN without TS approach outperformed the various models tested with the NN3 forecasting competition.

**Table 12.** Rank on SMAPE of FTSNN with TS and FTSNN without TS approach compared with NN3 11 time series data

| Participant | SMAPE |
|---|---|
| FTSNN with TS approach | 12.00 |
| FTSNN without TS approach | 12.30 |
| CI Benchmark - Theta AI (Nikolopoulos) | 13.07 |
| Stat. Benchmark - Autobox (Reily) | 13.49 |
| Stat. Benchmark - ForecastPro (Stellwagen) | 13.52 |
| Yan | 13.68 |
| Stat. Benchmark - Theta (Nikolopoulos) | 13.70 |
| llies, Jäger, Kosuchinas, Rincon, Sakenas, Vaskevcius | 14.26 |
| Chen, Yao | 14.46 |
| Yousefi, Miromeni, Lucas | 14.49 |
| Ahmed, Atiya, Gayar, El-Shishiny | 14.52 |
| Flores, Anaya, Ramirez, Morales | 15.00 |

## 5. Conclusion

This study proposed a novel Fuzzy Time Series Model using Neural Network with Tracking Signal (FTSNN with TS) approach. It is proposed to forecast one-step-ahead closing index of stock market and it is applied to two real time series data set namely BSE100 and TAIEX. It has analyzed the neuron number in the hidden layer, training time, convergence speed (epoch) and performance measure of SMAPE, RMSE, POCID and TS in the training dataset, validation dataset and test dataset. After the analysis of various fuzzy time series models using neural network, finally FTSNN without TS approach and FTSNN with TS approach identified the neuron numbers in the hidden layer for improving prediction accuracy and reduce over-fitting problem. This study recommends to increase the prediction accuracy, the best forecasting model is selected by the presence of tracking signal interval [-4, +4] in training set and validation set; and minimum error value in SMAPE of validation set.

The in-sample and the out-of-sample forecasting performance analyzed; and the results indicate that the in-sample model selection can be provide a reliable guide to out-of-sample performance and can be an apparent connection between in-sample model and out-of-sample model forecasting performance by using FTSNN with TS approach. The experimental result with BSE and TAIEX market of real datasets indicate that the proposed FTSNN with TS approach be an effective way in-order-to yield accurate prediction result. FTSNN with TS approach is perfectly fitted on stock market data range from small dataset to large dataset. In addition, the proposed approach has been tested on standard NN3 forecasting competition time series dataset and this approach outperforms the various models tested with the NN3 forecasting competition. This study is also found that the tracking signal is the best performance measure for time series data and it identifies the level of over-forecasting and under-forecasting in NN.

The proposed FTSNN with TS approach can be used as an alternative forecasting tool for time series forecasting. In this study, only single variable is taken for prediction; In future, multi variables will be taken for prediction to improve the accuracy of stock market; It will be applied to identify hidden neurons in the multiple hidden layer; and also it will be applied to different types of NN model for forecasting closing stock index/price of stock market data.

## 6. References

1. Wong WK, Min Xia, Chu WC. Adaptive neural network model for time-series forecasting..European Journal of Operational Research. 2010 Jun; 207(2): 807–16. Crossref
2. Kumar K, Bhattacharya S. Artificial neural network vs. linear discriminant analysis in credit ratings forecast: A comparative study of prediction performances. Review of Accounting and Finance. 2006; 5(3): 216–27.
3. Kanas A. Neural network linear forecasts for stock returns. International Journal of Finance and Economics. 2001 July; 6(3): 245–54. Crossref
4. Gronholdt L, Martensen A. Analysing customer satisfaction data: A comparison of regression and artificial neural networks. International Journal of Market Research. 2005 Feb; 47(2): 121–30.
5. Refenes AN, Azema-Barac M, Zapranis A D. Stock ranking: Neural networks vs. multiple linear regression. Proceedings

In IEEE international conference on neural networks. 1993 Mar; 1419–26. Crossref

6. Lean Y, Wang S, Lai KK. Forecasting foreign exchange rates using an SVR-based neural network ensemble. Advances in Banking Technology and Management: Impacts of ICT and CRM: Impacts of ICT and CRM. 2007; 261–77.
7. Palmer A, Monta-o JJ, Sesé A. Designing an artificial neural network for forecasting tourism time series. Tourism Management. 2006 Oct; 27(5): 781–90. Crossref
8. Ricardo de Araujo A, Tiago AE, Ferreira. An intelligent hybrid morphological-rank-linear method for financial time series prediction. Neurocomputing. 2009 Jun; 72(10-12):2507–24. Crossref
9. Khashei M, Bijari M. A novel hybridization of artificial neural networks and ARIMA models for time series forecasting. Applied Soft Computing. 2011 Mar;11(2): 2264–75. Crossref
10. Yu THK, Huarng KH. A neural network-based fuzzy time series model to improve forecasting. Expert Systems with Applications. 2010Apr; 37(4):3366–72. Crossref
11. Qi M, Zhang GP. An investigation of model selection criteria for neural network time series forecasting. European Journal of Operational Research. 2001August; 132(3):666–80. Crossref
12. Timothy Masters, Practical neural network recipes in C++. Morgan Kaufmann; 1993.
13. Heaton J. Introduction to Neural Networks for Java. 2nd ed. Heaton Research, Inc. 2008; 159–60.
14. Balestrassi PP, Popova E, Paiva AP, Maran gon Lima JW. Design of experiments on neural network training for nonlinear time series forecasting. Neuro computing. 2009 Jan; 72(4-6):1160–78. Crossref
15. Vahedi A. The Predicting Stock Price using Artificial Neural Network. Journal of Basic and Applied Scientific Research. 2012 Mar; 2(3): 2325–8.
16. IebelingKaastra A, Milton Boyd B. Designing a neural network for forecasting financial and economic time series. Neurocomputing.1996 Apr; 10(3): 215–36. Crossref
17. Ashok Kumar D, Murugan S. Performance Analysis of Indian Stock Market Index using Neural Network Time series Models. IEEE Explore Digital Library. 2013 Feb; 72–8.
18. Ashok Kumar D, Murugan S. Performance Analysis of MLPFF Neural Network Back propagation Training Algorithms for Time Series Data. IEEE Explore Digital Library. 2014 Mar; 114–9.
19. Problem about getting optimum output in Neural Network MATLAB 2012a. 2016 November 08. Available from: http://in.mathworks.com/matlab central/ newsreader/view thread/331714.
20. AdebiyiAyodele A, Ayo Charles K, Adebiyi Marion O, Otokiti Sunday O. Stock Price Prediction using Neural Network with Hybridized Market Indicators. Journal of Emerging Trends in Computing and Information Sciences. 2012 Jan; 3(1):1–9.
21. Guresen E, Kayakutlu G, Tugrul, Daim U. Using artificial neural network models in stock market index prediction. Expert Systems with Applications. 2011 Aug; 38(8):10389–97. Crossref
22. Yu L, Wang S, Lai KK. Adaptive Smoothing Neural Networks in Foreign Exchange Rate Forecasting. Proceeding Computational Science ICCS 2005 5th International. 2005 May; 523–30. Crossref
23. Rahman M, Islam M, Yao X. Layered Ensemble Architecture for Time Series Forecasting. IEEE Transaction cybernetics. 2015 Jan; 46(1): 270–83. Crossref PMid:25751882
24. Suresh Kumar KK. Performance analysis of stock price prediction using artificial neural network. Global Journal of Computer Science and Technology. 2012 Jan; 12(1): 154–70.
25. Measuring Forecast Accuracy Approaches to Forecasting A Tutorial. 2016 November 08. Available from: http://scm.ncsu.edu/scm-articles/article/measuring-forecast-accuracy.
26. Sivanandam SN, Deepa SN. Principles of soft computing. 1st ed. Wiley India (P) Ltd; 2008;79–80.
27. Fildes R, Makridakis S. The impact of empirical accuracy studies on time series analysis and forecasting international statistical review. 1995 Dec; 63(3): 289–308.
28. Bombay Stock Exchange India. 2016 November 08. Available from: http://www.bseindia.com/indices/ Index Archive Data.aspx.
29. Cronel SF, Nikolopoulos K, Hibon M. Automatic Modelling and Forecasting with Artificial Neural Networks a forecasting competition evaluation. IIF/SAS Grant 2005/6 Research Report. 2008 Apr; 81(3):1–52.