Juni Khyat
(UGC Listed Journal)
ISSN: 2278-4632
Vol-9 Issue-1 No.01 January 2019

Using Novel Multi-Layer Feed Forward Neural Network Analysis of Performance in Indian Bombay Stock Exchange Market based Fuzzy Time Series Model with Tracking Signal Approach

Prakash Chandra Dash[1], Prakash Chandra Jena[2], Bhagyadhar Mallick[3]

[1]Asst. Prof. Einstein Academy of Technology and Management, Bhubaneswar, India
[2]Asst. Prof. Einstein Academy of Technology and Management, Bhubaneswar, India
[3]Student, Einstein Academy of Technology and Management, Bhubaneswar, India

Abstract
This study, presents a novel multi-layer feed forward neural network based fuzzy time series model with tracking signal approach (MLFFNN-FTS with TS) for forecasting the closing index of the stock market. A novel approach strives to adjust the number of hidden neurons of a Multi-Layer Feed Forward Neural Network based Fuzzy Time Series (MLFFNN-FTS) model. It uses the Tracking Signal (TS) and rejects all models which result in values outside the interval of [-4, 4]. The effectiveness of the proposed approach is verified with one step ahead of Bombay Stock Exchange (BSE100) closing stock index of Indian stock market. This novel approach reduces the over-fitting problem, reduces the neural network structure and improves forecasting accuracy.

***Keywords**:* Neural Network, Fuzzy Time Series Data, Tracking Signal, Stock Index, BSE, Prediction.

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 may be 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 be 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 include time- series regression, Auto Regressive Integrated Moving Average (ARIMA) and exponential smoothing which 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 statistical model and hybrid models have been proposed during the last few years for obtaining accurate forecasting results in various applications. These were attempts to improve upon and better the conventional linear and nonlinear approaches. NNs are non-linear in nature and where most of the natural real world systems are non-linear in nature, so, NN are preferred over the traditional models [2]. The reason is that the NN is a global function approximation which can map any linear or non-linear functions. Although NNs have the advantages of accurate forecasting, Mehdi Khashei and Mehdi Bijari [3], Min Qi and Guoqiang Peter Zhang [4] reported that (i) there is no systematic rule to identify neuron counts in the hidden layer (ii) Neural Network suffers due to over-fitting and under- fitting problem. Timothy Master [5] proposed a geometric pyramid rule to solve the problem of neuron counts in the hidden layer issue with a three layer NN with m output and n input neurons, the hidden layer may have square root (m * n) neurons. Jeff Heaton [6] found that, a NN with 2N + 1 hidden neuron and one hidden layer is sufficient for N inputs, and observed that the optimum number of hidden layers and hidden neurons are highly problem dependent. As the accuracy of NN model depends on the careful NN model design, a detailed NN designing methodology and training process is reported in the literature ([7], [8], [9]). The performance of multi- layer feed forward neural network with various types of training algorithms ([10], [11]) found that 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 [12] suggests that design of ten neural networks with different types of random initial weights to mitigate the occasional bad random start. Adebiyi Ayodele et al. [13] suggest that training a great number of ANN with different configurations and selecting the optimum model will improve forecasting accuracy.

The data set in many applications is divided into two sets: training and testing set as observed by Mehdi Khashei and Mehdi Bijari [3], Tiffany Hui-Kuang Yu and Kun-Huang Huarng[2], Erkam Guresen et al. [14]. 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 [15]. The published research articles ([16], [17]) 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-fitting problem.

To solve the above mentioned problem, this paper recommends an iterative procedure to select optimum neural network based fuzzy time series model with tracking signal approach. The proposed approach systematically constructs different MLFFNN- FTS model from simple architecture to complex architecture; and the optimum MLFFNN-FTS model selection is based on the 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 based fuzzy time series model which reduces over-fitting or under-fitting problem.

Many research articles presented in the literature are related to selecting the optimal number neurons in a hidden layer of a neural network. These articles reported that the selection of optimal number neurons in a hidden layer is identified by sum of input and output variable for particular training function. The present study has developed 18 different neural network based fuzzy time series models with 15 different weights. The neural network based fuzzy time series model selection is normally based on trial and error method. The proposed approach has endeavored to select optimum neural network based fuzzy time series model by adjusting two important parameters, namely number of neuron in the hidden layer and different weight used in the neural network.

This study claimed that, after selecting the optimum neural network based fuzzy time series 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 above mentioned problem, this paper recommends a novel multi-layer feed forward neural network based fuzzy time series model with Tracking Signal (MLFFNN-FTS with TS) approach. TS is used to identify the presence of over- forecast or under-forecast in the NN model. The proposed MLFFNN-FTS with TS approach systematically constructs different MLFFNN-FTS model from simple architecture to complex architecture; and the optimum MLFFNN-FTS 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 [18] Cecil Bozarathreported that, the TS is 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 [15] Lean Yu 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 is used to analyze and select the best NN model after the NN training to improve forecasting accuracy.

The contribution of this study is seven folded: firstly, different MLFFNN-FTS model created for forecasting the closing stock index of the BSE100 market. Secondly, the performance measure Tracking Signal (TS) is introduced to select the optimum MLFFNN- FTS model which reduces the network complexity; faster in convergence; improves better forecast accuracy; and avoids over- forecast and under-forecast. Thirdly, 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 MLFFNN-FTS with TS approach. Fourthly, the neuron numbers in the hidden layer is identified for BSE100 stock market. Fifthly, the performance of the proposed approach is compared with MLFFNN-FTS without TS approach and it outperformed. Sixthly, the performance of the proposed approach is compared with MLFFNN with TS approach and it outperformed. Seventhly, unlike the report of Min Qi [4] 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 MLFFNN-FTS with TS approach; section 4 reports the experimental results attained by the MLFFNN-FTS without TS approach and MLFFNN with TS approach using real world financial time series such as BSE100. Finally this study is concluded in section 5.

2. Multi Layer Feed Forward Neural Network Model

MLFFNN consists of an input layer, one or more hidden layers and an output layer. The hidden layer receives weight from input layer. Each subsequent layer receives 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 one as shown in Figure 1.

ISSN: 2278-4632
Vol-9 Issue-1 No.01 January 2019

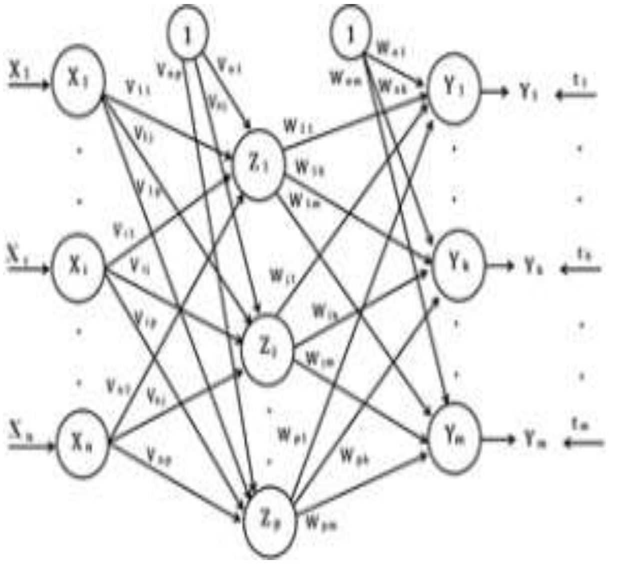

Figure 1. Multi Layer Feed Forward Neural Network

The bias term also acts as weights and it shows the architecture of Back Propagation Neural Network, depicting 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 [19].

Tracking Signal

In [18] Cecil reported that the Tracking Signal is calculated as the ratio of cumulative forecast error divided by the mean absolute deviation (MAD). It can be represented in the equation (3). 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 MAD and the Standard deviation in a normally distributed population is established as 1.25 MAD = 1 SD (standard deviation of the distribution).

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 [2].

$$U = \{u_1, u_2, \ldots . u_n\}.$$

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

$$f_A : U \rightarrow [0, 1]. f_A(u_i)$$

Fuzzy time series data are structured by fuzzy sets. Let U be the universe of discourse, such that Let us defined a fuzzy set A of U by where is the membership function of A, and $1 \leq i \leq n$. Tiffany Hui-Kuang Yu and Kun-Huang Huarng10 f_A
proposed a sequence of steps to design NNFTS model represented in Algorith

Forecasting performance measure

The forecasting performance is evaluated using the statistical measures, namely, symmetric mean absolute percentage error (SMAPE), percentage of change in direction (POCID) and Tracking Signal (TS). In each of the following measure yt is the actual value, ft is the forecasted value. et=yt- ft is the forecast error and n is the size of the test set. The global performance of a forecasting model is evaluated by the SMAPE [17] which is used in NN3, NN5 and NNGC1 forecasting competition. A smaller SMAPE value suggests the better forecasting accuracy. It can be expressed as

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

POCID (Percentage of Change in Direction) [16] maps the accuracy in the forecasting of the future direction of the time series. A larger POCID value suggests the better forecasting accuracy. It tends to 100 % is a perfect modeling. It can be represented as

$$POCID = 100\frac{\sum_{t=1}^{n} D_t}{n} \qquad (2)$$

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

In [18] Cecil reported that the Tracking Signal (TS) is used to pinpoint forecasting models that need adjustment. As long as the TS is between -4 and +4, assume the model is working correctly. The TS is a simple indicator that forecast bias is present in the forecast model. TS is used to verify the validity of the forecasting model. It can be represented as,

$$TS = \frac{\sum_{t=1}^{n} e_t}{MAD} \qquad (3)$$

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

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

3. Proposed Methodology

The proposed MLFFNN-FTS with TS approach is modified version of Neural Network Based Fuzzy Time Series Model (NNFTS) [2] uses BSE100 dataset and it differs in two ways when compared to [2]. First, NNFTS [2] model used 12-24-12 (12 input node - 24 neuron in 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. In the proposed approach, neuron number in the hidden layer is different for every year and it is identified by creating different NN (9 input nodes – neuron in the hidden layer starts from 1 to MAX_NEURON – 9 output nodes) architecture. MLFFNN-FTS with TS approach reduces the complexity of NN architecture. Second, the performance measure of RMSE of test data only reported in NNFTS [2] 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. Over fitting is one of the main issues in neural network modeling. In order to reduce over fitting problem, this study proposed a novel approach MLFFNN-FTS with TS which is used to forecast the closing index of the stock market. MLFFNN receives fuzzified data and trains different network by using different random initial weight with different neurons. TS measure is used to reject all MLFFNN-FTS model which results in values outside the interval of [- 4, +4] in training set and validation set of different neural networks.

In neural network modeling, training parameter and the weight play an important role to increase the forecasting accuracy. The proposed MLFFNN-FTS 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.

In this study, forecasting strategies are taken a step ahead of prediction. Let y1, y2, y3 ….. yt be a time series. As time t for t>=1, the next value yt+1 is predicted based on the observed realizations of yt, yt-1, yt-2 ….. y1. The resulting NN can be used for multi-step prediction by feeding the prediction back to the input of NN recursively. The MLFFNN-FTS with TS approach is represented in Figure 2.

ISSN: 2278-4632
Vol-9 Issue-1 No.01 January 2019

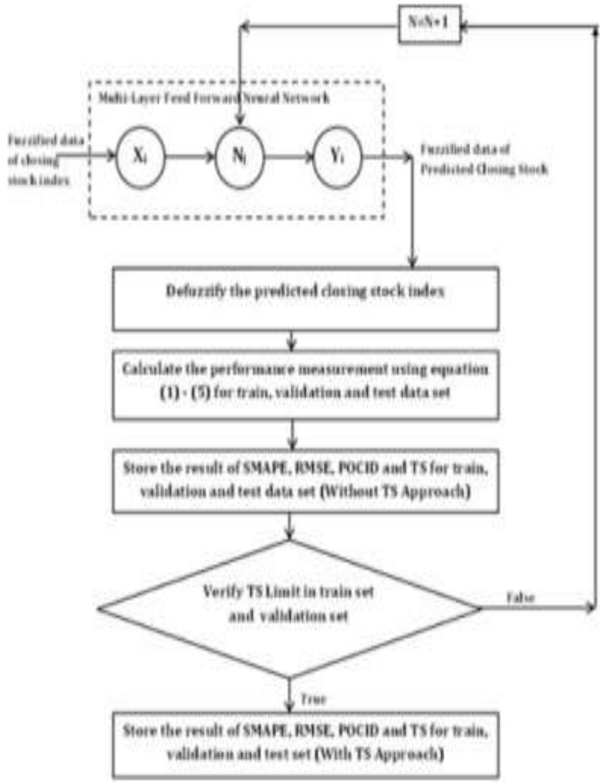

Figure 2. Fuzzy Time Series model using Neural Network with Tracking Signal Approach

In Figure 2, Xi is the fuzzified data of closing stock index vector, Yi is the fuzzified data of predicted closing stock index from neural network model and Nj 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 MLFFNN-FTS 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 (MLFFNN-FTS 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.

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.

1. **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}.

$$D_{min} = \min\{d'(t-1,t)\}, \forall t.$$
$$D_{max} = \max\{d'(t-1,t)\}, \forall t.$$

2. **Universe of discourse:** The Universe of discourse U is defined as [D -D1 D +D], where D and D are two proper positive numbers. The length of the interval is fix to l, then divide U into equal intervals and let it be u1,u2,u3, . . .

ISSN: 2278-4632
Vol-9 Issue-1 No.01 January 2019

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

$$u_2 = [D_{min} - D_1 + l, \quad D_{min} - D_1 + 2l],$$
$$u_3 = [D_{min} - D_1 + 2l, \quad D_{min} - D_1 + 3l],$$
$$\ldots\ldots$$
$$u_k = [D_{min} - D_1 + (k-1)l, \quad D_{min} - D_1 + kl],$$

Their corresponding midpoints are

$$m' = \frac{D_{min} - D_1 + D_{min} - D_1 + l}{2} = D_{min} - D_1 + \frac{l}{2}$$
$$m^2 = \frac{D_{min} - D_1 + l + D_{min} - D_1 + 2l}{2} = D_{min} - D_1 + \frac{3l}{2}$$
$$\ldots\ldots$$
$$m^k = D_{min} - D_1 + \frac{2 x (k-1) x l}{2}$$

Define the linguistic values of the fuzzy sets. Suppose A1, A2, A3, … are linguistic values. Label all the fuzzy sets by all possible linguistic values u1, u2, u3, …

3. **Fuzzification:** d' (t-1,t) can be fuzzified into a set of degrees of membership, V(t-1,t), where $V(t-1,t) = \mu^1_{t-1,t}, \mu^2_{t-1,t}, \cdots$

4. **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.

FOR NEURON = 1 TO MAX_NEURON

FOR TRIAL = 1 TO MAX_TRIAL

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.

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.

Train the NN using train function.

5. **Neural Network Forecasting:** With V(t-1,t) we 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.

6. **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 $\mu^k_{t-1,t}$ forecasted degrees of membership and m^k represents the corresponding midpoints of the interval $\mu^k_{t-1,t}$

7. **Forecasting:** After obtain the forecasted difference between t- 1 and t, find the forecast for t:

$$fd'(t-1,t) = fd(t-1,t) - const,$$
$$forecast(t) = fd'(t-1,t) + obs_{t-1}$$

8. **Performance Evaluation:** Calculate the performance measure SMAPE, POCID, RMSE and TS for train, validation and test set using equation (1) - (5).

9. 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 MLFFNN-FTS without TS approach.

10. Verify the interval [-θ,+θ] of Tracking Signal in training set (TStrain) and validation set (TSvalidation) from step 10, where θ = round (SD * 1.25). If (TStrain ≥-θ&&TStrain ≤ +θ) and (TSvalidation ≥-θ&& TSvalidation ≤ +θ) then go to step 12. Otherwise, go to step 5.2.

11. 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 MLFFNN-FTS with TS approach.

12. END// MAX_TRIAL

13. END // MAX_NEURON

14. From the step 10, select the optimum fuzzy time series neural network model, which provides less error in SMAPE for MLFFNN- FTS without TS approach.

15. From the step 12, select the optimum fuzzy time series neural network model, which provides less error in SMAPE for MLFFNN- FTS 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, MLFFNN-FTS model with tan sigmoidal function in the hidden layer and linear function in the output layer is used.

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 MLFFNN-FTS training process is represented in step 1 to step 10 of Algorithm1 is known as MLFFNN-FTS without TS approach and the remaining steps are known as MLFFNN-FTS with TS approach.

In MLFFNN-FTS 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 MLFFNN-FTS model. The optimum MLFFNN-FTS model selection is based on minimum forecasting error in validation set of SMAPE. After selecting the optimum model using MLFFNN-FTS 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 MLFFNN-FTS model with neuron 7 which is represented in Figure 3.

Test case 12 is identified as the optimum MLFFNN-FTS 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.

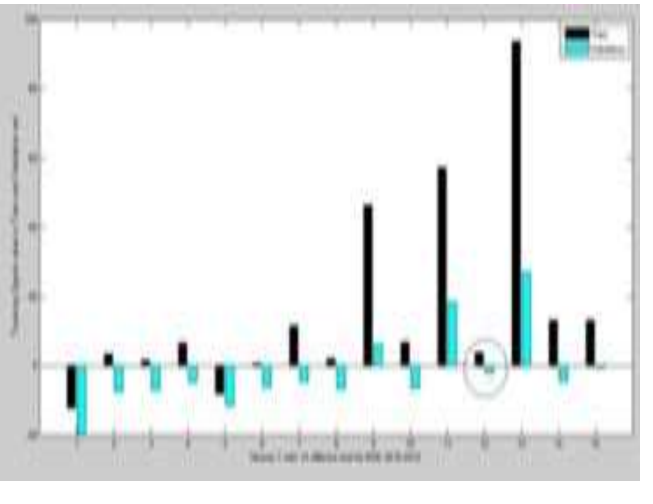

Figure3. Tracking Signal (TS) value in Training set andvalidation set for BSE 2010-2012.

MLFFNN-FTS with TS approach is used to assess the over-forecast or under-forecast in training dataset, validation dataset and test dataset. For every MLFFNN-FTS model, check the TS interval [- θ,+θ] in the training dataset and validation dataset, where θ=4 and SD=3. It rejects all MLFFNN-FTS model which results in values outside the interval of [-4, +4]; it accepts the MLFFNN-FTS 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 MLFFNN-FTS 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 Result

In this section, the effectiveness of the proposed MLFFNN-FTS without TS approach and MLFFNN-FTS with TS approach is verified for closing stock index forecasting. 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 MLFFNN-FTS 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 [20]. 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 MLFFNN-FTS model contains fifteen different random initial weight generations. From the eighteen architectures of different trial, some models are selected by the MLFFNN-FTS 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 MLFFNN- FTS 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 MLFFNN-FTS with TS approach is represented in Table 1.

Table 1. Model rejection and selection in MLFFNN-FTS withTS 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

The performance measure of SMAPE, RMSE, POCID and TS of training set, validation set and test set using MLFFNN-FTS with TS approach and MLFFNN-FTS 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 2. Performance measures of train, validation and test set for the year 2010 – 2012 of BSE 100 Index

Measure	MLFFNN-FTS With TS			MLFFNN-FTS 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

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 MLFFNN-FTS with TS approach is 0.46 and 36.90; the performance measure SMAPE and RMSE of test set in MLFFNN-FTS without TS approach is 0.47 and 37.40. It indicates that the forecasting error is minimum in the MLFFNN-FTS with TS approach when compared to MLFFNN-FTS without TS approach. In addition, it is observed that the forecasting error of SMAPE and RMSE in validation set is high in MLFFNN-FTS with TS approach when compared to MLFFNN-FTS without TS approach; MLFFNN-FTS 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 MLFFNN-FTS with TS approach is very close to each other when compared to the performance measure SMAPE and RMSE of training dataset and test dataset in MLFFNN-FTS 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 MLFFNN-FTS with TS approach is 92; the performance measure POCID of test set in MLFFNN-FTS without TS approach is 81.80. It indicates the correctness of the forecasting direction is very high in MLFFNN-FTS with TS approach when compared to MLFFNN-FTS 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 MLFFNN-FTS with TS approach; the performance measure TS of train and test set tracking signal value is -18.50 and 10.40 in MLFFNN-FTS 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 MLFFNN-FTS with approach is very low when compared to the value of tracking signal in the test dataset of MLFFNN-FTS 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 MLFFNN-FTS 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 MLFFNN-FTS 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 MLFFNN-FTS with TS approach when compared to MLFFNN-FTS without TS approach.

4.2 Comparison of MLFFNN with TS and MLFFNN-FTS withTS Approach

In this section, the effectiveness of the proposed MLFFNN-FTS with TS approach is compared with the MLFFNN with TS approach. MLFFNN with TS approach is similar to MLFFNN-FTS with TS approach and the main difference is MLFFNN with TS approach receives raw data between 0 and 1 instead of fuzzified data.

The performance measure of SMAPE, RMSE, POCID and TS of training set, validation set and test set using MLFFNN-FTS with TS approach and MLFFNN with 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 3 and best forecasting result are highlighted by bold face.

Table 3. Performance measures of train, validation and testset for the year 2010 – 2012 of BSE 100 Index

Measure	MLFFNN-FTS With TS			MLFFNN With TS		
	Train	Val	Test	Train	Val	Test
SMAPE	0.60	0.96	0.46	0.87	0.76	0.84
RMSE	46.10	62.40	36.90	60.40	51.80	57.30
TS	3.81	-2.04	15.10	0.00	0.17	0.04
POCID	92.20	95.20	92.00	77.60	73.30	76.50

The testset of RMSE and SMAPE is very low in MLFFNN-FTS with TS when compared to MLFFNN with TS approach. The test set of POCID is very high in MLFFNN-FTS with TS approach when compared to MLFFNN with

TS approach. From Table 3, it is observed that the MLFFNN-FTS with TS approach outperformed MLFFNN with TS approach with respect to SMAPE, RMSE and POCID. Minimum value in RMSE, SMAPE and maximum value in POCID represents best forecasting model.

5. Conclusion

This study proposed a novel Multi-Layer Feed Forward Neural Network based Fuzzy Time Series Model with Tracking Signal (MLFFNN-FTS with TS) approach. It is proposed to forecast one- step-ahead closing index of stock market and it is applied to real time series data set namely BSE100. 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 neural network based fuzzy time series models, finally MLFFNN-FTS without TS approach and MLFFNN- FTS 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 MLFFNN-FTS with TS approach. The experimental results with BSE market of real dataset indicate that the proposed MLFFNN-FTS with TS approach can be an effective way in-order-to yield accurate prediction result. 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. In addition, the proposed MLFFNN-FTS with TS approach is compared with MLFFNN with TS approach and it outperformed. The proposed MLFFNN-FTS 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.

References

[1]. W.K.Wong, Min Xia, and W.C. Chu, "Artificial neural network vs. linear discriminant analysis in credit ratings forecast: A comparative study of prediction performances", Review of Accounting & Finance, Vol. 5.3, pp.216-227, 2010.

[2]. Tiffany Hui-Kuang Yu and Kun-Huang Huarng, "A neural network-based fuzzy time series model to improve forecasting", Expert Systems with Applications, Vol. 37, pp.3366-3372, 2010.

[3]. Mehdi Khashei and Mehdi Bijari, "A novel hybridization of artificial neural networks and ARIMA models for time series forecasting", Applied Soft Computing, Vol. 11, pp. 2264-2275, 2011.

[4]. Min Qi and Guoqiang Peter Zhang, "An investigation of model selection criteria for neural network time series forecasting", European Journal of Operational Research, Vol. 3, pp. 666-680, 2001.

[5] Timothy Master (1993) 'Practical neural network recipes in C++', Morgan Kaufmann.

[6]. Jeff Heaton , "Introduction to Neural Networks for Java", Heaton Research, Inc, Second Edition, pp. 159-160, 2008.

[7]. P.P. Balestrassi, E. Popova, A. P. Paiva, and J. W. Marangon Lima, "'Design of experiments on neural network training for nonlinear time series forecasting", Neurocomputing, Vol. 72, pp. 1160-1178, 2009.

[8]. Abbas Vahedi, "The Predicting Stock Price using Artificial Neural Network", Journal of Basic and Applied Scientific Research, Vol. 2, pp. 2325-2328, 2012.

[9]. A. Iebeling Kaastra, and B. Milton Boyd, "Designing a neural network for forecasting financial and economic time series", Neurocomputing, pp. 215-236, 1996.

[10]. D. Ashok Kumar, and S. Murugan, "Performance Analysis of Indian Stock Market Index using Neural Network Time series Models", IEEE Explore Digital Library, pp. 72-78, 2013.

[11]. D. Ashok Kumar, and S. Murugan, "'Performance Analysis of MLPFF Neural Network Back propagation Training Algorithms for Time Series Data", IEEE Explore Digital Library, pp. 114-119, 2014.

[12]. Greg Heath, "Problem about getting optimum output in Neural Network MATLAB 2012a", http://in.mathworks.com/matlab central/ newsreader/view thread/331714.

[13]. A. Adebiyi Ayodele, K. Ayo Charles, O. Adebiyi Marion and O. Otokiti Sunday, "Stock Price Prediction using Neural Network with Hybridized Market Indicators", Journal of Emerging Trends in Computing and Information Sciences, Vol. 3, pp. 1- 9, 2012.

[14]. Erkam Guresen, Gulgun Kayakutlu, Tugrul, and U. Daim, " Using artificial neural network models in stock market index prediction", Expert Systems with Applications, Vol. 38, pp.10389-10397, 2012.

[15]. Lean Yu, Shouyang Wang and Kin Keung Lai, "Adaptive Smoothing Neural Networks in Foreign Exchange Rate Forecasting", Computational Science ICCS 2005 5th International, Vol. 3516, pp. 523-530, 2005.

[16]. Ricardo de A.Araujo, and Tiago A.E. Ferreira, “An intelligent hybrid morphological- rank-linear method for financial time series prediction”, Neurocomputing, Vol. 72, pp. 2507-2524, 2009.

[17]. Md. Mustafizur Rahman, Md. Monirul Islam and Xin Yao, “Layered Ensemble Architecture for Time Series Forecasting”, IEEE Transaction cybernetics, 2015.

[18]. Cecil Bozarth, “Measuring Forecast Accuracy Approaches to Forecasting A Tutorial”, http://scm.ncsu.edu/scm-articles/article/measuring-forecast-accuracy approaches-to-forecasting-a-tutorial.

[19]. S. N. Sivanandam, and S. N. Deepa,“Principles of softcomputing”, Wiley India, First Edition, pp. 79-80, 2008.

[20]. Bombay Stock Exchange India, http://www.bseindia.com.

