International Journal of Latest Research in Engineering and Technology (IJLRET)
ISSN: 2454-5031
www.ijlret.com || Volume 03 - Issue 03 || March 2017 || PP. 01-20

A Comparative Study on Stock Market Time Series Data Index Analysis by Various Neuro Fuzzy Models Using Tracking Signal Approach

Ashok Kumar D[a], MuruganS[b]
[a]*Department of Computer Science, Government Arts College, Trichirappalli-620 022,Tamilnadu, India.*
[b]*Department of Computer Science, Alagappa Government Arts College, Karaikudi-630 003,Tamilnadu, India.*

Abstract: This study proposed a novel Neuro Fuzzy model with Tracking Signal (TS) approach 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 (MLFFNN), Nonlinear Auto Regressive eXogenous Neural Network (NARXNN) and Fuzzy Time Series (FTS) model. It uses the Tracking Signal (TS) and rejects all models which result in values outside the interval. The effectiveness of the proposed approach is seen to be a step ahead of Bombay Stock Exchange (BSE100) closing index of Indian stock market. This novel approach reduces the over-fitting problem, neural network structure, training time, fast in convergence speed and improves forecasting accuracy. In addition, the present approach has been compared with different MLFFNN, NARXNN, FTS-MLFFNN and FTS-NARXNN model and identified the neuron counts in the hidden layer for every model which leads to reduce over-fitting and under-fitting problem. The experimental result shows that the FTS-NARXNN with TS approach outperformed other model and it produced 99.57% accuracy.
Keywords:NARXNN model, MLFFNN model, Performance Analysis, Fuzzy Time Series Data, Tracking Signal, Forecasting.

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 Wong, W.K., Min Xia, & Chu, W.C. (2010).

A number of neural network (NN) models (Kumar, K. & Bhattacharya, S. (2006),Kanas, A. (2001), Gronholdt, L. &Martensen, A. (2005), Lean Yu, Shouyang Wang, & Kin Keung Lai. (2007), Palmer, A., Montano, J.J. &Sese, A. (2006)) and hybrid models (Ricardo de .A Araujo (2009), Mehdi Khashei& Mehdi Bijari (2011), Tiffany Hui-Kuang Yu & Kun-Huang Huarng (2010)) have been proposed during the last few years for obtaining accurate forecasting results, in an attempt to outperform the conventional linear and nonlinear approaches. NNs are non-linear in nature and where most of the naturalreal world systems are non-linear in nature, so, NN are preferred over thetraditional models. Tiffany Hui-Kuang Yu & Kun-Huang Huarng (2010) found that applications of NN on credit ratings, Foreign exchange rate forecasting, Dow Jones Forecasting, stock ranking, customer satisfaction analysis and tourism demand was varied and effective. The reason is that the NN is a global function approximation which can mapping any linear or non-linearfunctions.

Although NNs have the advantages of accurate forecasting, the most important issues mentioned in the analyzed articles are as follows: (i) there is no systematic rule to identify neuron counts in the hidden layer MehdiKhashei& Mehdi Bijari (2011).(ii) Min Qi &Guoqiang Peter Zhang (2001) investigated and reported that thein-sample (training set) model selection criteria cannot offer a reliable guideto out-of-sample (testing set) performance and there is no apparent linking between in-sample (training set) model t and out-of-sample (test set) forecasting performance. NN model suffers due to under-fitting orover-fitting problems.

Timothy Master Timothy Master (1993) 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 of (m*n) neurons. Jeff Heaton (2008) 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(Balestrassi, P.P., Popova, E., Paiva, A.P., &MarangonLima J.W. (2009), Abbas Vahedi (2012), IebelingKaastra, A.& Milton Boyd, b. (1996)). The performance of various types of training algorithms (Ashok kumar, D. &Murugan, S. (2013), Ashok kumar, D. &Murugan, S. (2014)) 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 (2012) suggests that design of ten neural networks with different types of random initial weights to mitigate the occasional bad random start. AdebiyiAyodeleAdebiyiAyodele, A., Ayo Charles, K., Adebiyi Marion, O., &Otokiti Sunday O. (2012) suggests that training a great number of ANN with different configurations and selects the optimum model will improveforecasting accuracy.

The data set in many applications is divided into two sets: training and testing set as observed by (Mehdi Khashei& Mehdi Bijari (2011), Ti any Hui-Kuang Yu & Kun-Huang Huarng (2010), ErkamGuresen, GulgunKayakutlu, Tugrul, &Daim, U. (2011)). 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 Lean Yu, Shouyang Wang, & Kin Keung Lai. (2005). The published research articles (Ricardo de .A Araujo(2009), MustazurRahman, Md., Monirul Islam, Md. &Xin Yao (2015), Suresh Kumar, K.K. (2012) 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. Cecil Bozarth (2011)reported that, the TS is a statistical measure which is used to assess the presence of bias in the forecast model; and also it warns that there are unexpected outcomes from the forecast. Lean Yu et al. Lean Yu, Shouyang Wang, & Kin Keung Lai. (2005) proposed that adaptive smoothing approachis used to adjust the NN learning parameters automatically by TS underdynamic varying environments. In their study TS is used during the NN training.

Many research articles presented in the literature are related to selecting the optimal number neurons in a hidden layer of a neural network. Thesearticles 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 10 different neural network models with 15 different weights for single training function. 12 training functions were used in this study. 120 neural network models in total with differentweight are developed. After analysing the different neural network model,this study has brought out the results of optimum neural network model for every training function. The neural network model selection is normally based on trial and error method. The proposed approach has endeavoured to select optimum neural network model by adjusting two important parameters,namely number of neuron in the hidden layer and training function used inthe neural network. In addition, the present study is maiden e ort that the TS is used to analyze and select the best NN model after the NN training to improve forecasting accuracy.

The result of this study is seven folded: firstly, different MLFFNN, NARXNN, FTS-MLFFNN and FTS-NARXNN model was created for forecastingthe closing stock index of the BSE100 stock market. Secondly, the performance measure Tracking Signal (TS) is introduced to select the optimum neuralnetwork model which reduces the network complexity, training time; faster in convergence; improves better forecast accuracy; and reduce over-fitting and under-fitting problem. Thirdly, the in-sample (train set and validationset) and the out-of-sample (test set) forecasting performance analyzed and reported using different performance measure such as SMAPE, POCID and TS using different models with TS approach and different models withoutTS approach. Fourthly, the neuron count in the hidden layer is identified in different models with TS approach and different models without TS approachfor BSE100 stock market. Fifthly, the performance of the various models with TS approach is compared with the performance of the various modelswithout TS approach; the result indicates that the various models with TS approach outperformed various models without TS approach. Sixthly, unlike the report of Timothy Master, the investigations of this study revealthat, the neuron counts in the hidden layer cannot be identified by some rule of thumb and it can be identified by constructing different models with different parameters and selects the best one. Seventhly, unlike the report of Min Qi &Guoqiang Peter Zhang (2001), the investigation of this study proves that thein-sample (training and validation set) model selection criteria can provide are 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, NARXNN, FTS, TS and performance measures whichare used to assess the performance of the proposed approach; Section 3 describes the details of proposed model with TS approach and without TS approach; Section 4 reports the experimental results attained by the various models with TS approach and various models without TS approach using realworldfinancial time series BSE100 stock index dataset. Finally this study isconcluded in section 5.

2. Background

2.1. Multi Layer Feed Forward Neural Network

MLFFNN consists of an input layer, one or more hidden layer and an output layer. The hidden layer receives weight from input layer. Eachsubsequent 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. The biasterm also acts as weights and it shows the architecture of Back Propagation Neural Network, depicting only the direction of information flow for the feedforward 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 Sivanandam, S.N. &Deepa S.N.(2008).

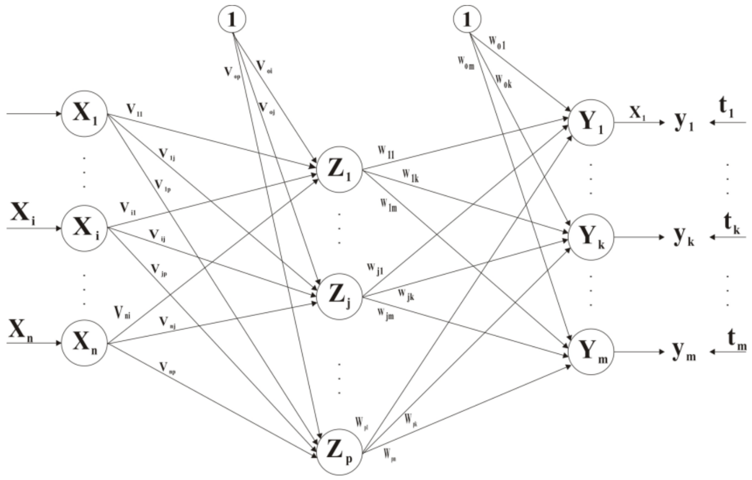

Figure 1: Feed Forward Neural Network Architecture

2.2. Non-linear Auto Regressive and eXogenous NN (NARXNN)

NARXNN architecture creates feed forward back propagation with feedback from output unit to input unit. The first hidden layer receives weight from input unit. Each subsequent layer receives weight from the previous layer.The NARX network can be carried out in one out of the following two modes: Series-Parallel (SP) Mode and Parallel mode. In SP mode, the output's regressor is formed only by actual values of the system's output. In ParallelMode, estimated outputs are fed back and included in the output's regressor. The P mode NARX NN architecture can be represented in Figure2.

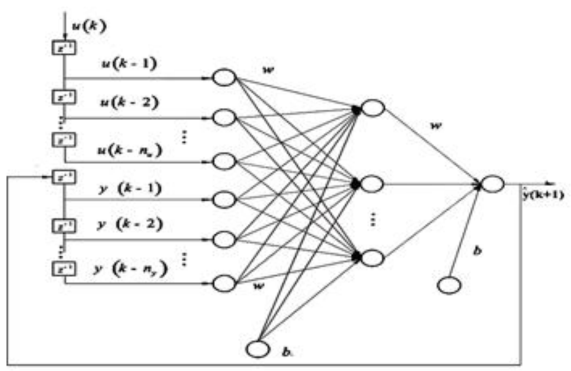

Figure **2**. NARX Neural Network

The dynamics of multi-layer perceptron (MLP) neural network consists of an input vector composed of past values of the NN input and output. This is the approach by which the MLP can be considered as a NARX model of the system. This way of introducing dynamics into a static network has the advantage of being simple to implement. To deduce the dynamic model of realized fNN system, NARX P-type NN model (Jose Maria, P., Junior, Guilherme, A., &Barreto. (2008)) can be represented as follows:

$$y(k+1) = f_{ANN}$$
$$= (y(k), y(k-1), \dots y(k-n+1), u(k), u(k-1), \dots u(k-m+1) + \in (k)$$

where y(k+1) is model predicted output, f_{ANN} is a non-linear function describing the system behavior, y(k), u(k), ε(k) are output, input and approximation error vectors at the time instances k, n and m the order of y(k) and u(k) respectively. The order of the process can be predicted from experience. Modeling by NN depends on the considerations of an approximate function of f_{ANN}. Approximate dynamic model is developed by adjusting a set of connection biases (b) and weight (W) via training function defined as MLP network.

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.

Fuzzy time series data are structured by fuzzy sets. Let U be the universe of discourse, such that$U = \{u_1, u_2, \dots . u_n\}$. Let us defined a fuzzy set A of U by $A = \frac{f_A(u_1)}{u_1} + \frac{f_A(u_2)}{u_2} + \; . \; . \; . \; . \frac{f_A(u_n)}{u_n}$where f_A is the membership function of A, and $f_A: U \to [0\,,1]$. $f_A(u_i)$ is the membership value of u_i in A, where $f_A(u_i) \epsilon\, [0\,,1]$and $1 \leq i \leq n$. Tiffany Hui-Kuang Yu and Kun-Huang Huarngproposed a sequence of steps to design NNFTS model.

2.4. Tracking Signal

Tracking Signal (TS) is used to pinpoint forecasting models that need adjustment. The calculation of TS (Cecil Bozarth (2011)) is represented in the equation (3). If the forecast value is lower than the actual value then themodel is 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 theforecast 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 theorywhich 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).

2.5. 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 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 the size of test set. The global performance of a forecasting model is evaluated by the SMAPE (MustazurRahman, Md., Monirul Islam, Md. &Xin Yao (2015)) which is used in NN3, NN5 and NNGC1 forecasting competition. A smaller SMAPE value suggests the better forecasting accuracy. It can be expressed as

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

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

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

POCID (Percentage of Change in Direction) (Ricardo de .A Araujo (2009))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(2011) 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.

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

3. Proposed Methodology

Neural Network based fuzzy time series model was proposed by by Tiffany Hui-Kuang Yu and Kun-HuangHuarng (2010). In their study, the neuron count in the hidden layer is set to the sum of the number of input and output nodes. The proposed approach of the present study systematically constructs different MLFFNN, NARXNN, neural network based fuzzy time series model such as FTS-MLFFNN and FTS-NARXNN model from simple architecture to complex architecture; and the optimum neural network and neural network based fuzzy time series model selection 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 model which reduces over-fitting or under-fitting problem. The modification of Neural Network based Fuzzy Time Series (NNFTS) model proposed by Ti any Hui-Kuang Yu & Kun-Huang Huarng (2010) can be represented in Algorithm 1 and its forecasting model represented in Figure 3. This algorithm contains four models which is enabled and executed based on the choice selection.

Algorithm 1.MLFFNN/NARXNN/FTS-MLFFNN/FTS-NARXNN without tracking signal Approach.
Input: Time series/Fuzzy time series data for the closing stock index vector
Output:Time series/Fuzzy time series data for predicted closing stock index vector

---

1. Select the forecasting model choice:
    1. MLFFNN without TS Approach
    2. FTS-MLFFNN without TS Approach
    3. NARXNN without TS Approach
    4. FTS-NARXNN without TS Approach
2. if (choice == 1) then goto step 7.
3. else if (choice == 2) then goto step 26.
4. else if (choice == 3) then goto step 7.
5. else if (choice == 4) then goto step 26.
6. end.
7. Read the input and target pair from the data file and normalizedforpre-process the data using mapminmaxfuction.
8. Set the maximum number of neuron count MAX NEURON in hidden layer and maximum number of trial MAX TRIAL (random initial weight) for random weight generation.
9. FOR NEURON = 1 TO MAX NEURON.
10. FOR TRIAL = 1 TO MAX TRIAL.

11. if (choice==1) then

    Create MLFFNN architecture here; specify the input and target vector, number of hidden layer, training function, transfer function used in the hidden and output layer.
12. elseif (choice==3) then
    Create NARXNN architecture here; specify the input and target vector, number of hidden layer, training function, transfer function used in the hidden and output layer.
13. end.
14. Select the data division ratio using divide function and divide the dataset into training dataset, validation dataset and test dataset usingdivideparam function. Training dataset and validation dataset are referred to as in-sample observation. Test dataset is referred to as out-of-sample observation.
15. Train the neural network using train function.
16. Simulate the neural network using sim function.

17. Denormalize or post-process the simulated neural network output data.

18. Calculate the performance measure SMAPE, POCID and TS for train, validation and test set using equation(1) - (4).

19. if (choice==1) then

    Record the result of neuron count, trial number, training time, epoch (convergence speed) and performance measure specified in step 18. It contains the performance of different MLFFNN model without TS approach.
20. elseif (choice==3) then

    Record the result of neuron count, trial number, training time, epoch (convergence speed) and performance measure specified in step 18. It contains the performance of different NARXNN model without TS approach.
21. end.
22. END for TRIAL.

23. END for NEURON.

24. From the step 19, select the optimum MLFFNN model, which providesless error in SMAPE of validation dataset using MLFFNN model withoutTS approach and goto step 49.
25. From the step 20, select the optimum NARXNN model, which provides less error in SMAPE of validation dataset using NARXNN model without TS approach and goto step 51.

26. 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.

27. 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, $D_{min}\ and D_{max}$.

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

28. 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 $u_1, u_2, u_3, \ . \ . \ .$

$$\text{Where } u_1 = [D_{min} - D_{1,} D_{min} - D_1 + l],$$
$$u_2 = [D_{min} - D_1 + l\,, \quad D_{min} - D_1 + 2l],$$

$u_3 = [D_{min} - D_1 + 2l, \quad D_{min} - D_1 + 3l],$

……

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

Their corresponding midpoints are

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

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

…..

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

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

29. Fuzzification: $d'(t-1,t)$ can be fuzzified into a set of degrees of membership, V(t-1,t), where $V(t-$ *1,t=μt−1,t,1μt−1,t,2….*

30. Neural Network Creation and Training: Before training the neural network, Set the maximum number of neuron counts MAX NEURONin hidden layer, maximum number of trial MAX TRIAL (random initialweight) for random weight generation and SD (Standard Deviation) value for assigning TS limit.

31. FOR NEURON = 1 TO MAX NEURON.

32. FOR TRIAL = 1 TO MAX TRIAL.

33. if (choice==2) thenCreate MLFFNN architecture here; specify the input and target vector (fuzzy time series data), number of hidden layer, training function, transfer function used in the hidden and output layer.

34. else if (choice==4) then

    Create NARX architecture here; specify the input and target vector (fuzzy time series data), number of hidden layer, training function, transfer function used in the hidden and output layer.

35. end.

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

37. Train the NN using train function.

38. Neural Network Forecasting: With V(t-1,t) we can proceed to forecastV(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.

39. Defuzzification:Defuzzify the degrees of membership:

$$fd(t-1,t) = \frac{\sum_{k=1} \mu_{t-1,t}^{k}\, X m^k}{\sum_{k=1} \mu_{t-1,t}^{k}}$$

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

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

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

41. Performance Evaluation: Calculate the performance measure SMAPE, POCID and TS for train, validation and test set using equation (1) -(4).

42. if (choice==2) thenRecord the result of neuron counts, trial number, training time, epoch (convergence speed) and performance measure specified in step 41. It contains the performance of different FTS-MLFFNN without TS approach.

43. else if (choice==4) thenRecord the result of neuron counts, trial number, training time, epoch (convergence speed) and performance measure specified in step 41. It contains the performance of different FTS-NARXNN without TS approach.

44. end.

45. END for Trial.

46. END for Neuron.

47. From the step 42, select the optimum FTS- MLFFNN model, which provides less error in SMAPE of validation dataset for FTS-MLFFNN without TS approach and go to step 50.

48. From the step 43, select the optimum FTS-NARXNN model, which provides less error in SMAPE of validation dataset for FTS-NARX without TS approach and goto step 52.

49. Forecast the future value using MLFFNN model without TS Approach.

50. Forecast the future value using FTS-MLFFNN model without TS Approach.

51. Forecast the future value using NARXNN model without TS Approach.

52. Forecast the future value using FTS-NARXNN model without TS Approach.

Over-fitting is one of the main issues in neural network modeling. In order to reduce over-fitting problem, this study is also proposed a MLFFNN, NARXNN, FTS-MLFFNN and FTS-NARXNN model with TS approach which is used to forecast the closing index of the stock market. All modelsare received closing stock index historical data and trains different networkby using different random initial weight with different neurons. TS measure is used to reject all NN model which results in values outside the interval [-4, +4] in training set and validation set of different neural networks to reduce NN structure which leads to avoid over-fitting or under-fitting problems. In neuralnetwork modeling, training parameter and the weight play an important roleto increase the forecasting accuracy. The proposed MLFFNN, NARXNN, FTS-MLFFNN and FTS-NARXNN model with TS approach tries to find optimal parameter, particularly, neuron counts in the hidden layer and optimalweight for the forecasting problem in time series.

Forecasting strategies are taken a step ahead of prediction in this study. Let $y_1, y_2, y_3, \ldots y_t$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}, $\ldots y_1$. The result out network can be used for multi-step prediction by feeding the prediction back to the input of network recursively.The proposed approach of forecasting model is represented in Figure 3.

In Figure 3, If the model selection is MLFFNN or NARXNN then X_i is the closing stock index vector. If the model selection is FTS-MLFFNN orFTS-NARXNN then X_i is the fuzzyfied closing stock index vector. If the model selection is MLFFNN or NARXNN then Y_i is the predicted closing stock index vector. If the model selection is FTS-MLFFNN orFTS-NARXNN then Y_i is the predicted fuzzified closing stock index vector from neural network model and N_j is neuron counts in hidden layer. For every model(NARXNN, FTS-MLFFNN and FTS-NARXNN), the presence of tracking signal interval [-4, +4] is verified in training set and validation set. If it is present, the model is considered as feasible model otherwise the model isrejected.

International Journal of Latest Research in Engineering and Technology (IJLRET)
ISSN: 2454-5031
www.ijlret.com || Volume 03 - Issue 03 || March 2017 || PP. 01-20

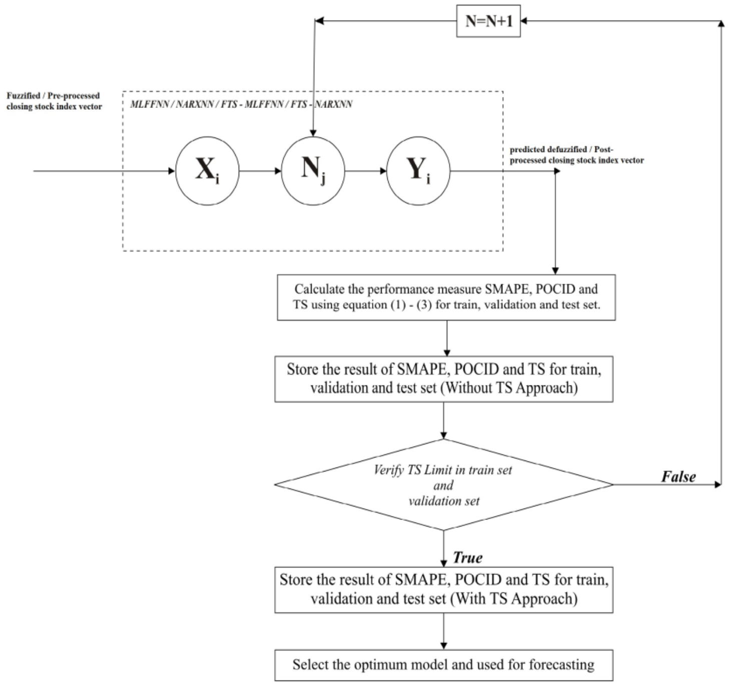

Figure 3: NN model selection Without TS and With TS Approach

This process is repeated until the specified trial number (random initial weight) and maximum neuron count is reached. The implementation procedure of MLFFNN, NARXNN, FTS-MLFFNN and FTS-NARXNN with TS approach is represented in Algorithm 2, and explained further as follows.

Algorithm 2.MLFFNN/NARXNN/FTS-MLFFNN/FTS-NARXNN withtracking signal Approach.
Input: Time series/Fuzzy time series data for the closing stock index vector
Output:Time series/Fuzzy time series data for predicted closing stock index vector

---

1. Select the forecasting model choice:
    1. MLFFNN with TS Approach
    2. FTS-MLFFNN with TS Approach
    3. NARXNN with TS Approach
    4. FTS-NARXNN with TS Approach

2. if (choice == 1) then goto step 7.

3. else if (choice == 2) then goto step 24.

4. else if (choice == 3) then goto step 7.

5. else if (choice == 4) then goto step 24.

6. end.

7. Repeat the Step 7 and Step 8 of Algorithm-1

8. FOR NEURON = 1 TO MAX NEURON

International Journal of Latest Research in Engineering and Technology (IJLRET)
ISSN: 2454-5031
www.ijlret.com || Volume 03 - Issue 03 || March 2017 || PP. 01-20

9. FOR TRIAL = 1 TO MAX TRIAL

10. Repeat the Step 11 to Step 21 of Algorithm-1.

11. if (choice==1) then

12. Verify the interval $[-\theta, +\theta]$ of Tracking Signal in training set (TStrain) and validation set (TSvalidation) from step 10, where θ =round (SD * 1.25).
If (TStrain $\geq -\theta$&&TStrain $\leq +\theta$) and (TSvalidation $\geq -\theta$&& TSvalidation $\leq +\theta$) then go to step 13. Otherwise, go to step 9.

13. Record the result of neuron count, trial number, training time, epoch convergence speed) and performance measure specified in step 18 of Algorithm-1. It contains the performance of different MLFFNN withTS approach. Goto step 19.

14. elseif (choice==3) then

15. Verify the interval $[-\theta, +\theta]$ of Tracking Signal in training set (TStrain) and validation set (TSvalidation) from step 10, where θ =round (SD * 1.25).
If (TStrain $\geq -\theta$&&TStrain $\leq +\theta$) and (TSvalidation $\geq -\theta$&& TSvalidation $\leq +\theta$) then go to step 16. Otherwise, go to step 9

16. Record the result of neuron count, trial number, training time, epoch (convergence speed) and performance measure specified in step 18 ofAlgorithm-1. It contains the performance of different NARXNN withTS approach. goto step 20.

17. END for TRIAL

18. END for NEURON

19. From the step 13, select the optimumMLFFNN model, which provides less error in SMAPE of validation dataset using MLFFNN model withTS approach. goto step 35.

20. From the step 16, select the optimum NARXNN model, which provides less error in SMAPE of validation dataset using NARXNN model with TS approach. goto step 37.

21. Repeat Step 26 to Step 30 of Algorithm-1.

22. FOR NEURON = 1 TO MAX NEURON.

23. FOR TRIAL = 1 TO MAX TRIAL.

24. Repeat the Step 33 to Step 41 of Algorithm-1.

25. if (choice==2) then

26. Verify the interval $[-\theta, +\theta]$ of Tracking Signal in training set (TStrain) and validation set (TSvalidation) from step 42 of Algorithm 1, where θ =round (SD * 1.25).
If (TStrain $\geq -\theta$&&TStrain $\leq +\theta$) and (TSvalidation $\geq -\theta$&& TSvalidation $\leq +\theta$) then go to step 27. Otherwise, go to step 23.

27. Record the result of neuron counts, trial number, training time, epoch(convergence speed) and performance measure specified in step 40. It contains the performance of different FTS-MLFFNN with TS approach. goto step 33.

28. elseif (choice==4) then

29. Verify the interval $[-\theta, +\theta]$ of Tracking Signal in training set (TStrain) and validation set (TSvalidation) from step 43 of Algorithm 1, where θ =round (SD * 1.25).
If (TStrain $\geq -\theta$&&TStrain $\leq +\theta$) and (TSvalidation $\geq -\theta$&& TSvalidation $\leq +\theta$) then go to step 16. Otherwise, go to step 34.

30. Record the result of neuron counts, trial number, training time, epoch(convergence speed) and performance measure specified in step 40. It contains the performance of different FTS-NARXNN with TS approach. goto step 20.

31. END for TRIAL.

32. END for NEURON.

33. From the step 27, select the optimum FTS-MLFFNN model, which provides less error in SMAPE of validation dataset for FTS-MLFFNN with TS approach and goto Step 36.

34. From the step 30, select the optimum FTS-NARXNN model, which provides less error in SMAPE of validation dataset for FTS-NARX withTS approach and goto Step 38.

35. Forecast the future value using MLFFNN model with TS Approach.

36. Forecast the future value using FTS-MLFFNN model with TS Approach.

37. Forecast the future value using NARXNN model with TS Approach.

38. Forecast the future value using FTS-NARXNN model with TS Approach.

Neural network training process is an iterative process. Before training the NN, the input data and target data should be normalized or preprocessed, if the model selection is either MLFFNN or NARXNN. During this process the input data is converted into -1 to +1. The input data and target data shouldbe converted into fuzzy data using the step 7 in Algorithm 2, if the model selection is either FTS-MLFFNN or FTS-NARXNN. The closing stock index vector or fuzzified closing stock index vector can be divided into three parts: a training, validation and test dataset. Training dataset can be used to t the models, validation dataset can be used to evaluate the forecasting errorfor model selection; test dataset can be used to assess the generalization error in the final model. Divide block method is used to distribute the datasetinto train, validation and test data set. After the division of data chosen, MLFFNN, NARXNN, FTS-MLFFNN and FTS-NARXNN model with tan sigmoidal function in the hidden layer and linear function in the output layeris used. The tan sigmoidal function and linear function is defined in equation (5) and (6).

$$tansig(x) = \frac{2}{1 + e^{-2x}} - 1 \qquad (5)$$

$$purelin(x) = x(6)$$

Levenberg Marquardt is used as a training function.After training the NN, simulate the NN if the model selection is either MLFFNN or NARXNN. Simulate the neural network and defuzzify the simulated output using the step 24 in Algorithm 2, if the model selection iseither FTS-MLFFNN or FTS-NARXNN. Finally analyze the performance of neural network using performance measure equation (1) - (4).

In MLFFNN,NARXNN, FTS-MLFFNN and FTS-NARX neural network without TS approach, after post-processing the data, store the results of performance measure SMAPE, POCID and TS of training set, validation set and test set for different MLFFNN, NARXNN, FTS-MLFFNN and FTS-NARX neural network model with different neuron counts. The optimum MLFFNN,NARXNN, FTS-MLFFNN and FTS-NARX neural network model selection is based on minimum forecasting error in validation set of SMAPE.

After selecting the optimum model using MLFFNN, NARXNN, FTS-MLFFNN and FTS-NARX neural network without TS approach, still, there existsover-forecast or under-forecast in training dataset, validation dataset andtest 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 FTS-NARX model with neuron count 13 and the training function is trainlm, which is represented in Figure 4.

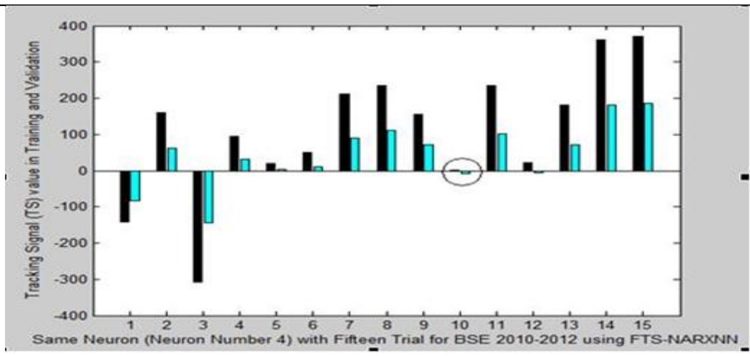

Figure 4: Level of over-forecast and under-forecast in FTS-NARX with Neuron 4 (Fifteen trial)

Test case 8is identified as the optimum FTSNARX neural network model by the TSmeasure marked with the circle in Figure 4, which contains 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. MLFFNN, NARXNN, FTS-MLFFNN and FTS-NARX with TS approach is used to assess the over-forecast or under-forecast in training dataset, validationdataset and test dataset.

For every MLFFNN, NARXNN, FTS-MLFFNN andFTS-NARX model, check the TS interval [-θ ; +θ] in the training dataset and validation dataset, where θ = 4 and SD=3. It rejects all MLFFNN, NARXNN,FTS-MLFFNN and FTS-NARX model which results in values outside the interval of [-4, +4]; it accepts the MLFFNN, NARXNN, FTS-MLFFNN andFTS-NARX 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, NARXNN, FTS-MLFFNN and FTS-NARX model selection is based on the TS interval value [-4, +4] in the training dataset andvalidation dataset which contain minimum forecasting performance error inSMAPE (Instead of SMAPE any other performance measure can be used) of validation set. The threshold of 4 is really a threshold of 3.75 (3 * SD). 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 ina normally distributed population is built as 1.25 MAD = 1 SD (standard deviation of the distribution). For this reason, this study selects the interval [-4, +4].

To highlight the importance of TS measure in neural network model selection, this study reports and compares the result using NN model withoutTS approach and NN model with TS approach. In NN model without TS approach, the TS measure did not used in neural network model selection but reports the results of TS in train, validation and test set. It shows how the selection of model affected by over-fit or under-fit without consideringTS measure. In NN model with TS approach, the TS measure is used to NNmodel selection and reports the results of TS in train, validation and test set. It shows how the selection of model reducing in over-fit or under-fit problem with considering TS approach.

4. Experimental Results

In this section, the excellence of proposed approach is verified and it isapplied toclosing stock index forecasting. The results were carried out inMATLAB 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 FTSNARX with TS approach is testedon 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 BSE (2012). For each NN created with different random initial weight for neuron 1 to neuron 18 with different training functions. The choice of random initial weight (trial) and maximum neuron counts is selected by user. Inthis study, random initial weight is 15 and maximum neuron count is 10 for BSE100 stock market index. The data division ratio is 50/25/25. Theresults of performance measure of 18 different models from 9-1-9 to 9-18-9 were generated for FTS-MLFFNN and FTS-NARXNN model. (Here 9-1-9,)first part represents neuron counts in input layer, second part representsneuron counts in hidden layer and third part represents the neuron counts in output layer). The results of performance measure of 10 different models from 1-1-1 to 1-10-1 were generated for MLFFNN and NARXNN model. Every model contains fifteen different random initial weight generations.

Fromthe different models with different trial, some models are selected by theMLFFNN, NARXNN, FTS-MLFFNN and FTS-NARXNN 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, NARXNN, FTS-

MLFFNN and FTS-NARXNN with TS approach which does not containthe interval [-4, +4] in the training dataset and validation dataset of trackingsignal. Rejection of model and selection of model using MLFFNN, NARXNN, FTS-MLFFNN and FTS-NARXNN with TS approach are represented in Table 1.

Table 1: NN model rejection and selection using di erent model with TS approach

Approach	Model Rejection	Model Selection
MLFFNN	1-2-1, 1-5-1, 1 -7-1, 1-10-1	1-1-1, 1-3-1, 1-4-1, 1-6-1, 1 -8-1, 1-9-1
FTSMLFFNN	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
NARXNN	1-1-1, 1-3-1, 1-4-1, 1-5-1, 1-6-1, 1-7-1, 1-8-1, 1-9-1, 1-10-1	1-2-1
FTSNARXNN	9-6-9 to 9-15-9, 9-18-9	9-1-9, 9-2-9, 9-3-9, 9-4-9, 9-5-9, 9-16-9, 9-17-9

The performance measure of SMAPE, POCID and TS of training set, validation set and test set of different NN model with TS approach anddifferent NN model without TS approach are reported in Table 2.

The optimum results of the test set of the proposed approach with different models are reported in seven aspects. (i) forecasting error of the NN with respect to SMAPE (ii) over-fitting or under- fitting problem with respect to TS (iii) correctness of the predicted direction with respect to POCID (iv)complexity of the NN with respect to neurons count in the hidden layer (v) convergence speed (epoch) of the NN (vi) and training time of the NN (vii)accuracy of the NN. (Note: Lower value in SMAPE, higher value in POCID indicates best prediction model.)

Table 2: Performance evaluation by SMAPE, POCID, TS and POA using NARX with TS and NARX without TS approach

NN/FTS Approaches	Measure	Without TS approach			% of Accuracy Test	With TS approach			% of Accuracy Test
		Train	Val	Test		Train	Val	Test	
MLFFNN	SMAPE	0.854	0.726	0.915	99.09	0.868	0.760	0.836	99.16
	TS	2.220	5.890	34.200		0.000	0.173	0.036	
	POCID	77.100	75.900	74.900		77.600	73.300	76.500	
FTS-MLFFNN	SMAPE	0.610	0.820	0.474	99.53	0.600	0.955	0.464	99.54
	TS	-18.500	-10.400	18.900		3.810	-2.040	15.100	
	POCID	85.000	82.400	81.800		92.200	95.200	92.000	
NARXNN	SMAPE	0.807	1.110	0.653	99.35	0.750	1.070	0.642	99.36
	TS	-0.376	-54.800	-4.880		1.730	-3.800	1.900	
	POCID	77.000	73.800	77.700		77.000	73.800	77.700	
FTS-NARXNN	SMAPE	0.720	0.922	0.882	99.12	0.683	0.913	0.428	99.57
	TS	218.000	79.400	157.000		1.440	-7.470	11.800	
	POCID	47.900	44.900	46.000		94.100	93.600	93.000	

From Table 2, the performance measure SMAPE of test set in different NNmodel with TS approach is low when compared to the performance measure SMAPE of test set in different NN model without TS approach. It indicates that the forecasting error is low in different NN model with TS approach than different NN model without TS approach. In addition, it is observed that the forecasting error of SMAPE in validation set is high in differentmodels with TS approach when compared to different models without TS approach; Different models with TS approach produce lowest forecasting error in SMAPE of test set even it produce highest forecasting error value in validation set.

The performance measure TS of test set in different NN model with TSapproach is extremely low when compared to the performance measure TS of test set in different NN model without TS approach. It indicates that the over-fitting or under- fitting problem can be reduced in different NN model with TS approach than different NN model without TS approach. Thepercentage of accuracy of testset in different NN model with TS approach ishigh when compared to the percentage of accuracy of test set in different NN model without TS approach. It indicates that the forecasting accuracy is high in different NN model with TS approach than different NN model without TS approach.

The performance measure POCID of test set in different NN modelwith TS approach is high when compared to the performance measure POCIDof test set in different NN model without TS approach. It indicates that the prediction direction is high in different NN model with TS approach than different NN model without TS approach. MLFFNN, NARXNN,FTS-MLFFNN and FTS-NARXNN with TS approach outperformed MLFFNN, NARXNN, FTS-MLFFNN and FTS-NARXNN without TS approach withrespect to SMAPE, TS and POCID measure.

Fildes, R., &Makridakis (1995) reported that if a close relationship between model fit (train set) and out of sample forecasts (test set) does not exist, then it is hard to argue that selection of NN model should be based on minimum model fitting errors. From Table 2, it is observed thatthere is a close relationship between train and test dataset in different NN model with TS approach when compared to different NN model without TS approach.

According to the Fildes and Makridakisstatement,different NN model with TS approach has close relationship between training set and test set of performance measure SMAPE, POCID and TS. It is observed that, thedifference between the performance measure SMAPE of training dataset and test dataset in different NN model with TS approach is slightly close to eachother when compared to the performance measure SMAPE of training dataset and test dataset in different NN model without TS approach. For example the train and test set of SMAPE in the FTSNARX without TS approach is0.72 and 0.88 respectively, the difference is 0.16; whereas the train and test set of MAPE in the FTSNARX with TS approach is 0.48 and 0.43 respectively,the difference is only 0.5. It indicates that FTS-NARXNN with TS approach has close relationship between train and test set of SMAPE when compared to FTSNARX without TS approach. This is the main purpose of using trackingsignal in this study. Unlike the report of Min Qi &Guoqiang Peter Zhang (2001), this closeness of training and testing performance measure of SMAPE indicates that the in-sample (training dataset) model selection criteria canprovide a reliable guide to out-of-sample (testing dataset) performance and an apparent connection between in-sample model t and out-of-sample modelforecasting performance. It happens due to the model selection based on tracking signal.

From Table 2, it is clearly observed that the result of performance measure TS of different NN model without TS approach is severely suffered by either over-fitting or under-fitting with respect to TS in training dataset and testdataset, whereas, the result of performance measure TS of different NN model with TS approach do not suffer due to under-fitting or over-fitting with respect to TS in training dataset and test dataset. For example, the train and test set of TS in the FTSNARX without TS approach is 218 and 157respectively; whereas the train and test set of TS in the FTSNARX with TSapproach is 1.44 and 11.80 respectively. It indicates the level of under-fitting or over-fitting is very high in NARX without TS approach when compared to NARX with TS; and also there is no close relationship between trainand test set of TS in FTSNARX without TS approach when compared to FTSNARX with TS approach.

Similarly, the train and test set of POCIDin the FTSNARX without TS approach is 47.90 and 46.00 respectively, the difference is 1.90; whereas the train and test set of TS in the FTSNARX with TS approach is 94.10 and 93.00 respectively, the difference is 1.10. It indicates the direction of prediction is low in FTSNARX without TS when compared to FTSNARX with TS; and also there is no close relationship between trainand test set of POCID in FTSNARX without TS approach when compared to FTSNARX with TS approach.With respect to lower value in SMAPE measure, FTS-NARXNN modeloutperformed other NN models with using TS approach; and FTS-MLFFNN outperformed other NN models without using TS approach. The maindifference is that differentNN model without TS approach suffers due to over-fitting or under-fitting problem; whereas the different NN model with TS approach reduces the over-fitting or under-fitting problem. MLFFNN, NARXNN, FTS-MLFFNN and FTS-NARXNN model with TS approach outperformed MLFFNN, NARXNN, FTS-MLFFNN and FTS-NARXNNwithout TS approach with respect to SMAPE, POCID and TS.

The test set of performance measure SMAPE of different NN model without TS and different NN model with TS approach are represented in Table 3 and their corresponding graph is represented in Figure 5. Best prediction performance is represented by boldface. Note: Lower value in SMAPE indicates best prediction model.

Table 3: Performance evaluation by SMAPE in test set

NN/FTS Approaches	Without TS	With TS
MLFFNN	0.915	**0.836**
FTS-MLFFNN	0.474	**0.464**
NARXNN	0.653	**0.642**
FTS-NARXNN	0.882	**0.428**

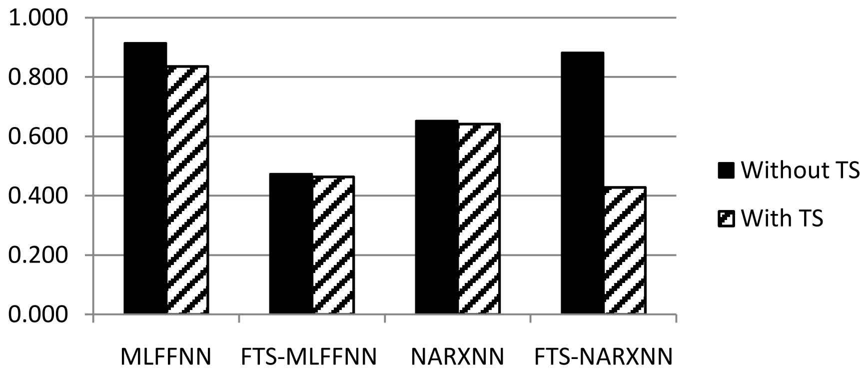

Figure 5: Performance evaluation by SMAPE in testset

From Table 3 and Figure 5, it is observed that the performance measureSMAPE of different NN model with TS approach is low when compared to the performance measure SMAPE of all other different NN model without TS approach.The test set of performance measure TS of different NN model without TS and different NN model with TS approach are represented in Table 4 and their corresponding graph is represented in Figure 6. Best results are represented by boldface.

Table 4: Performance evaluation by TS in test set

NN/FTS Approaches	Without TS	With TS
MLFFNN	34.200	**0.036**
FTS-MLFFNN	18.900	**15.100**
NARXNN	-4.880	**1.900**
FTS-NARXNN	157.000	**1.800**

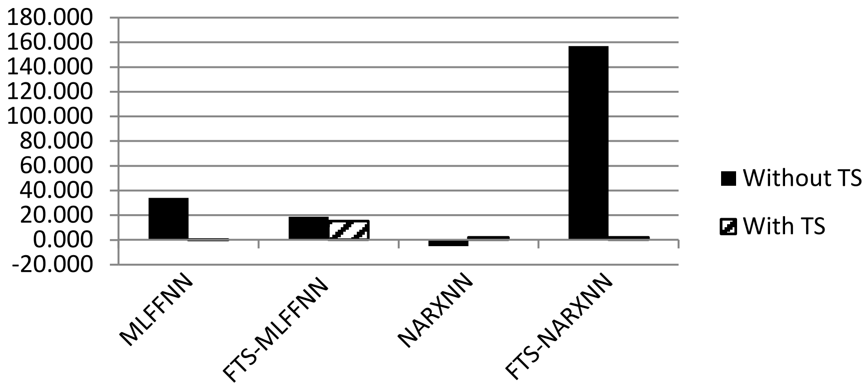

Figure 6: Performance evaluation by TS in testset

From Table 4 and Figure 6, it is clearly observed that the value of testset of TS in different NN model with TS approach is low when compared to the value of test set of TS in different NN model without TS approach. It indicates that the level of under-fitting or over-fitting is reduced in different NN model with TS approach when compared to different NN model without TS approach. Particularly, the test set of FTS-NARXNN with TS approachis very low which is 1.80 when compared to the test set of FTS-NARXNNwithout TS approach, which is 157. If the model selection is based on TS measures then it can reduce the over-fit or under-fit in the test dataset.The test set of performance measure POCID of different NN model without TS and different NN model with TS approach are represented inTable 5 and their corresponding graph is represented in Figure 7. Best results are represented by boldface. Note: Higher value in POCID indicates best prediction model.

Table 5: Performance evaluation by POCID in test set

NN/FTS Approaches	Without TS	With TS
MLFFNN	74.90	**76.50**
FTS-MLFFNN	81.80	**92.00**
NARXNN	77.70	**77.70**
FTS-NARXNN	46.00	**93.00**

From Table 5 and Figure 7, it is clearly observed that the value of test set of POCID in different NN model with TS approach is high when compared tothe value of test set of POCID in different NN model without TS approach. It indicates that the prediction direction is high in FTS-NARX with TS when compared to FTS-NARX without TS.

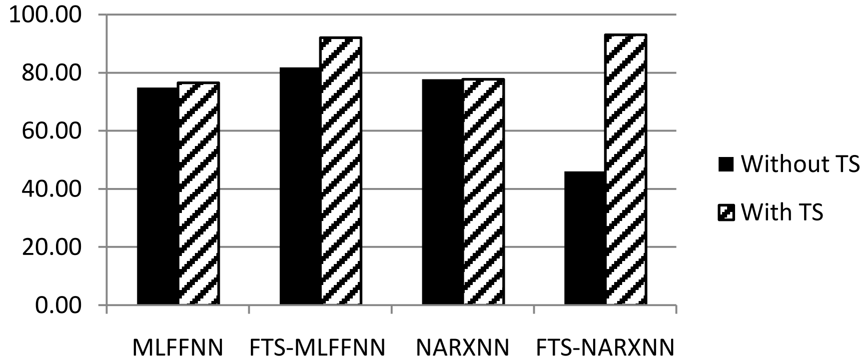

Figure 7: Performance evaluation by POCID in testset

The test set of percentage of accuracy of different NN model without TS and different NN model with TS approach are represented in Table 6and their corresponding graph is represented in Figure 8. Best results are represented by boldface.

Table 6: Performance evaluation by percentage of accuracy in test set

NN/FTS Approaches	Without TS	With TS
MLFFNN	99.09	**99.16**
FTS-MLFFNN	99.53	**99.54**
NARXNN	99.35	**99.36**
FTS-NARXNN	99.12	**99.57**

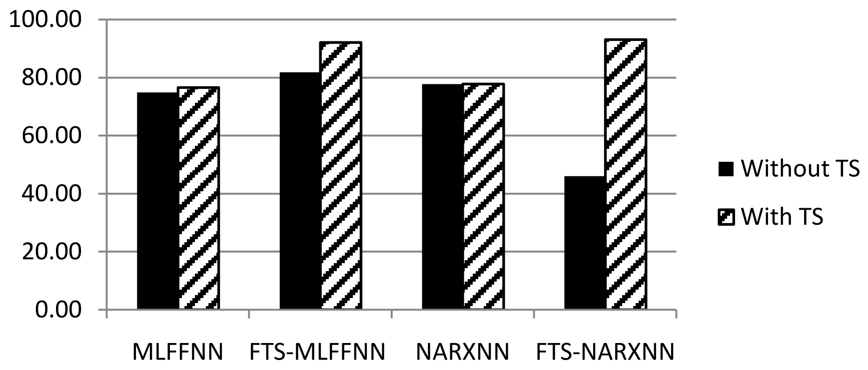

Figure 8: Performance evaluation by percentage of accuracy in testset

From Table 6 and Figure 8, it is clearly observed that the value of test set of percentage of accuracy in different NN model with TS approach is high when compared to the value of test set of percentage of accuracy in differentNN model without TS approach. It indicates that the percentage of accuracy is high in different NN model with TS approach.

Neuron counts in the hidden layer is identified for different NN model without TS and different NN model with TS approach are represented in Table 7 and their corresponding graph is represented in Figure 9. From Table 7 and Figure 9, the NN complexity is reduced in different NN model with TS approach when compared to different NN model without TS approach. The best NN model represented by boldface.

Table 7: Performance evaluation by neuron counts in the hidden layer

NN/FTS Approaches	Without TS	With TS
MLFFNN	4	**1**
FTS-MLFFNN	13	**7**
NARXNN	9	**2**
FTS-NARXNN	15	**4**

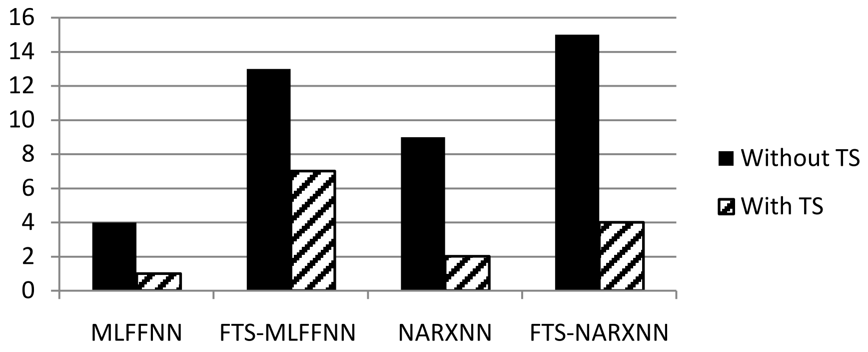

Figure 9: Performance evaluation by neuron counts in the hidden layer

According to the Timothy Master (1993) Pyramid rule, for a three layerwith m output and n input neurons, the hidden layer may have square 755 root (n * m) neurons. In this study, n=9 and m=9 for FTS-MLFFNN and FTS-NARXNN, square root (9*9) = 9; n=1 and m=1 for MLFFNN and NARXNN, square root(1*1)=1. Unlike the report of Master, the investigation of this study report that, neuron counts in the hidden layer cannot be determined by some rule of thumb and it can be identified by modifying various neural network parameters. This study tries to achieve best prediction result by modifying two important parameter neuron counts in the hidden layer and training function. It is also noted that Pyramid rule satisfied for MLFFNN with TS approach.NN training time for different NN model without TS and different NN model with TS approach is represented in Table 8 and their correspondinggraph is represented in Figure 10.

From Table 8 and Figure 10, it is observed that, The NN training time isreduced in different NN model with TS approach when compared to differentmodel without TS approach. The lowest training time for different NN models are represented by boldface.

Table 8: Performance Evaluation by training time (in seconds)

NN/FTS Approaches	Without TS	With TS
MLFFNN	0.69	**0.46**
FTS-MLFFNN	1.90	**1.50**
NARXNN	0.52	**0.41**
FTS-NARXNN	21.00	**1.95**

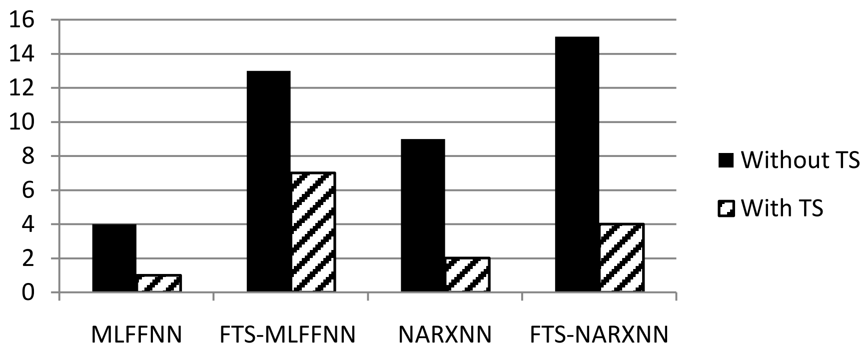

Figure 10: Performance evaluation by training time

NN convergence speed for different NN model without TS and different NN model with TS approach are represented in Table 9 and their corresponding graph is represented in Figure 11.

Table 9: Performance evaluation by convergence speed

NN/FTS Approaches	Without TS	With TS
MLFFNN	10	**7**
FTS-MLFFNN	18	**7**
NARXNN	13	**21**
FTS-NARXNN	5	**2**

From Table 9 and Figure 11, it is observed that, the convergence speed of FTS-NARXNN with TS approach is very fast, which is completed in 2 epochs; the convergence speed of NARXNN with TS approach is slow, which is completed in 21 epochs. The convergence speed of FTS-NARXNN withTS approach is very fast, which is completed in 5 epochs; the convergence speed of FTS-MLFFNN is very slow, which is completed in 18 epochs. Thefastest convergence speed for different NN model is represented by boldface.

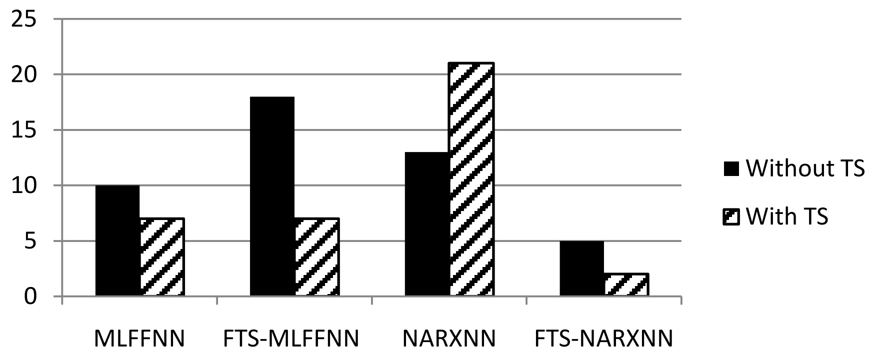

Figure 11: Performance evaluation by convergence speed

5. Conclusion

This study proposed a novel Neural Network (NN) and Fuzzy Time Series model with TS approach which strives to adjust the number of hidden neurons of a different Neural Network. It proposes to forecast one-step-ahead closing index of stock market and it is applied to real time series data set, BSE100. It has analyzed the neuron counts in the hidden layer, training time, convergence speed (epoch) and performance measure of SMAPE, POCID and TS in the training dataset, validation dataset and test dataset. After the analysis of various neural network and fuzzy time series models, finally different NNmodel without TS approach and different NN model with TS approach identified the neuron counts in the hidden layer for improving prediction accuracy and reduce over-fitting problem. The experimental result shows that FTS-NARXNN with TS approach outperformed other neural network model with respect to the performance measure SMAPE, TS and POA. 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 setand validation set; and minimum error value in SMAPE of validation set. The experimental result with BSE market of real datasets indicates that the proposed Tracking Signal approach can be an effective way in-order-to yieldaccurate prediction result.

The proposed NN model 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 toimprove the accuracy of stock market; in this study only two parameters are adjusted to reach the tracking signal limit in the training set and validation set. In future, other important parameters such as lag variable, learning rate and momentum, etc., will be used to reach the TS interval [-4, +4] in the training dataset and validation dataset.

References

[1]. Abbas Vahedi, 2012. The predicting stock price using articial neural network. Journal of Basic and Applied Scientific Research 2 (3), 2325-2328.

[2]. AdebiyiAyodele, A., Ayo Charles, K., Adebiyi Marion, O., &Otokiti Sunday O., 2012.Stockprice prediction using neural network with hybridized marketindicators.Journal of Emerging Trends in Computing and Information Sciences 3 (1), 1-9.

[3]. Ashok kumar, D. &Murugan, S., 2013. Performance analysis of indian stock market index using neural network time series models. IEEE Explore Digital Library ISBN : 978-1-4673-5843-9, 72-78.
[4]. Ashok kumar, D. &Murugan, S., 2014. Performance analysis of mlp neural network back propagation training algorithms for time series data. IEEE Explore Digital Library ISBN: 978-1-4799-2876-7, 114-119.
[5]. Balestrassi, P.P., Popova, E., Paiva, A.P., &MarangonLima J.W., 2009.Design of experiments on neuralnetworks training for nonlinear time seriesforecasting.Neurocomputing 72, 1160-1178.
[6]. BSE, 2012.http : \\www:bseindia:com\indices\indexarchivedata:aspx?expandable = 3.
[7]. Cecil Bozarath 2011, Measuring forecast accuracy approaches to forecastings a tutorial. http://scm.ncsu.edu/scm-article/measuring-forecast-accuracy-approaches-to-forecasting-a-turorial
[8]. ErkamGuresen, GulgunKayakutlu, Tugrul, &Daim, U., 2011. Using artificial neural network models in stock market index prediction. Expert Systems with Applications 38, 10389-10397.
[9]. Fildes, R., &Makridakis, 1995. The impact of empirical accuracy studies on time series analysis and forecasting. international statistical review 63, 289-308.
[10]. Greg Heath, 2012. http://in:mathworks:com=matlabcentral=newsreader =viewthread=331714.
[11]. Gronholdt, L. &Martensen, A., 2005.Analysing customer satisfaction data: A comparison of regression and articial neural networks. International Journal of Market Research, 121-130.
[12]. IebelingKaastra, A. & Milton Boyd, b., 1996. Designing a neural network for forecasting financial and economic time series. Neurocomputing, 215-236.
[13]. Jeff Heaton, 2008. Introduction to neural networks for java, 159-160.
[14]. Jose Maria, P., Junior, Guilherme, A., &Barreto., 2008. Long-term time series prediction with the narx network: An empirical evaluation. Neurocomputing 71, 3335-3343.
[15]. Kanas, A., 2001.A novel hybridization of artificial neural networks and arimamodels for time series forecasting. International Journal of Finance and Economics, 245-254.
[16]. Kumar, K. & Bhattacharya, S., 2006. A novel hybridization of artificial neural networks and arima models for time series forecasting, Applied Soft Computing 11, 2264-2275.
[17]. Lean Yu, Shouyang Wang, & Kin Keung Lai., 2005. Adaptive smoothing neural networks in foreign exchange rate forecasting. Computational Science ICCS 2005 5th International 3516, 523-530.
[18]. Lean Yu, Shouyang Wang, & Kin Keung Lai., 2007. Forecasting foreign exchange rates using ansvr-based neural network ensemble. Advances inBanking Technology and Management: Impacts of ICT and CRM, 121-130.
[19]. Mehdi Khashei& Mehdi Bijari, 2011.A novel hybridization of artificial neural networks and arima models for time series forecasting. Applied Soft Computing 11, 22264-2575.
[20]. Min Qi &Guoqiang Peter Zhang, 2001.An investigation of model selectioncriteria for neural network time series forecasting. European Journal of Operational Research 3, 666-680.
[21]. MustazurRahman, Md., Monirul Islam, Md. &Xin Yao, 2015.Layered ensemble architecture for time series forecasting.IEEE TRANSACTIONS ON CYBERNETICS.
[22]. Palmer, A., Montano, J.J. &Sese, A., 2006. Designing an articial neural network for forecasting tourism time series. Tourism Management, 781-790.
[23]. Ricardo de .A Araujo, 2009. An intelligent hybrid morphological-rank-linear method for financial time series prediction. Neurocomputing 72, 2507-2524.
[24]. Sivanandam, S.N. &Deepa S.N., 2008.Principles of softcomputing.WileyIndia First Edition, 79-80.
[25]. Suresh Kumar, K.K., 2012. Performance analysis of stock price prediction using artificial neural network.Global Journal ofComputer Science and Technology, 154-170.
[26]. Tiffany Hui-Kuang Yu & Kun-Huang Huarng, 2010. A neural network-basedfuzzy time series model to improve forecasting. Expert Systems with Applications 37, 3366-3372.
[27]. Timothy Master, 1993. Practical neural network recipes in c++. Morgan Kaufmann.
[28]. Wong, W.K., Min Xia, & Chu, W.C., 2010. Artificial neural network vs. lineardiscriminant analysis in credit ratings forecast: A comparative study of prediction performances. Review of Accounting & Finance 5.3, 216-227.