RESEARCHERID
THOMSON REUTERS

[Ashok Kumar *et al.*, 6(7): July, 2017]
IC™ Value: 3.00

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

PERFORMANCE ANALYSIS OF NARX NEURAL NETWORK BACK PROPAGATION ALGORITHM BY VARIOUS TRAINING FUNCTIONS WITH TRACKING SIGNAL APPROACH FOR TIME SERIES DATA

Ashok Kumar [1], Murugan [*2]

[1] Department of Computer Science, Government Arts College, Trichirappalli - 620 022, Tamilnadu, India.

[*2] Department of Computer Science, Alagappa Government Arts College, Karaikudi - 630 003, Tamilnadu, India.

DOI: 10.5281/zenodo.827506

ABSTRACT

This study proposed a novel Nonlinear Auto Regressive eXogenous Neural Network (NARXNN) with Tracking Signal (TS) approach and seeks to investigate the various training functions to forecast the closing index of the stock market. A novel approach strives to adjust the number of hidden neurons of a NARXNN model with different training functions. 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 at convergence speed and improves forecasting accuracy. In addition, the present approach has been tested with different training functions and identified the neuron counts in the hidden layer for every training function which leads to reduce over-fitting or under-fitting problem.

KEYWORDS: NARX Neural Network, Time Series Data, Training Functions, Closing Stock Index, Tracking Signal, Forecasting, Performance Analysis

I. 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 a developer on 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 between the past values of the forecast variable and therefore non-linear patterns cannot be captured by these models [1].

A number of neural network (NN) models [2-6] and hybrid models [7-9] 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 natural real world systems are non-linear in nature, so, NN are preferred over the traditional models. In [9] reported that the 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-linear functions. 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 [8]. (ii) Min Qi and Guoqiang Peter Zhang [10] investigated and reported that the in-sample (training set) model selection criteria cannot offer a reliable guide to out-of-sample (testing set) performance and there is no apparent linking between in-sample (trainining set) model fit and out-of-sample (test set) forecasting performance. NN model suffers due to under-fitting or over-fitting problems.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

Timothy Master [10] 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 [11] 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 [12-14]. The performance of various types of training algorithms [16-17] 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 [18] suggests that design often neural networks with different types of random initial weights to mitigate the occasional bad random start. Adebiyi Ayodele et al. [19] suggests that training a great number of ANN with different configurations and selects 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 [8, 9, 20]. 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 the training set) and test set [21]. The published research articles [7, 22, 23] reported that the optimum NN model selection is based on minimum forecasting error on validation set of any performance measure (SMAPE, NMSE, RMSE, etc.) and reports its corresponding results in test set to avoid over-fitting problem. Cecil Bozarth [24] 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. [5] 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.

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 of the 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 different weight are developed. After analyzing 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 the number of neurons in the hidden layer and training function used in the neural network. In addition, the present study is maiden effort 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 NARXNN architecture was created for forecasting the closing stock index of the BSE100 stock market. Secondly, the performance measure Tracking Signal (TS) is introduced to select the NARXNN model with different training functions which reduces the network complexity, training time; faster in convergence; improves better forecast accuracy; and reduce 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 and TS using NARXNN with TS approach and NARXNN without TS approach. Fourthly, the neuron counts in the hidden layer are identified with different training functions using NARXNN with TS approach and NARXNN without TS approach for BSE100 stock market. Fifthly, the performance of the various training functions using NARXNN with TS approach was compared with the performance of the various training functions using NARXNN without TS approach; the result indicates that the proposed NARXNN with TS approach outperformed NARXNN without TS approach. Sixthly, unlike the report of Timothy Master [11], the investigations of this study reveal that, the neuron counts in the hidden layer cannot be identified by some rule of thumb and it can be identified by constructing different NN with different parameter and selects the best one. Seventhly, unlike the report of Min Qi and Guoqiang Peter Zhang [10] the investigation of this study proves that the in-sample (training and validation set) model selection criteria can 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 NARXNN model, training functions, TS and performance measures which are used to assess the performance of the proposed approach; Section 3 describes the details of proposed NARXNN with TS approach and NARXNN without TS approach; Section 4 reports the experimental results attained by the NARXNN with TS approach and NARXNN without TS approach using real world financial time series BSE100 stock market dataset. Finally, this study is concluded in section 5.

II. BACKGROUND

Non-linear Auto Regressive eXogenous Neural Network (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 NARXNN 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 regression is formed only by the actual values of the system's output. In Parallel (P) Mode, estimated outputs are fed back and included in the output's regression. The P mode NARXNN architecture can be represented in Figure 1.

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 NN system, NARX P-type NN model [25] 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.

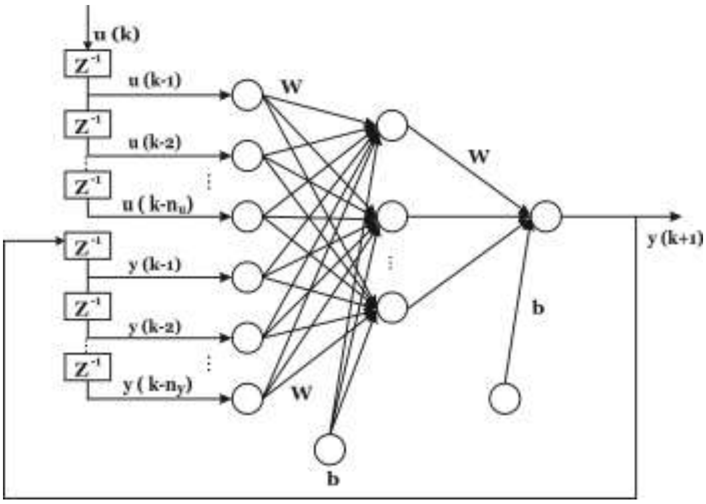

Figure 1. NARX Neural Network

Training Functions

Several different training functions are available for feed forward networks; some of the training functions in matlab and their associated training parameter [16, 17] are listed in Table 1. The accuracy of the training function depends on the number of data used in the training set, the number of biases and weights in the network and the error goal, etc.

Tracking Signal

The calculation of the TS [24] is 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 the Standard deviation and MAD in a normally distributed population is built as 1.25 MAD = 1 SD (standard deviation of the distribution).

Forecasting Performance

The forecasting performance is evaluated using the statistical measures, namely, symmetric mean absolute percentage error (SMAPE), Root Mean Square Error (RMSE) percentage of accuracy (POA). 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 the test set.

The global performance of a forecasting model is evaluated by the SMAPE [22] which is used in NN3, NN5 and NNGC1 forecasting competition. A smaller SMAPE value suggests the better forecasting accuracy. It can be expressed as

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

Table 1. Training functions and their parameters

Training Algorithm	Training function	Parameters
Levenberg-Marquardt	trainlm	mu, mu_dec, mu_inc, mu_max, epochs, show, goal, time, max_fail, min_grad, mu, mu_dec, mu_inc, mu_max and mem_reduc.
BFGS Quasi-Newton	trainbfg	epochs, show, goal, time, min_grad, srchFcn, scal_tol, alpha, beta, delta, gama, max_fail, low_lim, up_lim, minstep, maxstep and bmax.
Resilient Back propagation	trainrp	epochs, show, goal, time, min_grad, delta0, max_fail, delt_inc, delt_dec and deltamax.
Scaled Conjugate Gradient	trainscg	epochs, show, goal, time, min_grad, sigma, max_fail and lambda.
Conjugate Gradient with Powell/Beale Restarts	traincgb	epochs, show, goal, time, min_grad, srchFcn, max_fail, scal_tol, gama, alpha, beta, delta, up_lim, low_lim, maxstep, minstep and bmax.
Fletcher-Powell Conjugate Gradient	traincgf	epochs, show, goal, time, min_grad, srchFcn, max_fail, scal_tol, gama, alpha, beta, delta, up_lim, low_lim, maxstep, minstep and bmax.
Polak-Ribiére Conjugate Gradient	traincgp	epochs, show, goal, time, min_grad, srchFcn, max_fail, scal_tol, gama, alpha, beta, delta, up_lim, low_lim, maxstep, minstep, and bmax.
One Step Secant	trainoss	epochs, show, goal, time, min_grad, srchFcn, max_fail, scal_tol, gama, alpha, beta, delta, up_lim, low_lim, maxstep, minstep, and bmax.
Gradient Descent	traingd	epochs, show, goal, time, max_fail, min_grad and lr.
Gradient Descent with Adaptive Learning Rate	traingda	epochs, show, goal, time, min_grad, lr, mc, max_perf_inc, lr_dec, lr_inc and max_fail.
Gradient Descent with Momentum	traingdm	epochs, show, goal, time, max_fail, min_grad, lr and mc.
Variable Learning Rate	traingdx	epochs, show, goal, time, min_grad, lr, mc, max_perf_inc, lr_dec, max_fail and lr_inc.

Percentage of Accuracy (POA) [26] is one of the forecast bias measurements. If the ratio of POA is 100 percent, then it indicates the forecast is unbiased. The value of POA is 95 to 110% indicates the better forecast model. The value of POA is closer to 100% indicates the best forecast model.

$$\text{POA}=\frac{\sum_{t=1}^{n} f_t}{\sum_{t=1}^{n} y_t} X100 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Cecil bozrath [24] 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,

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

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

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

III. METHDOLOGY

Over fitting is one of the main issues in neural network modeling. In order to reduce the over fitting problem, this study proposes a novel NARXNN with TS approach which is used to forecast the closing index of the stock market. NARXNN receives closing stock index historical data and trains different network by using different random initial weight and different training functions with different neurons. TS measure is used to reject all NARXNN model which results in values outside the interval of [-4, +4] in the training set and validation set of different neural networks to reduce NARXNN structure which leads to avoid over-fitting or under-fitting problems.

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

Forecasting strategies have taken a step ahead of prediction in this study.

Let y_1, y_2, y_3 ….. y_t be a time series. As time t for t>=1, the next value y_{t+1} is predicted based on the observed realizations of y_t, y_{t-1}, y_{t-2} ….. y_1. The resultant network can be used for multi-step prediction by feeding the prediction back to the input of network recursively. The proposed approach of NARXNN is represented in Fig. 2.

In Fig. 2, X_i is the closing stock index vector, Y_i is the predicted closing stock index of the neural network model and N_j is neurons count in the hidden layer. For every NARXNN model, the presence of tracking signal interval [-4, +4] is verified in the training set and a validation set. If it is present, the model is considered as a feasible model otherwise the model is rejected. This process is repeated until the specified trial number (random initial weight) and maximum neuron count is reached. The implementation procedure of NARXNN with TS approach is represented in Algorithm 1, and explained further.

Algorithm 1. Nonlinear Auto Regressive eXogenous Neural Network with Tracking Signal Approach.
Input: Time series data for the closing stock index vector
Output:Time Series data for predicted closing stock index vector

---

1. Read the input and target pair from the data file and normalize or pre-process the data using mapminmax fuction.
2. Set the maximum number of neuron count MAX_NEURON in the hidden layer, maximum number of trial MAX_TRIAL (random initial weight) for random weight generation and SD (Standard Deviation) value for assigning TS limit.
3. FOR NEURON = 1 TO MAX_NEURON
4. FOR TRIAL = 1 TO MAX_TRIAL
5. Create NARX neural network architecture here; specify the input and target vector, the number of hidden layer, training function, transfer function used in the hidden and output layer.
6. Select the data division ratio using divide function and divide the data set into training data set, validation data set 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.
7. Train the neural network using train function.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

8. Simulate the neural network using sim function.
9. Denormalize or post-process the simulated neural network output data
10. Calculate the performance measure SMAPE, POA and TS for train, validation and test set using equation 1 - 3.
11. Record the result of neuron count, trial number, training time, epoch (convergence speed) and performance measure specified in step 10. It contains the performance of different NARXNN model without TS approach.
12. Verify the interval $[-\theta, +\theta]$ of Tracking Signal in the training set (TStrain) and validation set (TSvalidation) from step 11, 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 5.
13. Record the result of neuron count, trial number, training time, epoch (convergence speed) and performance measure specified in step 10. It contains the performance of different NARXNN with TS approach.
14. END for TRIAL
15. END for NEURON
16. From the step 11, select the optimum NARXNN model, which provides less error in SMAPE of a validation dataset using a NARXNN model without TS approach.
17. From the step 13, select the optimum NARXNN model, which provides less error in SMAPE of a validation dataset using a NARXNN model with TS approach.

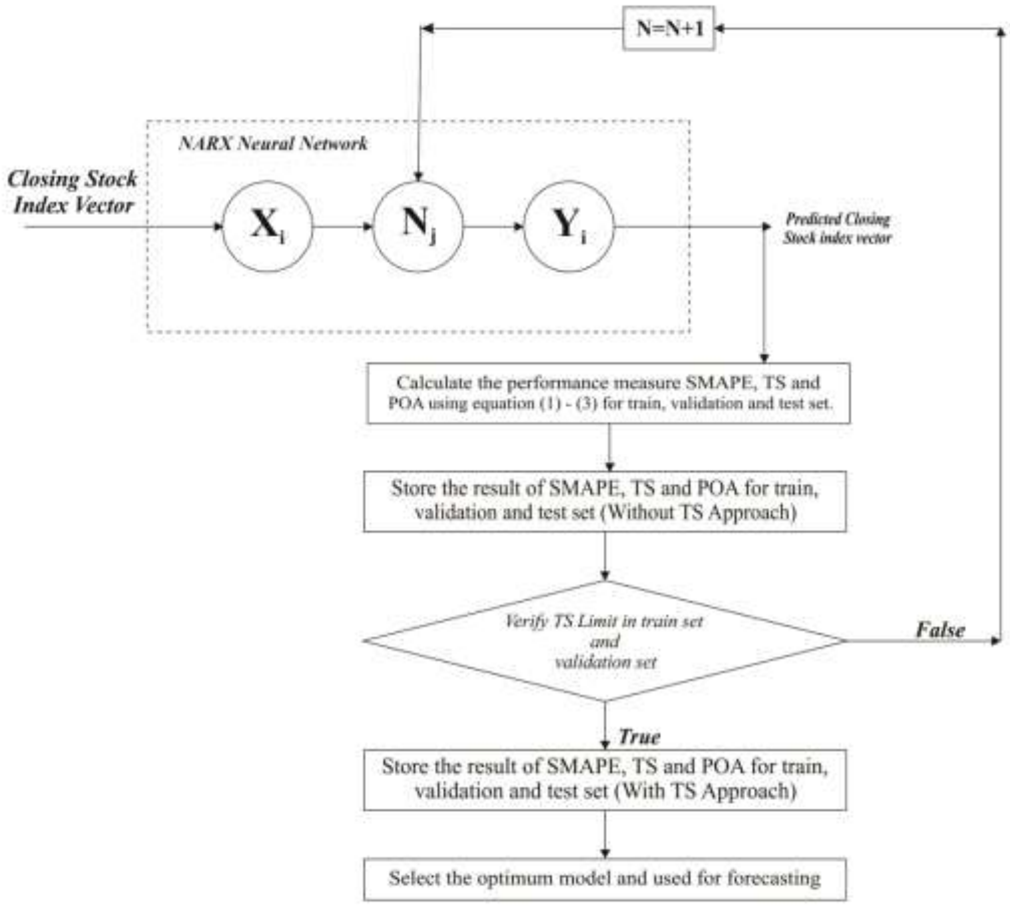

Fig. 2. NARXNN with TS Approach

Neural network training process is an iterative process. The input data and target data should be normalized or preprocessed before training the NN. In this process the input data is converted into -1 to +1. The preprocessed 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 in 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. NARXNN model with tan sigmoidal function in the hidden layer and linear function in the output layer is used after the division of data chosen. The tan sigmoidal function and linear function are defined in equation (5) and (6).

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

$$purelin(x) = x \qquad (6)$$

Levenberg Marquardt is used as a training function. After training the NN, simulate the NN and post process the simulated output. Finally, analyze the performance of the neural network using performance measure equation (1) - (4). The NARXNN training process is represented in step 1 to step 11 of Algorithm1 is known as NARXNN without TS approach and the remaining steps are known as NARXNN with TS approach. In NARXNN without TS approach, after post-processing the data, store the results of performance measure SMAPE, POA and TS of training set, validation set and test set for the different NARXNN model with different training functions. The optimum NARXNN model selection is based on minimum forecasting error on validation set of SMAPE. After selecting the optimum model using NARXNN without TS approach, still, there exists over-forecast or under-forecast in the training data set, validation data set and test data set. For example, the level of over-forecast and under-forecast in the training data set and validation dataset of BSE100 stock market with fifteen test cases (trial) of NARXNN model with neuron 7 which is represented in Fig. 3.

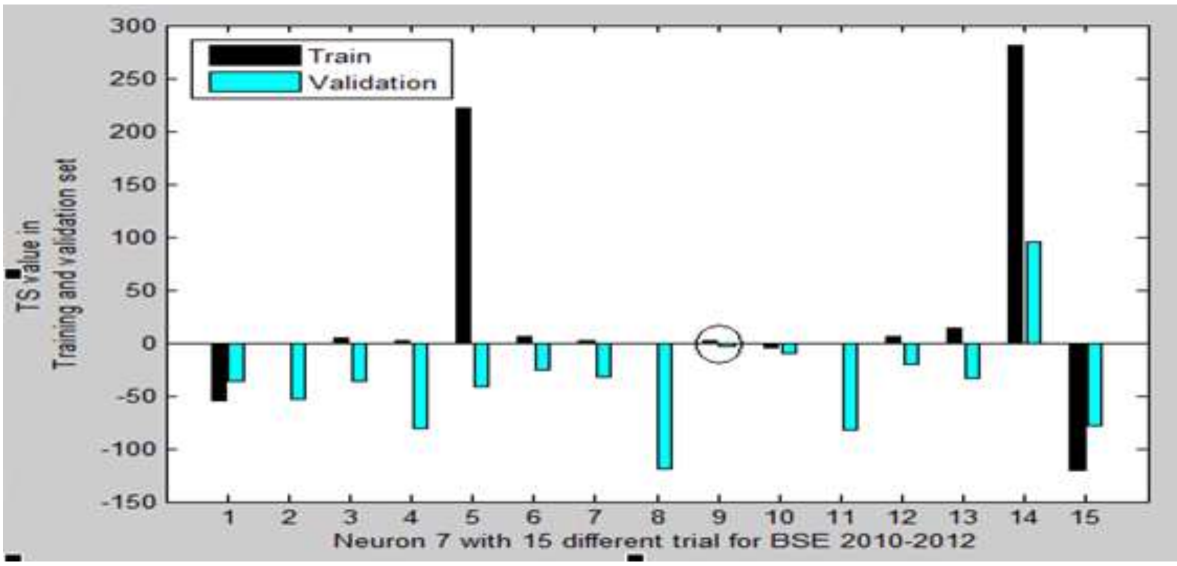

Fig. 3. Level of over-forecast or under-forecast in NARXNN with Neuron 7 (15 Trial)

Test case 9 is identified as the optimum NARXNN model by the TS measure marked with the circle in Fig. 3, which contains the TS interval value [-4, +4] in the training and validation set. Remaining test cases are rejected, which contains beyond the TS interval value [-4, +4] in the training and validation set. NARXNN with TS approach is used to assess the over-forecast or under-forecast in the training data set, validation data set. For every NARXNN model, check the TS interval $[-\theta,+\theta]$ in the training data set and validation data set, where $\theta=4$ and SD=3. It rejects all NARXNN models which results in values outside the interval of [-4, +4]; it accepts the NARXNN 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 NARXNN model selection is based on the interval value [-4, +4] in the training data set and validation data set which contains the minimum forecasting performance error in SMAPE (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 in a normally distributed

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

population is built as 1.25 MAD = 1 SD (standard deviation of the distribution). For this reason, this study selects the interval [-4,+4].

IV Experimental Results

In this section, first verify the excellence of the proposed approach, then it is applied to 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.

BSE100 Index

The effectiveness of the proposed NARXNN 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 [27]. For each NN created with different random initial weight for neuron 1 to neuron 10 with different training functions. The choice of random initial weight (trial) and maximum neuron count is selected by the user. In this study, random initial weight is 15 and the maximum neuron count is 10 for BSE100 stock market index. The data division ratio is 50/25/25.

The results of performance measure of 10 different models from 1-1-1 to 1-10-1 with different training functions were generated (Here 1-1-1, first part represents neuron counts in the input layer, second part represents neuron counts in the hidden layer and third part represents the neuron counts in output layer). Every NARXNN model contains fifteen different random initial weight generations. From the ten architectures with different training functions of different trial, some models are selected by the NARXNN with TS approach which contains the interval [-4, +4] in the tracking signal of the training data set and validation dataset; and some models are rejected by the NARXNN with TS approach which does not contains the interval [-4, +4] in the training data set and validation dataset of tracking signal. Rejection of the model and selection of model using NARXNN with TS approach is represented in Table 2.

Table 2. NN Model Rejection and Selection Using NARX with TS Approach

Training Functions	**Model Rejection**	**Model Selection**
trainscg	1-1-1, 1-2-1, 1-3-1, 1-4-1, 1-5-1, 1-6-1, 1-7-1, 1-9-1, 1-10-1	1-8-1
trainrp	1-3-1, 1-4-1, 1-6-1, 1-7-1, 1-8-1, 1-9-1, 1-10-1	1-1-1, 1-2-1, 1-5-1
trainoss	1-1-1, 1-2-1, 1-3-1, 1-4-1, 1-5-1, 1-6-1, 1-8-1, 1-9-1, 1-10-1	1-7-1
trainlm	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
traingdx	1-1-1, 1-2-1, 1-4-1, 1-5-1, 1-6-1, 1-7-1, 1-9-1, 1-10-1	1-3-1
traingdm	1-1-1, 1-2-1, 1-4-1, 1-5-1, 1-6-1, 1-7-1, 1-9-1, 1-10-1	1-3-1
traingda	1-1-1, 1-2-1, 1-3-1, 1-4-1, 1-5-1, 1-6-1, 1-7-1, 1-9-1, 1-10-1	1-8-1
traingd	1-1-1, 1-2-1, 1-3-1, 1-6-1, 1-8-1, 1-9-1, 1-10-1	1-4-1,1-5-1,1-7-1
traincgp	1-1-1, 1-2-1, 1-3-1, 1-4-1, 1-5-1, 1-6-1, 1-8-1, 1-9-1, 1-10-1	1-7-1
traincgf	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
traincgb	1-1-1, 1-2-1, 1-3-1, 1-4-1, 1-5-1, 1-6-1, 1-7-1, 1-8-1, 1-10-1	1-9-1
trainbfg	1-1-1, 1-2-1, 1-3-1, 1-4-1, 1-5-1, 1-6-1, 1-7-1, 1-10-1	1-8-1, 1-9-1

. The performance measure of SMAPE, TS and POA of training set, validation set and a test set of NARXNN with TS approach and without TS approach using different training functions are reported in Table 3.

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

Table 3. Performance Evaluation by SMAPE, TS and POA

Training Functions	Measure	NARX Without TS			NARX With TS		
		Train	Val	Test	Train	Val	Test
trainscg	SMAPE	0.97	1.19	0.80	0.85	1.11	0.70
	TS	94.40	-101.00	27.00	-0.22	-2.01	15.70
	POA	99.03	98.81	99.20	99.15	98.89	99.30
trainrp	SMAPE	1.49	1.16	1.09	0.84	1.10	0.68
	TS	310.00	51.10	133.00	-3.48	-1.80	4.80
	POA	98.51	98.84	98.91	99.16	98.90	99.32
trainoss	SMAPE	1.03	1.38	0.91	0.98	1.29	0.84
	TS	99.20	20.30	57.00	2.61	-3.69	13.30
	POA	98.97	98.62	99.09	99.02	98.71	99.16
trainlm	SMAPE	0.81	1.11	**0.65**	0.75	1.07	**0.64**
	TS	-0.38	-54.80	-4.88	1.73	-3.80	1.90
	POA	99.19	98.89	99.35	99.25	98.93	99.36
traingdx	SMAPE	1.88	1.41	1.46	1.01	1.34	0.87
	TS	-216.00	-41.90	-119.00	2.95	-0.89	2.23
	POA	98.12	98.59	98.54	98.99	98.66	99.13
traingdm	SMAPE	1.00	1.15	0.86	0.90	1.09	0.74
	TS	148.00	-24.50	62.80	3.24	-1.60	11.40
	POA	99.00	98.85	99.14	99.10	98.91	99.26
traingda	SMAPE	1.10	1.50	0.89	0.96	1.40	0.84
	TS	247.00	-45.40	11.70	3.39	1.40	6.34
	POA	98.90	98.50	99.11	99.04	98.60	99.16
traingd	SMAPE	1.21	1.17	0.93	0.88	1.10	0.71
	TS	4.27	-41.20	60.90	2.55	3.80	49.20
	POA	98.79	98.83	99.07	99.12	98.90	99.29
traincgp	SMAPE	0.98	1.19	0.77	0.81	1.12	0.69
	TS	-33.20	30.60	22.80	-3.55	-2.13	36.20
	POA	99.02	98.81	99.23	99.19	98.88	99.31
traincgf	SMAPE	3.55	2.15	2.14	2.23	1.62	1.42
	TS	21.40	125.00	-52.60	-3.24	3.68	-14.40
	POA	96.45	97.85	97.86	97.77	98.38	98.58
traincgb	SMAPE	0.85	1.14	0.74	0.87	1.09	0.70
	TS	15.80	-19.10	-0.90	3.47	2.21	11.90
	POA	99.15	98.86	99.26	99.13	98.91	99.30
trainbfg	SMAPE	0.87	1.13	0.71	0.85	1.10	0.68
	TS	49.90	-39.30	24.90	-2.91	-1.30	12.50
	POA	99.13	98.87	99.29	99.15	98.90	99.32

From Table 3, the performance measure SMAPE of the test set in every training function using NARX with TS approach is low when compared to the performance measure SMAPE of test set in every training function using NARX without TS approach. It indicates that the forecast error is low in NARX with TS approach than NARX without TS approach. The performance measure TS of test set in every training function using NARX with TS approach is extremely low when compared to the performance measure TS of test set in every training function

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

using NARX without TS approach. It indicates that the over-fitting or under-fitting problem can be reduced in NARX with TS approach than NARX without TS approach. The performance measure POA of test set in every training function using NARX with TS approach is high when compared to the performance measure POA of test set in every training function using NARX without TS approach. It indicates that the forecasting accuracy is high in NARX with TS approach than NARX without TS approach. The NARX with TS approach outperformed NARX without TS approach with respect to SMAPE, TS and POA measure.

Fildes and Makridakis [29] 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 the selection of NN model should be based on minimum model fitting errors. From Table 3, it is observed that there is a close relationship between train and test dataset in different training functions of NARXNN with TS approach when compared to different training functions of NARXNN without TS approach. According to the Fildes and Makridakis statement, the NARXNN with TS approach has a close relationship between training set and a test set of performance measurement SMAPE.

From Table 3, it is observed that, the difference between the performance measure SMAPE of the training data set and test dataset in NARXNN with TS approach is slightly close to each other when compared to the performance measure SMAPE of the training data set and test dataset in NARXNN without TS approach. For example the train and test set of SMAPE in the NARXNN without TS approach of training function trainscg is 0.97 and 0.80 respectively; whereas the train and test set of SMAPE in the NARXNN with TS approach of training function trainscg is 0.85 and 0.70 respectively. It indicates that there is no close relationship between the train and test set of SMAPE in NARXNN without TS approach when compared to NARXNN with TS approach. This is the main purpose of using the tracking signal in this study. Unlike the report of Min Qi and Guoqiang Peter Zhang [10], this closeness of training and testing performance measure of SMAPE indicates that the in-sample (training dataset) model selection criteria can provide a reliable guide to out-of-sample (testing dataset) performance and an apparent connection between in-sample model fit and out-of-sample model forecasting performance. It happens due to the model selection based on tracking signal.

From Table 3, it is clearly observed that the result of performance measure TS of different training functions using NARXNN without TS approach is severely suffered by either over-fitting or under-fitting with respect to TS in training data set and validation data set, whereas, the result of performance measure TS of different training functions using NARXNN with TS approach do not suffer due to under-fitting or over-fitting with respect to TS in training data set and validation data set. For example, the train and test set of TS in the NARXNN without TS approach of training functions trainscg is 94.40 and -101.00 respectively; whereas the train and test set of TS in the NARXNN with TS approach of training function trainscg is -0.22 and 2.01 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 the train and test set of TS in NARXNN without TS approach when compared to NARXNN with TS approach.

Table 4. Performance Evaluation by SMAPE in Test Set

Training function	SMAPE	
	Without TS	**With TS**
trainscg	0.80	**0.70**
trainrp	1.09	**0.68**
trainoss	0.91	**0.84**
trainlm	0.65	**0.64**
traingdx	1.46	**0.87**
traingdm	0.86	**0.74**
traingda	0.89	**0.84**
traingd	0.93	**0.71**
traincgp	0.77	**0.69**
traincgf	2.14	**1.42**
traincgb	0.74	**0.70**
trainbfg	0.71	**0.68**

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

Levenberg-Marquardt (trainlm) training function is outperforming other training functions using NARX without TS approach and NARX with TS approach. The main difference is NARX without TS approach suffers due to over-fitting or under-fitting problem; whereas the NARX with TS approach reduces the over-fitting or under-fitting problem.

The test set of performance measurement SMAPE of different training functions using NARXNN without TS and NARXNN with TS approach is represented in Table 4. The Lowest value in SMAPE represents (boldface) best prediction result.

From Table 4, it is observed that the performance measure SMAPE of different training functions using NARXNN with TS approach is low when compared to the performance measure SMAPE of all other training functions using NARXNN without TS approach.

The test set of performance measuremnt TS of different training functions using NARXNN without TS and NARXNN with TS approach is represented in Table 5.

Table 5 Performance Evaluation by TS in Test Set

Training function	**TS**	
	Without TS	**With TS**
trainscg	27.00	**25.70**
trainrp	133.00	**4.80**
trainoss	57.00	**13.30**
trainlm	-14.88	**11.90**
traingdx	-119.00	**2.23**
traingdm	62.80	**47.40**
traingda	11.70	**6.34**
traingd	60.90	**49.20**
traincgp	22.80	**16.20**
traincgf	-52.60	**-14.40**
traincgb	-10.90	**1.90**
trainbfg	24.90	**12.50**

From Table 5, it is clearly observed that the value of test set of TS in different training functions using NARXNN with TS approach is low when compared to the value of test set of TS in different training functions using NARXNN without TS approach. It indicates that the level of under-fitting or over-fitting is reduced in NARXNN with TS approach when compared to NARXNN without TS approach.

Table 6 Performance Evaluation by POA inTtest Set

Training function	**POA**	
	Without TS	**With TS**
trainscg	99.20	**99.30**
trainrp	98.91	**99.32**
trainoss	99.09	**99.16**
trainlm	99.35	**99.36**
traingdx	98.54	**99.13**
traingdm	99.14	**99.26**
traingda	99.11	**99.16**
traingd	99.07	**99.29**
traincgp	99.23	**99.31**
traincgf	97.86	**98.58**
traincgb	99.26	**99.30**

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

trainbfg	99.29	**99.32**

Particularly, the test set of training function traingdx is very low, which is 2.23 in NARXNN with TS approach when compared to the test set of training function traingdx, which is -119 in NARXNN without TS approach. If the model selection is based on TS measures, then it can reduce the over-fit or under-fit in the test data set.

The test set of performance measuremnt POA of different training functions using NARXNN without TS and NARXNN with TS approach is represented in Table 6. Best results are represented by boldface. The value of POA is 95 to 110% indicates the better prediction model.

From Table 6, it is clearly observed that the value of test set of POA in different training functions using NARXNN with TS approach is high when compared to the value of test set of POA in different training functions using NARXNN without TS approach. It indicates that the percentage of accuracy is high in NARXNN with TS when compared to NARXNN without TS approach.

Neuron counts in the hidden layer are identified for different training functions using NARXNN without TS and NARXNN with TS approach is represented in Table 7.

Table 7 Performance Evaluation by Neuron Count

Training function	**Neuron**	
	Without TS	**With TS**
trainscg	9	8
trainrp	6	5
trainoss	10	7
trainlm	9	**2**
traingdx	7	3
traingdm	4	3
traingda	10	8
traingd	9	4
traincgp	8	7
traincgf	6	2
traincgb	10	9
trainbfg	10	9

From Table 7, the NN complexity is reduced in all training functions using NARXNN with TS approach when compared to NARXNN without TS approach. The best NN model for every training function is represented by boldface. According to the Timothy Master's [11] Pyramid rule, for a three layer NN with m output and n input neurons, the hidden layer may have a square root (m*n) neurons.

Table 8 Performance evaluation by training Time

Training function	Training Time	
	Without TS	With TS
trainscg	0.31	**0.24**
trainrp	0.28	0.29
trainoss	0.33	0.32
trainlm	0.52	0.41
traingdx	0.30	**0.24**
traingdm	0.31	0.38
traingda	0.51	0.37
traingd	0.60	0.51
traincgp	0.30	0.25
traincgf	0.32	0.25

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

traincgb	0.33	0.26
trainbfg	0.37	0.34

In this study, m=1 and n=1, square root(1*1) = 1. Unlike the report of the Master, the investigation of this study reported that, neuron counts in the hidden layer cannot be determined by some formulas or rule of thumb and it can be identified by modifying various neural network parameters. This study tries to achieve best prediction results by modifying two important parameter neuron counts in the hidden layer and training functions.

NN training time for different training functions using NARXNN without TS and NARXNN with TS approach is represented in Table 8.

From Table 8, it is observed that, the NN training time is reduced in all training functions using NARXNN with TS approach except the training function trainrp and traingdm when compared to NARXNN without TS approach. The NN training time is reduced in trainrp which has taken 0.28 seconds; and the NN training time is increased in traingd which has taken 60 seconds using NARXNN without TS approach. The NN training time is reduced in trainscg and traingdx which has taken 0.24 seconds; the NN training time is increased in traingd which has taken 0.51 seconds using NARXNN with TS approach. The lowest training time for every training function is represented by boldface.

NN convergence speed for different training functions using NARXNN without TS and NARXNN with TS approach is represented in Table 9.

Table 9 Performance evaluation by convergence speed

Training function	**Convergence Speed**	
	Without TS	**With TS**
trainscg	10	**7**
trainrp	5	9
trainoss	9	**3**
trainlm	13	**11**
traingdx	9	14
traingdm	18	**13**
traingda	109	**43**
traingd	157	**110**
traincgp	3	5
traincgf	3	9
traincgb	6	**4**
trainbfg	6	**5**

From Table 9, it is observed that, the convergence speed of NN with traincgp and traincgf is very fast, which is completed in 3 epochs; the convergence speed of NN with traingd is very slow, which is completed in 157 epochs using NARXNN without TS approach. The convergence speed of NN with trainoss is very fast, which is completed in 3 epochs; the convergence speed of NN with traingdm is very slow, which is completed in 110 epochs using NARXNN with TS approach. The fastest convergence speed for every training function is represented by boldface.

IV. CONCLUSION

This study proposed a novel NARXNN with TS approach which strives to adjust the number of hidden neurons of a NARX Neural Network (NARXNN) with different training functions. 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 and TS in the training data set, validation data set and test dataset. After the analysis of various NARXNN models, finally NARXNN without TS approach and NARXNN 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 NARXNN with TS approach outperformed NARXNN without TS approach with respect to the performance measure SMAPE, TS and POA. This study recommends to increase the prediction accuracy and reduce over-fitting or under-fitting problem, the best forecasting model is selected by the presence of tracking signal interval [-4, +4] in the training set and validation set; and minimum error value in SMAPE of validation set. The experimental result with BSE market of real data sets indicates that the proposed NARXNN with TS approach can be an effective way in-order-to yield accurate prediction result.

The proposed NARXNN with TS approach can be used as an alternative forecasting tool for time series forecasting. In this study, the only single variable is taken for prediction; In the future, multi variables will be taken for prediction to improve the accuracy of the stock market; 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., may be used to reach the TS interval [-4, +4] in the training data set and validation data set.

V. REFERENCES

[1] W.K.Wong, M. Xia and W.C. Chu, "Adaptive neural network model for time-series forecasting", European Journal of Operational Research, Vol. 207, No. 2, pp. 807–816, 2010.
[2] K. Kumar and S. Bhattacharya "A novel hybridization of artificial neural networks and ARIMA models for time series forecasting", Applied Soft Computing, Vol. 11, pp.2264-2275, 2006.
[3] A. Kanas, "Neural network linear forecasts for stock returns", International Journal of Finance and Economics, Vol.6, No.3, pp. 245–254, 2001.
[4] L. Gronholdt and A. Martensen, "Analysing customer satisfaction data: A comparison of regression and artificial neural networks", International Journal of Market Research. Vol. 47, No.2, pp. 121–130, 2005.
[5] Y. Lean, W. Shouyang and K.L. Kin, "Adaptive Smoothing Neural Networks in Foreign Exchange Rate Forecasting", Computational Science ICCS 2005 5th International, Vol.3516, pp.523-530, 2005.
[6] A. Palmer, J.J.Montano and A.Sese, "Designing an artificial neural network for forecasting tourism time series", Tourism Management, Vol. 27, No. 5, pp. 781–790, 2006.
[7] A. Ricardo de Araujo, A.E. Tiago and Ferreira, "An intelligent hybrid morphological-rank-linear method for financial time series prediction", Neurocomputing, Vol. 72, No. 10, pp. 2507-2524, 2009.
[8] M. Khashei and M. Bijari, "A novel hybridization of artificial neural networks and ARIMA models for time series forecasting", Applied Soft Computing, Vol. 11, No. 2, ,pp. 2264-2275, 2011.
[9] T.H.K.Yu and K.H. Huarng, "A neural network-based fuzzy time series model to improve forecasting", Expert Systems with Applications, Vol. 37, No.4, pp. 3366-3372, 2010.
[10] M. Qi and G.P. Zhang, "An investigation of model selection criteria for neural network time series forecasting", European Journal of Operational Research. Vol. 132, No.3, pp. 666-80, 2001.
[11] T. Masters, Practical neural network recipes in C++, Publication Company: Morgan Kaufmann, 1993.
[12] J Heaton, "Introduction to Neural Networks for Java. Publication Company", Heaton Research, Inc, Second Edition, pp. 159-160, 2008.
[13] 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", Neuro computing, Vol. 72, pp. 1160-1178, 2009.
[14] A. Vahedi, "The Predicting Stock Price using Artificial Neural Network", Journal of Basic and Applied Scientific Research, Vol. 2, No. 3, pp. 2325-2328, 2012.
[15] A. Iebeling Kaastra and B. Milton Boyd, "Designing a neural network for forecasting financial and economic time series", Neurocomputing, Vol. 10, No.3, pp. 215-236, 1996.
[16] D. Ashok Kumar and S. Murugan, "Performance analysis of Indian Stock Market Index using neural network time series model", pp.72–78, 2013. International Conference on Pattern Recognition, Informatics and Mobile Engineering (PRIME 2013), Periyar University, Salem, India, 21st and 22nd February 2013.
[17] D. Ashok Kumar and S. Murugan, "Performance Analysis of MLPFF Neural Network Back propagation Training Algorithms for Time Series Data", pp.114–119, 2014. World Congress on Computing and Communication Technologies (WCCCT 2014), St. Joseph's College (Autonomous), Tiruchirappalli, India, February 27,28 and March 1, 2014.
[18] Greg Heath "Problem about getting optimum output in Neural Network MATLAB 2012a", [online] http://in.mathworks.com/matlabcentral/newsreader/view thread/331714 (Accessed 4 December, 2016).
[19] A. AdebiyiAyodele, 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, No. 1, pp. 1-9, 2012.
[20] E. Guresen, G. Kayakutlu, Tugrul and U. Daim, "Using artificial neural network models in stock market index prediction", Expert Systems with Applications, Vol. 38, No. 8, pp. 10389–10397, 2012.

[21] L.Yu, S.Wang and K.K.Lai, "Forecasting foreign exchange rates using an SVR-based neural network ensemble", Advances in Banking Technology and Management: Impacts of ICT and CRM, India, pp. 261-277, 2007.

[22] M. Rahman, M. Islam and X. Yao, "Layered Ensemble Architecture for Time Series Forecasting", IEEE Transaction Cybernetics, Vol. 46, No. 1, pp. 270-283, 2016.

[23] K.K. Suresh Kumar, Performance analysis of stock price prediction using artificial neural network, Global Journal of Computer Science and Technology, Vol. 12, No.1, pp. 154-170, 2012.

[24] Cecil Bozarth, "Measuring Forecast Accuracy Approaches to Forecasting A Tutorial", http://scm.ncsu.edu/scm-articles/article/measuring-forecast-accuracy. Date accessed: 08/11/2016.

[25] Jose Maria P. Junior and Guilherme A. Barreto, "Narx reference: Long-Term Time Series Prediction with the NARX Network: An Empirical Evaluation", Neurocomputing, Vol . 71, pp. 3335-3343, 2008.

[26] R. Adhikari and R.K. Agrawal, "An Introductory Study on Time Series Modeling and Forecasting", LAP Lambert Academic Publishing, Germany, pp.1-67, 2007.

[27] Bombay Stock Exchange India, http://www.bseindia.com/indices/ Index Archive Data.aspx. Date accessed: 08/11/2016.

[28] R. Fildes and S. Makridakis, "The impact of empirical accuracy studies on time series analysis and forecasting international statistical review", Vol. 63, No.3, pp. 289-308, 1995.

CITE AN ARTICLE

Kumar, Ashok, and Murugan. "PERFORMANCE ANALYSIS OF NARX NEURAL NETWORK BACK PROPAGATION ALGORITHM BY VARIOUS TRAINING FUNCTIONS WITH TRACKING SIGNAL APPROACH FOR TIME SERIES DATA ." *INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY* 6.7 (2017): 312-26. Web. 15 July 2017.