(IJTBM) 2018, Vol. No.8, Issue No. IV, Oct-Dec e-ISSN: 2231-6868, p-ISSN: 2454-468X

DEVELOPING A NEURO-GENETIC MODEL TO EFFECTIVELY PREDICT STOCK PRICES IN BSE SENSEX, 2018

Drishti Arora

ABSTRACT

In this research work, we are taking into consideration the Stock price prediction of BSE SENSEX information using neuro-genetic approach. In this research we are taking BSE SENSEX dataset as input parameter and forecast the output of few days. in our research various algorithm has been implemented and result is being compared. The best neural system model is further being exposed to synaptic weight advancement utilizing the Genetic Algorithm. The different models are then exposed to testing over a time of 15 days, to acquire the most exact model The proposed framework applies variations of Back Propagation (BP) learning calculation on a Multi-Layer Perceptron arrange (MLP) which is prepared to utilize four years' BSE Sensex information. The presentation of the system is estimated by Normalized Mean Squared Error (NMSE). It is seen that strong back spread calculation with log sigmoid actuation work gives the most reduced NMSE of 0.003745. The exploration work likewise utilizes a Genetic Algorithm (GA) for weight advancement. BP experiences the risk of stalling out in neighbourhood minima. This is maintained a strategic distance from by utilizing GA to choose the best synaptic loads and hub edges at first and afterward continuing with the preparation of MLP utilizing BP. It is seen that this half breed model gives improved outcomes. So as to validate the model proposed, tests are first directed without utilizing GA. The aftereffects of this general BP MLP model are then contrasted and that of the GA-BP MLP model and dissected. NMSE for the GA-BP MLP model is 0.003092121.

1. INTRODUCTION

Stock market prediction is an important task for stock traders, applied researchers and stock investors. Various methods have been devised for the same. They are fundamental analysis, technical analysis, traditional time series forecasting, and Artificial Neural Networks. Technical analysis uses historical data such as stock prices and volume information for prediction. It is highly subjective in nature and has been shown to be statistically invalid. Nevertheless, it is preferred by many stock traders. Traditional time series techniques require a large amount of high-quality data and are suited for short-term forecasting only1.

There are some distinguishing features that make Artificial Neural Network (ANN) a preferred technique for prediction as compared to other models of forecasting. ANN gaining increasing acceptance in the business area due to its ability to learn and detect relationships among nonlinear variables. Also, it allows deeper analysis of large sets of data, especially those that have the tendency to fluctuate within a short of period of time2.

A neural network system is a highly parallel circulated processor made up of straightforward preparing units that has a characteristic penchant for putting away experimental information and making it accessible for use3. ANNs are fit for taking care of non-straight frameworks viably. Securities exchange is a mayhem system4. Tumult is a non-direct deterministic framework which just seems arbitrary as a result of its unpredictable variances. These frameworks are dynamic, an intermittent, confused and hard to manage ordinary systematic techniques. The neural systems are viable in adapting such non-direct fuzzy frameworks since they make not very many presumptions about the practical type of the basic unique conditions and their underlying conditions. However, the majority of the non-straight factual procedures require that the non-direct model be indicated before the estimation of the parameters and by it happens that the pre-determined non-straight models may neglect to watch the basic highlights of the critical framework under investigation. ANN has developed out to be a superior strategy in catching the basic connection between a stock's exhibition and its determinant factors more precisely than numerous other factual techniques5–8.

Genetic algorithms, based on Darwinian survival of the fittest, are self-adaptive globally optimistic search procedures9–11. Genetic Algorithm (GA) is an iterative algorithm that is parallel and global. It is also a soft computing method. GAs are good at taking larger, potentially huge search spaces and navigating them looking for optimal combinations of things and solutions which we might not find in a life time11.

2. BACKGROUND

Prediction of stock prices is a classic problem12. A number of researchers are exploring the different neural network architectures for stock market related applications. In13 explains the use of neural network to model financial and economic time series. In14 used ANN for stock market trend prediction. In15 used genetic algorithm with back propagation for achieving optimization. In16 applied neural network with genetic algorithm to the stock exchange of Singapore to predict the market direction. In17 used feed forward neural network architecture for stock market trend prediction. A comprehensive review of neural network concepts and principals can be found in18,19. An overview of financial and investment applications of neural networks can be found in20,21.

3. ARTIFICIAL NEURAL NETWORK

Artificial Neural Network is a soft computing technique, which is used to solve problems that cannot be solved by the conventional rule-based programming algorithms. An important aspect of this method is training the neurons to enrich it with knowledge so as to guide it in its analysis, predictions or classifications.

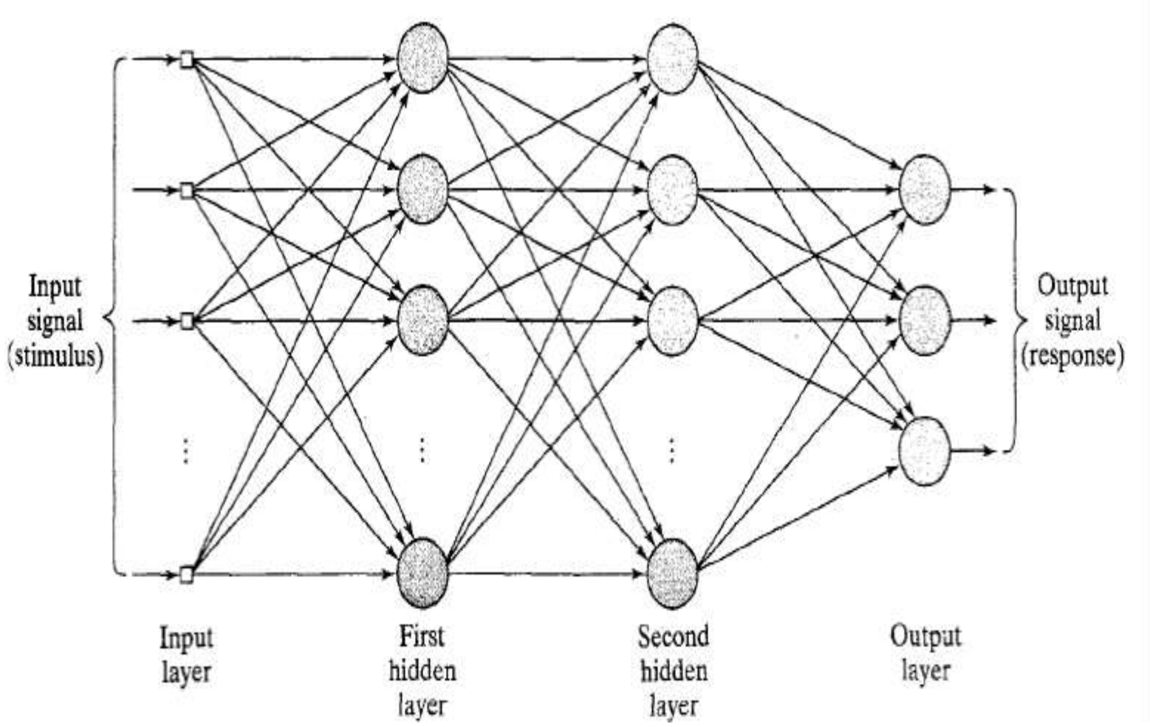

Figure 1. Neural network architecture.

Input neurons are those that act as sensors, and receive the input information from the environment, or user. Hidden neurons are those that process the input information received, depending on their weights. Output neurons are those that act as actuators and receive the processed value from the hidden neurons and produce output for the user. There can be more than one hidden layer.

4. GENETIC ALGORITHM

Genetic Algorithm is the mimicry of the natural selection process of biological evolution. The algorithm repeatedly modifies a population of individual solutions. The input parameters are considered as population, from which the Genetic Algorithm randomly selects data, which act as parents. The parents are modified to produce children for the next generation. Over successive generations, the population evolves towards optimized solutions. Genetic Algorithm can be used to solve both constrained and unconstrained problems of optimization. Thus, discontinuous, non-differentiable, stochastic or highly non-linear problems functions can also be optimized using Genetic Algorithm.

By using Artificial Neural Network for stock prediction, the Neural Network is trained to establish all the patterns within the input dataset and hence predict future values, considering past uncertainties as well. Furthermore, the Neural Network architecture is improvised using Genetic Algorithm to obtain optimized weights, for a faster and more accurate training of the Neural Network.

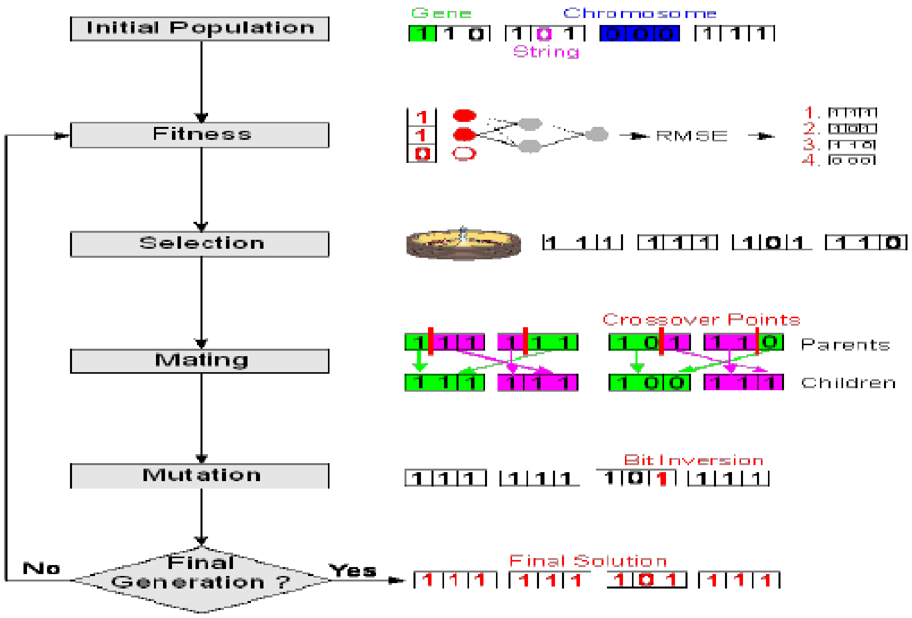

Figure 2. Genetic algorithm.

BSE Sensex historical dataset for 4 years is collected from the official website of BSE SENSEX. Input parameters are collected from 24th February 2011 to 7th April 2015. With the help of this dataset, the future day closing price is determined as the output. The dataset comprises of:

Open: • The price at which a security first trades upon the opening of an exchange on a given trading day. A security's opening price is an important marker for that day's trading activity, especially for those interested in measuring short-term results, such as day traders. Additionally, securities, which experience very large intra-day gains and losses, will have those swings measured relative to their opening price for the day.

Close: • The final price at which a security is traded on a given trading day. The closing price represents the most up-to-date valuation of a security until trading commences again on the next trading day.

Low: • The lowest price to which the stock value drops in the course of the day. It determines the maximum dip in the stock price.

High: • The highest price to which the stock value rises in the course of the day. It determines the highest peak in the stock price.

Volume: • The number of stocks that have been traded during the day.

MATLAB is used to create network architecture for the prediction, train it and then predict future values accordingly.

5. DATA PRE-PROCESSING

Data pre-processing is an essential part of prediction models in Neural Network. It is essential that normalization of data is carried out prior to subjecting the data into the learning phase. The information pre-processing of the info variable of the neural system model encourages de-inclining of the information and feature fundamental relationship, in order to encourage the appropriate system learning process.

The following linear scaling function is used in order to scale down the input values to an interval between

[0, 1]:

$$X^{*}_{k,n} = \frac{X_{k,n} - \min(X_k)}{\max(X_k) - \min(X_k)}$$

Where X_k is the data series of each type of input parameters taken separately and $X_{k,n}$ is the nth day's input parameter of that type. 6. Training Process

In the training process, the pre-processed data is fed into the Neural Network architecture. This phase is called the learning phase. The neural network uses weights and biases to update the input according to the desired output. The patterns in the data are established and the neural network is trained to be able to predict recurring instances of such patterns.

For this model, a 5-4-1 Neural Network architecture has been adopted. It comprises of 5 inputs, namely Open, Close, High, Low and Volume of the Sensex. There is one hidden layer, which comprises of 4 hidden neurons and there is one output neuron that gives closing Sensex value.

5.1 Training Algorithm

The following training algorithms have been considered for network training:

5.1.1 Train GDA

It's a backpropagation gradient descent training method with adaptive learning, to prevent over-learning or under-learning in a network. This method expands the learning rate, yet just to the degree that the system can learn without enormous mistake increments. Accordingly, a close ideal learning rate is acquired for the nearby territory. At the point when a bigger learning rate could bring about stable learning, the learning rate is expanded. At the point when the learning rate is too high to even think about guaranteeing a diminishing in blunder, it gets diminished until stable learning resumes23.

Backpropagation is used to calculate derivatives of performance dperf with respect to the weight and bias variables X. Each variable is adjusted according to gradient descent

5.1.2 TrainGDX

Traingdx is an extension of traingda. Traingdx is a network training function that updates weight and bias values according to gradient descent momentum and an adaptive learning rate. It has an additional parameter, momentum constant (mc), which helps in setting the momentum of the learning rate. By default, the value of mc is 0.9.

5.1.3 TrainRP

In sigmoid function (additionally called 'squashing capacities'), interminable information go is compacted into a limited yield extend. At the point when steepest plummet strategy is utilized to prepare a multilayer connect with sigmoid capacities, the inclination can have a little size; and along these lines, cause little changes in the loads and predispositions, despite the fact that the loads and predispositions are a long way from their ideal values23.

5.2 Activation Functions

The following activation functions have been considered for the layers:

5.2.1 Tan-Sigmoid Activation Function

Tan-sigmoid function generates values between -1 and +1. Mathematically, it can be written as:

$$\sigma(t) = \tanh(t) = \frac{e^{t} - e^{-t}}{e^{t} + e^{-t}}$$

5.2.2 Log-Sigmoid Activation Function

Log-sigmoid function generates values between 0 and +1 (not inclusive). Mathematically, it can be written as:

$$\sigma(t) = \frac{1}{1 + e^{-\dagger t}}$$

5.2.3 Pure-Linear Activation Function

In the event that direct yield neurons are utilized in the last layer of the multilayer organize, the system yields can take on any esteem, in contrast to that of sigmoid capacities. Scientifically, it can be written as: $f(x) = x$

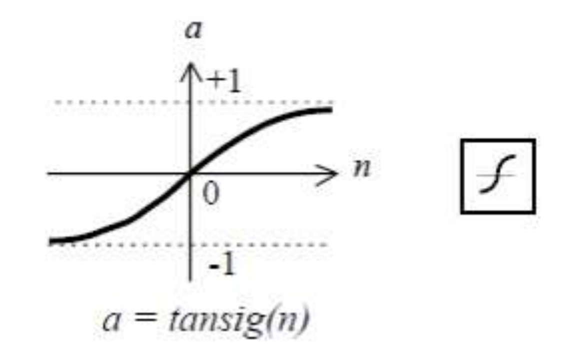

Fig 3: **Tan Sigmoid**

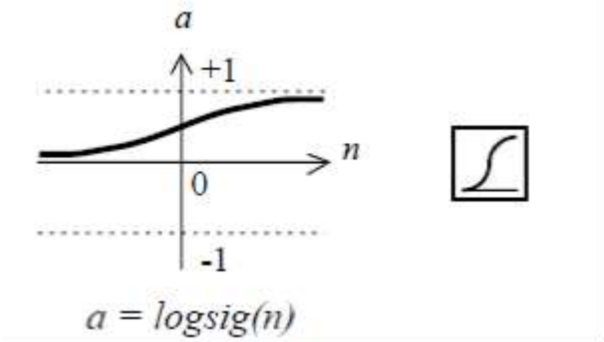

Fig 4: **Log Sigmoid**

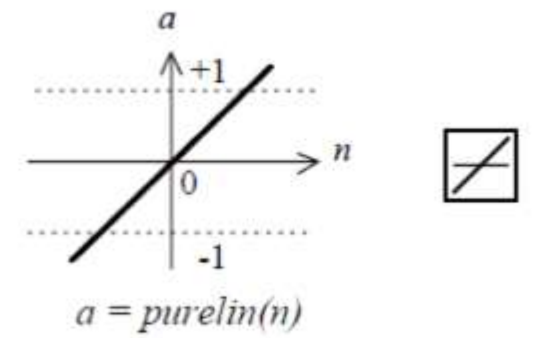

Fig 5: **Pure- liner**

6. EXPERIMENTS AND RESULTS

The yield got from preparing the system is contrasted and the real qualities. The presentation of the system is estimated by NMSE (Normalized Mean Squared Error)24. Scientifically,

$$\mathrm{NMSE} = \frac{\sum_{t=1}^{N1} (P_i - O_i)^2}{\sum_{t=1}^{N1} (P_i - \overline{P}_i)^2}$$

Where Pt is the genuine estimation of the pre-prepared information arrangement, Ot is the watched worth or the anticipated an incentive for that day shutting costs of the file and Pt is the mean of the real qualities.

When the presentation estimation produces an ideal outcome for specific system design and preparing calculation mix, the system is utilized for future worth expectation. The network is simulated to predict the (n+1)th day's closing index value, which is then compared to the actual obtained value, the next day. Hence, prediction is done for a particular period of time to test the consistency of the model. The accuracy of the predicted values is what will determine the reliability of this model.

The results obtained from various models are compared and analysed. The model which provides the lowest Normalized Mean Squared Error is the most accurate model. This model is chosen over others for Stock Market prediction.

NMSE for traingda algorithm using tansig as activation function = 0.020845

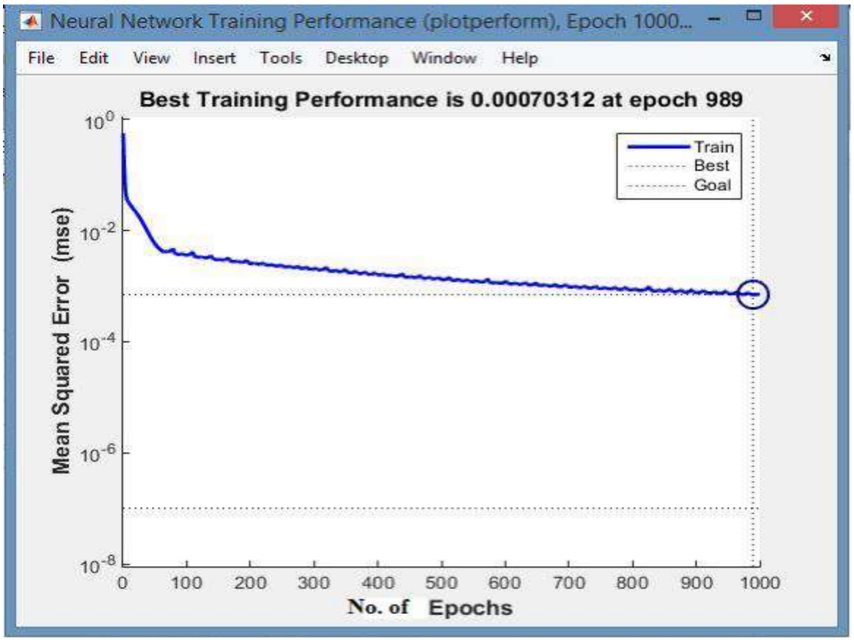

Figure 6. Training performance graph for tansig, traingda.

As seen in the training graph above, there's a steep decrease in the Mean Squared Error (mse) at the 0th epoch itself. After that, there's a gradual decline in the slope till it reaches the given epoch limit. Also, the training is not smooth as it is marked by crests and troughs throughout the training. Thus, even though it achieves an NMSE of 0.020845, the network is still not trained completely in compliance with the given data. This uncertainty is the main drawback of using traingda algorithm with tan sigmoid as the activation function.

NMSE for traingdx algorithm using tansig as activation function = 0.006599

As can be seen from the performance graph, traingdx algorithm, when using tan sigmoid as the activation function, provides a smoother training of the Neural network, as indicated by the absence of crests and troughs. There's absence of smoothness in the initial iterations, which slightly reduces the accuracy of the Neural Network. However, in comparison to traingda, traingdx is much more efficient for Stock Prediction.

NMSE for trainrp algorithm using tansig as activation function = 0.003917076

As seen in the above graph, the training achieves a very low MSE in the beginning itself. The training curve is smooth, indicating proper learning by the Neural Network. Thus trainrp training algorithm, by far, has been the most efficient in obtaining the lowest NMSE, and as evient from the graph, it facilitates faster and more accurate training.

NMSE for traingda algorithm using logsig as activation function = 0.01466

As seen in the training graph above, there's a steep decrease in the Mean Squared Error (MSE) in the beginning. After that, it is marked by numerous crests and troughs till it reaches the epoch limit. It achieves an NMSE of 0.01466, which indicates that the network is still not trained completely in compliance with the given data. Thus, traingdx and traingda have better accuracy when predicting Stock Market.

NMSE for traingdx algorithm using logsig as activation function = 0.010221

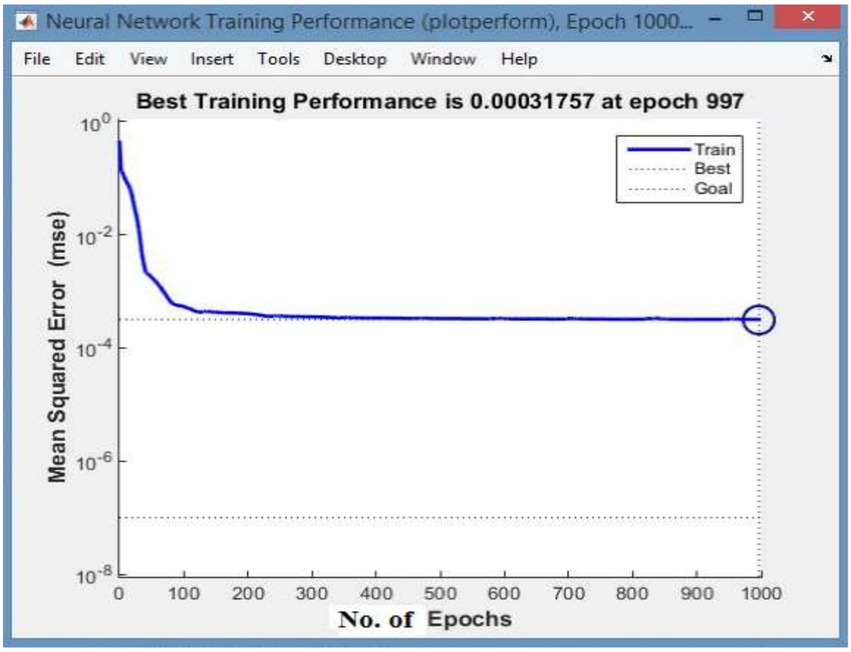

Figure 7. Training performance graph for tansig, traingdx.

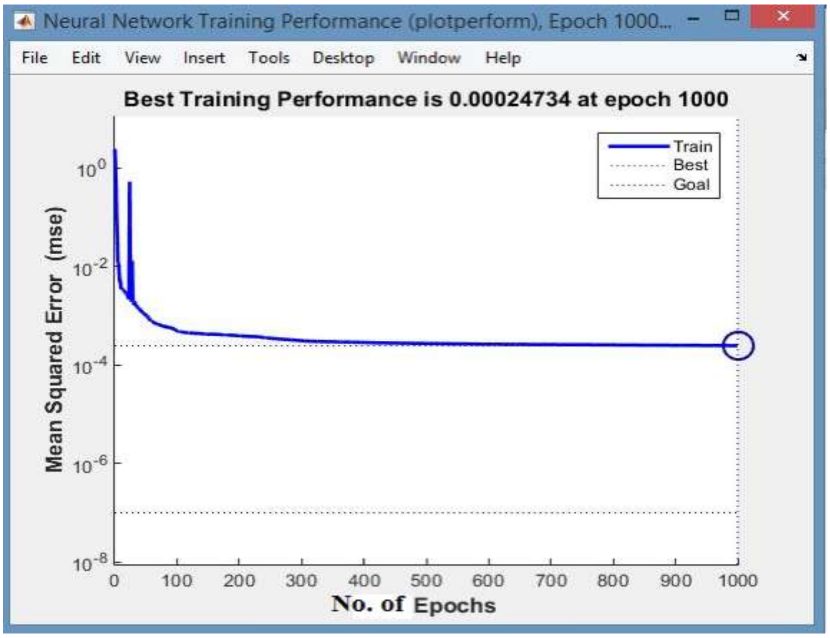

Figure 8. Training performance graph for tansig, trainrp.

It is seen that traingdx algorithm, with log sigmoid as the activation function, provides erratic training for the Network. The 0th iteration indicates a steep drop, followed by a bulging drop, which slightly reduces the accuracy of the Neural Network. Also, certain discontinuity exists in the higher epochs at certain places. This considerably reduces its accuracy, and hence the NMSE.

NMSE for trainrp algorithm using logsig as activation function = 0.003745

As seen in the above graph, it provides the smoothest training curve. The training achieves a very low MSE in

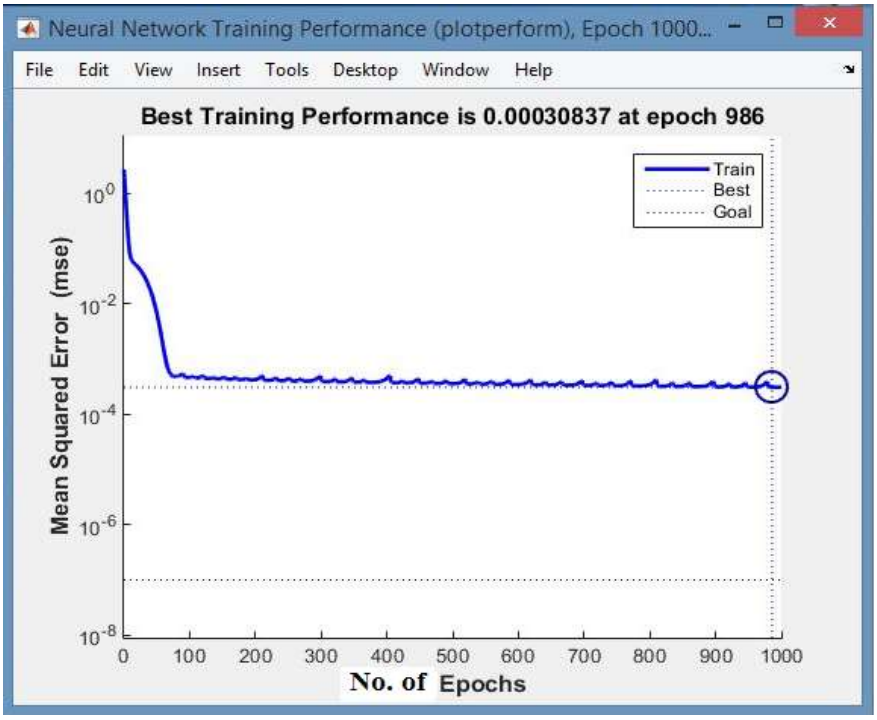

Figure 9. Training performance graph for logsig, traingda.

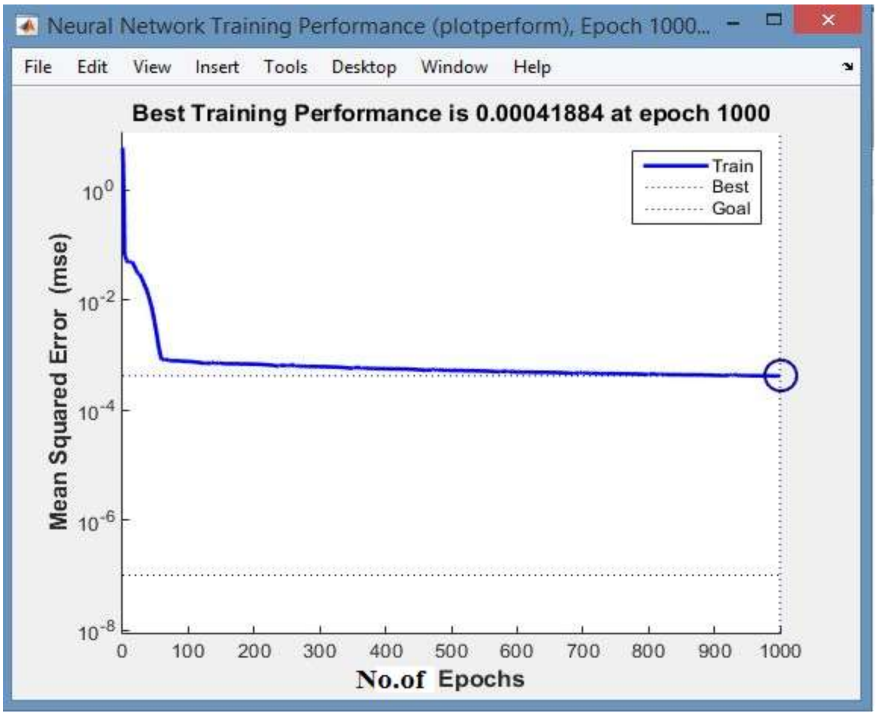

Figure 10. Training performance graph for logsig, traingdx.

the beginning itself, without any erratic training behavior or discontinuity. The training curve is smooth, indicating proper learning by the Neural Network. Thus trainrp training algorithm along with log-sigmoid is the most efficient in obtaining the lowest NMSE, and as evident from the graph, it facilitates faster and more accurate training.

6.1 Optimization using Genetic Algorithm

When stock prediction is carried out using Artificial Neural Networks, the initial values for synaptic weights are randomly chosen. However, if the initial weights as taken by the system are not in synchronization with the direction of the training, the training process will be lengthened. In order to choose the best values for the synaptic weights Genetic Algorithm is used. In this model, genetic crossover is used. An initial set of random population of size 400 is taken, and is subjected to crossover, by increasing its fitness value, where the fitness function is the Mean Squared Error (MSE) for the population. The lesser the MSE, the more is the fitness value of the individual belonging to the population. The algorithm terminates when the average change in the fitness value is less than 1e-08. The MLP network is now trained by using the optimized synaptic weight values using trainrp and logsig.NMSE for training using Genetic Algorithm = 0.003092121

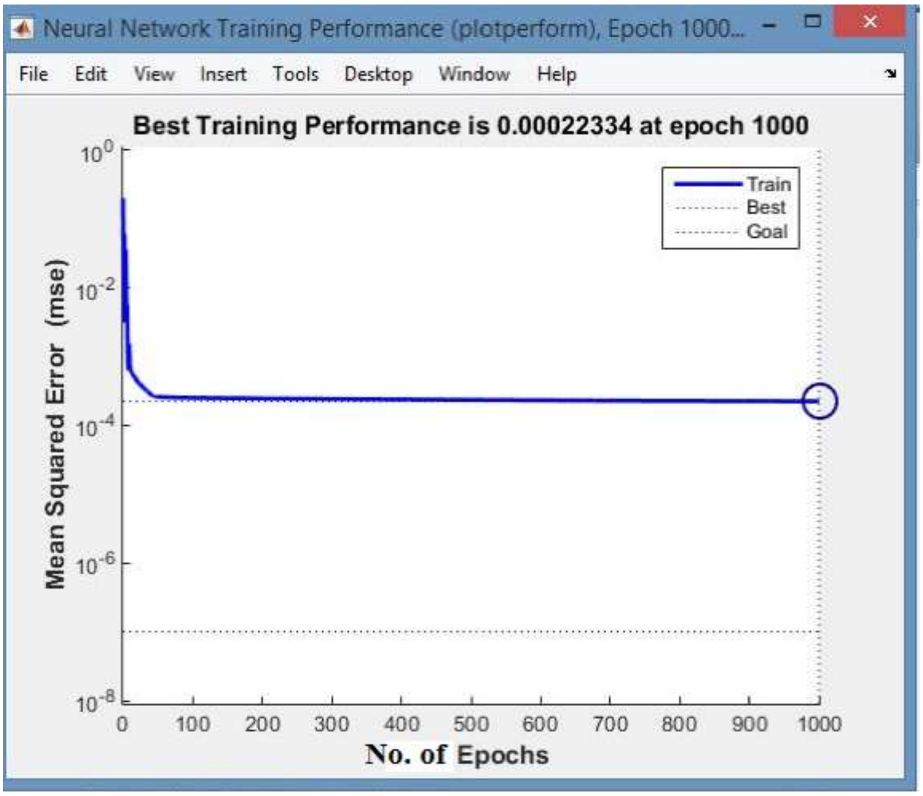

Figure 11. Training performance graph for logsig, trainrp.

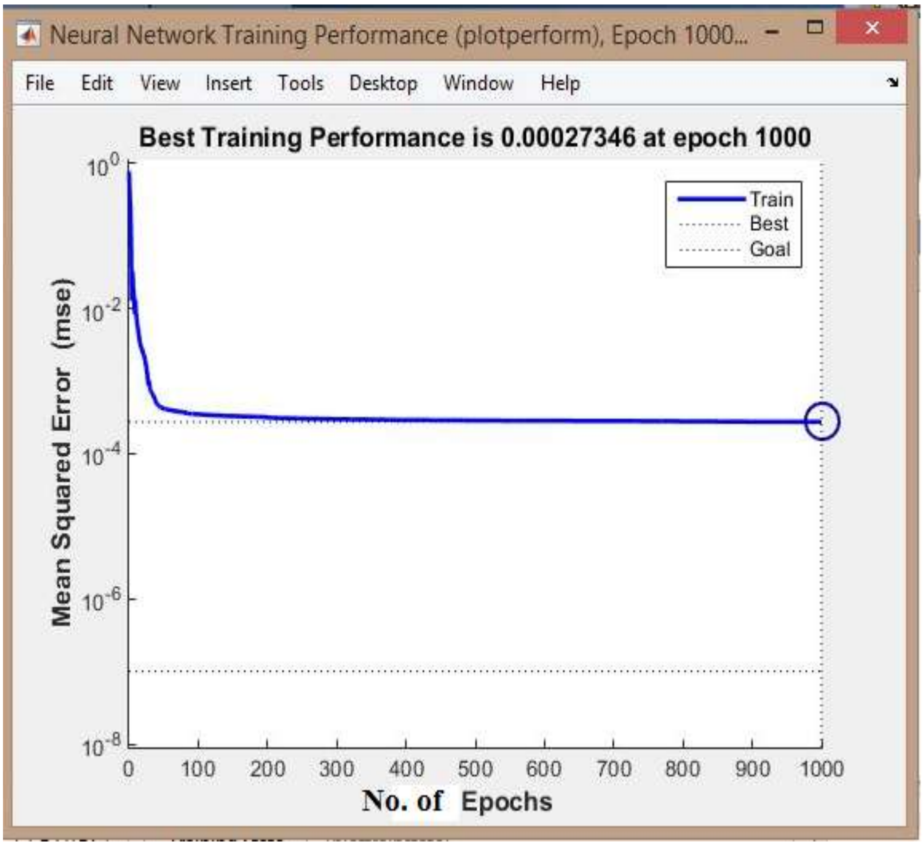

Figure 12. Training performance graph for logsig, trainrp after GA.

It is observed that the training is smoother and faster, and hence more efficient.

Once the Neural Network is trained, it is tested for future stock market predictions. The nth day's input is entered, and the network is simulated with this input data. Upon doing so, the network gives the prediction for next day's closing value. This is tabulated in the excel sheet. When the actual value is obtained the next day, it is compared against the predicted value and its accuracy is analyzed.

6.2 Testing

Testing is carried out for the networks using trainrp as the training algorithm, with tan-sigmoid and log-sigmoid as the activation functions. The data for all working days of April, is predicted for the purpose of testing.

6.2.1 Testing for Trainrp in Combination with Tan-Sigmoid

Upon testing for tan-sigmoid with Trainrp, the network provides somewhat accurate training in the beginning, with a minimum difference between the actual and predicted value being 10.5509. However, the difference increases with passing days, except for one instance where the difference reduces to 15.2722. It has a NMSE of 0.276605.

6.2.2 Testing for Trainrp in Combination with Log-Sigmoid

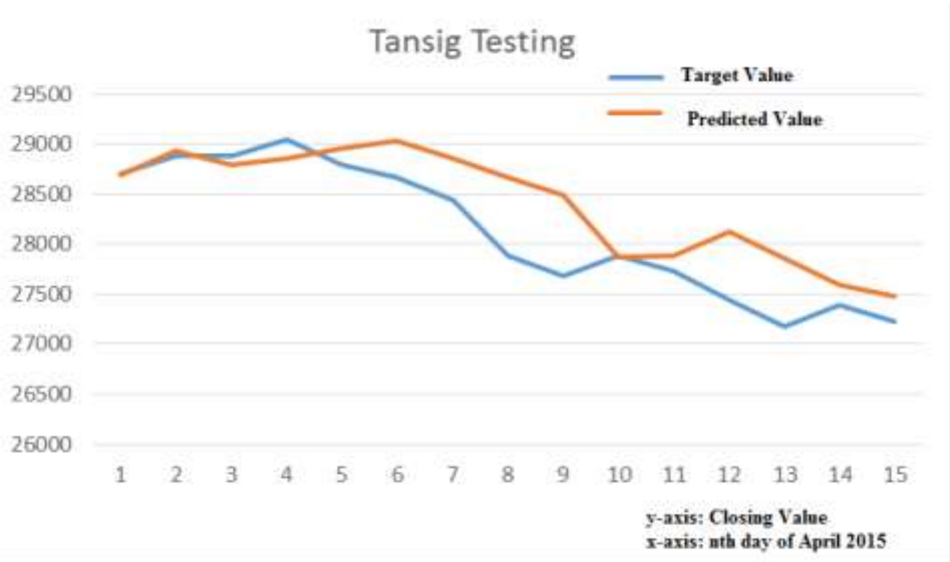

Figure 13. Testing graph for tansig, trainrp.

Upon testing for log-sigmoid with Trainrp, the network provides a more accurate result than tan-sigmoid. The minimum difference between the actual and predicted value is 5.05754187. As seen in the graph, the predicted trend almost reflects the actual trend of the stock market index. Given that human factors also play a role in deter mining the trends of stock market (which is beyond the understanding of machine), the Neural Network architecture adopted for stock market prediction more or less predicts the next day's value as close to the original value as possible. It has a NMSE of 0.053318.

6.2.3 Testing for Trainrp with Log-Sigmoid using Genetic Algorithm

Upon testing with trainrp while including Genetic Algorithm, the following graph is obtained, showing two non-overlapping lines. The values predicted by this network do not coincide with the original values completely. Hence, the NMSE is 0.248067946. However, the network very efficiently follows the trends of the stock market index. The highs and lows, the gradient of the slopes, are all accurately depicted by improvising with the help of Genetic Algorithm. This indicates that the network training successfully inculcates the nature of the stock market and hence is able to accurately predict future trends.

6.3 Comparison between Genetic Algorithm and Non-Genetic

Algorithm Model

The Figure below shows the results of the Genetic Algorithm Model and the non-Genetic Algorithm model overlapped simultaneously over the actual closing value. As seen in the Figure, the graph corresponding to Non GA model (Trainrp with log-sigmoid model) almost converges with the actual closing value on Day 4 and Day 11. It is very close on Day 2, 3, 5 and 10. However, it does not completely follow the trend of the actual stock index. For instance, on Day 6, the actual closing price value is shown to be decreasing from the previous day's value. Hence there's a downward slope in the graph from Day 5 to Day 6. But, the predicted value is misleading, as it shows that the closing value of Day 6 is more than that of Day 5. Similar anomalies are visible in Day 7, 9 and 13.

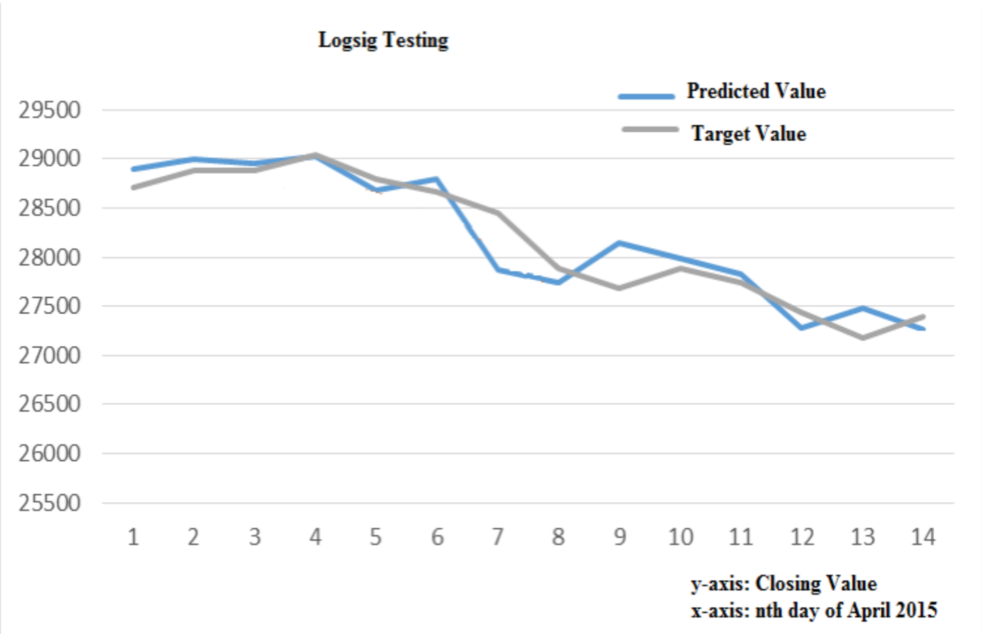

Figure 14. Testing graph for logsig, trainrp

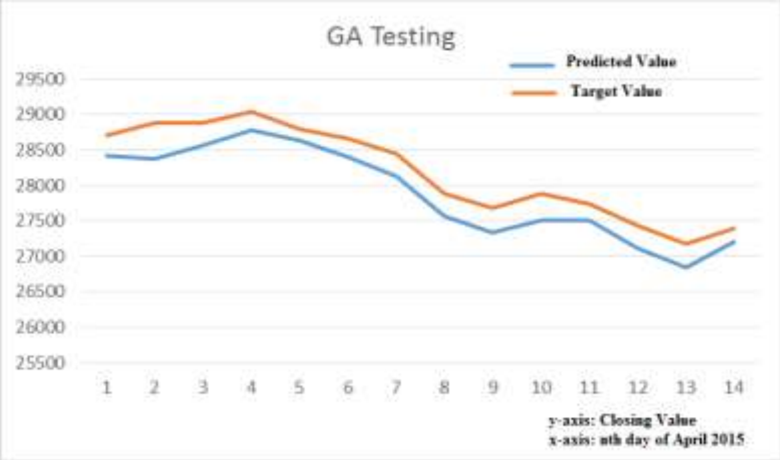

Figure 15. Testing graph for logsig, trainrp with GA.

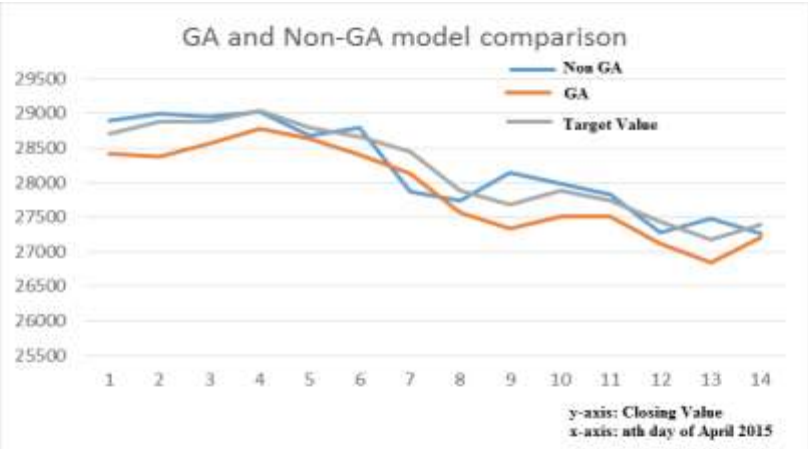

Figure 16. Comparison graph for logsig, trainrp with GA and without GA models

In contrast to this, the Genetic algorithm model, which also implements Trainrp with log sigmoid, is in compliance with the actual stock market trend. The highs and lows in the graph are in agreement with the actual closing prices. Even though the values do not converge at any point, the difference in the values is a small percentage of the actual value of the closing prices. It is the trend of the stock market which is more important to investors.

7. CONCLUSION

Artificial Neural Network is a successful soft-computing technique to predict the trends of the stock market index. Various training algorithms based on backpropagation of errors have been tried, out of which the Resilient Propagation training algorithm (trainrp) provides the most efficient training results, as it undoes the error that occurs due to squashing of inputs. Two activation functions, tan-sigmoid and log-sigmoid have been used in the model, out of which log-sigmoid fares better than tan-sigmoid, according to the results obtained.

The Neural Network model is further improvised by using Genetic Algorithm to initialize synaptic weights. By introducing Genetic Algorithm, the network is more effectively trained.