International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 6 (2014) pp. 257 - 269


# Impact of Data Normalization on Stock Index Forecasting

**S. C. Nayak[1], B. B. Misra[2] and H. S. Behera[3]**

[1] Department of Computer Science & Engineering, VSS University of Technology,
Burla, Sambalpur, India
*sarat_silicon@yahoo.co.in*

[2] Department of Computer Science and Engineering, Silicon Institute of Technology
Silicon Hills, Patia, Bhubaneswar, India
*misrabijan@gmail.com*

[3] Department of Computer Science & Engineering, VSS University of Technology,
Burla, Sambalpur, India
*hsbehera_india@yahoo.com*

***Abstract*: Forecasting the behavior of the financial market is a nontrivial task that relies on the discovery of strong empirical regularities in observations of the system. These regularities are often masked by noise and the financial time series often have nonlinear and non-stationary behavior. With the rise of artificial intelligence technology and the growing interrelated markets of the last two decades offering unprecedented trading opportunities, technical analysis simply based on forecasting models is no longer enough. To meet the trading challenge in today's global market, technical analysis must be redefined. Before using the neural network models some issues such as data preprocessing, network architecture and learning parameters are to be considered. Data normalization is a fundamental data preprocessing step for learning from data before feeding to the Artificial Neural Network (ANN). Finding an appropriate method to normalize time series data is one of the most critical steps in a data mining process. In this paper we considered two ANN models and two neuro-genetic hybrid models for forecasting the closing prices of Indian stock market. The present pursuit evaluates impact of various normalization methods on four intelligent forecasting models i.e. a simple ANN model trained with gradient descent (ANN-GD), genetic algorithm (ANN-GA), and a functional link artificial neural network model trained with GD (FLANN-GD) and genetic algorithm (FLANN-GA). The present study is applied on daily closing price of Bombay stock exchange (BSE) and several empirical as well as experimental result shows that these models can be promising tools for the Indian stock market forecasting and the prediction performance of the models are strongly influenced by the data preprocessing method used.**

***Keywords***: artificial neural networks, back propagation, normalization, functional link artificial neural network, gradient descent**.**

## I. Introduction

Artificial neural network is one of the important approaches in machine learning methods. ANNs are software constructs designed to mimic the way the human brain learns. The neural network can imitate the process of human's behavior and solve nonlinear problems, which have made it widely used in calculating and predicting complicated system. The quality of non linearity mapping achieved in ANN is difficult with the conventional calculating approaches. The nerve cell, which is the base of neural network, has the function of processing information [1]. From the range of Artificial Intelligence techniques (AI), the one that deals best with uncertainty is Artificial Neural Network (ANN). Dealing with uncertainty with ANNs forecasting methods primarily involves recognition of patterns in data and using these patterns to predict future events. ANNs have been shown to handle time series problems better than other AI techniques because they deal well with large noisy data sets. Unlike expert systems, however, ANNs are not transparent, thus made them difficult to interpret. ANN and its hybridization with other soft computing techniques have been successfully applied to the potential corporate finance applications which are not limited to the followings:

- Financial and Economic forecasting.
- Detection of Regularities in security price movement.
- Prediction of default and bankruptcy.
- Credit Approval.
- Security / Asset portfolio management.
- Mortgage Risk Assessment.
- Predicting Investor's Behavior.
- Risk rating of Exchange-traded, fixed income investments etc.

Neural networks extensively used in medical applications such as image/signal processing [2], pattern and statistical classifiers [3] and for modeling the dynamic nature of biological systems. Some of the systems employing neural networks are developed for decision support purposes in diagnosis and patient management. The use of neural networks for diabetic classification has also attracted the interest of the medical informatics community because of their ability to model the nonlinear nature. Gradient based methods are one of the most widely used error minimization methods used to train back propagation networks. Back propagation algorithm is a classical domain dependent technique for supervised training. It works by measuring the output error calculating the gradient of this error, and adjusting the ANN weights and biases in the descending gradient direction. Back propagation is the most commonly used and the simplest feed forward algorithm used for classification.

The goal of this study is to develop some efficient soft computing models to forecast the Indian stock market effectively and efficiently. Secondly various statistical normalization procedures have been proposed as an essential data preprocessing task to improve the prediction accuracy, out of which the best data normalization methods are suggested. The data set prepared after normalization; enhances the learning capability of the network with minimum error.

Rest of the paper is organized as follows. Section II covers the work related to this field. The background of the stock index prediction is covered is Section III. In Section IV, preprocessing of the input data is discussed. Section V presents the different types of normalization methods. The architecture of the different models are presented is Section VI. Experimental study and the analysis of the results are covered in Section VII, followed by the concluding remarks.

## II. Research background

The Artificial Neural Network (ANN) has recently been applied to many areas such as Data mining, Stock market analysis, medical and many other fields. Performance of hybrid forecasting models (ARFIMA-FIGARCH) have been compared with traditional ARIMA models and shown their superiority [4]. In order to reducing the over fitting tendency of genetic algorithm, the authors in [5] proposes an approach, neighborhood evaluation, which involves evaluation for neighboring points of genetic individuals in fitness landscape. Suitable forms of neighborhoods were discussed on the performance of the genetic searches. In [6] the methods of ensemble of nonlinear architectures in the computational paradigm have been carried out to forecast the foreign exchange rates and shown enhanced results. The research work done by Z. Mustaffa and Y. Yusof incorporated LS-SVM and Neural Network Model (NNM) for future dengue outbreak [7]. They investigated the use of three normalization techniques such as Min-Max, Z-Score and Decimal Point normalization and suggested that the above models achieve better accuracy and minimized error by using Decimal Point Normalization compared to the other two techniques [8]. Realizing the importance of data preprocessing in mining algorithms, L. AI Shalabi and Z. Shaaban proposed a different normalization techniques which is experimented on ID3 methodology [9]. Their experimental result shows that min-max Normalization produces best result with the highest accuracy than that of Z-Score and decimal scaling normalization. In [10], various normalization methods used in back propagation neural networks for diabetes data classification; they found that the results are dependent on the normalization methods. Junita Mohamad-saleh and Brain applied the Principle Component Analysis (PCA) method to the Electricity Capacitance Tomography data which are highly correlated due to overlapping sensing areas [11]. The findings suggest that PCA data processing method useful for improving the performance of MLP systems. In [12], an Adaptive Normalization (AN) was proposed, which has been tested with an ANN and the result shows that AN improve ANN accuracy in both short and long term prediction. Another traditional approach to handle non-stationary time series uses the sliding window technique [13]. This works well for time series with uniform volatility, but most time series present non-uniform volatility, i.e. high volatility for certain time periods and low for others.

## III. Stock market forecasting background

### *A. Stock index forecasting problem*

Recently forecasting stock market return is gaining more attention, because of the fact that if the direction of the market is successfully predicted the investors may be better guided and also monetary rewards will be substantial. In recent years, many new methods for the modeling and forecasting the stock market has been developed [14]. Another motivation for research in this field is to highlight many theoretical and experimental challenges. The most important of these is the efficient market hypothesis which proposes that profit from price movement is very difficult and unlikely. In an efficient stock market, prices fully reflect available information about the market and its constituents and thus any opportunity of earning excess profit ceases to exist any longer. So it is ascertained that no system is expected to outperform the market predictably and consistently. The Neuro-Genetic hybrid networks gain wide application in the aspect of nonlinear forecast due to its broad adaptive and learning ability [15]. At present, the most widely used neural network is back propagation neural network (BPNN), but it has many shortcomings such as the slow learning rate, large computation time, gets stuck to local minimum. RBF neural networks is also a very popular method to predict stock market, this network has better calculation and spreading abilities, it also has stronger nonlinear mapped ability [16]. The new hybrid iterative evolutionary learning algorithm is more effective than the conventional algorithm in terms of learning accuracy and prediction accuracy [17]. Many researchers have adopted a neural network model, which is trained by GA [18]-[19]. The Neural Network-GA model to forecasting the index value has got wide acceptance. ANNs have data limitation and no definite rule exists for the sample size requirement of a given problem. The amount of data for network training depends on the network structure, the training method, and the complexity of the particular problem or the amount of noise in the data on hand [20]. Using hybrid models, by combining several models, has become a common practice to improve the forecasting accuracy since forty years ago [21]. Clemen has provided a comprehensive review in this area [22]. Many studies can be reported in the literature; for example combination of ARIMA models and Support Vector Machines (SVMs) [23]-[24], hybrid of Grey and Box-Jenkins Auto-Regressive Moving Average (ARMA) models [25], integration of ANN with Genetic Algorithms (GAs) [26]-[28], combination of ANN with Generalized Linear Auto-Regression (GLAR) [29], hybrid of Artificial Intelligence (AI) and ANN [30], integration of ARMA with ANN [31], combination of Seasonal ARIMA with Back Propagation ANN (SARIMABP) [32], combination of several ANNs [33], hybrid model of Self Organization Map (SOM) neural network and GAs and Fuzzy Rule Base (FRB) [34], combination of fuzzy logic and Nonlinear Regression (NLR) [35], hybrid of fuzzy techniques with ANNs [36]-[37], combination of ARIMA and fuzzy logic [38], and hybrid based on Particle Swarm Optimization (PSO), Evolutionary Algorithm (EA) and Differential Evolution (DE) and ANN [39].

### *B. Efficient Market Hypothesis (EMH)*

EMH is one of the most controversial and influential arguments in the field of academic finance research and financial market prediction. EMH is usually tested in the form of a random walk model. The crux of the EMH is that it should be impossible to predict trends or prices in the market through fundamental analysis or technical analysis. A random walk is the path of a variable over time that exhibits no predictable patterns. If a price time series y(t) moves in a random walk, the value of y(t) in any period will be equal to the value of the previous period y(t-1), plus some random variable. The random walk hypothesis states that successive price changes between any two time periods are independent, with zero mean and variance proportional to the time interval between the two. More and more recent studies on financial markets argue that the market prices do not follow a random walk hypothesis.

### *C. Stock Market Forecasting Methods*

There are three common methods regarding stock market predictions: Technical Analysis, Fundamental Analysis, and Machine Learning Algorithms. Fundamental analysis is a security valuation method that uses financial and economic analysis to predict the movement of security prices. Fundamental analysts examine a company's financial conditions, operations, and/or macroeconomic indicators to derive the intrinsic value of its common stock [40].Technical analysis is a security analysis technique that claims the ability to forecast the future direction of prices through the study of past market data, primarily price and volume. [41] There are three key assumptions in technical analysis: 1. Market action (price, volume) reflects everything. 2. Prices move in trends. 3. History repeats itself. Because of the large exchange volume of the stock market, more and more people believe that machine learning is an approach that can scientifically forecast the tendency of the stock market. Machine learning refers to a system capable of the autonomous acquisition and integration of knowledge. Our forecasting task is mainly based on artificial neural networks (ANNs) algorithm as it has been proven repeatedly that ANNs can predict the stock market with high accuracy. ANNs are non-linear statistical data modeling tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data. ANNs has ability to extract useful information from large set of data therefore ANNs play very important role in stock market prediction. Artificial neural networks approach is a relatively new, active and promising field on the prediction of stock price behavior. Artificial neural networks can be used for financial prediction in one of the three ways. It can be provided with inputs, which enable it to find rules relating the current state of the system being predicted to future states. Secondly, it can have a window of inputs describing a fixed set of recent past states and relate those to future states. Finally it can be designed with an internal state to enable it to learn the relationship of an indefinitely large set of past inputs to future states, which can be accomplished via recurrent connections.

Prediction on stock price by neural network consists of two steps training or fitting of neural network and forecast. In the training step, network generates a group of connecting weights and bias values, getting an output result through positive spread, and then compares this with expected value. If the error has not reached expected minimum, it turns into negative spreading process, modifies connecting weights of network to reduce errors. Output calculation of positive spread and connecting weight calibration of negative spread are doing in turn. This process lasts till the error between calculated output and expected value meets the requirements, so that the satisfactory connecting weights and threshold can be achieved. Network prediction process is to input testing sample to predict, through stable trained network (including training parameters), connecting weights and threshold. Due to rapid growth in economics and globalization, it is nowadays a common notion that vast amounts of capital are traded through the stock markets all around the globe. National economies are strongly linked and heavily influenced by the performance of their stock markets. Moreover, recently the markets have become a more accessible investment tool, not only for strategic investors but for common people as well. Consequently they are not only related to macroeconomic parameters, but they influence everyday life in a more direct way. Therefore they constitute a mechanism which has important and direct social impacts. The characteristic that all stock markets have in common is the uncertainty, which is related to their short and long term future state. This feature is undesirable for the investor but it is also unavoidable whenever the stock market is selected as the investment tool. The best that one can do is to try to reduce this uncertainty. Stock market prediction is one of the instruments in this process which is a critical issue. The main advantage of artificial neural networks is that they can approximate any nonlinear function to an arbitrary degree of accuracy with a suitable number of hidden units.

### *D. Time Series Data*

Real world problems with time dependencies require special preparation and transformation of data, which are, in many cases, critical for successful data mining. Here we considered the simple single feature measured over time such as stock daily closing prices, temperature measured every hour, or sales of a product recorded every day. These are the classical univariate time series problem, where it is expected that the value of the variable X at a given time be related to previous values. Because the time series is measured at fixed units of time, the series of values can be expressed as

$X = \{t(1), t(2), t(3), \ldots\ldots\ldots, t(n)\}$

Where t(n) is the most recent value.

For many time series problems, the goal is to forecast t(n+1) from previous values of the feature, where these values are directly related to the predicted value. While the typical goal is to predict the next value in time, in some applications, the goal can be modified to predict values in the future, several time units in advance. More formally, given the time-dependent values t(n-i), ……, t(n), it is necessary to predict the value t(n+j), where j is the number of days in advanced.

Forecasting in time series is one of the more difficult and less reliable task in data mining. The goal for a time series can be changed from predicting the next value in time series to classification into one of predefined categories. Instead of predicted output value t(i+1), the new classified output will be binary 1 for t(i+1) threshold value and 0 for t(i+1) threshold value. The time units can be relatively small, enlarging the number of artificial features in a tabular representation of time series for the same time period. The resulting problem of high dimensionality is the price paid for precision in the standard representation of the time series data. In practical applications, many older values of a feature may be historical relics that are no more relevant and should not be used for analysis. Therefore, for many business and social applications, new trends can make older data less reliable and less useful. This leads to a greater emphasis on recent data, possibly discarding the oldest portion of the time series.

Besides standard tabular representation of time series, sometimes it is necessary to additionally preprocess raw data and summarize their characteristics before application of data mining techniques. Many times it is better to predict the difference t(n+1) – t(n) instead of the absolute value (n+1) as the output. Also, using a ratio, t(n+1) / t(n), which indicates the percentage of changes, can sometimes give better prediction results. These transformations of the predicted values of the output are particularly useful for logic-based data-mining methods such as decision trees or rules. When differences or ratios are used to specify the goal, features measuring differences or ratios for input features may also be advantageous.

## IV. Preprocessing input data

In raw datasets missing values, distortions, incorrect recording, and inadequate sampling is expected. Raw data is highly susceptible to noise, missing values, and in consistency. The quality of data affects the data mining results. In order to help improve the quality of the data and consequently, of the mining results raw data is preprocessed so as to improve the efficiency and ease of the mining process. Data preprocessing is one of the most critical steps in a data mining process which deals with the preparation and transformation of the initial data set. Data preprocessing methods include data cleaning, data integration, data transformation and data reduction. Data that is to be analyze by data mining techniques can be incomplete (lacking attribute values or certain attributes of interest, or containing only aggregate data), noisy (containing errors, or outlier values which deviate from the expected), and inconsistent (e.g., containing discrepancies in the codes used to categories items). Incomplete, noisy, and inconsistent data are commonplace properties of large, real-world databases and data warehouses. Incomplete data can occur due to various reasons. Attributes of interest may not always be available. Other data may not be included because it was not considered important at the time of collection. Relevant data may not be recorded due to a misunderstanding or because of equipment malfunctions. Data that were inconsistent with other recorded data may have been deleted. Missing data, particularly for tuples with missing values for some attributes, may need to be inferred. Data can be noisy, having incorrect attribute values due to human or computer errors occurring at data entry, errors in data transmission or technology limitations. Data cleaning methods work to clean the data by filling in missing values, smoothing noisy data, identifying or removing outliers, and resolving inconsistencies. Most of the mining routines concentrate on avoiding over fitting the data to the function being modeled. Therefore, care should be taken to adopt an appropriate preprocessing technique to clean the raw dataset. On the other hand data integration combines data from multiple sources into a coherent data store. These sources may include multiple databases, data cubes, or flat files. There are a number of issues to consider during data integration. Transformation and normalization are two widely used preprocessing methods. Data transformation involves manipulating raw data inputs to create a single input to a net, while normalization is a transformation performed on a single data input to distribute the data evenly and scale it into an acceptable range for the network. Data transformation involves normalization, smoothing, aggregation and generalization of the data. Complex data analysis and mining on huge amounts of data may take a very long time making such analysis impractical or infeasible. Data reduction techniques have been helpful in analyzing reduced representation of the data set without compromising the integrity of the original data and producing the quality knowledge. The concept of data reduction involves either reducing the volume or reducing the dimensions of datasets. Strategies for data reduction includes data cube aggregation, dimension reduction, data compression, numerosity reduction, discretization and concept hierarchy generation. Knowledge of the domain is important in choosing preprocessing methods to highlight underlying features in the data, which can increase the network's ability to learn the association between inputs and outputs. Some simple preprocessing methods include computing differences between or taking ratios of inputs. This reduces the number of inputs to the network and helps it learn more easily. In financial forecasting, transformations that involve the use of standard technical indicators should also be considered. For example, moving average is utilized to smooth the price data. When creating a neural net to predict tomorrow's close, a five-day simple moving average of the close can be used as an input to the net. This benefits the net in two ways. First, it gives useful information at a reasonable level of detail; and second, by smoothing the data, the noise entering the network has been reduced. This is important because noise can obscure the underlying relationships within input data from the network, as it must concentrate on interpreting the noise component. The only disadvantage is that worthwhile information might be lost in an effort to reduce the noise, but this tradeoff always exists when attempting to smooth noisy data. The data preprocessing methods are shown in Fig. 1.

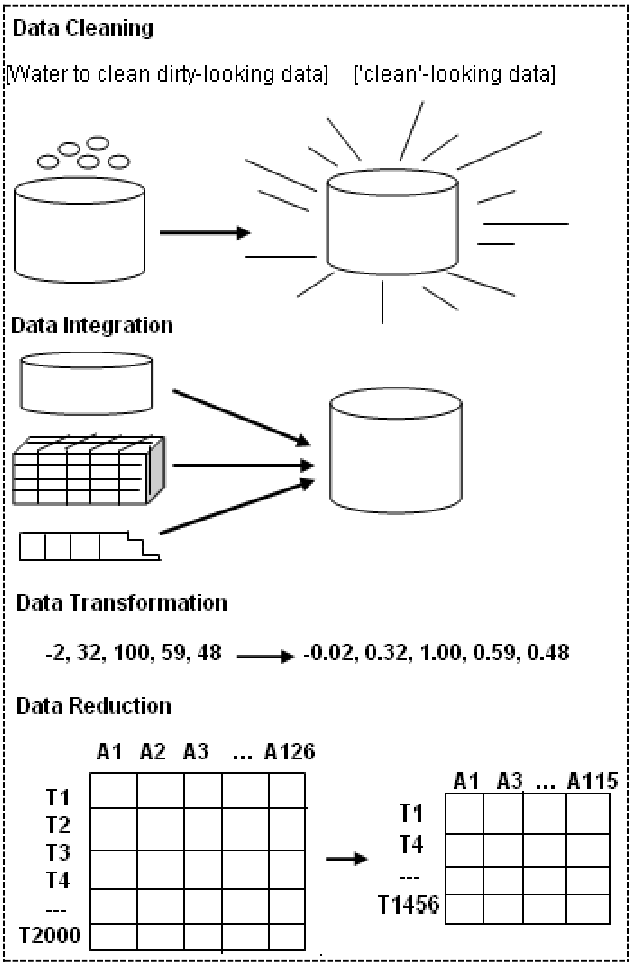


Figure 1. Data preprocessing methods

## V. Data Normalization Techniques

The main goal of data normalization is to guarantee the quality of the data before it is fed to any learning algorithm. There are many types of data normalization. It can be used to scale the data in the same range of values for each input feature in order to minimize bias within the neural network for one feature to another. Data normalization can also speed up training time by starting the training process for each feature within the same scale. It is especially useful for modeling application where the inputs are generally on widely different scales. Different techniques can use different rules such as min-max, Z-score, decimal scaling, Median normalization and so on. Some of the techniques are discussed here.

### *A. Min-Max Normalization*

In this technique the data inputs are mapped into a predefined range [0, 1] or [-1, 1]. The min-max method normalizes the values of the attributes $A$ of a data set according to its minimum and maximum values. It converts a value $a$ of the attribute $A$ to $\hat{a}$ in the range [*low*, *high*] by computing:

$$\hat{a} = low + \frac{(high - low) * (a - \min A)}{\max A - \min A} \qquad (1)$$

The main problem of using the min-max normalization method in time series forecast is that the minimum and maximum values of out-of-sample data set are unknown. A simple way to overcome this problem is to consider the minimum, $\min A$ and the maximum $\max A$ values presented in the sample data set, and then map all out-of-sample values below $\min A$ and above $\max A$ to $low$ and $high$, respectively. However, this approach leads to significant information loss and to a concentration of values on certain parts of the normalized range [42], which implies more computational effort and loss of quality in learning techniques [43]-[44].

### *B. Decimal Scaling Normalization*

In this method, the decimal point of the values of an attribute $A$ is moved to its maximum absolute value. The number of decimal points moved depends on the maximum absolute value of the data set $A$. Hence, a value $a$ of $A$ is normalized to $\hat{a}$ by computing:

$$\hat{a} = \frac{a}{10^d} \qquad (2)$$

where $d$ is the smallest integer such that $\max(|\hat{a}|) < 1$. This method also depends on knowing the maximum values of a time series and has the same problems of min-max when applied to time series data.

### *C. Z-Score Normalization*

In this normalization method the values of an attribute $A$ are normalized according to their mean and standard deviation. A value $a$ of $A$ is normalized to $\hat{a}$ by computing:

$$\hat{a} = \frac{a - \mu(a)}{\partial(a)} \qquad (3)$$

Where $\mu(a)$ is the mean value and $\partial(a)$ is the standard deviation of attribute $A$, respectively.
This method is useful in stationary environments when the actual minimum and maximum values of $A$ are unknown, but it cannot deal well with non-stationary time series since the mean and standard deviation of the time series vary over time.

### *D. Median Normalization*

The median method normalizes each sample by the median of raw inputs for all the inputs in the sample. It is a useful normalization to use when there is a need to compute the ratio between two hybridized samples. Median is not influenced by the magnitude of extreme deviations. It can be more useful when performing the distribution.

$$\hat{a} = \frac{a}{median(a)} \qquad (4)$$

### *E. Sigmoid Normalization*

This normalization method is the simplest one used for most of the data normalization. The data value $a$ of attribute $A$ is normalized to $\hat{a}$ by computing:

$$\hat{a} = \frac{1}{1 + e^{a}} \qquad (5)$$

The advantage of this normalization method is that it does not depend on the distribution of data, which may be unknown at the time of training the model.

*F. Median and Median Absolute Deviation*

The median and median absolute deviation (MAD) is a robust measure of the variability of a univariate sample of quantitative data. MAD is a measure of statistical dispersion and more resilient to outliers in a data set than the standard deviation. The data value *a* of attribute *A* is normalized to â by computing:

$$\hat{a} = \frac{a - median(a)}{MAD} \tag{6}$$

Where $MAD = median(\,(\mathrm{abs}(\{a_k\} - median(A)))$

*G. Tanh estimators*

Tanh estimator is a robust and highly efficient method to normalize time series data. This was introduced by Hampel. The data value $a\cdot$ of attribute *A* is normalized to â by computing:

$$\hat{a} = 0.5\left[\tanh\left[\frac{0.01(a-\mu)}{\sigma}\right]+1\right] \tag{7}$$

Where μ is the mean value and σ is the standard deviation of the data set respectively.

## VI. Architecture of Models used

The system architecture for this research work is shown in Fig. 3. The work starts with collecting the raw data comprising the daily closing price of BSE. Different normalization methods are applied on the data set, the normalized data set has been generated as output. The data set are then used for the training and testing purpose by four expert forecasting systems.

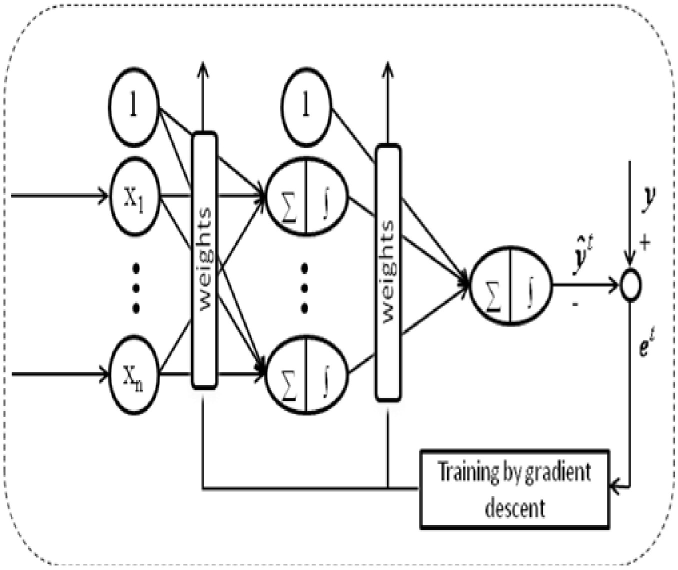



Figure 2. Architecture of ANN-GD

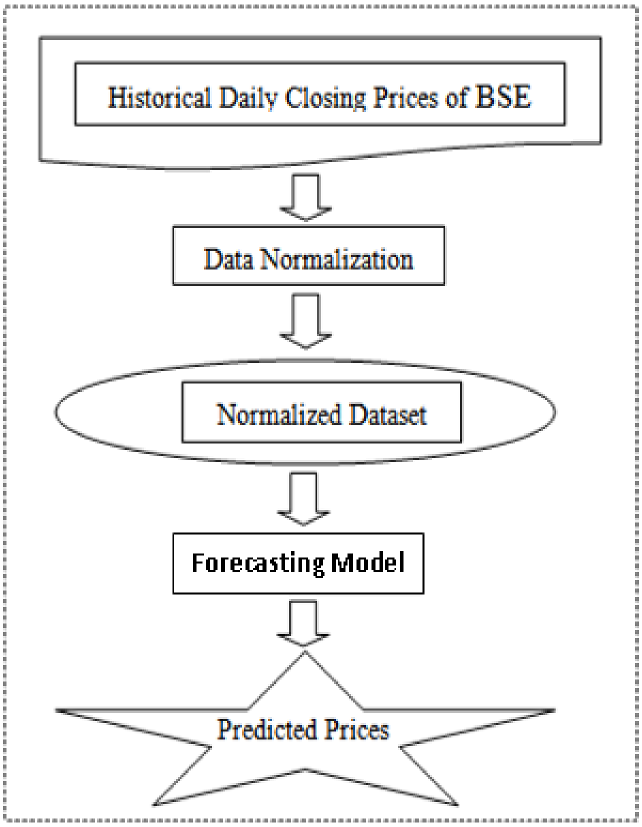



Figure 3. System Architecture

*A. ANN-GD model*

The feed forward neural network model considered here consists of one hidden layer only. The architecture of the ANN model is presented in Fig. 2. This model consists of a single output unit. The neurons in the input layer use a linear transfer function, the neurons in the hidden layer and output layer use sigmoid function.

$$y_{out} = \frac{1}{1+e^{-\lambda y_{in}}}$$

Where $y_{out}$ the output of the neuron, λ is is the sigmoidal gain and $y_{in}$ is the input to the neuron. The back propagation algorithm is described as follows:

**Back Propagation Algorithm**

Step-I: Initialize weight to a small random value.

Step-II: While stopping condition, do Steps III-X

Step-III: for each training pair do Steps IV-X

%%***Feed forward***

*Step-IV:* each input neuron receives the input signal xi and transmits this signal to all units in the layer above.

Step-V: each hidden unit (zj, j=1, …, p) sums its weighted input signals.

$z_{inj} = v_{oj} + \sum x_i v_{ij}$

$z_j = f(\,Z_{inj})$ and send this signal to all units in the layer above.

Step-VI: each output unit ($y_k$, $k$=1, ……, $m$) sums its weighted signals

$y_{ink} = w_{ok} + \sum z_j w_{jk}$ and applies its activation function to calculate the output signals.

$Y_k = f(y_{ink})$

%%***Back propagation of errors***

Step-VII: each output unit ($y_k$, $k$=1,…,$m$) receives a target pattern corresponding to an input pattern and error term is calculated.

$\delta_k = t_k - y_k$

Step-VIII: each hidden unit ($z_j$, $j$=1,……, $n$) sums its delta inputs from units in the layer above.

$\delta_{inj} = \sum \delta_j w_{jk}$ , The error is calculated as

$\delta_j = \delta_{inj} f(z_{inj})$

%%***Updation of weight and biases***

Step-IX: for each output neuron, the weight correction term is given by

$\Delta W_{jk} = \alpha\ \delta\ z_j$ and bias correction term is

$\Delta W_{ok} = \alpha\ \delta_k$

Therefore $W_{jk}\ (new) = W_{jk}\ (old) + \Delta W_{jk}$ and

$W_{ok}\ (new) = W_{ok}\ (old) + \Delta W_{ok}$

each hidden neuron updates its bias and weights

$\Delta V_{ij} = \alpha\ \delta_j\ x_i$ and bias correction term

$\Delta V_{oj} = \alpha\ \delta_j$

Therefore $V_{ij}\ (new) = V_{ij}\ (old) + \Delta V_{ij}$ and

$V_{oj}\ (new) = V_{oj}\ (old) + \Delta V_{oj}$

Step-X: Test the stopping condition.

### *B. ANN-GA model*

Because the BP neural network learning is based on gradient descent, the existence of a problem is that the network training speed is slow, easily falling into local minima, global search capability are poor. Hence it will affect the accuracy of model predictions. However the genetic algorithm search over the whole solution space, easier to obtain the global optimal solution, and does not require the objective function continuous, differentiable, and even does not require an explicit function of the objective function, just only requires problem computable. It's not often trapped in local minima, and especially in the error function is not differentiable or no gradient conditions.

The problem of finding an optimal parameter set to train an ANN could be seen as a search problem into the space of all possible parameters. The parameter set includes the weight set between the input-hidden layer, weight set between hidden-output layer, the bias value and the number of neurons in the hidden layer. This search can be done by applying genetic algorithm using exploitation and exploration. The important issues for GA are: which parameters of the network should be included into the chromosome i.e. solution space, how these parameters are codified into the chromosomes i.e. encoding schema and finally how each individual chromosome is evaluated and translated into the fitness function.

The architecture of the ANN-GA model is presented in Fig. 5. This model employs the ANN-GD network presented at Fig. 2 with a single hidden layer. The chromosomes of GA represent the weight and bias values for a set of ANN models. Input data along with the chromosomes values are fed to the set of ANN models. The fitness is obtained from the absolute difference between the target $y$ and the estimated output $\hat{y}$. The less the fitness value of an individual, GA considers it better fit. The overall steps of genetic algorithm are described as follows:

**Pseudo code for Genetic Algorithm**

```
  Choose an initial population of chromosomes;
while termination condition not satisfied do
  repeat
    if cossover condition satisfied then
    {
      Select parent chromosomes;
      Choose crossover parameters;
      Perform crossover;
    }
    if mutation condition satisfied then
    {
      Choose mutation point;
    Perform mutation;
    }
  Evaluate fitness of offspring
  until sufficient offspring created;
 Select new population;
end while.
```

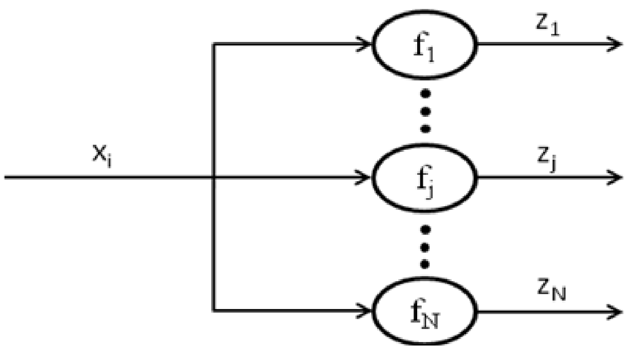



Figure 4. Functional expansion (FE) unit

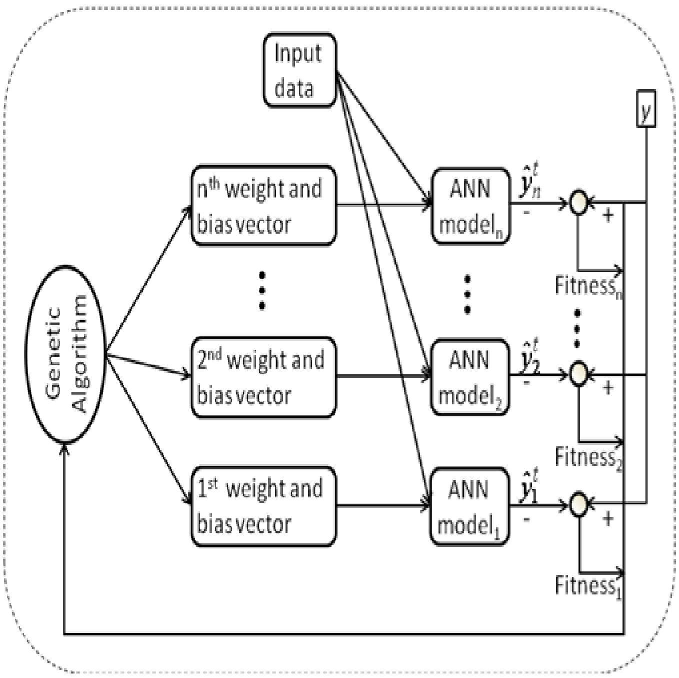



Figure 5. Architecture of ANN-GA

### *C. FLANN-GD model*

The FLANN introduces higher-order effects through nonlinear functional transforms via links rather than at nodes [45-46]. The FLANN architecture uses a single layer feed forward neural network without hidden layers. The functional expansion effectively increases the dimensionality of the input vector and hence the hyper plane generated by the FLANN provides greater discrimination capability in the input pattern space.

Let each element of the input pattern before expansion be represented as $x_i, 1 < i < d$, where each element $x_i$ is functionally expanded as $z_{ij} = f_j(x_i), 1 < \mathrm{j} < \mathrm{N}$, where N is the number of expanded points for each input element and $d$ represents the number of attributes in the data set. For designing the network, at the first instance a functional expansion (FE) unit expands each input attribute of the input data. Fig. 4 represents the FE unit of the FLANN model.

Fig. 6 shows the architecture of the FLANN-GD model. The attributes of each input pattern is passed through the FE unit. The sum of the output signals of the FE units multiplied with weights is passed on to the sigmoidal activation function of the output unit. The estimated output is compared with the target output and error signal is obtained. This error signal is used to train the model using GD technique.

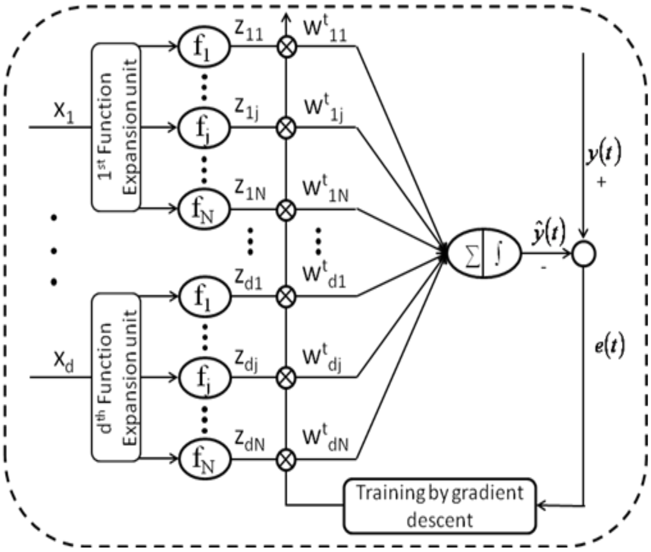



Figure 6. Architecture of FLANN-GD Network

*D. FLANN-GA model*

Fig. 7 illustrates the architecture of FLANN-GA model. The binary GA technique is employed for FLANN-GA. This model generates a set of weight and bias vectors representing a set of FLANN models. The input vector along with weights and bias values are passed on to the respective FLANN model. The absolute difference between the target $y$ and the estimated output $\hat{y}$ is treated as the fitness of the respective FLANN models. Lower fitness value is considered as the better fit individuals for the purpose of selection of individual to the next generation by the GA.

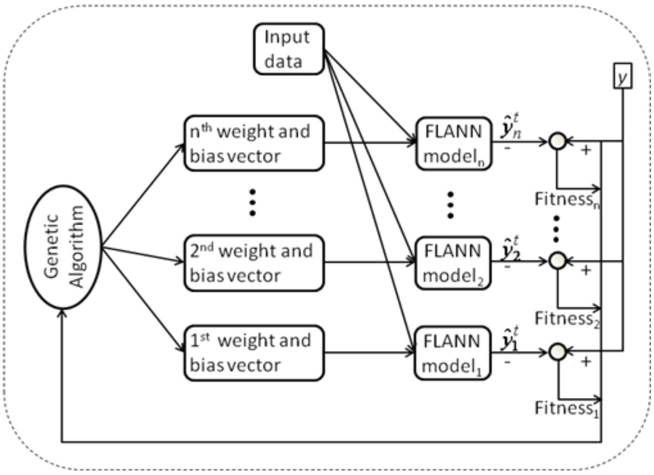



Figure 7. Architecture of FLANN-GA Network

## VII. Experimental Results and Analysis

The experiment has been carried out by a system with Intel ® core TM i3 CPU, 2.27 GHz and 2.42 GB memory. The programming language used MATLAB-2009 Version-7.8.0.347.

The networks have been employed to use the input data as the daily closing index price (S&P100) which was collected from the historical values for the BSE indices. The index values are collected during the financial year 2006 to 2010. The data are preprocessed by using different normalization methods such as Median normalization, Min-Max, normalization, Z-score, sigmoid normalization, and Decimal Scaling normalization. The normalized data are then used to form a training bed for the network model. The sliding window technique is a popular approach to deal with time series data, in where $N$ inputs and a single output slide over the whole time series. For example, given a time series $x_1, x_2, \ldots, x_{t-1}, x_t$, input vectors: $X_1 = (x_1, x_2, x_3, \ldots, x_N)$, $X_2 = (x_2, x_3, x_4, \ldots, x_{N+1})$, …, $X_i = (x_i, x_{i+1}, x_{i+2}, \ldots, x_t)$ are generated. Each input vector and a single output target value constitute one training sample for machine learning. The percentage of average absolute error is considered for the performance evaluation. The error for the $i^{th}$ data pattern is given by

$$e_i = \mathrm{abs}(t_i - y_i)$$

where $e_i$ is the error signal, $t_i$ is the target signal and $y_i$ is the calculated output. Therefore, the original financial time series will be transformed into a pattern set depending on the number of input nodes $k$ for a particular network, and each pattern will consists of the following:

- $k$ input values correspond to $k$ normalized previous values of period t: $N_{t-1}, N_{t-2}, \ldots, N_{t-k}$.
- one output value: $N_t$ (desired index value)

The models are simulated for 10 times and then average error is considered for comparative analysis of results.

Uniform crossover operator is employed for the simulation of the models associated with GA. To ensure survival of some of the best fit individuals, the elitist model of GA algorithm is employed. 10% of the individuals better fit in the generation than others are selected through elitism, and rest 90% of the individuals are selected through binary tournament selection. The parameters considered for simulation of ANN-GD, FLANN-GD, ANN-GA and FLANN-GA models are presented in Table I-Table-IV respectively.

*TABLE I. Simulated parameters of ANN-GD Model*

| Parameters | Value |
|---|---|
| Learning rate (η) | 0.03 |
| Momentum factor (α) | 0.001 |
| No. of itteration | 500 |

*TABLE II. Simulated parameters of FLANN-GD Model*

| Parameters | Value |
|---|---|
| Learning rate (η) | 0.05 |
| Momentum factor (α) | 0.001 |
| No. of itteration | 1000 |

*TABLE III. Simulated parameters of ANN-GA Model*

| Parameters | Value |
|---|---|
| Population Size | 40 |
| Maximum generation | 200 |
| Crossover probability | 0.7 |
| Mutation probability | 0.3 |

All the models associated with FLANN employees the following functional expansions.

$$z_{i1} = x_i,\ z_{i2} = \sin(x_i),\ z_{i3} = \cos(x_i),\ z_{i4} = \sin(\pi x_i),$$
$$z_{i5} = \cos(\pi x_i),\ z_{i6} = \sin(2\pi x_i),\ z_{i7} = \cos(2\pi x_i)$$

The parameters considered for simulation of FLANN-GA models are presented in Table II.

*TABLE IV. Simulated parameters of FLANN-GA Model*

| Parameters | Value |
|---|---|
| Population Size | 70 |
| Maximum Generation | 1000 |
| Crossover probability | 0.6 |
| Mutation probability | 0.04 |

Table V shows the average absolute error obtained from ANN-GD in different financial years employing different normalization technique. For 2006 data, min-max method yield minimum error. Similarly, for 2007, 2008 and 1010 data Z-score, and for 2009 data, Decimal scaling performs better than other normalization methods. For the financial year data 2008 and 2010, tanh estimator also gives better accuracy, which is 0.8930 and 0.9330 respectively.

*TABLE V. Average absolute error from ANN-GD Model*

| | 2006 | 2007 | 2008 | 2009 | 2010 |
|---|---|---|---|---|---|
| **Min-Max** | 0.0251 | 9.7928 | 0.1575 | 3.7621 | 0.2041 |
| **Decimal Scaling** | 37.3082 | 26.1459 | 25.8242 | 0.9996 | 3.4632 |
| **Z-Score** | 1.4464 | 0.3263 | 0.4126 | 1.9861 | 1.9674 |
| **Median** | 9.5643 | 2.3182 | 1.3791 | 3.1246 | 2.4632 |
| **Sigmoid** | 2.4883 | 1.4291 | 1.1209 | 1.9726 | 2.7072 |
| **Tanh Estimator.** | 1.4085 | 1.0537 | 0.8930 | 1.6733 | 0.9330 |
| **MAD** | 5.8253 | 14.0218 | 9.2381 | 11.7129 | 3.2833 |

The average absolute error obtained from ANN-GA in different financial years employing different normalization technique is presented in Table VI. Here, for financial year 2006, decimal scaling, median, sigmoid and thah estimator give remarkable result. For 2009 data Decimal Scaling, median and sigmoid can be appropriate choice. For financial year 2007, 2008, and 2010 data sigmoid normalization technique perform better in comparison to other methods.

*TABLE VI. Average absolute error from ANN-GA Model*

| | 2006 | 2007 | 2008 | 2009 | 2010 |
|---|---|---|---|---|---|
| **Min-Max** | 24.1156 | 10.7342 | 25.2532 | 34.7466 | 41.6255 |
| **Decimal Scaling** | 0.0748 | 21.4685 | 1.2292 | 0.1643 | 0.9501 |
| **Z-Score** | 19.0459 | 9.2909 | 17.9199 | 17.1231 | 15.0047 |
| **Median** | 0.2427 | 0.5573 | 0.2960 | 0.4108 | 0.0980 |
| **Sigmoid** | 0.3084 | 0.3337 | 0.2100 | 0.4957 | 0.0544 |
| **Tanh Estimator.** | 0.9785 | 1.5530 | 0.0936 | 1.3723 | 1.0535 |
| **MAD** | 7.4247 | 12.6218 | 12.2381 | 10.5120 | 2.2632 |

The performance of FLANN-GD in different financial years employing different normalization technique is presented in Table VII. For the data of all the years sigmoid normalization method yields better prediction accuracy in comparison to others.

*TABLE VII. Average absolute error from FLANN-GD model*

| | 2006 | 2007 | 2008 | 2009 | 2010 |
|---|---|---|---|---|---|
| **Min-Max** | 18.906 6 | 2.9862 | 23.9514 | 22.9164 | 34.6894 |
| **Decimal Scaling** | 15.851 1 | 5.8506 | 16.3984 | 16.8349 | 16.4095 |
| **Z-Score** | 14.286 5 | 5.5712 | 16.2320 | 15.7592 | 14.7994 |
| **Median** | 4.2594 | 3.3645 | 1.3629 | 0.8611 | 3.3567 |
| **Sigmoid** | 0.2315 | 0.3380 | 0.3835 | 0.3946 | 1.8378 |
| **Tanh Estimator.** | 0.6382 | 0.0239 | 1.3961 | 0.9763 | 0.0351 |
| **MAD** | 7.8651 | 10.1218 | 17.1382 | 13.0124 | 5.1834 |

Table VIII shows the performance of FLANN-GA in different financial years employing different normalization techniques. For the data of all the years, Decimal scaling method shows better prediction capabilities than others.

*TABLE VIII. Average absolute error from FLANN-GA Model*

| | 2006 | 2007 | 2008 | 2009 | 2010 |
|---|---|---|---|---|---|
| **Min-Max** | 6.5898 | 5.3525 | 3.0595 | 10.4887 | 11.7636 |
| **Decimal Scaling** | 0.0058 | 0.1049 | 0.0055 | 0.0061 | 0.0111 |
| **Z-Score** | 1.1345 | 0.1624 | 2.6011 | 2.1954 | 2.0296 |
| **Median** | 0.0990 | 2.8062 | 0.2037 | 0.2042 | 0.0958 |
| **Sigmoid** | 0.1008 | 0.1078 | 0.1858 | 0.1759 | 0.1762 |
| **Tanh Estimator.** | 0.0735 | 1.0092 | 0.0936 | 0.3928 | 0.9544 |
| **MAD** | 6.1244 | 10.0258 | 8.5380 | 10.3392 | 1.2334 |

The average absolute errors generated by all normalization techniques for five financial year data from 2006 to 2010 are added. The summed error is then multiplied by 100. The relative performance of each normalization method with others is then calculated. The relative errors of normalization techniques generated from each forecasting models are shown in Fig. 8-11. Fig. 8 presents the relative error generated by ANN-GD model. It is observed that on an average both Z-Score and tanh give minimal relative error i.e. 3% in comparison to others. Decimal scaling method may not be adopted for this model. which yields maximum relative error i.e. 49%. For ANN-GA, the model performance by using different normalization methods is shown in Fig. 9. In this case the best choice can be sigmoid which gives 1.4022 followed by median and tanh estimator. The min-max method which yields inferior result may not be adopted for this model. The relative error generated from normalization techniques by FLANN-GD model is presented in Fig. 10. Again the sigmoid beats other method by giving nominal relative error i.e. 1% only. The tanh and median can be the alternate choice with 1% and 4% relative error respectively. Surprisingly the decimal scaling when used in case of FLANN-GA gives superior result i.e. 0.1334 followed by sigmoid which gives 1% error and tanh with 3%. The min-max method may not be used for both FLANN-GD and FLANN-GA with 33% and 42% error respectively.

Similarly the average absolute errors generated by all four models using a particular normalization technique are added and then the sum is multiplied by 100. The relative performance of each normalization method in comparison to others is calculated and the result is presented in Fig. 12. The sigmoid and tanh may be the first choice with 2% relative error each followed by the median normalization with 4%.

The performance of each forecasting model is analyzed in Fig. 13. In the same way the average error of all normalization techniques are added and then multiplied by 100. The performance of each model in comparison to other model is calculated and shown in Fig. 13. The FLANN-GA model gives consistent result for all the financial year data set as well as by considering all normalization methods used in this experimental work with average error of 10%. Hence the FLANN-GA model may be claimed as the prominent forecasting model for this stock index forecasting problem.

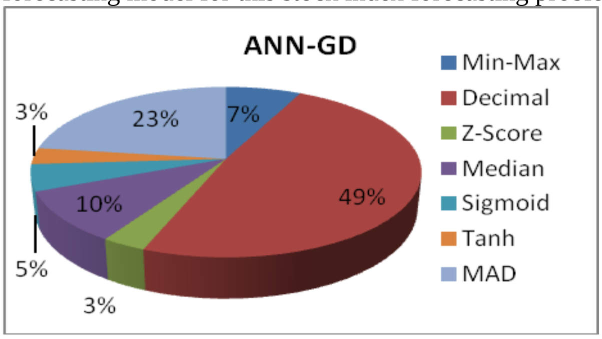



Figure 8. Relative error from ANN-GD model

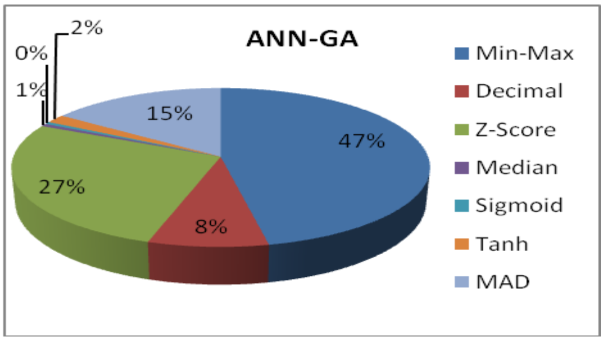



Figure 9. Relative error from ANN-GA model

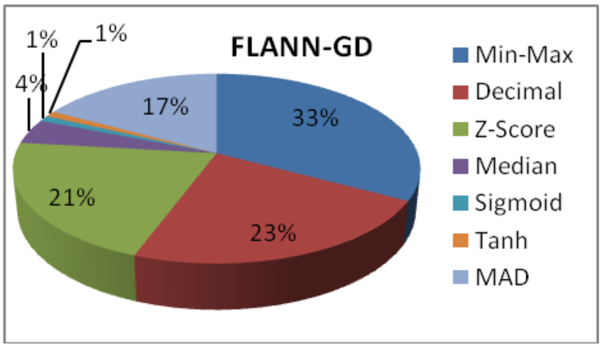



Figure 10. Relative error from FLANN-GD model

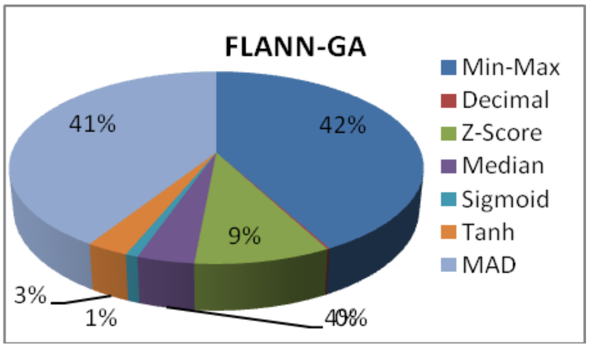



Figure 11. Relative error from FLANN-GA model

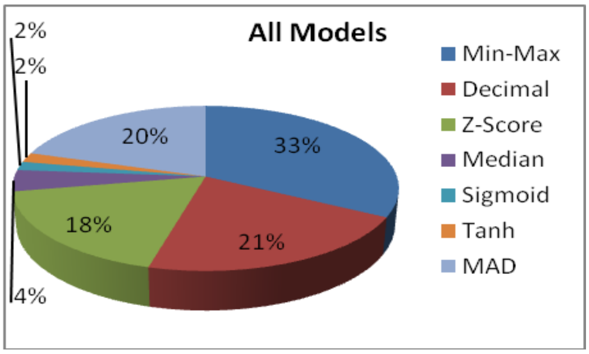



Figure 12. Comparision of relative errors from all models

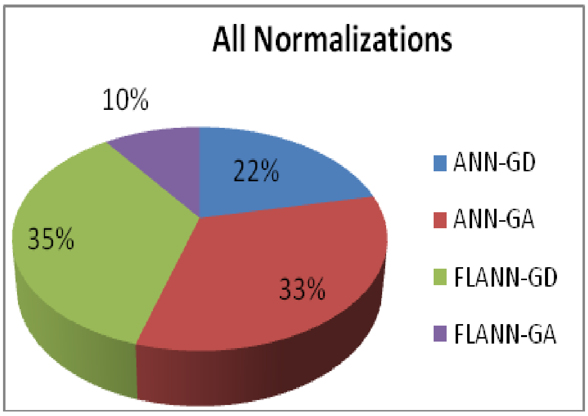



Figure 13. Relative performance of all models considering all normalization methods

The above four forecasting models have been implemented and then employed to forecast each year data and the graphs are obtained. For example, four such graphs shown in Fig. 14-17 are obtained by employing ANN-GD, ANN-GA, FLANN-GD and FLANN-GA respectively. In this case the above models used 2009 financial year data normalized by the sigmoid method.

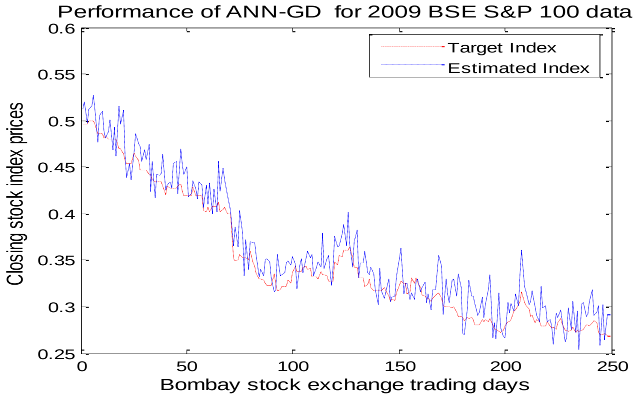



Figure 14. Average percentage of error by ANN-GD model for financial year 2009

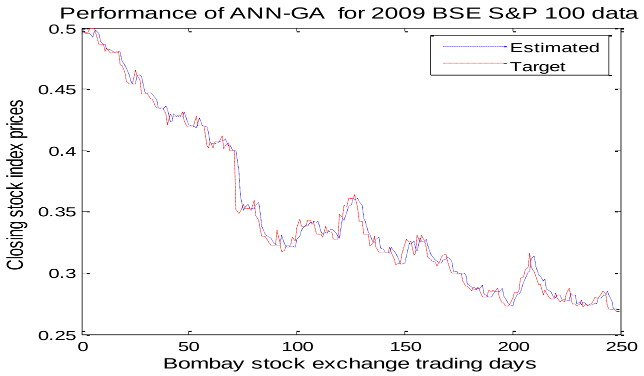



Figure 15. Average percentage of error by ANN-GA model for financial year 2009

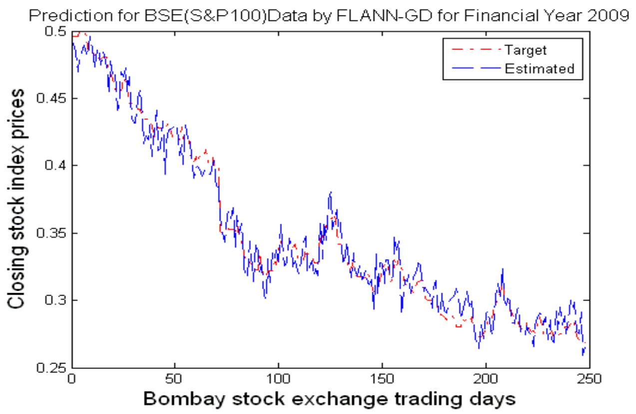



Figure 16. Average percentage of error by FLANN-GD model for financial year 2009

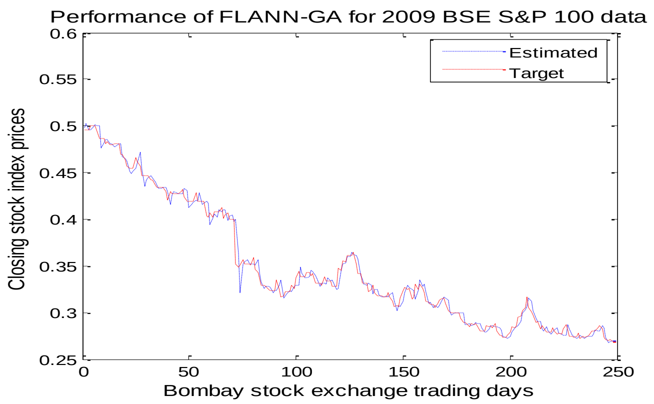



Figure 17. Average percentage of error by FLANN-GA model for financial year 2009

## VIII. Conclusion

In this research work, four ANN based forecasting models are considered for forecasting the Indian stock market. Various normalization methods such as Decimal Scaling, Median, Z-score, Min-Max, Sigmoid, Tanh estimator and MAD have been considered for the evaluation of prediction accuracy of two ANN models as well as two hybrid forecasting models. The investigation has been carried out on five years daily closing price of Bombay Stock Exchange. The four different models are termed as ANN-GD, ANN-GA, FLANN-GD and FLANN-GA. The overall impact of data preprocessing specifically data normalization has been systematically evaluated. Theoretically as well as empirically it is observed that, data preprocessing affects prediction accuracy significantly [48]. From the experimental study, it is observed that a single normalization method is not superior over all others. However, methods like Z-Score, MAD and Min-Max performed poorly most of the time and may not be preferred for models other than ANN-GD. The sigmoid and tanh estimator method not also performed better but also shows a consistent performance on the data over the years for different models. Again, sigmoid method does not perform better than decimal scaling for FLANN-GA model. Therefore, it may not wise for the users to adopt a single normalization technique, rather alternate methods should be taken into consideration to obtain better result. This empirical and experimental work can be concluded with the fact that, raw data may be preprocessed, particularly when predicting non-stationary data before feeding it to the training model and appropriate data normalization method should be adopted carefully in order to improving the predictability of the soft computing models.

Finally, further research may includes, application of other data preprocessing approaches that can be used to analyze impacts on forecasting models.

## References

[1] P. Zuo, Y. Li, and J. M. SiLiang, "Analysis of Noninvasive Measurement of Human Blood Glucose with ANN-NIR Spectroscopy," *International Conference on Neural Network and Brain (ICNN&B'05)*, vol.3, pp.1350-1353, 13-15 Oct. 2005.

[2] A. S. Miller, B. H. Blott, and T. K. Hames, " *Review of neural network applications in medical imaging and signal processing*," Med. Biol. Eng. Comput., pp. 449-464, 1992.

[3] N. Maglaveras, T. Stamkopoulas, C. Pappas, and M. Strintzis, "*An Adaptive back-propagation neural network for real time ischemia episodes detection Development and performance using the European ST-T databases*," IEEE Trans., Biomed Engng. pp. 805-813, 1998.

[4] P. B. Sivakumar, V. P. Mohandas, "Performance Analysis of Hybrid Forecasting models with Traditional ARIMA models – A Case Study on Financial Time Series Data," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 2, pp. 187-211, 2010.

[5] K. Matsui, and H. Sato, "Neighborhood Evaluation in Acquiring Stock Trading Strategy Using Genetic Algorithms," *International Journal of Computer*

*Information Systems and Industrial Management Applications*, vol. 4, pp. 366-373, 2012.

[6] V. Ravi, R. Lal, and N. R. Kiran, " Foreign Exchange Rate Prediction using Computational Intelligence Methods", *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 4, pp. 659-670, 2012.

[7] Z. Mustaffa and Y. Yusof, "A Comparision of Normalization Techiques in Predicting Dengue Outbreak", *International Conference on Business and Economics Research, IACSIT Press*, vol. 1, 2011.

[8] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, "Data Preprocessing for Supervised Learning," *Computer Science*, vol. 1, pp. 1306-4428, 2006.

[9] L. AI Shalabi and Z. Shaaban, "Normalization as a Preprocessing Engine for Data Mining and the Approach of Preference Matrix," *International Conference in Dependability of Computer Systems*, pp. 207-214, 2006.

[10] T. Jayalakshmi, and A. Santhakumaran, "Statistical Normalization and Back Propagation for Classification," *International Journal of Computer Theory and Engineering*, vol. 3, no.1, pp.1793-8201, Feb.2011.

[11] J. Mohammad-saleh, and B. S. Hoyle, "Improved Neural Network performance using Principle Component Analysis on Matlab," *International Journal of the Computer, the Internet and Management*, vol.16, No. 2, pp. 1-8, 2008.

[12] E. Ogasawara, L. C. Martinez, D. de Oliveira, G. Zimbrao, G. L. Pappa, and M. Mattoso, "Adaptive Normalization: A Novel Data Normalization Approach for Non-Stationary Time series," *International Joint Conference on Neural Network (IJCNN)*, IEEE, 2010.

[13] S. Haykin, *Neural Networks and Learning Machines*, 3 ed. Prentice Hall, 2008.

[14] W. Shangfei, Z. Peiling, "Application of radial basis function neural network to stock forecasting," *Forecasting*, vol. 6, pp. 44-46, 1998.

[15] Y. K. Kwon, B. R. Moon, "A Hybrid Neuro Genetic Approach for Stock Forecasting," *IEEE trans. on Neural Networks*, vol. 18, no. 3, May 2007.

[16] Z. Guangxu, "RBF based time-series forecasting," *Journal of Computer Application*, vol. 9, 2005, pp. 2179-2183.

[17] L. Yu and Y. Q. Zhang, "Evolutionary Fuzzy Neural Networks for Hybrid Financial Prediction," *IEEE trans. on Systems, Man, and Cybernetics—Part C: Applications and Reviews*, vol. 35, no. 2, May 2005.

[18] S. C. Nayak, B. B. Misra, and H. S. Behera, "Index Prediction Using Neuro-Genetic Hybrid Networks: A Comparative Analysis of Perfermance," *International Conference on Computing Communication and Application, (ICCCA-2012)*, IEEE Xplore, 2012, doi: 10.1109 / ICCCA.2012.6179215.

[19] S. C. Nayak, B. B. Misra, and H. S. Behera, "Stock Index Prediction with Neuro-Genetic Hybrid Techniques," *International Journal of Computer Science and Informatics, (IJCSI)*, vol. 2, Issue-3, pp. 27-34, 2012.

[20] G. Zhang, B. E. Patuwo, and M, Y. Hu, "Forecasting with artificial neural network: the state of the art," *International Journal of Forecasting*, Vol. 14, pp. 35-62, 1998.

[21] J. M. Bates, and W. J. Granger, "The combination of forecasts," *Operation Research*, Vol. 20, pp. 451-468, 1969.

[22] R. Clemen, "Combining forecasts: a review and annotated bibliography with discussion," *International Journal of Forecasting*, Vol. 5, pp. 559-608, 1989.

[23] P. F. Pai, and C. S. Lin, "A hybrid ARIMA and support vector machines model in stock price forecasting," *Omega*, Vol. 33, pp. 497-505, 2005.

[24] K. Y. Chen, and C. H. Wang, "A hybrid ARIMA and support vector machines in forecasting the production values of the machinery industry in Taiwan," *Expert Systems with Applications*, Vol. 32, pp. 254-264, 2007.

[25] Z. J. Zhou, and C. H. Hu, "An effective hybrid approach based on grey and ARMA for forecasting gyro drift," *Chaos, Solitons and Fractals*, Vol. 35, pp. 525-529, 2008.

[26] L. Tian, and A. Noore, "On-line prediction of software reliability using an evolutionary connectionist model," *Journal of Systems and Software*, Vol. 77, pp. 173-180, 2005.

[27] G. Armano, M. Marchesi, and A. Murru, "A hybrid genetic-neural architecture for stock indexes forecasting," Information Science, Vol. 170, pp. 3-33, 2005.

[28] H. Kim, and K. Shin, "A hybrid approach based on neural networks and genetic algorithms for detecting tmporal patterns in stock markets," *Applied Soft Computing*, Vol. 7, pp. 569-575, 2007.

[29] L. Yu, S. Wang, and K. K. Lai, "A novel nonlinear ensemble forecasting model incorporating GLAR and ANN for foreign exchange rates," *Comput. Oper. Res.*, Vol. 32, pp.2523-2541, 2005.

[30] R. Tsaih, Y. Hsu, and C. C. Lai, "Forecasting S&P 500 stock index futures with a hybrid AI system," *Decis. Support System*, Vol. 23, pp. 161-174, 1998.

[31] D. K. Wedding, and K. J. Cios, "Time series forecasting by combining RBF networks, certainty factors, and Box-Jenkins model," *Neurocomputing*, Vol. 10, pp. 149-168, 1996.

[32] F. M. Tseng, H. C. Yu, and G. H. Tzeng, "Combining neural network model with seasonal time series ARIMA model," *Technol. Forecast. Soc. Changes*, Vol. 69, pp. 71-78, 2002.

[33] I. Ginzburg, and D. Horn, "Combining neural networks for time series analysis," *Adv. Neural Inform. Process. Syst.*, Vol. 6, pp. 224-231, 1994.

[34] P. Chang, C. Liu, and Y. Wang, "A hybrid model by clustering and evolving fuzzy rules for sales decission supports in printed circuit board industry," *Decis. Support Syst.*, Vol. 42, pp. 1254-1269, 2006.

[35] Y. Lin, and W. G. Cobourn, "Fuzzy system models combining with nonlinear regression for daily ground-level ozones predictions," *Atomspheric Environment*, Vol. 41, pp. 3502-3513, 2007.

[36] K. Huarng, and T. H. K. Yu, "The application of neural networks to forecast fuzzy time series," *Physica A.*, Vol. 336, pp. 481-491, 2006.

[37] M. Khashei, M. Bijari, and G. A. Raissi Ardali, "Improvement of auto-regressive integrated moving average models using fuzzy logic and

artificial neural networks," *Neurocomputing*, Vol. 72, pp. 956-967, 2009.

[38] F. M. Tseng, G. H. Tzeng, H. C. Yu, and B. J. C. Yuan, "Fuzzy ARIMA model for forecasting the foreign exchange market," *Fuzzy Sets Syst*., Vol. 118, pp. 9-19, 2001.

[39] P. Roychowdhury, and K. K. Shukla, "Incorporating fuzzy concepts along with dynamic tunneling for fast and robust training of multilayer perceptrons," *Neurocomputing*, Vol. 50, pp. 319-340, 2003.

[40] B. Graham and D. Dodd, *Security Analysis. New York and London*, McGraw-Hill, 1934.

[41] Kirkpatrick and Dahlquist, *Technical Analysis: The Complete Resource for Financial Market Technicians*, Financial Times Press, pp. 3, 2006.

[42] P.Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*, Addison Wesley, 2005.

[43] J. Sola, and J. Sevilla, "Importance of Input Data Normalization for the Application of Neural Networks to Complex Industrial problems," *IEEE Transactions on Nuclear Science*, vol.44, No.3, pp.1464-1468, 1997.

[44] D. E. Goldberg, *Genetic Algorithms in Search Optimization and Machine Learning*, Pearson Education, 2003.

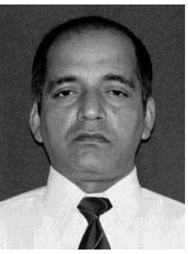
[45] M. Klassen, Y. H. Pao, and V. Chen, "Characteristics of the functional link net: a higher order delta rule net," *IEEE International Conference on Neural Networks*, 1988, pp. 507 – 513.

[46] Pao YH, Takefuji Y (1992) Functional link net computing: theory, system, architecture and functionalities. *IEEE Comput*, pp 76–79.

[47] S. C. Nayak, B. B. Misra and H. S. Behera, "Evaluation of Normalization methods on neuro-genetic models for stock index forecasting," *World congress on Information and Communication Technologies, WICT-2012*, IEEE Xplore, 2012, doi: 10.1109/WICT.2012.6409147.

## Author Biographies

**S. C. Nayak** has received a M. Tech (Computer Science) degree from Utkal University, Bhubaneswar in 2011. He is pursuing PhD (Engineering) at VSS University of Technology, Burla, India. Currently he is working in Computer Science & Engineering department at Silicon Institute of Technology, Bhubaneswar, India. His area of research interest includes data mining, soft computing, evolutionary computation, financial time series forecasting. He has published 6 research papers in various journals and conferences of National and International repute.



**B. B. Misra** has completed his B. Text. degree in 1984 from Kanpur University, UP, India, M. Tech. (Computer Science) in 2002 from Utkal University, Bhubaneswar, Orissa, India and Ph.D. (Engineering) in 2011 from Biju Pattanaik University of Technology. His areas of interests include data mining, soft computing, wireless sensor network, bioinformatics. He has published two books, three book chapters, and more than 60 papers in different journals and conferences of National and International repute. Presently he is continuing as Dean Research at Silicon Institute of Technology, Bhubaneswar.

**H. S. Behera** has completed his M. E (Computer Science & Engineering) from National Institute of Technology, Rourkela, India, and PhD (Engineering) from Biju Pattanaik University of Technology. His area of interest includes data mining and soft- computing, software engineering, distributed system. Dr. Behera has more than 50 research publications in National and International journals and conferences. Currently he is working as Reader, department of computer science & engineering, VSS University of Technology, Burla, India. He has more than 16 years experiences in teaching as well as research.