A Synopsis

on

"Stock Price Direction Prediction Applying Soft Computing"

Submitted to:

Gujarat Technology University

For the Degree of

Doctor of Philosophy

In

Computer Engineering

By:

Amit M. Panchal

Enrollment No. 119997107006

Computer Engineering

Guided By

Dr. Jayesh M. Patel

Associate Professor MCA Programme, Acharya Motibhai Patel Institute of Computer Studies, Ganpat University

Foreign Co-Supervisor:

Dr. Ramakrishna Thurimella

Department of Computer Science, University of Denver, U.S.A.

Table of Contents

Research Title

Stock Price Direction Prediction Applying Soft Computing

Abstract

The stock market is a complex, non-linear, non-stationary, chaotic and dynamic system. This research work proves that properly tuned artificial neural network model can have the capability to efficiently solve non-linear time-series such as stock market indices. This research work proposes generalized stock direction prediction model (GSDPM) that will be applicable to any stock market index and/or stock securities. The design of the model is simple, which will give an advantage to other researchers for easily developing their own model. Feed forward three-layer back-propagation network, Elman back-propagation network, and cascade-forward back-propagation network are selected to develop the model. The collected datasets are divided into three subsets, training dataset, holdout dataset and testing dataset, which will be used for model training, testing model reliability and testing model prediction performance respectively. The research work proposes a heuristic based technique to select a reliable tuned model using holdout dataset. Due to multi-parameters involved, different architectures and a huge variety of stock securities across the globe, it is difficult to find an identical model in literature for model performance comparison, and hence researchers found difficult to compare the prediction performance validity of their model. Here we have proposed Succession Prediction Ratio that will help where the identical model does not available for validating the model. The research work proposes the model prediction performance validation using Success Prediction Ratio that will be helpful to other researchers to validate their model's prediction performance.

1. Brief description on the state of the art of the research topic

Stock security price is highly dependent on various parameters such as company fundamental, stock demand supply, government policy, global policy, inflation, interest rate etc. There is always uncertainty about these parameters, and hence predicting the stock market is a challenging task. Even though financial time-series is random, there may exist certain fraction, in form of consistent patterns, which are not random. If we can identify this non-random pattern, then the prediction is possible even in random series.

Widely used techniques to predict stock securities are fundamental analysis and technical analysis. Fundamental analysis is the material study of a company in terms of its product sales, company balance sheet, manpower, quality, infrastructure etc. to understand its influence in the stock market and ultimately its profit on investment [1]. Fundamental analysis is costly due to the involvement of human and economic resources as well as suitable only for long-term investment, so that it is not suitable for this research work. Technical indicators are a fundamental part of technical analysis. The technical indicator has been derived by applying a mathematical formula to security historical price data. Any combination of the open, high, low or close over a period of time can be included in price data. This is a very popular approach used to predict the market because of its only required historical price data. However, the problem of this analysis is that the trading rules extracted from chart analysis by technical analyst are highly subjective. As a result different technical analysts extract different trading rules studying the same charts. This research work uses technical analysis approach where the artificial neural network (ANN), one of the popular soft computing tools, does the job of technical analyst more accurately. This will remove the limitation of subjective analysis of different human technical analysts.

It has been found from many literatures, soft computing based proposed stock prediction model will apply to specific stock index and/or stock securities. ANN approaches have suffered from difficulties with generalization and producing models that can over-fit the data [2]. This may be due to random nature of stock securities and multi-parameters architecture of the artificial neural network. In ANN, selecting proper parameters is always challenging for researchers. Changing one or more parameters of selected ANN model will affect model prediction performance. Unfortunately, there is no absolute method available for specifying of ANN’s parameters. The number of hidden layer’s node stores important knowledge regarding prediction. In this research work, total number of hidden layer’s nodes are empirically selected for ANN by applying one epoch training on holdout dataset. Proposed GSDPM output performance is tested against percentage prediction and percentage rate of return (ROR). The Validity of models is tested against random generated data buy/sell signal.

2. Literature Review

2.1. Work done

The study[3] has concluded that soft computing methods successfully solve non-linear problems compared to hard computing. Artificial neural network, fuzzy logic and probabilistic methods make a solid set (either in its pure form or hybrid form). These methods can effectively solve non-linear problems where ARMA or GARCH econometric modeling failed. There exist literatures [4]–[8] which proved that artificial neural network (ANN) can reasonably predict stock price direction. The research study [9] by Breazak, measures forecasting performances of feed-forward neural network and recurrent neural networks (NN) by training with different learning algorithms. The authors compared forecasting performance of these NN with performance of NN using the Mackey-Glass nonlinear chaotic time series. Breazak, once again confirmed potential efficiency of NN in modeling and predicting nonlinear problem such as financial time-series. The stock market prices prediction done by Ramon Lawrence [6], reveals the ability of NN to find out patterns in nonlinear time-series more accurately that other current forecasting tools. The study proposes that in 91 per cent cases NN correctly predicted the future price trend as compared to 74 percent using multiple discriminate analysis (MDA).

2.2. Methodology

The research study [4] by G. Atsalakis and K. Valavanis have surveyed more than 100 published articles that focus on soft computing techniques applied to forecast the financial market. Authors made three classifications: input data, forecasting methodology and performance measures. Five summary tables were represented in this study viz. author's respective model for stock market (sample data), input variables to the stock market model (technical indicators and/or fundamental parameters), author's specific approach (processing techniques), summarized comparisons made against other techniques and summary of performance measure techniques. This study expressed that predicting market is possible in many ways, but no identical model is available. Yakup Kara [10] predicted the direction of movement in the daily Istanbul Stock Exchange (ISE) National 100 Index. The model was based on two ANN architectures: ANN and support vector machines (SVM). Kara compared the prediction performance of ANN with SVM and concluded that average performance of ANN model (75.74%) was better than that of SVM (71.52%). Patel et al. [11] study compared four predictive models: naive Bayes, random forests, ANNs and SVM on Indian stock exchange benchmark index CNX Nifty, S&P BSE Sensex indices. This study used two approaches for these models. First approach focused on input data involvement in computation of ten technical parameters using stock trading data (open, high, low & close prices). On the other hand, second approach focused on representing trend deterministic data of those technical parameters. Author concluded that, the performance of all the prediction models improve when those technical parameters have represented as trend deterministic data.

2.3. Observation found

Soft computing techniques effectively solve stock market prediction problem [3], [12]. More than 100 researchers survey [4] proves that there is no absolute model for stock market prediction. Back-propagation (BP) based neural network is one of the most popular techniques [13]–[20] in the field of artificial neural network for analysis of stock data, which we have applied here to develop GSDPM model in this research work. With changing market, the nature and complexity of time series also differs. The prediction accuracy highly depends on the selection of input variables, model topology and model parameters. There is a possibility to make generalized price prediction model, which will be giving more returns compared to random buy/sell.

2.4. Limitation of the existing solution

It has been found from various research studies, that to predict stock securities, applied soft computing model stores pattern by learning from historical data. The approaches used by these models are either technical analysis or fundamental analysis. The technical analysis approach uses historical data, assuming that history predicts the future. The technical analysis becomes less accurate or sometime fail when market changes drastically. Fundamental analysis involves human resource, the limitation of a human such as bias, human error to interpretation data, lack of team coordination, etc., that will makes less accurate prediction. The soft computing model requires large training data. If enough historical data is not available, prediction performance will be affected. Also there is an assumption that testing dataset has co-relation with training dataset. If market participant changes or market condition changes, then re-training is required for consitent performance.

3. Objectives of the research work

- To review work done in the field of predicting price direction of stock securities.
- To identify an important feature of stock securities, which will be helpful in predicting direction of stock price.
- To design GSDPM based on soft computing technique, which can guide stock market participants to buy/sell stock securities.
- To identify number of hidden layer's node for ANN.
- To identify the reliable tuned model for GSDPM.
- To measure and test the performance of price direction prediction model.
- To validate GSDPM model output performance.

4. Scope of the research work

The scope of this research work is related to financial domain and soft computing technique for predicting stock securities price direction. In this research, artificial neural network architecture is used to develop price direction predicting model. The neural network parameters are selected by an empirically and prominent research study. Across the globe, popular benchmark indices are selected for testing prediction performance of proposed model. The stock data used, has been the daily closing price of respective stock securities. Model prediction performance is measured in terms of percentage prediction and rate of return. The research work is not limited to stock indices but it also provides scale to predict any stock securities.

5. Original contribution by the thesis

The research identifies important feature of stock securities, which will help to predict direction of stock price. This research work provides effective data sampling distribution for the model training, tuned model's reliability testing and model prediction testing. The research proposes heuristic based technique to find reliable tuned model using holdout dataset. Here we propose simplified design of developing generalized stock direction prediction model. Proposed GSDPM has proved that only changing few parameters and retrain other parameter of the artificial neural network model, it is possible to develop generalized prediction model, which will efficiently predict stock securities price direction. This will give advantages to others researchers to save their time for selection of model parameter. After developing generalized stock security direction prediction model, the real challenge is to test its prediction validity because of two reasons. First, it is difficult to find an identical model to compare their validity and second, due to a variety of stock securities. To overcome this problem here we propose SPR, which will help to validate the model against randomize prediction.

6. Research Methodology

Soft computing based artificial neural network architecture and technical analysis are used as an approach to this research work. First, we collect data samples from reputed stock exchanges. The research data are the daily closing price of trade stock securities. The entire collected data sample divides to subset as per model's processing requirement. We train model until it tunes properly. Using holdout dataset, the reliable tuned (properly tuned) model is selected from generated tuned models. Testing data is applied to reliable tuned model and output of the model is evaluated. Based on the output of processing unit, the decision will be taken whether stock security direction is up or down. For testing model prediction performance, testing data is fed to model and model performance is analyzed. Proposed GSDPM performance will test against percentage prediction and percentage rate of return (ROR). To validate the model, Success Prediction Ratio (SPR) is calculated and validity of model will test against random generated buy/sell signal.

6.1. Generalized Stock Direction Prediction Model design

The GSDPM design is shown in Figure 1. The processing data samples are from financial domain. Model design contains six units: 1) Stock Database Selection, 2) Raw Input Data Sample, 3) Features Extraction, 4) Data Sampling 5) Processing Framework 6) Output

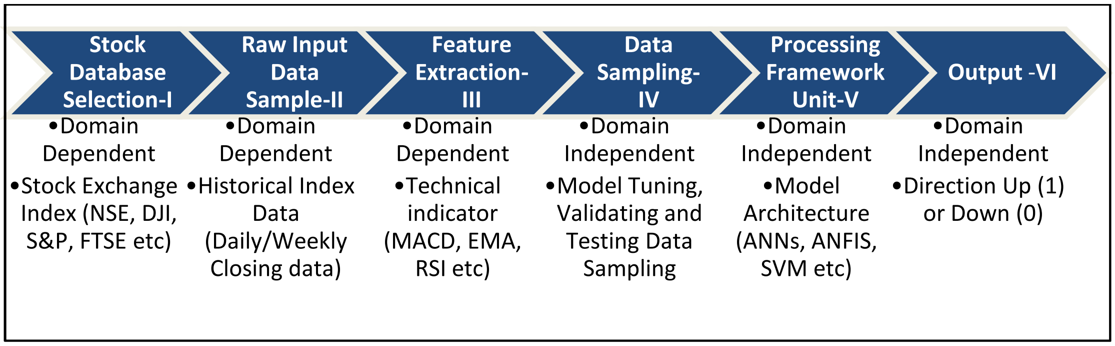

Stock Database Selection-I
•Domain Dependent
•Stock Exchange Index (NSE, DJI, S&P, FTSE etc)
Raw Input Data Sample-II
•Domain Dependent
•Historical Index Data (Daily/Weekly Closing data)
Feature Extraction-III
•Domain Dependent
•Technical indicator (MACD, EMA, RSI etc)
Data Sampling-IV
•Domain Independent
•Model Tuning, Validating and Testing Data Sampling
Processing Framework Unit-V
•Domain Independent
•Model Architecture (ANNs, ANFIS, SVM etc)
Output -VI
•Domain Independent
•Direction Up (1) or Down (0)

Figure 1. Generalized Stock Direction Prediction Model (GSDPM)

6.1.1. Stock database selection

Here we can select benchmark index and/or traded stock securities that are domain specific to selected stock exchange. The database may be from the single stock exchange or from multiple stock exchanges. In this research, we used benchmark stock indices from multiple stock exchanges.

6.1.2. Raw input data sample

This unit contains processing of raw price data sample. The raw price data sample contains the closing price of security data with specific interval. The interval may be minute, hours, day, weekly etc. every stock exchange provides real time as well as historical data to authorized users. The real-time data will used to initiate trade while historical market data can be used to project pricing trends. Each closing data sample of traded securities includes minimum attributes: high price, low price, open price, closing price and traded volume. These attributes of stock securities contain important features of time series.

6.1.3. Features extraction

Technical indicators are selected feature of the model. Stock raw data point does not have any relation with previously raw data point because they are independent with each other. As price moves up or down, accordingly technical indicators generates various patterns. These patterns contain important knowledge regarding the future movement of price. Technical indicator makes relation between these raw data points. At a data point, combining multiple technical indicators form a specific pattern. Technical analyst assumes that these patterns are repeats and hence, accordingly future price will follow. The model learns this knowledge during training and stores into its memory.

6.1.4. Data sampling

This unit is responsible for distribution of entire collected dataset. Distributed datasets are used for model learning, model over-fitting testing and model prediction validation by applying domain independent method. The prediction performance of the model depends on how data distribution is done. In this research, the entire collected data is divided into three independent subsets. The first subset, containing larger data samples, is used for training model. Second, holdout dataset is used for setting the model parameters and to find reliable tuned model. The third testing dataset is used for testing prediction performance.

6.1.5. Processing framework

The processing framework consists of domain independent soft computing based architecture. The architecture may be either specific or hybrid. This will be capable of solving the non-linear problem. This unit involves selection of architecture parameters, model learning, model over-fit testing, identification of the reliable tuned model and generation of an output based on input patterns appeared. Inputs are fed to model, the model will match input pattern with their stored memory and output is given. The model processing speed, prediction accuracy, and validity to generalize acceptance depend on the selection of these parameters. In this research work, artificial neural networks are selected as soft computing architecture.

6.1.6. Output

After processing input data, output of the model is used to predict next day price direction by some domain independent technique. Depending on the type of output produced by the model, appropriate method will be applied, which will directly or indirectly predict the future direction of stock security.

6.2. Generalized Stock Direction Prediction Model implementation

6.2.1. Select stock database

In this research work, ten different top benchmark stock indices are selected from reputed stock exchanges across the globe as illustrated in Table 1.

Table 1. Benchmark index data collection and description.

Sr. No.	Index	Description	Country	Total data sample (Years)	Data collected from
1.	NIFTY	National Stock Exchange of India	India (Asia)	3368	www.nseindia.com
2.	SENSEX	Bombay Stock Exchange	India (Asia)	3325	www.nseindia.com
3.	NIKKIE	Nikkei Stock Average	Japan (Asia)	3325	www.stooq.com
4.	SSE	Shanghai Stock Exchange	China (Asia)	3294	www.stooq.com
5.	SNP500	Standard & Poor's 500	United State	3913	www.stooq.com
6.	NASDAQ	NASDAQ Composite	United State	3915	www.stooq.com
7.	DJI	Dow Jones Industrial Average	United State	3913	www.stooq.com
8.	FTSE	London Stock Exchange	England (Europe)	3683	Yahoo finance
9.	CAS	French stock market index	France(Europe)	3733	Yahoo finance
10.	DAX	DeutscherAktien Index	Germany(Europe)	3706	www.nseindia.com

6.2.2. Raw input data sample

The research input raw data sample used in this study is the daily closing price of benchmark indices from their respective stock exchange. Each data sample contains basic attributes: high, low, open, close and traded volume of a respective stock security.

6.2.3. Feature extraction

Five technical indicators for each case are used as input variables to the neural network model compared to research study [10], [21]–[23], where more than nine technical indicators were used as input to the neural network. The selection of five technical indicators is done by the review of domain experts and prior researches [24]–[26]. Technical indicators are broadly classified trend following, price oscillators and price momentum. Some technical indicators are effective under cyclical markets or consolidation market and others perform better during trading markets [27]. While selecting technical indicators as input to the model here we make sure that, each category of indicators is covered and redundancy is avoided. Table 2 shows the description for selected five technical indicators and their categories as an input variable to the neural network for this research work.

Table 2. Technical Indicators Description

Indicator	Description	Category	Formula
MACD	Difference of two EMAs that shows a stock's momentum and direction	Trend oscillator	$\text{MACD}(n)_{t-1} + \frac{2}{n+1}\ X\ (\text{DIFF}_t\text{-MACD}(n)_{t-1})$
Relative Strength Index (RSI)	Shows how strongly a stock is moving in its current direction	Trend Strength oscillator.	$100 - \frac{100}{1 + \frac{\sum_{i-0}^{n-1} U_{t-1}/n}{\sum_{i-0}^{n-1} D_{t-1}/n}}$
William %R	Uses Stochastic to determine overbought and oversold levels	Stochastic	$\frac{H_n - C_t}{H_n - L_n}\ X\ (-100)$
Accumulator/Distribution	Combines price and volume to show how money may be flowing into or out of a stock	momentum oscillator Indicator	$\frac{(C_t - L_t) - (H_t - C_t)}{(H_t - L_t)}\ X\ V_t$
On Balance Volume (OBV)	Combines price and volume in a very simple way to show how money may be flowing into or out of a stock	momentum Indicator	$O_P + \begin{cases} V_t & , \text{if } C > C_P \\ 0, \text{if } C = C_P \\ -V_t & , \text{if } C < C_P \end{cases}$

DIFF$_t$: EMA(12)$_t$ - EMA(26)$_t$, EMA is exponential moving average, EMA(k)$_t$: EMA(k)$_{t-1}$ + α X(Ct - EMA(k)$_{t-1}$), α smoothing factor: 2/(1 + k), k is time period of k day exponential moving average, Up$_t$ means the upward price change, Dw$_t$ means the downward price change at time t. C$_t$ is the closing price, L$_t$: the low price, H$_t$: the high price at time t. [10]

6.2.4. Data sampling

Entire data distribution is illustrated in Figure 2. The entire collected raw dataset divided into three independent subsets: tuning dataset, holdout dataset, and prediction testing dataset. The first dataset, known as tuning dataset, is used during training. The tuning dataset is further divided into three subsets: model training dataset, model testing dataset and model validating dataset.

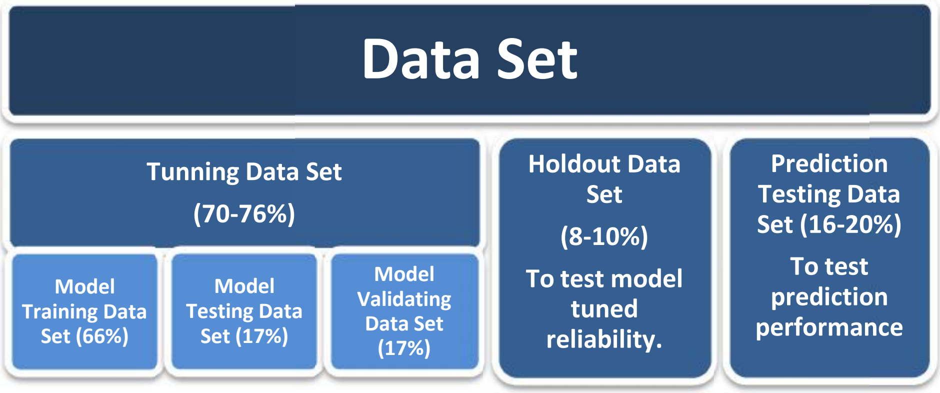

Data Set
Tunning Data Set
(70-76%)
Model Training Data Set (66%)
Model Testing Data Set (17%)
Model Validating Data Set (17%)
Holdout Data Set
(8-10%)
To test model tuned reliability.
Prediction Testing Data Set (16-20%)
To test prediction performance

Figure 2. General flows for model data distribution

The entire tuning dataset is divided into different groups having 7 data samples. Every sixth data sample of each group is combined to form model testing dataset and every seventh data sample of each group is combined to form a model validating dataset. The remaining data samples of each group are combined to form model training dataset. The process of getting model training dataset, model testing dataset and model validating dataset is illustrated in Figure 3. In a way, 66% of entire tuning dataset belongs to model training dataset, 17% belong to the model testing dataset and remaining 17% belong to model validating dataset.

Approximately one-year duration (250 data sample) before last year are used in holdout dataset to validate model whether model properly tunes or not. The third dataset of last two years (500 data sample), known as prediction testing dataset, is used to test prediction performance of the model. The testing dataset and holdout dataset, which is independent of the training dataset, is divided continuously. The training dataset is used to train the model.

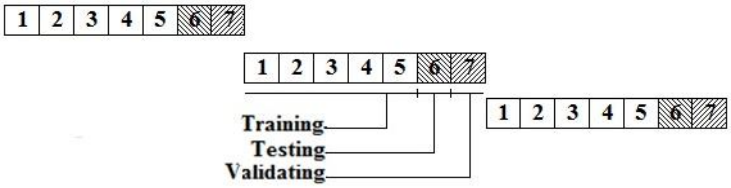


Figure 3. Tuning dataset distribution process

6.2.5. Processing framework

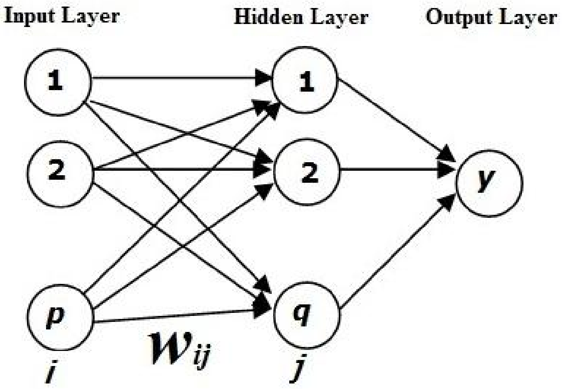


Figure 4. Simplified three-layer feed-forward neural network.

In this research, the data processing model consists of artificial neural network. A simplified view of the three layers feed-forward back-propagation artificial neural network (FFBPNN) is shown in Figure 4. The relation between output y_k and the inputs (x1, x2,….,xp) as expressed in Eq. (1).

$$y_k = w_0 + \sum_{j=1}^{q} w_j \cdot f\left(w_{0,j} + \sum_{i=1}^{p} w_{i,j} \cdot x_{i,j}\right) \qquad \text{Eq. (1)}$$

Where *f* is a nonlinear function to be approximated, $w_{i,j}$ (i=0,1,2,…,p, j=0,1,2,…,q) and w_j (j=0,1,2,…,p, j=0,1,2,…,q) are the weights, *p* is the number of inputs neurons in the input layer, and *q* is the number of neurons in hidden layer. The weight $w_{i,j}$ is updated such that total squared error of the output computed by the net is minimized. The total square error E is expressed in Eq. (2).

$$E=\frac{1}{2}\sum_k\left[t_k\text{-}y_k\right]^2 \qquad \text{Eq. (2)}$$

Where, t_k is targeted output.

Model parameters setting

The input layer consists of five neurons, where the hidden layer neurons are selected empirically and output layer has one neuron. The initial values of weight are randomly assigned. Levenberg-Marquardt [28], [29] back-propagation learning algorithm is used to train the ANN. Among various training algorithm, Levenberg-Marquardt produces the best result with a minimum number of the epoch, less time in execution and lowest prediction error in performance [30]. For smaller size input, Levenberg-Marquardt algorithm is faster and has achieved better performance than other algorithms in training [31]. The mean square error is used to evaluate the performance of the ANN model. The hyperbolic tangent sigmoid transfer function is used at both hidden layer and output layer. We initialize a layer's weights and biases according to the Nguyen-Widrow initialization algorithm.

The number of neurons (*h*) in the hidden layer of model’s parameters will be efficiently determined. Five levels of *h* (10, 15, 20, 25, and 30) were tested in the parameter setting experiments. It has observed from various experiments, that one epoch X-training on M models involved less computation time than X-training of a single M model. For each level of h neuron, the twenty-trained model are generated by applying only one epoch training on holdout dataset for each index. The model output is measured in terms of the percentage prediction. The similar treatment applies to all indices and model parameters are evaluated. The parameter setting experiments yield 5 x 20 =100 treatments for each index. For each level of h neuron, percentage prediction is evaluated for all generated model. Select the level of h neuron, where model having max percentage prediction. Table 3 illustrates minimum, average and maximum percentage prediction output of FFBPNN on holdout dataset for parameters setting experiment. The selected FFBPNN’s number of hidden layer’s node for each index is illustrated in Table 4.

Table 3. Prediction performance (%) of FFBPNN in parameter setting.

	Hidden layer's neurons: 10			**Hidden layer's neurons: 15**			**Hidden layer's neurons: 20**			**Hidden layer's neurons: 25**			**Hidden layer's neurons: 30**		
Index	Min.	Avg.	Max.	Min.	Avg.	Max.	Min.	Avg.	Max.	Min.	Avg.	Max.	Min.	Avg.	Max.
NIFTY	40.4	50.1	61.2	40	47.1	59.6	40	49	60	39.6	49.3	59.6	40	48.02	64.8
SENSEX	40.4	50.12	59.2	37.6	47.76	60.4	40.8	47.38	60.4	37.2	48.94	60.4	40	48.28	60
NIKKIE	44.8	49.08	54.8	45.2	49	55.2	44.8	49.1	55.2	43.2	48.76	55.6	44	48.16	56
SSE	42.8	50.94	59.6	43.2	50.7	57.2	42.8	52.26	58	43.2	52.02	57.2	42.4	51.54	58.8
SNP500	43.2	50.64	55.2	43.2	50.4	55.2	45.2	52.14	56.8	44.8	50.76	56.4	44.8	50.4	56.4
NASDAQ	42.4	50.56	57.6	42	49.56	57.6	41.2	50.52	57.6	42.4	51.06	57.6	40.4	48.62	57.6
DJI	44.8	49.2	56.4	43.6	49.56	55.2	44	50.66	55.2	44.4	50.26	55.2	45.2	51.22	56.4
FTSE	46	50.16	52.8	47.2	51.1	53.6	46.4	50.38	52.8	47.2	50.4	56	46.4	51.58	55.6
CAS	43.6	49.82	54.8	44	49.46	59.6	43.2	49.44	55.2	42	49.02	56	40.8	48.98	54
DAX	40	50.74	60.4	40	50.76	59.6	43.2	50.6	60	40	48.72	60	40	46.84	60

Table 4. Selected number of FFBPNN's hidden layer's node.

Index	**Best prediction (%)**	**Best hidden layer's node**
NIFTY	64.8	30
SENSEX	60.4	15
NIKKIE	56	30
SSE	59.6	10
SNP500	56.8	20
NASDAQ	57.6	10
DJI	56.4	10
FTSE	56	25
CAS	59.6	15
DAX	60.4	10

Find reliable tuned model

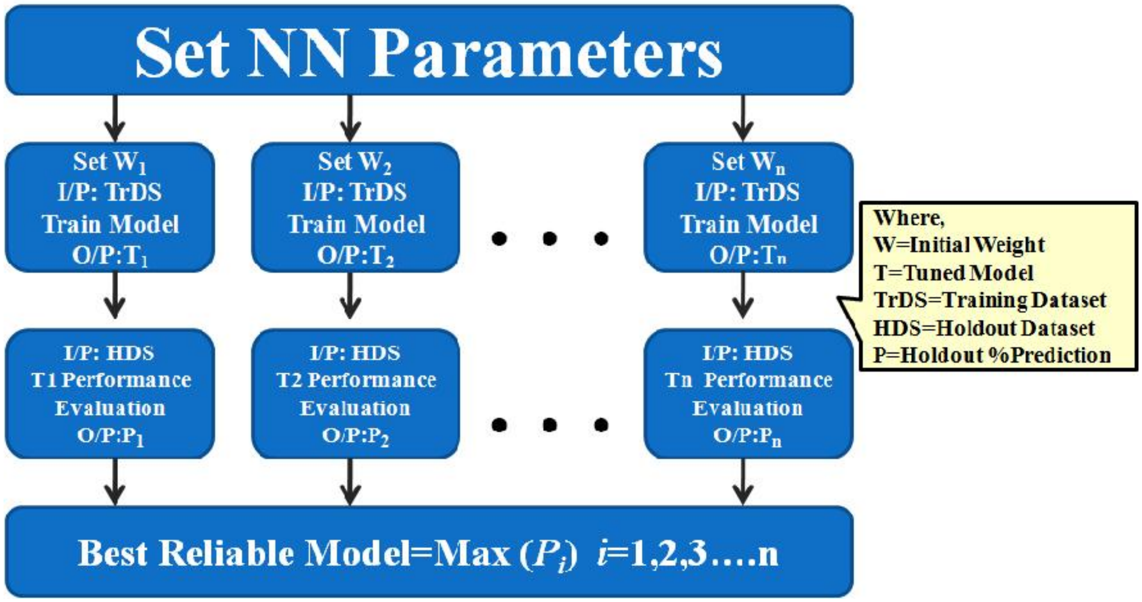
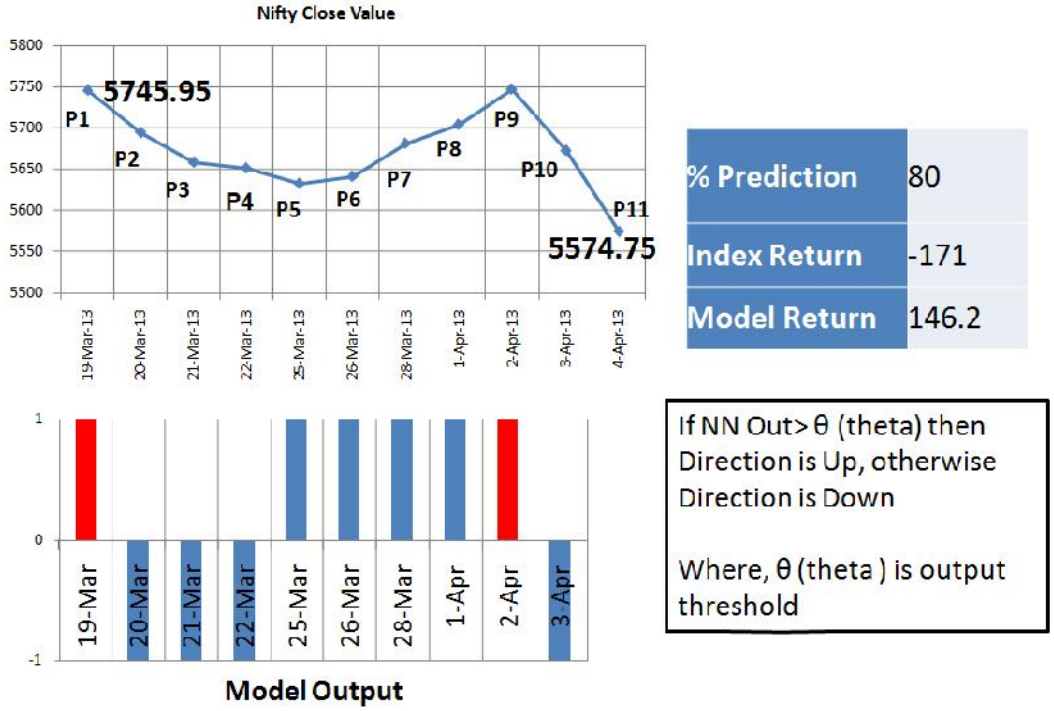

Set NN Parameters
Set W_1 I/P: TrDS Train Model O/P:T_1
Set W_2 I/P: TrDS Train Model O/P:T_2
Set W_n I/P: TrDS Train Model O/P:Tn
I/P: HDS T1 Performance Evaluation O/P:P_1
I/P: HDS T2 Performance Evaluation O/P:P_2
I/P: HDS Tn Performance Evaluation O/P:P_n
Where,
W=Initial Weight
T=Tuned Model
TrDS=Training Dataset
HDS=Holdout Dataset
P=Holdout %Prediction
Best Reliable Model=Max (P_i) i=1,2,3….n

Figure 5. The process to select reliable tuned model

After setting model parameters, the neural network is trained until one of these conditions is true: training epoch reaches to 100 or goal reach to 0.01 or maximum validation fail to reach count six. Here we generated *n* tuned models *(T_1, T_2 …T_n)* by training dataset. Apply holdout dataset on all *n* tuned models to get percentage prediction (P_1, P_2…P_n) of model’s output. From generated tuned models, we select that model as a reliable tuned model whose percentage prediction output is maximum. The selection of the reliable tuned model process is illustrated in Figure 5.

6.2.6. Output

The hyperbolic tangent sigmoid transfer function is used at output layer of ANN, which will give output value between 0 and 1. If the model gives output less than 0.5, then next day price direction is considers as “0” (down), otherwise next day price direction is considered as “1” (Up). If next day prediction output of the model is up, then buy trade will be initiate. If next day prediction output of the model is down, then sell trade will initiated. Model percentage prediction and model return will calculated based on hit and miss trade as illustrates in Figure 6.



% Prediction	80
Index Return	-171
Model Return	146.2

If NN Out>θ (theta) then Direction is Up, otherwise Direction is Down

Where, θ (theta) is output threshold

Figure 6. Hit rate calculation: measure the prediction performance the model

7. Experiment result and analysis

Testing data sample of approximately last two years from respective benchmark index fed to selected reliable tuned model and model output performance is measured. To change initial weight and retaining other parameters, tuned model parameters will settle in a different way and hence, newly tuned model is generated. Applying above process, different tuned models are generated by initializing weight value and preparing five different cases from CASE1 to CASE5 for all indices. This way test all global benchmark indices and analyze whether model outperforms with the respective performance measure. We have prepared result set in form of result matrix according to generated GSDPM's output.

7.1. Calculation of Rate of Return (ROR)

A rate of return is the total gain/loss on investment over a specified time interval. The following Table 5 and Table 6 demonstrate calculation of rate of return (ROR) for five data point.

Table 5. Representation of GSDPM generated output value on five data sample.

Data Point	Date	Close	Absolute Change	Model Out	Target	Output	Hit	Gain	Loss
x1	19-Dec-16	8104.35	21.95	0	0	0.25971	1	21.95	0
x2	20-Dec-16	8082.4	21.1	0	0	0.267467	1	21.1	0
x3	21-Dec-16	8061.3	82.2	0	0	0.239435	1	82.2	0
x4	22-Dec-16	7979.1	6.65	1	0	0.228713	0	0	6.65
x5	23-Dec-16	7985.75	77.5	0	0	0.439072	1	77.5	0
x6	26-Dec-16	7908.25	124.6	1	0	0.254586	0	0	124.6
Total		-196.1						202.75	131.25

Table 6. ROR calculation of Table 5 data point.

ROR Calculation
Absolute Change: Xn=Xn+1-Xn
Hit: if Output>0.5 then 1, else 0
Gain: if Hit=1 then absolute Change
Loss: if Hit=0 then absolute Change
Absolute Return: Gain-Loss (202.75-131.25=71.5)
% ROR: absRoR*100/absolute Index (Start Point)
%ROR: (71.5*100/8104.35)=0.88%

7.2. Calculation of Succession Prediction Ratio

Table 7. Generalized representation of experiment's result matrix.

Index	CASE$_1$	CASE$_2$	CASE$_m$	Total Cases With (PM>β)
Ind$_1$	$\square_{1,1}$	$\square_{1,2}$	$\square_{1,m}$	$\sum_{j-1}^{m}(\square_{1,j} > \beta)$
Ind$_2$	$\square_{2,1}$	$\square_{2,2}$	$\square_{2,m}$	$\sum_{j-1}^{m}(\square_{2,j} > \beta)$
Ind$_s$	$\square_{s,1}$	$\square_{s,2}$	$\square_{s,m}$	$\sum_{j-1}^{m}(\square_{s,j} > \beta)$
Total Cases With (PM >β)	$\sum_{i-1}^{s}(\square_{i,1} > \beta)$	$\sum_{i-1}^{s}(\square_{i,2} > \beta)$	$\sum_{i-1}^{s}(\square_{i,m} > \beta)$	$S = \sum_{i-1}^{s}\sum_{j-1}^{m}(\square_{i,j} > \beta)$

Where s is Total stock securities, m is Total generated cases, □ is Model output value, β is Constant value, PM is Performance Measure and ST is Total success prediction test.

SPR is the ratio of total success prediction test vs. total prediction test. The formula for calculating SPR as expressed Eq. (3).

$$SPR = \frac{ST}{s \times m} \qquad \text{Eq. (3)}$$

Where s is total stock securities, m is total cases and ST is total success prediction test.

SPR value 0 indicates 100% failed in all test and SPR value 1 indicates 100% success in all test. The value of SPR higher than 0.5 suggests that model will outperform with respect to performance measure and less than 0.5 suggest that model will underperform. Table 7 shows model output in form of generalized representation result matrix for various treatments. This research work uses ten different benchmark indices and five cases are prepared for each index.

7.3. Model output performance

Model performance is measured in terms of percentage prediction and model positive percentage ROR. When we trade x sample, there are two possible prediction outcomes; hit or miss. Both outcomes are equally likely hence, the probability of each outcome is 50%. Higher than 50% probability suggests outperforms in terms of percentage prediction. If model percentage prediction is higher than 50%, then considered model is successfully predicting the direction of respective stock securities. Based on the model output hit rate, percentage ROR is calculated. If percentage ROR is positive, then considered model will successfully outperform in term of percentage ROR. Table 8 illustrates the result of FFBPNN model percentage prediction by applying testing dataset on selected reliable model. In Table 8, success test greater than 50% prediction is total 41 and total cases are 50, which evaluates SPR value 0.82. SPR value 0.82 of Table 8 is higher than 0.5 shows that, model will outperform in term of percentage prediction. Table 9 illustrates the result of FFBPNN model percentage ROR. SPR value 0.76 of Table 9 shows that model will outperform in term of positive ROR (%).

Table 8 FFBPNN model percentage prediction

s=10, m = 5, PM is prediction (%), α= prediction (%), β is 50 (%), ST=41, SPR=0.82.

Index	**CASE1**	**CASE2**	**CASE3**	**CASE4**	**CASE5**	**Total Cases>50**
NIFTY	50.4	50.8	51.4	50	52.4	4
SENSEX	51	52	51.4	52.4	50.6	5
NIKKIE	51	48.8	53.6	52.4	50	3
SSE	51.4	49	44.2	45.8	53.4	2
SNP500	51	50.8	51.6	49.6	51.4	4
NASDAQ	53	52	53.4	51.4	53.6	5
DJI	52.4	50.8	51.8	53	51	5
FTSE	52.6	54.8	50.8	48.8	51	4
CAS	52.4	53.8	54.6	51	49.2	4
DAX	52.8	51.2	51.6	54.8	53	5
Total Indices>50	10	8	9	6	8	41

Table 9 FFBPNN model ROR.

s=10, m = 5, PM is ROR (%), α= ROR (%), β is 0, ST=38, SPR=0.76.

Index	CASE1	CASE2	CASE3	CASE4	CASE5	Total Cases>0
NIFTY	-4.56	15.55	-4.96	7.26	-1.66	2.00
SENSEX	-0.76	15.48	11.94	17.87	1.80	4.00
NIKKIE	-8.13	14.69	3.35	60.37	-10.53	3.00
SSE	-5.33	50.87	8.72	10.18	28.51	4.00
SNP500	11.80	9.87	13.75	9.89	17.34	5.00
NASDAQ	-3.30	2.82	17.64	11.16	17.26	4.00
DJI	15.07	16.15	24.40	17.93	7.48	5.00
FTSE	15.65	55.11	1.95	11.49	-10.83	4.00
CAS	16.01	41.19	46.76	34.58	-10.94	4.00
DAX	0.20	-25.36	-2.80	18.96	0.84	3.00
Total Indices >0	5.00	9.00	8.00	10.00	6.00	38.00

7.4. Model validation

Model prediction validation is done against randomly generated buy/sell trade. Instead of following the model buy/sell trade, simple random buy/sell trade is initiated on a number of the testing data sample and result sets are generated. This randomly generated output has a value between 0 and 1, with 0.5 mean and 0.2 standard deviation values. The output of random data compared with targeted output and hit rate is calculated. Based on hit rate, the percentage prediction and percentage ROR is calculated for random data. Randomly generated data percentage prediction of indices is shown in Table 10 with SPR (0.48) which is below 0.5. Table 11 is shown percentage ROR for random testing data sample. By comparing SPR value of Table 8 and Table 10, we conclude that model percentage prediction is higher than random data percentage prediction.

Model validation in terms of percentage prediction (Table 8) against random data percentage prediction (Table 10) is illustrated in Table 12. The result shows that model percentage prediction effectively outperforms against random data percentage prediction. The model validation in terms of percentage ROR (Table 9) against random data percentage ROR (Table 11) is illustrated in Table 13. The experiment result shows that the model percentage ROR effectively outperforms against random data percentage ROR.

Table 10 Random buy/sell percentage prediction.

s=10, m = 5, PM is prediction (%), α= prediction (%), β is 50 (%), ST=24, SPR=0.48.

Index	CASE1	CASE2	CASE3	CASE4	CASE5	Total Cases>50
NIFTY	50.6	48.6	48.8	53	54.2	3
SENSEX	46.2	46.6	49.6	49.2	53.8	1
NIKKIE	51.2	53.4	49.2	52.2	48.8	3
SSE	52.6	48	46.8	50.6	53.8	3
SNP500	48.8	56.4	50.8	50.6	49.8	3
NASDAQ	53.8	49	48.6	51	50.8	3
DJI	52.2	55.2	50.2	49	47.4	3
FTSE	50	48.2	46.8	49.4	54.2	1
CAS	49.4	48.4	49.2	54.8	53	2
DAX	49.4	52	43.8	49.8	54.4	2
Total Indices>50	5	4	2	6	7	24

Table 11 Random buy/sell ROR.

(s=10, m = 5, PM is % prediction, α= ROR (%), β is 0, ST=23, SPR=0.46)

Index	CASE1	CASE2	CASE3	CASE4	CASE5	Total Cases>0
NIFTY	-14.95	-9.27	-0.09	23.33	17.58	2.00
SENSEX	-48.87	-35.36	-5.75	-8.83	20.57	1.00
NIKKIE	-17.40	58.24	-10.57	-14.64	-5.31	1.00
SSE	27.39	-57.57	-57.13	0.42	6.86	3.00
SNP500	-6.22	35.67	1.47	-15.96	-19.67	2.00
NASDAQ	12.52	-23.24	6.21	1.26	-8.78	3.00
DJI	17.40	35.91	-21.95	0.48	2.39	4.00
FTSE	3.87	-14.23	-20.48	-12.47	22.67	2.00
CAS	-1.85	-38.84	-35.39	46.78	-4.45	1.00
DAX	33.22	35.63	-40.19	12.19	56.78	4.00
Total Indices>0	5.00	4.00	2.00	6.00	6.00	23.00

Table 12 FFBPNN model prediction> Random data prediction

ST=30, SPR=0.60

Index	CASE1	CASE2	CASE3	CASE4	CASE5	Total
NIFTY	0	1	1	0	0	2
SENSEX	1	1	1	1	0	4
NIKKIE	0	0	1	1	1	3
SSE	0	1	0	0	0	1
SNP500	1	0	1	0	1	3
NASDAQ	0	1	1	1	1	4
DJI	1	0	1	1	1	4
FTSE	1	1	1	0	0	3
CAS	1	1	1	0	0	3
DAX	1	0	1	1	0	3
Total	6	6	9	5	4	30

Table 13 FFBPNN model ROR > Random data ROR.

ST=33, SPR=0.66.

Index	CASE1	CASE2	CASE3	CASE4	CASE5	Total
NIFTY	1	1	0	0	0	2
SENSEX	1	1	1	1	0	4
NIKKIE	1	0	1	1	0	3
SSE	0	1	1	1	1	4
SNP500	1	0	1	1	1	4
NASDAQ	0	1	1	1	1	4
DJI	0	0	1	1	1	3
FTSE	1	1	1	1	0	4
CAS	1	1	1	0	0	3
DAX	0	0	1	1	0	2
Total	6	6	9	8	4	33

For testing model’s design robustness, GSDPM is implemented with other neural networks; Elman back-propagation neural network and cascade-forward back-propagation neural network. Following above methodologies, output performance of the GSDPM is evaluated. Table 14 summarized various performance measures of GSDPM for all indices and all cases on different neural networks. The experimental result of tested neural network is shown in Table 14. SPR value of Table 14 proves that, GSDPM effectively outperforms in term of percentage prediction, ROR and against randomize generated data buy/sell prediction.

Table 14 Summary of various performance measures for all indices and all cases on different NN.

Result test	**Feed forward back-propagation neural network**		**Elman back-propagation neural network**		**cascade-forward back-propagation neural network**	
	Total success test	SPR	Total success test	SPR	Total success test	SPR
Model Prediction (%) >50	41	0.82	40	0.8	40	0.8
Model ROR (%) > 0	38	0.76	39	0.78	42	0.84
Random data prediction (%) > 50	24	0.48	26	0.52	22	0.44
Random data ROR (%) >0	23	0.46	23	0.46	26	0.52
Model prediction (%) Vs. Random data prediction (%)	30	0.6	33	0.66	36	0.72
Model ROR (%) Vs Random data ROR (%)	33	0.66	32	0.64	36	0.72

8. Conclusion

The output of GSDPM result from Table 14 proved that artificial neural network has the capability to effectively predict price direction of stock securities. The GSDPM result shown in Table 8 and Table 9 proved that GSDPM model outperforms with respect to percentage prediction and percentage positive return respectively. Instead of following random buy/sell trade, following model output will always outperform respect to percentage prediction as well as percentage ROR. Empirical results from Table 10 to Table 13 proved that predicted buy/sell signal by proposed model outperforms in terms of percentage return with respect to random buy/sell percentage return as well as with respect to ROR. Table 14 proved the robust design of GSDPM and its significant performance in predicting the direction of global indices. The proposed methodology to select reliable model will help research community to select their reliable model, where soft computing model may generate multiple solutions. The model validation methodology adopted will help other researchers to validate their prediction model performance whose model have multiple parameters along with input data having random nature. A new term, called Success Prediction Ratio, is proposed to assess the validity of the model. SPR can be useful in validation of other models, where identical models are not available for model's output comparison. GSDPM is not limited to indices; other researchers can apply other indices and/or stock security and test model prediction performance. There is future scope for other researchers to improve prediction quality. The research is not limited to artificial neural network architecture, indeed it provides an opportunity to other researchers to use other soft computing technique to test the prediction performance and validity of the model.

Paper published related to Ph.D study

- A. M. Panchal and J. M. Patel, “Role of soft computing techniques in predicting stock market direction,” *Artificial Intelligence Research*, vol. 1, no. 2, pp. 198–202, Dec. 2012.
- A. M. Panchal and J. M. Patel, “Stock Index Direction Prediction Applying Artificial Neural Network: The Sample of the National Stock Exchange,” *Current Trends in Systems & Control Engineering*, vol. 2, no. 2–3, pp. 33–42, Dec. 2012.
- A. M. Panchal and J. M. Patel, “National Stock Exchange Stock and Index Price Direction Prediction using Backpropagation Artificial Neural Network,” *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 4, no. 11, pp. 11133–11138, Nov. 2015.
- A. M. Panchal and J. M. Patel, “A Design, Development and Validation of Generalized Stock Index Prediction Model based on Soft Computing Technique,” - under review to be published in *Expert Systems with Applications* ISSN: 0957-4174

References

[1] S. Agrawal, M. Jindal, and G. N. Pillai, “Momentum analysis based stock market prediction using adaptive neuro-fuzzy inference system (ANFIS),” in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 2010, vol. 1.

[2] W. Zeng-min and W. Chong, “Application of Support Vector Regression Method in Stock Market Forecasting,” in *2010 International Conference on Management and Service Science (MASS)*, 2010, pp. 1–4.

[3] A. M. Panchal and J. M. Patel, “Role of soft computing techniques in predicting stock market direction,” *Artificial Intelligence Research*, vol. 1, no. 2, pp. 198–202, Dec. 2012.

[4] G. S. Atsalakis and K. P. Valavanis, “Surveying stock market forecasting techniques – Part II: Soft computing methods,” *Expert Systems with Applications*, vol. 36, no. 3, Part 2, pp. 5932–5941, Apr. 2009.

[5] T. Kimoto, K. Asakawa, M. Yoda, and M. Takeoka, “Stock market prediction system with modular neural networks,” in *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*, 1990, pp. 1–6.

[6] R. Lawrence, “Using neural networks to forecast stock market prices,” *University of Manitoba*, 1997.

[7] M. Karaatli, I. Gungor, Y. Demir, and S. Kalayci, “Estimating stock market movements with neural network approach,” *Journal of Balikesir University*, vol. 2, no. 1, pp. 22–48, 2005.

[8] C. S. Vui, G. K. Soon, C. K. On, R. Alfred, and P. Anthony, “A review of stock market prediction with Artificial neural network (ANN),” in *Control System, Computing and Engineering (ICCSCE), 2013 IEEE International Conference on*, 2013, pp. 477–482.

[9] D. Brezak, T. Bacek, D. Majetic, J. Kasac, and B. Novakovic, “A comparison of feed-forward and recurrent neural networks in time series forecasting,” in *Computational Intelligence for Financial Engineering & Economics (CIFEr), 2012 IEEE Conference on*, 2012, pp. 1–6.

[10] Y. Kara, M. Acar Boyacioglu, and Ö. K. Baykan, “Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange,” *Expert Systems with Applications*, vol. 38, no. 5, pp. 5311–5319, May 2011.

[11] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, “Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques,” *Expert Systems with Applications*, vol. 42, no. 1, pp. 259–268, Jan. 2015.

[12] A. Mochón, D. Quintana, Y. Sáez, and P. Isasi, “Soft computing techniques applied to finance,” *Appl Intell*, vol. 29, no. 2, pp. 111–115, Jun. 2007.

[13] W. Bing-hui and H. Jian-min, “The trend analysis of China’s stock market based on fractal method and BP neural network model,” in *2014 International Conference on Management Science Engineering (ICMSE)*, 2014, pp. 1258–1266.

[14] Y. Longguang and W. Qing, “Predicting the stock price based on BP neural network and big transaction,” in *2012 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, 2012, pp. 2957–2960.

[15] W. Ling-zhi and Q. Fa-jin, “Stock market forecasting model based on semi-parametric smoothing regression,” in *Control Conference (CCC), 2012 31st Chinese*, 2012, pp. 7220–7223.

[16] A. M. Panchal and J. M. Patel, “National Stock Exchange Stock and Index Price Direction Prediction using Backpropagation Artificial Neural Network,” *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 4, no. 11, pp. 11133–11138, Nov. 2015.

[17] W. Ming-Tao and Y. Yong, “The Research on Stock Price Forecast Model Based on Data Mining of BP Neural Networks,” in *2013 Third International Conference on Intelligent System Design and Engineering Applications (ISDEA)*, 2013, pp. 1526–1529.

[18] J. Zhang and Y. Yang, “BP Neural Network Model Based on the K-Means Clustering to Predict the Share Price,” in *2012 Fifth International Joint Conference on Computational Sciences and Optimization (CSO)*, 2012, pp. 181–184.

[19] L. Zhang, B. Zhang, and Y. Gu, “Application of Improved LM-BP Neuron Network in stock prediction,” in *2012 2nd International Conference on Computer Science and Network Technology (ICCSNT)*, 2012, pp. 1655–1658.

[20] Y. Ma, Y. Chang, and C. Xia, “Applied research on stock forcasting model based on BP neural network,” in *2011 International Conference on Electronic and Mechanical Engineering and Information Technology (EMEIT)*, 2011, vol. 9, pp. 4578–4580.

[21] M. M. Aldin, H. D. Dehnavi, and S. Entezari, “Evaluating the employment of technical indicators in predicting stock price index variations using artificial neural networks (case study: Tehran stock exchange),” *International Journal of Business and Management*, vol. 7, no. 15, p. p25, 2012.

[22] B. B. Nair, V. P. Mohandas, and N. R. Sakthivel, “A decision tree—rough set hybrid system for stock market trend prediction,” *International Journal of Computer Applications*, vol. 6, no. 9, pp. 1–6, 2010.

[23] T. Manojlovic and I. Stajduhar, “Predicting stock market trends using random forests: A sample of the Zagreb stock exchange,” in *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2015, pp. 1189–1193.

[24] J. Yao, C. L. Tan, and H.-L. Poh, “Neural networks for technical analysis: a study on KLCI,” *International journal of theoretical and applied finance*, vol. 2, no. 2, pp. 221–241, 1999.

[25] K. Kim, “Financial time series forecasting using support vector machines,” *Neurocomputing*, vol. 55, no. 1, pp. 307–319, 2003.

[26] G. Armano, M. Marchesi, and A. Murru, “A hybrid genetic-neural architecture for stock indexes forecasting,” *Information Sciences*, vol. 170, no. 1, pp. 3–33, 2005.

[27] T. Takahashi, R. Tamada, and K. Nagasaka, “Multiple line-segments regression for stock prices and long-range forecasting system by neural network,” in *SICE’98. Proceedings of the 37th SICE Annual Conference. International Session Papers*, 1998, pp. 1127–1132.

[28] K. Levenberg, “A method for the solution of certain problems in least squares,” *Quarterly of Applied Mathematics*, vol. 2, pp. 164–168, 1944.

[29] D. W. Marquardt, “An algorithm for least-squares estimation of nonlinear parameters,” *Journal of the Society for Industrial & Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[30] D. A. Kumar and S. Murugan, “Performance Analysis of MLPFF Neural Network Back Propagation Training Algorithms for Time Series Data,” in *2014 World Congress on Computing and Communication Technologies (WCCCT)*, 2014, pp. 114–119.

[31] K. Kim and I. Han, “Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index,” *Expert systems with Applications*, vol. 19, no. 2, pp. 125–132, 2000.