Accepted Manuscript

A Hybrid Stock Trading Framework Integrating Technical Analysis with Machine Learning Techniques

Rajashree Dash, Assistant Professor, Dr. Pradipta Kishore Dash, Director of Research

PII: S2405-9188(15)30017-9

DOI: 10.1016/j.jfds.2016.03.002

Reference: JFDS 9

To appear in: *The Journal of Finance and Data Science*

Received Date: 30 December 2015

Revised Date: 3 March 2016

Accepted Date: 8 March 2016

A Hybrid Stock Trading Framework Integrating Technical Analysis with Machine Learning Techniques

Author 1

Rajashree Dash,
Assistant Professor,
Computer Science & Engineering Department,
ITER, Siksha 'O' Anusandhan University,
Bhubaneswar, Odisha 751030, INDIA
rajashree_dash@yahoo.co.in

Rajashree Dash is working as an Asst. Professor in ITER, Siksha 'O' Anusandhan University. She has completed her M.Tech in Computer Science and Engineering. Now she is pursuing her P.H.D. Her research topics include data mining, time series prediction, evolutionary computation etc.

Author 2

Dr. Pradipta Kishore Dash,
Director of Research,
Siksha 'O' Anusandhan University,
Bhubaneswar, Odisha 751030, INDIA
pkdash.india@gmail.com

Professor P.K.Dash holds D.SC.,Ph.D. degrees in Eleectrical Engineering and currently is the director of research at the Siksha O Anusandhan University, Bhubaneswar, India. He had several visiting appointments at the Universities in USA, Canada, Singapore, and Malaysia. He has published more than 450 papers in reputed International journals and conferences. His research interests include Power Engineering, Renewable Energy, Signal Processing, Computational intelligence, Evolutionary Computing, and Information Technology.

A Hybrid Stock Trading Framework Integrating Technical Analysis with Machine Learning Techniques

Abstract

In this paper, a novel decision support system using a computational efficient functional link artificial neural network (CEFLANN) and a set of rules is proposed to generate the trading decisions more effectively. Here the problem of stock trading decision prediction is articulated as a classification problem with three class values representing the buy, hold and sell signals. The CEFLANN network used in the decision support system produces a set of continuous trading signals within the range 0 to 1 by analyzing the nonlinear relationship exists between few popular technical indicators. Further the output trading signals are used to track the trend and to produce the trading decision based on that trend using some trading rules. The novelty of the approach is to engender the profitable stock trading decision points through integration of the learning ability of CEFLANN neural network with the technical analysis rules. For assessing the potential use of the proposed method, the model performance is also compared with some other machine learning techniques such as Support Vector Machine (SVM), Naive Bayesian model, K nearest neighbor model (KNN) and Decision Tree (DT) model.

Key words: Stock trading, Stock trend analysis, Technical indicators, CEFLANN

1. Introduction

With the era of economic globalization and the facility of digital technology, generation and accumulation of financial data has reached at an unprecedented rate. The rapidly growing volume of data has far exceeded the ability of a human being to analyze them manually. Again financial time series data are more complicated than other statistical data due to the long term trends, cyclical variations, seasonal variations and irregular movements. These are highly affected by many external factors, such as many highly interrelated economic, political, social and even if the psychological behavior of the investor. The continuous growth of such highly fluctuating and irregular data has put forth the critical need for developing more automated approaches for efficient analysis of such massive financial data to extract meaningful statistics from that. Being a process of exploring useful hidden knowledge, Data mining has carved its own niche in financial time series analysis. It provides pathways for investors to take proactive and knowledge-driven decisions in order to achieve successful gain with less investment risk.

Gaining high profit is the ultimate goal of an investor participating in financial market. There are so many investment opportunities like trading (i.e. buying and selling) bonds, shares, foreign exchanges and precious metals etc. present in a financial market. Trading in stock market is one of the popular channels of financial investment. Investors in the stock Market can maximize their profit by buying or selling their investment at proper time. The key to realize high profits in stock trading is to find out the suitable trading time with the minimum risk of trading. But it is always hard to decide the best time to buy or sell due to the highly fluctuating and dynamic behavior of stock market. Technical indicators are the primary interest for most of the researchers to monitor the stock prices and to assist investors in setting up trading rules for buy–sell–hold decisions. Technical indicators are produced based on historical stock data. So trading decision taken based on particular technical indicators may not always be more profitable. In literature various data mining and artificial intelligence tools has been applied to analyze technical indicators in an attempt to find the best trading signals (Kablan, 2009; Teixeira, & De Oliveira, 2010; Rodríguez-González et al., 2011; Brasileiro et al., 2013). Gaining profit or loss from stock trading ultimately depends on analysis of future movement of highly fluctuating and irregular stock price values. Successful classification of up and down movements in stock price index values may not only helpful for the investors to make effective trading strategies, but also for policy maker to monitor stock market. Keeping track of upswings and downswings over the history of individual stocks will reduce the uncertainty associated to investment decision making. Investors can choose the best times to buy and sell the stock through proper analysis of the stock trends. In literature a number of models combining technical analysis with computational intelligent techniques are available for prediction of stock price index movements (Kara, Boyacioglu, & Baykan, 2011; Patel et al., 2015; Wysocki, & Lawrynczuk, 2013; Patra, Thanh, & Meher, 2009).

In this study, the problem of stock trading decision prediction is articulated as a classification problem with three class values representing the buy, hold and sell signals. The foremost objective of this study is to develop a novel decision support system using a computational efficient functional link artificial neural network (CEFLANN) and a set of rules based on technical analysis, to generate the trading decisions more effectively. Instead of training the CEFLANN network using traditional back propagation algorithm, the ELM learning is proposed for the network. Six popular technical indicators calculated from the historical stock index price values are used as the input features for the proposed model. The CEFLANN network is applied to capture the nonlinear relationship exists between the technical indicators and trading signals. Instead of using discrete class values during training of the network, a continuous trading signal within range 0 to 1 are fed to the network. The new trading signals in the range 0 to 1 can provide more detailed information regarding stock trading related to the original price variations. Further the outputs from the CEFLANN model is transformed in to a simple trading strategy with buy, hold and sell signals using suitable rules. The model performance is evaluated based on profit percentage obtained during test period. The CEFLANN model is also compared with some other known machine learning techniques like support vector machine (SVM) (Kara, Boyacioglu, & Baykan, 2011; Patel et al., 2015; Wen et al., 2010; Ballings et al., 2015), Naive Bayesian model, K nearest neighbor model (KNN) (Teixeira, & De Oliveira, 2010; Ballings et al., 2015) and decision tree (DT) (Wu et al., 2006) model.

The remainder of the paper is organized in to following sections; Section 2 highlights relevant reviews on different machine learning techniques used in stock trading. Section 3 specifies the details of CEFLANN network followed by the details of ELM Learning in Section 4. Section 5 describes the detailed steps of the decision support system for generating stock trading decision points. Section 6 shows experimental results obtained from the comparative analysis. Finally section 7 contains the concluding remarks.

2. Literature Survey

Though most of the financial time series analysis involve prediction of stock price or it fluctuation, but trading the stock market is another popular research area. Gaining profit or loss from stock trading ultimately depends on analysis of future movement of highly fluctuating and irregular stock price values. In literature a number of models combining technical analysis with computational intelligent techniques are available for prediction of stock price index movements and for stock trading. In (Lee et al., 2007) a new trading framework enhancing the performance of reinforcement learning based trading systems is proposed to make buy and sell suggestions for investors in their daily stock trading so as to maximize their profit in the dynamic stock market.

In (Chang, Fan, & Liu, 2009) a new model using Piecewise Linear Representations (PLR) and Artificial Neural Networks (ANNs) is proposed to analyze the nonlinear relationships between the stock closed price and various technical indexes, and to capture the knowledge of trading signals that are hidden in historical data. The learned ANN model is used to predict the future trading signals on a daily basis. Secondly, a trading decision is triggered by developing a dynamic threshold decision system. Another forecasting model integrating the case based dynamic window (CBDW) and the neural network is applied by (Chang et al., 2009) to predict the right turning points in stock trading, so as to maximize the investing revenue. In (Teixeira, & De Oliveira, 2010) a method using together the well known k-NN classifier and some common tools of technical analysis, like technical indicators, stop loss, stop gain and RSI filters is proposed with the purpose of investigating the feasibility of using an intelligent trading system in real market conditions, considering real companies of São Paulo Stock Exchange and transaction costs. An effective trading signal detection system using Piecewise Linear Representations (PLR) and Artificial Neural Networks (ANNs) is proposed in (Chang et al., 2011) to capture the knowledge of trading signals hidden in historical prices by analyzing the nonlinear relationships between the stock closed price and various technical indexes. The trading decision in the model is further triggered by a dynamic threshold bound which helps to gain significant profit amount during trading. In (Rodríguez-González et al., 2011) a trading system based on fundamental or chartist analysis is designed to improve the investment techniques. The main idea of the system is to generate trading points based on a financial indicator namely, relative strength index which is further calculated by a feed forward neural network. Another intelligent trading system using technical analysis, the Artificial Bee Colony Algorithm (ABC), a selection of past values, nearest neighbor classification (k-NN) and its variation, the Adaptive Classification and Nearest Neighbor is discussed in (Brasileiro et al., 2013). In (Chen, 2014) a high-order fuzzy time series model based on entropy-based partitioning and adaptive expectation model has shown its superiority compared to other conventional fuzzy time series models in generating decision rules as investment references for stock investors.

3. Computational Efficient Functional Link Artificial Neural Network (CEFLANN)

Computationally efficient FLANN is a single layer ANN having two components such as Functional expansion component and Learning component. In this network a set of highly nonlinear trigonometric basis functions are used in functional expansion block (FEB) that helps to capture the nonlinearity involves in the input space and to produce corresponding output space. The underlying betterment of the network lies in the fact of using an efficient FEB for input output mapping instead of using several hidden layers like traditional Multi Layer Perceptron network. Due to the single FEB, it possesses higher rate of convergence and lesser computational cost than those of a MLP structure. Unlike earlier FLANNS, where each input in the input pattern is expanded through a set of nonlinear functions, here all the inputs of the input pattern passes through a few set of nonlinear functions to produce the expanded input pattern (Bebarta, Biswal, & Dash, 2012; Dash, Dash, & Bisoi, 2014, Rout et al., 2015). Fig.1 depicts the single layer computationally efficient FLANN architecture. With the order n any d dimensional input pattern $X=[x_1, x_2 \dots x_d]^T$ is expanded to a m dimensional pattern CX by Trigonometric functional expansion as $CX=[cx_1,cx_2 \dots cx_d, cx_{d+1},cx_{d+2}, \dots cx_m]^T = [x_1, x_2 \dots x_d, cx_{d+1},cx_{d+2}, \dots cx_m]^T$ where m= d+n. For each order n, a trigonometric block (TB) containing a summation and tanh ( ) function is present in the FEB. Along with n number of TBs, a linear block is also present in the FEB which simply transfers the input features to the first d components of the expanded pattern. For each order n, the weighted sum of the components of the original input pattern is passed through the hyperbolic tangent (tanh ( )) function to produce an output o which is stored in cx_l (with d+1<=l<= m). Each o_i is obtained using the following formula.

$$o_i = \tanh\left(a_{i0} + \sum_{i=1,\ j=1}^{i=n,\ j=d} a_{ij} \times x_j \right) \tag{1}$$

Where a_{ij} is the associated parameter

After expansion, weight is initialized for each expanded unit and then the weighted sum of the components of the enhanced input pattern produces the output y using the following equation.

$$y_k = \sum_{j=1}^{m} w_{jk} cx_j \tag{2}$$

Where w_{jk} is the connection weight between jth expanded input and kth output layer node. The error obtained by comparing the output with desired output is used to update the weights of the FLANN structure by a weight updating algorithm.

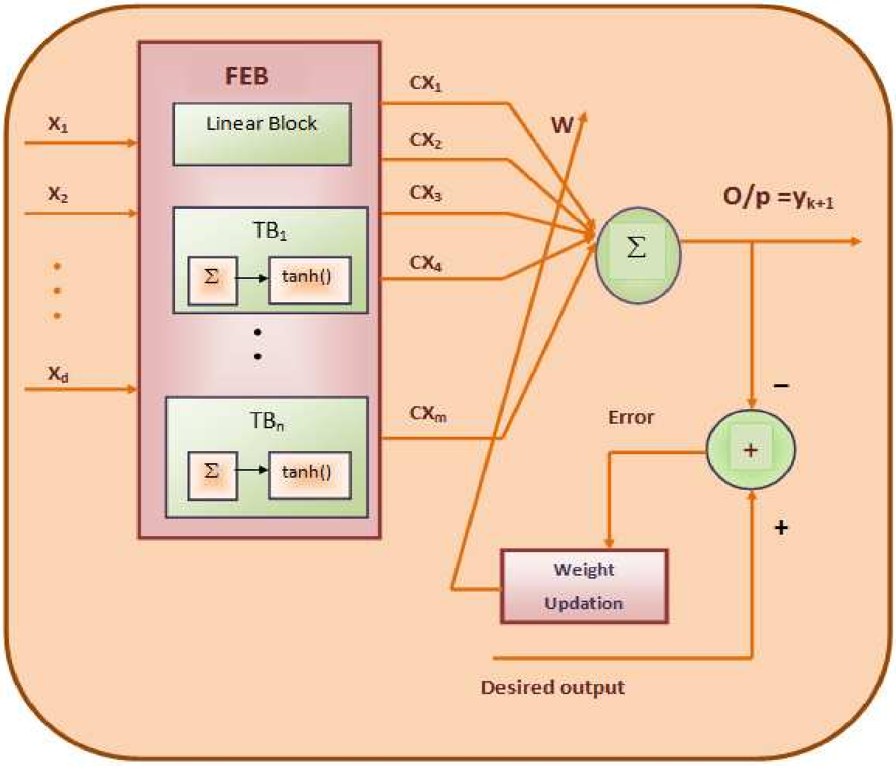

Fig. 1 Architecture of Computational Efficient FLANN (CEFLANN)

4. Extreme learning machine

Extreme learning machine is a recently introduced learning algorithm for single-hidden layer feed-forward neural networks (SLFNs) which randomly chooses the weights of connections between the input variables and neurons in the hidden layer and the bias of neurons in the hidden layer and analytically determines the output weights instead of iterative tuning (Huang, Zhu, & Siew, 2006). ELM not only has the capability of extremely fast learning and testing speed but also tends to achieve better generalization performance. The main advantage of ELM is that the hidden layer of SLFNs need not be tuned and it can work with a wide range of activation functions including piecewise continuous functions (Huang, Wang, & Lan, 2011; Huang et al., 2012; Bueno-Crespo et al., 2013; Luo, & Zhang, 2014). With a given a set of N training dataset D = (x_i, y_i), i=1 to N where each x_i is a d dimensional input pattern and y_i is the desired output, activation function for hidden layer nodes, H number of hidden layer nodes and a linear activation function in the output neuron, the output function of ELM for SLFN can be represented as:

$$y_i = \sum_{j=0}^{H} w_j h_j(x_i) \tag{3}$$

Where hj(x) is the activation function of hidden layer and W is the weight vector connecting the hidden layer neurons to output layer neuron. Equation (1) can be written as

$$Y=MW \tag{4}$$

Where M is a *N X (H+1)* hidden layer feature mapping matrix in which ith row specifies the hidden layer's output vector for an instance xi. Equation (2) being a linear system can be solved by

$$W = M^{\Psi} Y = (M^T M + \alpha I)^{-1} M^T Y \tag{5}$$

Where M^{Ψ} is the Moore-Penrose generalized inverse of matrix M,

α>0 is the regularization parameter and I is the M$\times$M identity matrix.

The proposed approach uses the same concept of the ELM where the output weights are obtained analytically using a robust least squares solution including a regularization parameter. The use of least squares method with regularization parameter will help in improving the performance of ELM in presence of noisy data.

5. Detailed Steps of Stock Trading

In this section, a novel decision support system using a computational efficient functional link artificial neural network (CEFLANN) and a set of rules based on technical analysis is proposed to generate the trading decisions more effectively. Instead of training the CEFLANN network using traditional back propagation algorithm, the ELM learning is proposed for the network. Six popular technical indicators calculated from the historical stock index price values are used as the input features for the proposed model. Here predicting trading decisions is cast as a classification problem with three class values representing the buy, hold and sell signals. Instead of using discrete class values during training of the network, a continuous trading signal within range 0 to 1 are fed to the network. The new trading signals in the range 0 to 1 helps to provide more detailed information regarding stock trading related to the original price variations. Further the outputs from the CEFLANN model is transformed in to a simple trading strategy with buy, hold and sell signals using suitable rules. The framework figure of the proposed model is shown in Fig. 2.

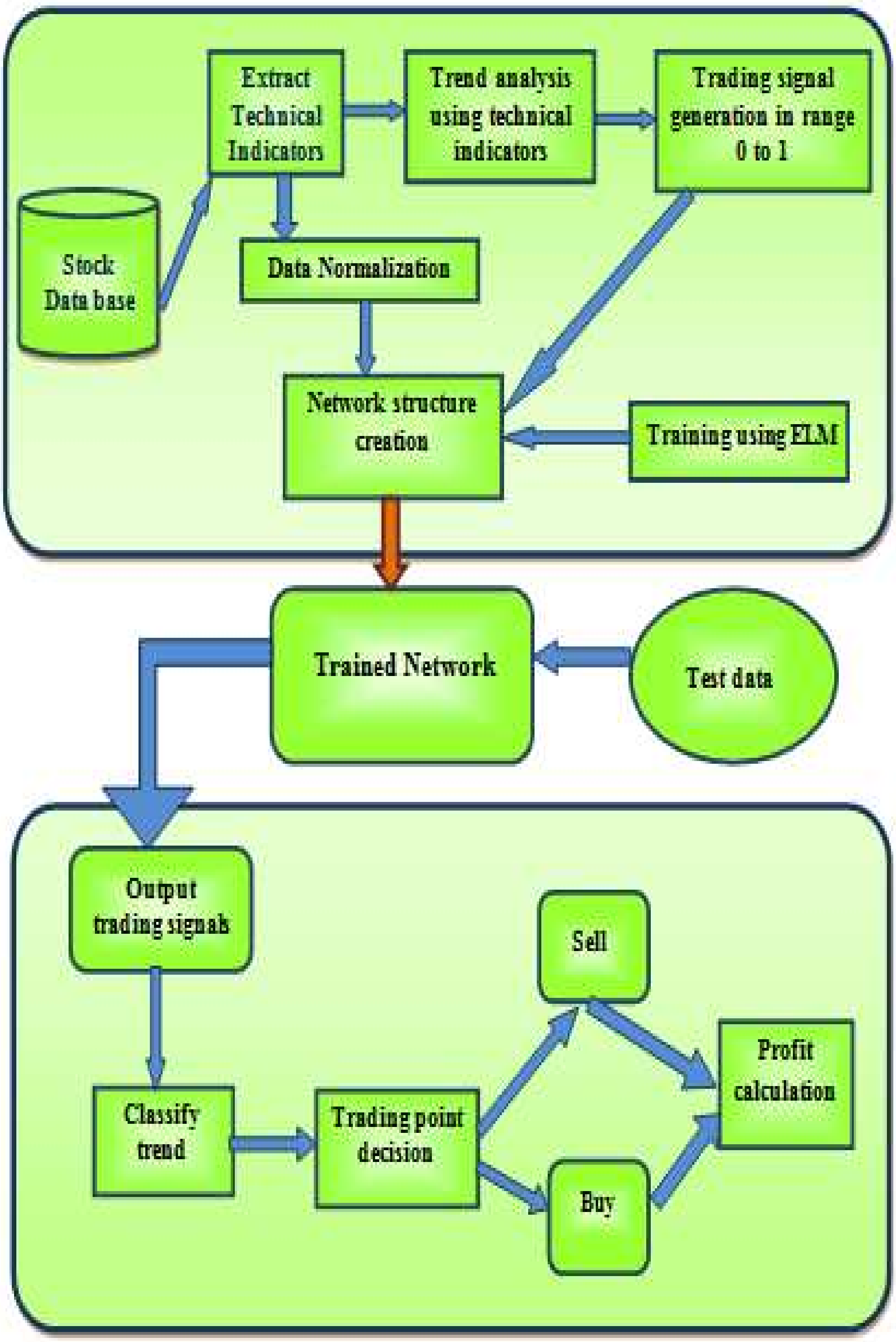

Fig. 2 Proposed Model for Stock Trading

The detailed steps of stock trading using CEFLANN model trained with ELM are as follows:

Step 1: Extract Technical indicators

In literature, researchers have used different types of technical indicators to monitor the future movement of stock prices and in setting up trading rules for buy–sell–hold decisions. In this study, six popular technical indicators i.e. MA_{15}, $MACD_{26}$, K_{14}, D_3, RSI_{14}, WR_{14} are chosen as input to the proposed model.

The technical indicators are calculated from historical prices as follows:

Simple Moving Average (MA):

It is the simple statistical mean of previous n day closing price, that normally smoothies out the price values. In this study value of t is set to 25.

$$MA_t = \frac{1}{t}\sum_{i=1}^{t} cp(i) \qquad (6)$$

Where cp(i) is the closing price.

Moving Average Convergence and Divergence (MACD):

The MACD shows the relationship between two exponential moving averages of prices.

$$\begin{aligned} &MACD = EMA_{12} - EMA_{26} \\ &EMA(i) = (CP(i) - EMA(i-1)) \times Multiplier + EMA(i-1) \\ &where \quad Multiplier = 2/(no\ of\ days\ to\ be\ considered + 1) \end{aligned} \qquad (7)$$

Stochastic KD:

Stochastic provides a mean of measuring price movement velocity. K% measures the relative position of current closing price in a certain time range, whereas D% specifies the three day moving average of K%.

$$\begin{aligned} &K\%(i) = \frac{cp(i) - L_t}{H_t - L_t} \times 100 \\ &D\%(i) = (K\%(i-2) + K\%(i-1) + K\%(i))/3 \end{aligned} \qquad (8)$$

Where cp(i) is the closing price, L_t is the lowest price of last t days, H_t is the highest price of last t days.

Relative Strength Index (RSI):

RSI is a momentum indicator calculated as follows:

$$RSI = 100 - \frac{100}{1 + RS}$$
$$where \quad RS = \frac{Average \quad of \quad t \quad day's \; up \; closes}{Average \quad of \quad t \quad day's \; down \quad closes} \tag{9}$$

Larry William's R%:

William's R% is a stochastic oscillator, calculated as follows:

$$R\%(i) = \frac{H_t - cp(i)}{H_t - L_t} \times 100 \tag{10}$$

Where cp(i) is the closing price, L_t is the lowest price of last t days, H_t is the highest price of last t days.

Step 2: Trend Analysis using Technical Indicators

Gaining profit or loss from stock trading ultimately depends on analysis of future movement of stock price values. In literature different technical indicators are used for successful classification of up and down movements in stock price index values. In this study rules using MA are used for classifying the stock market movement as upward (Uptrend) or downward (downtrend) as follows:

- If closing price value leads its MA_{15} and MA_{15} is rising for last 5 days then trend is Uptrend i.e. trend signal is 1.
- If closing price value lags its MA_{15} and MA_{15} is falling for last 5 days then trend is Downtrend i.e. trend signal is 0.

However, if none of these rules are satisfied then stock market is said to have no trend.

To illustrate the use of the above rules, Table 1 represents a sample time series data set representing the closing price and calculated moving average values. Using the specified rules the trend analyzed are specified in the third column of the table. The trend analysis of the sample dataset using moving average is also given in Fig. 3.

Table 1 Example of trend analysis on sample data set

Time series	**closing price**	**MA**	**Trend**
1	1877.7	1947.537	down
2	1886.76	1925.195	down
3	1904.01	1920.275	down
4	1941.28	1918.208	no
5	1927.11	1916.938	no
6	1950.82	1917.248	no
7	1964.58	1917.027	no

8	1961.63	1916.814	no
9	1985.05	1920.144	up
10	1982.3	1921.038	up
11	1994.65	1925.467	up
12	2039.68	1998.583	up
13	2038.25	2005.992	up
14	2039.33	2011.893	up
15	2039.82	2016.909	up
16	2041.32	2022.221	up
17	2063.5	2038.003	up
18	2060.31	2062.456	no
19	2059.82	2063.689	no
20	2026.14	2061.979	down
21	2035.33	2061.086	down
22	2002.33	2057.725	down

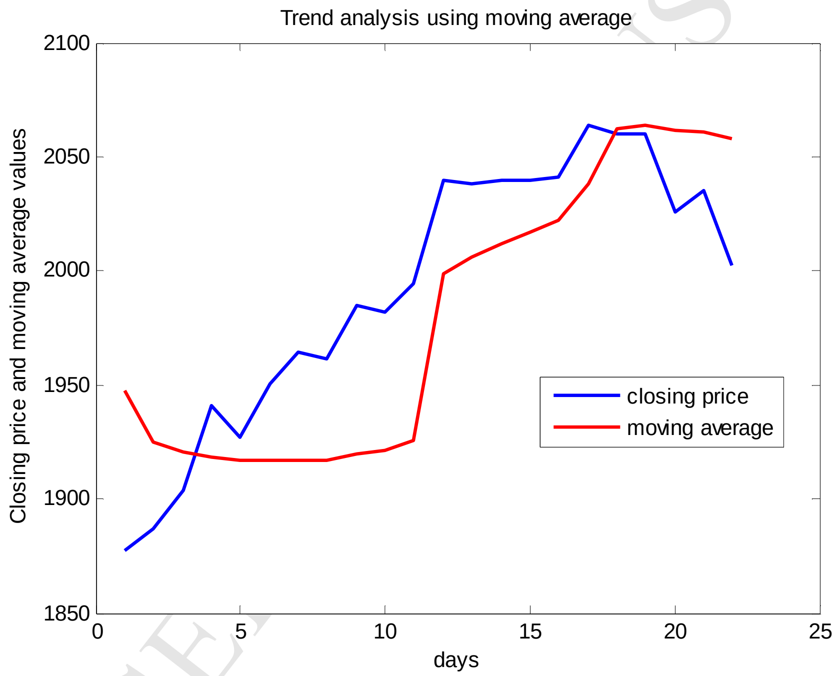

Fig. 3 Example of trend analysis on sample data set

Step 3: Trading Signal Generation from Trend Analysis

Neural network models are supervised models, which need to be trained on existing input and output pattern. In our study the six technical indicators are taken as input and the model will produce trading decision of buy, sell or hold from the trend analysis. As buy, sell and hold are the discrete values and normally output of neural network is continuous values within range 0 to 1, so instead of using the discrete signal in training of network, trading signals in range 0 to 1 are generated using momentum of the stock prices. With reference to (Chang, Fan, & Liu, 2009; Chang et al., 2009; Chang et al., 2011) a new trading signal is generated in range 0 to 1 reflecting the price variation. It also provides more detailed information to make precise stock trading decision. It provides more insightful information related to the movement of the stock price. The continuous trading signals Tr_i are generated using following rules:

For up trend:

$$Tr_i = \frac{[cp_i - \min cp]}{[\max cp - \min cp]} \times 0.5 + 0.5 \quad (11)$$

$$\min cp = \min(cp_i, cp_{i+1}, cp_{i+2})$$

$$\max cp = \max(cp_i, cp_{i+1}, cp_{i+2})$$

For down trend:

$$Tr_i = \frac{[cp_i - \min cp]}{[\max cp - \min cp]} \times 0.5 \quad (12)$$

Where cp_i , cp_{i+1} , cp_{i+2} , are the closing price of the i^{th}, $(i+1)^{th,}$ $(i+2)^{th}$ trading days respectively.

Table 2 represents the trading signal generated from the trend analysis for the sample data set using the equation (11) and (12).

Table 2 Example of trading signal generation from trend analysis for the sample data set

Time series	closing price	MA	Trend	Trading Signal
1	1877.7	1947.537	down	0
2	1886.76	1925.195	down	0
3	1904.01	1920.275	down	0
4	1941.28	1918.208	no	0.2988
5	1927.11	1916.938	no	0
6	1950.82	1917.248	no	0
7	1964.58	1917.027	no	0.0629
8	1961.63	1916.814	no	0
9	1985.05	1920.144	up	0.6113
10	1982.3	1921.038	up	0.5
11	1994.65	1925.467	up	0.5
12	2039.68	1998.583	up	1
13	2038.25	2005.992	up	0.5
14	2039.33	2011.893	up	0.5
15	2039.82	2016.909	up	0.5
16	2041.32	2022.221	up	0.5

17	2063.5	2038.003	up	1
18	2060.31	2062.456	no	1
19	2059.82	2063.689	no	1
20	2026.14	2061.979	down	0.3607
21	2035.33	2061.086	down	0.5
22	2002.33	2057.725	down	0.5

Step 4: Data Normalization

Originally the six technical indicator values represent continuous values in different ranges. So the input data is scaled in the range 0 to 1 using the min max normalization as follows:

$$y = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{13}$$

Where y = normalized value.

x = value to be normalized

x_{min} = minimum value of the series to be normalized

x_{max} = maximum value of the series to be normalized

Scaling the input data ensures that larger value input attributes does not overwhelm smaller value inputs.

Step 5: Network Structure Creation and Training using ELM

CEFLANN is a single layer neural network with only an output layer. In this study we have used a network with six inputs representing the normalized six technical indicator values and one output neuron for producing the trading signals. The network performance varies based on the selected expansion order and learning technique used. So initially with a suitable expansion order and random values of associated parameters used in expansion, the network is created. Further with the ELM learning, the output weights of the network is obtained analytically using a robust least squares solution including a regularization parameter. The regularization parameter value is set through a parameter selection process.

The working principle of the Computationally Efficient Functional Link Artificial Neural Network trained using ELM is detailed as follows:

Step 1: Choose a suitable expansion order n for functional expansion of CEFLANN.

Step 2: For each order n, randomly initialize the associated parameters and find a corresponding output component o_i by passing the weighted sum of the components of the original input pattern through a tanh( ) using equation (1) .

Step 3: Obtain the expanded input pattern by joining the s dimensional input vector with n number of o_i components as the output of the functional expansion block (FEB) and represent it as matrix M specified in equation (4).

Step 4: Find the output weight matrix W using robust least square solution as specified in equation (5).

Step 6: Finally generate the output of the network as the weighted sum of the components of the enhanced input pattern CX using the equation (2).

Step 6: Trend Determination from Output Trading Signal

After the training process, a new set of test data is applied to the trained network to produce a set of outputs. Output value of the network is the trading signal (OTr), i.e., the continuous value in range 0 to 1. To make trading decision, it is first required to track the trend and decide when to trade. The uptrend and down trend are classified from the output trading signals (OTr_i) using the following rules:

$$\text{If } OTr_i > \text{mean (Tr) predicted trend is up (1)} \quad \text{else predicted trend is down (0)} \tag{14}$$

Table 3 represents the output trading signal obtained from the network and the determined trend using equation (14) for the sample data set. Here the mean (Tr) is considered as 0.5.

Table 3 Example of trend determination from output trading signal over the sample data set

Time series	**OTr**	**Trend**	Trading Decision
1	0.1394	down	
2	0.0828	down	
3	0.0756	down	
4	0.3178	down	
5	0.0601	down	
6	0.153	down	
7	0.0445	down	
8	0.6975	up	**buy**
9	0.8558	up	
10	0.7318	up	
11	0.6696	up	
12	0.9103	up	

13	0.7158	up	
14	0.8436	up	
15	0.5323	up	
16	0.7417	up	
17	0.86442	up	
18	0.8907	up	
19	0.8433	up	
20	0.2254	down	**sell**
21	0.3474	down	
22	0.4195	down	

Step 7: Trading Point Decision from Predicted Trend

After obtaining the stock movement direction, trading points are obtained using straightforward trading rules as follows:

If the next day trend=Uptrend then decision is BUY (15)
If Buy decision exists then HOLD
If the next day trend=Downtrend then decision is SELL
If SELL decision exists then HOLD

The fourth column of Table 3 represents the trading decision taken from the predicted trend using rules given in equation (15) for the sample data set.

Step 8: Profit Calculation

The main parameter adopted for performance evaluation is the profit percentage obtained during the test period. The profit percentage is generated from a combination of buy and sells transactions as follows:

$$\mathrm{Profit}\,\% = \sum_{i=1}^{k} (cp_{s_i} - cp_{b_i}) / cp_{b_i} \times 100 \qquad (16)$$

$$Where\; k = number\; of\; transactions$$

$$cp_{s_i} = selling\; price\; of\; i^{th}\; transaction$$

$$cp_{b_i} = buying\; price\; of\; i^{th}\; transaction$$

6. Empirical Study

In this section the performance of the proposed model is validated for stock trading problem by applying it on two benchmark stock index data sets. The model performance is also compared with some other known classifiers like support vector machine (SVM), Naive Bayesian model, K nearest neighbor model (KNN) and decision tree (DT) model.

6.1 Dataset Description

Five years of historical stock index price values of two stock indices (BSE SENSEX and S&P 500) are used in this study. Initially both the data sets are divided into training and testing sets. For BSE data set the training set consists of 1000 patterns and remaining 208 patterns are used for testing and for S&P dataset the training set consists of 1000 patterns leaving the 221 patterns for testing.

The detail of the data set is given in Table 4. Initially the six technical indicators are extracted from the historical prices and normalized using min max normalization to be fed as input to the network. Table 5 summarizes the statistical analysis of the selected technical indicators for both the stock indices. As the aim of the study is to derive short term trading points from trend analysis, so MA_{15} has used for finding initial up down movements of the stock prices. Instead of using discrete value as output during training of the network, continuous trading signals are generated from the trend and used during training process.

Table 4 Data set Description

Data Set	Period
BSE SENSEX	4- Jan- 2010 to 31-Dec-2014
S&P 500	4- Jan- 2010 to 31-Dec-2014

Table 5 Summary statistics of selected technical indicators

Data set	Technical Indicators	Min	Max	Mean	Std
BSE SENSEX	SMA	1.5782e+004	2.8200e+004	1.9515e+004	2.9019e+003
	MACD	-562.6344	597.0689	59.0154	218.5564
	K%	1.6954	99.3916	56.2792	32.1840
	D%	4.4787	97.3337	56.2341	29.2545
	RSI	5.0617	95.4554	53.9376	18.0426
	LW R%	-98.3046	-0.6084	-43.7208	32.1840
S&P 500	SMA	1.0739e+003	2.0565e+003	1.4705e+003	286.2986
	MACD	-42.4684	25.6214	5.1836	12.0395
	K%	0	100	64.7144	31.4893
	D%	1.5690	99.1513	64.6599	28.1762
	RSI	9.3436	99.2987	57.1475	16.5881
	LW R%	-100	0	-35.2856	31.4893

6.2 Parameter Setup

For CEFLANN with order n, input size s, the number of associated parameters used in functional expansion is n(s+1) and number of weights between expanded pattern and output neuron is (n+s). Hence total number of unknown parameters need to be tuned by a learning algorithm is (n+s) + n(s+1). Using ELM the n(s+1) number of associated parameters are chosen randomly, and the remaining (n+s) number of parameters are obtained using the equation (5). The performance of the proposed model depends on different factors like the order of expansion, value of regularization parameter, input space size and so on. So initially through a number of simulations the controlling parameters of the model are derived.

6.3 Experimental Result Analysis

The generalization ability of the network is measured through a 5 fold cross validation approach applied on the initial 1000 samples taken as training dataset. The training dataset is divided into 5 groups, where first 4 randomly chosen groups are used for training and remaining one is used for validation. The average performance out of the 20 independent runs is considered for both the dataset. Finally the trained network is applied on the test pattern i.e., the out of sample data, which has not been used during training and validation. Fig. 4 and 5 show the trading signals generated from the CEFLANN model for both the dataset. The black line represents the average value of the trading signals obtained from the network, which is used as the threshold value for finding up trend and down trend. Further using the rules of equation (15), the trading points are generated.

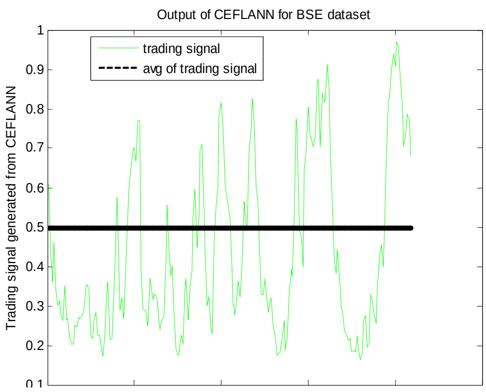

Fig. 4 Output trading signal obtained from CEFLANN model for BSE SENSEX data set

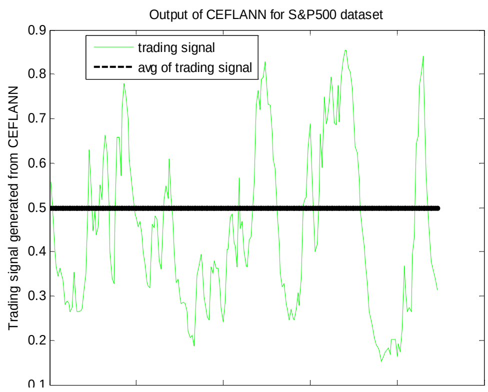

Fig. 5 Output trading signal obtained from CEFLANN model for S&P500 data set

Fig. 6 and 7 represents the initial trading points generated using the technical indicator MA_{15}. Trading points predicted by using the proposed model for both the data set are shown in Fig. 8 and 9. The overall performance of the model compared to other soft computing techniques like SVM, Naïve Bayesian, KNN and decision tree (DT) for the two dataset are shown in Table 6 and 7 respectively. Through a series of experimental tests, the proposed model consistently generates highest profit among others.

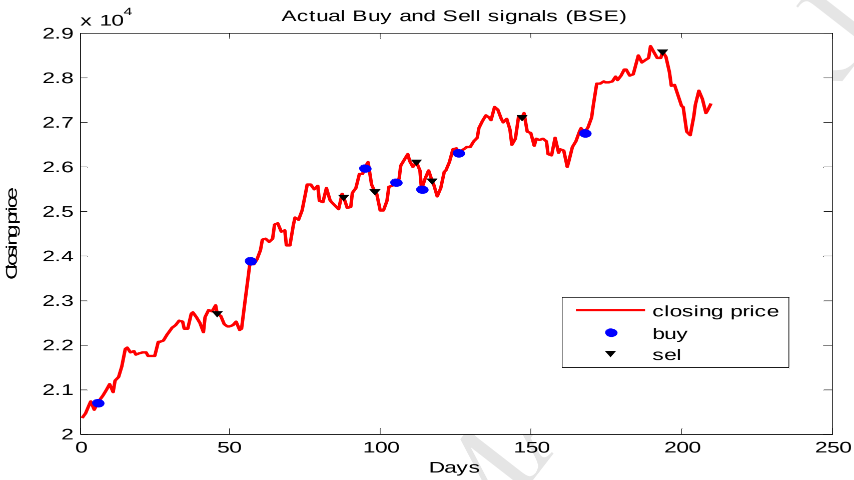

Fig. 6 Initial Trading points generated using MA_{15} for BSE SENSEX data set

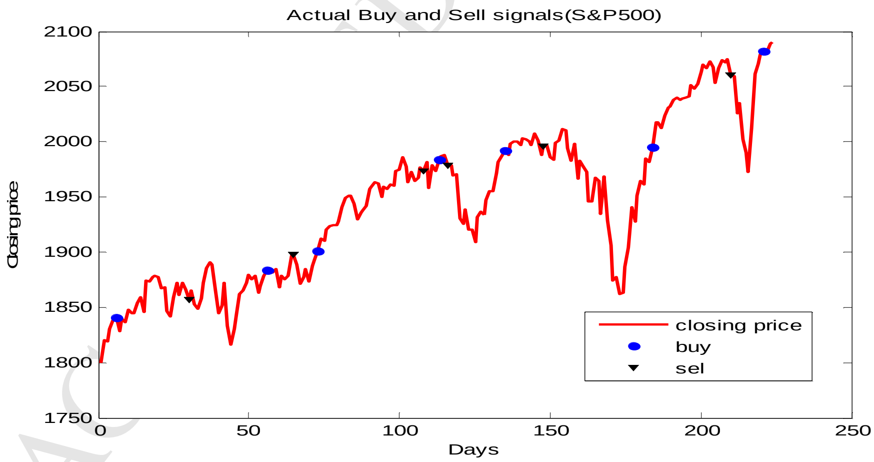

Fig. 7 Initial Trading points generated using MA_{15} for S&P500 data set

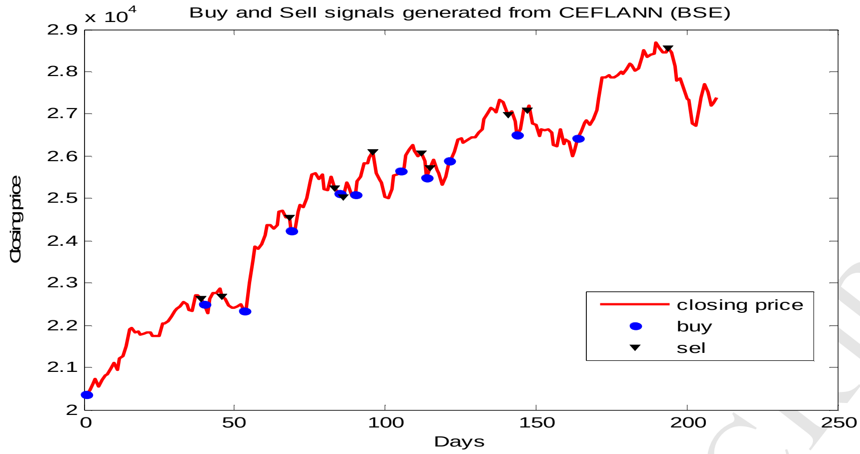

Fig. 8 Trading points from CEFLANN model for BSE SENSEX data set

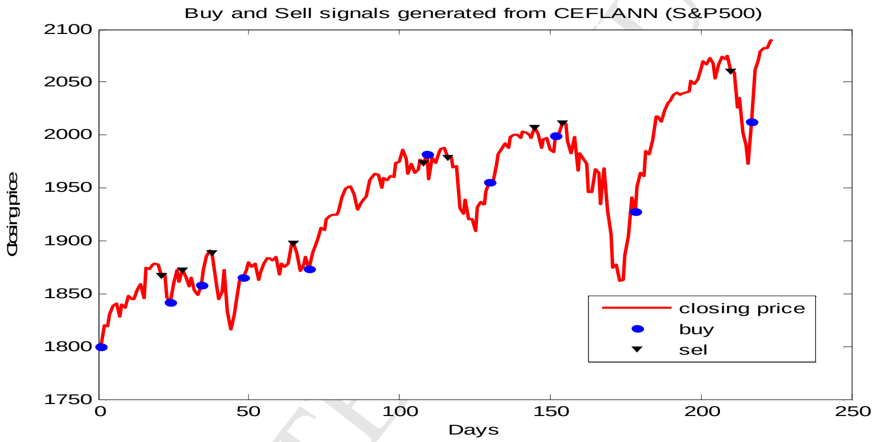

Fig. 9 Trading points from CEFLANN model for S&P500 data set

Table 6 Performance comparison of stock trading models on BSE SENSEX data set

Performance Metrics	Actual (Using MA_{15})	CEFLANN	SVM	Naïve Bayesian	KNN	DT
No. of Buy signals	7	11	6	8	13	9
No. of hold signals	194	186	196	192	182	190
No. of sell signals	7	11	6	8	13	9
Profit %	25.8282	**47.2007**	35.8099	42.3267	30.8015	33.4523

Table 7 Performance comparison of stock trading models on S&P500 data set

Performance Metrics	Actual (Using MA_{15})	CEFLANN	SVM	Naïve Bayesian	KNN	DT
No. of Buy signals	6	9	9	10	16	13
No. of hold signals	209	203	203	201	189	195
No. of sell signals	6	9	9	10	16	13
Profit %	8.6474	**24.2872**	17.5163	22.1668	13.6787	15.5161

7. Conclusion

This study has proposed a novel decision support system for developing efficient stock trading strategies, which may provide attractive benefits for investors. The model has integrated technical analysis with machine learning techniques for efficient generation of stock trading decisions. In this study the problem of stock trading decision generation is cast as a classification task. A classification model using the computational efficient functional link artificial neural network (CEFLANN) with ELM learning approach is proposed for generating the stock trading decisions. The outputs from the CEFLANN model is transformed in to a simple trading strategy with buy, hold and sell signals using suitable rules. From the experimental result analysis it is clearly apparent that the proposed model provides superior profit percentage compared to some other known classifiers such as support vector machine (SVM), Naive Bayesian model, K nearest neighbor model (KNN) and decision tree (DT) model. Hence instead of taking trading decision based on particular technical indicators, it is more profitable to take trading decision using combination of technical indicators with computational intelligence tools.

Further the work can be extended by validating the proposed model over more real world datasets. More work will be done on structure optimization of the model by using efficient optimization algorithms such as differential evolution, harmony search, shuffled frog leaping algorithm and soon. More technical analysis will also be explored in future.

References

Bebarta, D. K., Biswal, B., & Dash, P. K. (2012). Comparative study of stock market forecasting using different functional link artificial neural networks. *International Journal of Data Analysis Techniques and Strategies*, *4*(4), 398-427.

Brasileiro, R. C., Souza, V. L., Fernandes, B. J., & Oliveira, A. L. (2013, June). Automatic method for stock trading combining technical analysis and the Artificial Bee Colony Algorithm. *IEEE Congress on* Evolutionary *Computation (CEC)*, (pp. 1810-1817). IEEE.

Bueno-Crespo, A., García-Laencina, P. J., & Sancho-Gómez, J. L. (2013). Neural architecture design based on extreme learning machine. *Neural Networks*, *48*, 19-24.

Ballings, M., Van den Poel, D., Hespeels, N., & Gryp, R. (2015). Evaluating multiple classifiers for stock price direction prediction. Expert Systems with Applications, 42(20), 7046-7056.

Chang, P. C., Fan, C. Y., & Liu, C. H. (2009). Integrating a piecewise linear representation method and a neural network model for stock trading points prediction. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, *39*(1), 80-92.

Chang, P. C., Liu, C. H., Lin, J. L., Fan, C. Y., & Ng, C. S. (2009). A neural network with a case based dynamic window for stock trading prediction. *Expert Systems with Applications*, *36*(3), 6889-6898.

Chang, P. C., Liao, T. W., Lin, J. J., & Fan, C. Y. (2011). A dynamic threshold decision system for stock trading signal detection. *Applied Soft Computing*, *11*(5), 3998-4010.

Chen, M. Y. (2014). A high-order fuzzy time series forecasting model for internet stock trading. *Future Generation Computer Systems*, *37*, 461-467.

Dash, R., Dash, P. K., & Bisoi, R. (2014). A self adaptive differential harmony search based optimized extreme learning machine for financial time series prediction. Swarm and Evolutionary Computation, 19, 25-42.

Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2006). Extreme learning machine: theory and applications. *Neurocomputing*, *70*(1), 489-501.

Huang, G. B., Wang, D. H., & Lan, Y. (2011). Extreme learning machines: a survey. *International Journal of Machine Learning and Cybernetics*, *2*(2), 107-122.

Huang, G. B., Zhou, H., Ding, X., & Zhang, R. (2012). Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *42*(2), 513-529.

Kablan, A. (2009). Adaptive neuro-fuzzy inference system for financial trading using intraday seasonality observation model. *World Academy of Science, Engineering and Technology*, *58*, 479-488.

Kara, Y., Boyacioglu, M. A., & Baykan, Ö. K. (2011). 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*, *38*(5), 5311-5319.

Lee, J. W., Park, J., Lee, J., & Hong, E. (2007). A multiagent approach to Q-learning for daily stock trading. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 37*(6), 864-877.

Luo, M., & Zhang, K. (2014). A hybrid approach combining extreme learning machine and sparse representation for image classification. *Engineering Applications of Artificial Intelligence*, *27*, 228-235.

Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications*, *42*(1), 259-268.

Rodríguez-González, A., García-Crespo, Á., Colomo-Palacios, R., Iglesias, F. G., & Gómez-Berbís, J. M. (2011). CAST: Using neural networks to improve trading systems based on technical analysis by means of the RSI financial indicator. *Expert systems with Applications*, *38*(9), 11489-11500.

Rout, A. K., Dash, P. K., Dash, R., & Bisoi, R. (2015). Forecasting financial time series using a low complexity recurrent neural network and evolutionary learning approach. *Journal of King Saud University-Computer and Information Sciences*.

Teixeira, L. A., & De Oliveira, A. L. I. (2010). A method for automatic stock trading combining technical analysis and nearest neighbor classification. *Expert systems with applications*, *37*(10), 6885-6890.

Wen, Q., Yang, Z., Song, Y., & Jia, P. (2010). Automatic stock decision support system based on box theory and SVM algorithm. Expert Systems with Applications, 37(2), 1015-1022.

Wysocki, A., & Lawrynczuk, M. (2013, September). An investment strategy for the Stock Exchange using neural networks. *Federated Conference on* Computer *Science and Information Systems (FedCSIS), (*pp. 183-190). IEEE.

Wu, M. C., Lin, S. Y., & Lin, C. H. (2006). An effective application of decision tree to stock trading. Expert Systems with Applications, 31(2), 270-274.