Expert Systems with Applications 42 (2015) 259–268

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques

Jigar Patel, Sahil Shah, Priyank Thakkar *, K Kotecha

Computer Science & Engineering Department, Institute of Technology, Nirma University, Ahmedabad, Gujarat, India

ARTICLE INFO

Article history:
Available online 6 August 2014

Keywords:
Naive-Bayes classification
Artificial neural networks
Support vector machine
Random forest
Stock market

ABSTRACT

This paper addresses problem of predicting direction of movement of stock and stock price index for Indian stock markets. The study compares four prediction models, Artificial Neural Network (ANN), Support Vector Machine (SVM), random forest and naive-Bayes with two approaches for input to these models. The first approach for input data involves computation of ten technical parameters using stock trading data (open, high, low & close prices) while the second approach focuses on representing these technical parameters as trend deterministic data. Accuracy of each of the prediction models for each of the two input approaches is evaluated. Evaluation is carried out on 10 years of historical data from 2003 to 2012 of two stocks namely Reliance Industries and Infosys Ltd. and two stock price indices CNX Nifty and S&P Bombay Stock Exchange (BSE) Sensex. The experimental results suggest that for the first approach of input data where ten technical parameters are represented as continuous values, random forest outperforms other three prediction models on overall performance. Experimental results also show that the performance of all the prediction models improve when these technical parameters are represented as trend deterministic data.

1. Introduction

Predicting stock and stock price index is difficult due to uncertainties involved. There are two types of analysis which investors perform before investing in a stock. First is the fundamental analysis. In this, investors looks at intrinsic value of stocks, performance of the industry and economy, political climate etc. to decide whether to invest or not. On the other hand, technical analysis is the evaluation of stocks by means of studying statistics generated by market activity, such as past prices and volumes. Technical analysts do not attempt to measure a security' s intrinsic value but instead use stock charts to identify patterns and trends that may suggest how a stock will behave in the future. Efficient market hypothesis by Malkiel and Fama (1970) states that prices of stocks are informationally efficient which means that it is possible to predict stock prices based on the trading data. This is quite logical as many uncertain factors like political scenario of country, public image of the company will start reflecting in the stock prices. So, if the information obtained from stock prices is pre-processed efficiently and appropriate algorithms are applied then trend of stock or stock price index may be predicted.

Since years, many techniques have been developed to predict stock trends. Initially classical regression methods were used to predict stock trends. Since stock data can be categorized as non-stationary time series data, non-linear machine learning techniques have also been used. Artificial Neural Networks (ANN) and Support Vector Machine (SVM) are two machine learning algorithms which are most widely used for predicting stock and stock price index movement. Each algorithm has its own way to learn patterns. ANN emulates functioning of our brain to learn by creating network of neurons. Hassan, Nath, and Kirley (2007) proposed and implemented a fusion model by combining the Hidden Markov Model (HMM), Artificial Neural Networks (ANN) and Genetic Algorithms (GA) to forecast financial market behavior. Using ANN, the daily stock prices were transformed to independent sets of values that become input to HMM. Wang and Leu (1996) developed a prediction system useful in forecasting mid-term price trend in Taiwan stock market. Their system was based on a recurrent neural network trained by using features extracted from ARIMA analyses. Empirical results showed that the networks trained using 4-year weekly data was capable of predicting up to 6 weeks market trend with acceptable accuracy. Hybridized soft computing techniques for automated stock market forecasting and trend analysis was introduced by Abraham, Nath, and Mahanti (2001). They used Nasdaq-100 index of Nasdaq stock

* Corresponding author.
E-mail addresses: priyank.thakkar@nirmauni.ac.in (P. Thakkar), director.it@nirmauni.ac.in (K Kotecha).

http://dx.doi.org/10.1016/j.eswa.2014.07.040
0957-4174/

market with neural network for one day ahead stock forecasting and a neuro-fuzzy system for analysing the trend of the predicted stock values. The forecasting and trend prediction results using the proposed hybrid system were promising. Chen, Leung, and Daouk (2003) investigated the probabilistic neural network (PNN) to forecast the direction of index after it was trained by historical data. Empirical results showed that the PNN-based investment strategies obtained higher returns than other investment strategies examined in the study like the buy-and-hold strategy as well as the investment strategies guided by forecasts estimated by the random walk model and the parametric GMM models.

A very well-known SVM algorithm developed by Vapnik (1999) searches for a hyper plane in higher dimension to separate classes. Support vector machine (SVM) is a very specific type of learning algorithms characterized by the capacity control of the decision function, the use of the kernel functions and the scarcity of the solution. Huang, Nakamori, and Wang (2005) investigated the predictability of financial movement direction with SVM by forecasting the weekly movement direction of NIKKEI 225 index. They compared SVM with Linear Discriminant Analysis, Quadratic Discriminant Analysis and Elman Backpropagation Neural Networks. The experiment results showed that SVM outperformed the other classification methods. SVM was used by Kim (2003) to predict the direction of daily stock price change in the Korea composite stock price index (KOSPI). Twelve technical indicators were selected to make up the initial attributes. This study compared SVM with back-propagation neural network (BPN) and case-based reasoning (CBR). It was evident from the experimental results that SVM outperformed BPN and CBR.

Random forest creates n classification trees using sample with replacement and predicts class based on what majority of trees predict. The trained ensemble, therefore, represents a single hypothesis. This hypothesis, however, is not necessarily contained within the hypothesis space of the models from which it is built. Thus, ensembles can be shown to have more flexibility in the functions they can represent. This flexibility can, in theory, enable them to over-fit the training data more than a single model would, but in practice, some ensemble techniques (especially bagging) tend to reduce problems related to over-fitting of the training data. Tsai, Lin, Yen, and Chen (2011) investigated the prediction performance that utilizes the classifier ensembles method to analyze stock returns. The hybrid methods of majority voting and bagging were considered. Moreover, performance using two types of classifier ensembles were compared with those using single baseline classifiers (i.e. neural networks, decision trees, and logistic regression). The results indicated that multiple classifiers outperformed single classifiers in terms of prediction accuracy and returns on investment. Sun and Li (2012) proposed new Financial distress prediction (FDP) method based on SVM ensemble. The algorithm for selecting SVM ensemble's base classifiers from candidate ones was designed by considering both individual performance and diversity analysis. Experimental results indicated that SVM ensemble was significantly superior to individual SVM classifier. Ou and Wang (2009) used total ten data mining techniques to predict price movement of Hang Seng index of Hong Kong stock market. The approaches include Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), K-nearest neighbor classification, Naive Bayes based on kernel estimation, Logit model, Tree based classification, neural network, Bayesian classification with Gaussian process, Support Vector Machine (SVM) and Least Squares Support Vector Machine (LS-SVM). Experimental results showed that the SVM and LS-SVM generated superior predictive performance among the other models.

It is evident from the above discussions that each of the algorithms in its own way can tackle this problem. It is also to be noticed that each of the algorithm has its own limitations. The final prediction outcome not only depends on the prediction algorithm used but is also influenced by the representation of the input. Identifying important features and using only them as the input rather than all the features may improve the prediction accuracy of the prediction models. A two-stage architecture was developed by Hsu, Hsieh, Chih, and Hsu (2009). They integrated self-organizing map and support vector regression for stock price prediction. They examined seven major stock market indices. Specifically, the Self Organizing Map (SOM) was first used to decompose the whole input space into regions where data points with similar statistical distributions were grouped together, so as to contain and capture the non-stationary property of financial series. After decomposing heterogeneous data points into several homogenous regions, Support Vector Regression (SVR) was applied to forecast financial indices. The results suggested that the two stage architecture provided a promising alternative for stock price prediction. Genetic programming (GP) and its variants have been extensively applied for modeling of the stock markets. To improve the generalization ability of the model, GP have been hybridized with its own variants (Gene Expression Programming (GEP), Multi Expression Programming (MEP)) or with the other methods such as neural networks and boosting. The generalization ability of the GP model can also be improved by an appropriate choice of model selection criterion. Garg, Sriram, and Tai (2013) worked to analyse the effect of three model selection criteria across two data transformations on the performance of GP while modeling the stock indexed in the New York Stock Exchange (NYSE). It was found that FPE criteria had shown a better fit for the GP model on both data transformations as compared to other model selection criteria. Nair et al. (2011) predicted the next day's closing value of five international stock indices using an adaptive artificial neural network based system. The system adapted itself to the changing market dynamics with the help of genetic algorithm which tuned the parameters of the neural network at the end of each trading session. The study by Ahmed (2008) investigated the nature of the causal relationships between stock prices and the key macro-economic variables representing real and financial sector of the Indian economy for the period March, 1995–2007 using quarterly data. The study revealed that the movement of stock prices was not only the outcome of behavior of key macro-economic variables but it was also one of the causes of movement in other macro dimension in the economy. Mantri, Gahan, and Nayak (2010) calculated the volatilities of Indian stock markets using GARCH, EGARCH, GJR-GARCH, IGARCH & ANN models. This study used Fourteen years of data of BSE Sensex & NSE Nifty to calculate the volatilities. It was concluded that there was no difference in the volatilities of Sensex, & Nifty estimated under the GARCH, EGARCH, GJR GARCH, IGARCH & ANN models. Mishra, Sehgal, and Bhanumurthy (2011) tested for the presence of nonlinear dependence and deterministic chaos in the rate of returns series for six Indian stock market indices. The result of analysis suggested that the returns series did not follow a random walk process. Rather it appeared that the daily increments in stock returns were serially correlated and the estimated Hurst exponents were indicative of marginal persistence in equity returns. Liu and Wang (2012) investigated and forecast the price fluctuation by an improved Legendre neural network by assuming that the investors decided their investing positions by analysing the historical data on the stock market. They also introduced a random time strength function in the forecasting model. The Morphological Rank Linear Forecasting (EMRLF) method was proposed by Araújo and Ferreira (2013). An experimental analysis was conducted and the results were compared to Multilayer Perceptron (MLP) networks and Time-delay Added Evolutionary Forecasting (TAEF) method.

This study focuses on comparing prediction performance of ANN, SVM, random forest and naive-Bayes algorithms for the task

of predicting stock and stock price index movement. Ten technical parameters are used as the inputs to these models. This paper proposes Trend Deterministic Data Preparation Layer which converts continuous-valued inputs to discrete ones. Each input parameters in its discrete form indicates a possible up or down trend determined based on its inherent property. The focus is also to compare the performance of these prediction models when the inputs are represented in the form of real values and trend deterministic data. All the experiments are carried out using 10 years of historical data of two stocks Reliance Industries and Infosys Ltd. and two indices S&P BSE Sensex and CNX Nifty. Both stocks and indices are highly voluminous and vehemently traded in and so they reflect Indian economy as a whole.

The remainder of this paper is organized into following sections. Section 2 describes research data, the pre-processing of data and computation of financial parameters which serves as inputs. It also discusses about preparation of trend deterministic data. Four prediction models which are used in this study are discussed in Section 3. Section 4 shows experimental results. Discussions on the results achieved are reported in Section 5. Section 6 concludes the study.

2. Research data

Ten years of data of total two stock price indices (CNX Nifty, S&P BSE Sensex) and two stocks (Reliance Industries, Infosys Ltd.) from Jan 2003 to Dec 2012 is used in this study. All the data is obtained from <http://www.nseindia.com/> and <http://www.bseindia.com/> websites. These data forms our entire data set. Percentage wise increase and decrease cases of each year in the entire data set are shown in Table 1.

This study uses 20% of the entire data as the parameter selection data. This data is used to determine design parameters of predictor models. Parameter selection data set is constructed by taking equal proportion of data from each of the ten years. The proportion of percentage wise increase and decrease cases in each year is also maintained. This sampling method enables parameter setting data set to be better representative of the entire data set. This parameter selection data is further divided into training and hold-out set. Each of the set consists of 10% of the entire data. Table 2 depicts the number of increase and decrease cases for parameter selection data set. These statistics is for S&P BSE Sensex. Similar data analysis is done for CNX Nifty, Reliance Industries and Infosys Ltd.

Optimum parameters for predictor models are obtained by means of experiments on parameter selection data. After that, for comparing ANN, SVM, random forest and naive-Bayes, comparison data set is devised. This data set comprises of entire ten years of data. It is also divided in training (50% of the entire data) and hold-out (50% of the entire data) set. Details of this data set for S&P BSE SENSEX is shown in Table 3. These experimental settings are same as in Kara, Acar Boyacioglu, and Baykan (2011).

There are some technical indicators through which one can predict the future movement of stocks. Here in this study, total ten technical indicators as employed in Kara et al. (2011) are used. These indicators are shown in Table 4. Table 5 shows summary statistics for the selected indicators of two indices and two stocks. Two approaches for the representation of the input data are employed in this study. The first approach uses continuous value representation, i.e., the actual time series while the second one uses trend deterministic representation (which is discrete in nature) for the inputs. Both the representations are discussed here.

2.1. Continuous representation – the actual time series

Ten technical indicators calculated based on the formula as discussed in the Table 4 are given as inputs to predictor models. It is evident that each of the technical indicators calculated based on the above mentioned formula is continuous-valued. The values of all technical indicators are normalized in the range between [−1, +1], so that larger value of one indicator do not overwhelm the smaller valued indicator. Performance of all the models under study is evaluated for this representation of inputs.

2.2. Discrete representation – trend prediction data

A new layer of decision is employed which converts continuous valued technical parameters to discrete value, representing the trend. We call this layer "Trend Deterministic Data Preparation

Table 1
The number of increase and decrease cases percentage in each year in the entire data set of S&P BSE SENSEX.

Year	Increase	%	Decrease	%	Total
2003	146	58.63	103	41.37	249
2004	136	54.18	115	45.82	251
2005	147	59.04	102	40.96	249
2006	148	59.92	99	40.08	247
2007	139	55.82	110	44.18	249
2008	114	46.72	130	53.28	244
2009	127	52.70	114	47.30	241
2010	134	53.39	117	46.61	251
2011	116	47.15	130	52.85	246
2012	128	51.82	119	48.18	247
Total	1335	53.94	1139	46.06	2474

Table 2
The number of increase and decrease cases in each year in the parameter setting data set of S&P BSE SENSEX.

Year	Training			Holdout		
	Increase	Decrease	Total	Increase	Decrease	Total
2003	15	10	25	15	10	25
2004	14	11	25	14	11	25
2005	15	10	25	15	10	25
2006	15	10	25	15	10	25
2007	14	11	25	14	11	25
2008	11	13	24	11	13	24
2009	13	11	24	13	11	24
2010	13	12	25	13	12	25
2011	12	13	25	12	13	25
2012	13	12	25	13	12	25
Total	135	113	248	135	113	248

Table 3
The number of increase and decrease cases in each year in the comparison data set of S&P BSE SENSEX.

Year	Training			Holdout		
	Increase	Decrease	Total	Increase	Decrease	Total
2003	73	52	125	72	52	124
2004	68	58	126	67	58	125
2005	74	51	125	73	51	124
2006	74	50	124	73	50	123
2007	70	55	125	69	55	124
2008	57	65	122	57	65	122
2009	64	57	121	63	57	120
2010	67	59	126	66	59	125
2011	58	65	123	58	65	123
2012	64	60	124	63	60	123
Total	669	572	1241	661	572	1233

Table 4
Selected technical indicators & their formulas (Kara et al., 2011).

Name of indicators	Formulas
Simple n(10*here*)-day Moving Average	$\frac{C_t+C_{t-1}+\ldots+C_{t-9}}{n}$
Weighted n(10*here*)-day Moving Average	$\frac{(10)C_t+(9)C_{t-1}+\ldots+C_{t-9}}{n+(n-1)+\ldots+1}$
Momentum	$C_t - C_{t-9}$
Stochastic K%	$\frac{C_t-LL_{t-(n-1)}}{HH_{t-(n-1)}-LL_{t-(n-1)}} \times 100$
Stochastic D%	$\frac{\sum_{i=0}^{n-1} K_{t-i}}{10}\%$
Relative Strength Index (RSI)	$100 - \frac{100}{1+\left(\sum_{i=0}^{n-1} UP_{t-i}/n\right)/\left(\sum_{i=0}^{n-1} DW_{t-i}/n\right)}$
Moving Average Convergence Divergence (MACD)	$MACD(n)_{t-1} + \frac{2}{n+1} \times (DIFF_t - MACD(n)_{t-1})$
Larry William's R%	$\frac{H_n-C_t}{H_n-L_n} \times 100$
A/D (Accumulation/Distribution) Oscillator	$\frac{H_t-C_{t-1}}{H_t-L_t}$
CCI (Commodity Channel Index)	$\frac{M_t-SM_t}{0.015D_t}$

C_t is the closing price, L_t is the low price and H_t the high price at time t, $DIFF_t = EMA(12)_t - EMA(26)_t$, EMA is exponential moving average, $EMA(k)_t = EMA(k)_{t-1} + \alpha \times (C_t - EMA(k)_{t-1})$, α is a smoothing factor which is equal to $\frac{2}{k+1}$, k is the time period of k-day exponential moving average, LL_t and HH_t implies lowest low and highest high in the last t days, respectively. $M_t = \frac{H_t+L_t+C_t}{3}$, $SM_t = \frac{\left(\sum_{i=1}^{n} M_{t-i+1}\right)}{n}$, $D_t = \frac{\left(\sum_{i=1}^{n} |M_{t-i+1}-SM_t|\right)}{n}$, UP_t means upward price change while DW_t is the downward price change at time t.

Table 5
Summary statistics for the selected indicators.

Indicator	Max	Min	Mean	Standard deviation
Nifty				
SMA	6217.37	935.38	3789.68	1047.47
EMA	6214.38	940.35	3789.69	1054.51
MOM	748.40	−1372.70	17.81	7.55
STCK%	99.14	1.84	60.51	76.33
STCD%	97.90	4.08	60.50	57.75
MACD	277.17	−357.33	13.52	−13.39
RSI	100.00	1.42	56.40	46.28
WILLR%	−0.86	−98.16	−39.49	−23.67
A/D Osc	98.24	1.91	53.31	86.74
CCI	333.33	−270.50	22.84	96.39
BSE-Sensex				
SMA	20647.77	2957.11	12602.94	3263.12
EMA	20662.52	2964.00	12603.00	3280.12
MOM	2362.24	−4139.84	59.22	17.58
STCK%	100.00	1.10	60.04	75.10
STCD%	97.79	5.17	60.02	56.60
MACD	921.17	−1146.29	45.05	−33.42
RSI	100.00	1.07	56.48	45.96
WILLR%	0.00	−98.90	−39.96	−24.90
A/D Osc	100.00	1.78	50.79	94.47
CCI	333.33	−247.49	23.09	97.32
Infosys				
SMA	3432.13	337.98	1783.73	543.61
EMA	3425.35	341.62	1783.74	550.85
MOM	340.10	−493.90	6.49	16.70
STCK%	100.00	0.67	55.40	92.65
STCD%	96.57	3.13	55.39	75.88
MACD	108.04	−145.99	5.05	−11.59
RSI	98.13	1.27	53.55	57.78
WILLR%	0.00	−99.33	−44.60	−7.35
A/D Osc	100.00	1.41	50.24	86.37
CCI	330.44	−314.91	16.65	140.39
Reliance				
SMA	3073.36	265.02	1102.55	278.16
EMA	3065.97	265.95	1102.55	281.48
MOM	483.90	−1122.20	2.03	2.40
STCK%	99.30	0.89	53.14	46.57
STCD%	98.02	2.88	53.13	41.97
MACD	162.91	−276.50	1.50	−4.54
RSI	100.00	4.31	53.22	42.58
WILLR%	−0.70	−99.11	−46.86	−53.43
A/D Osc	572.88	−350.48	46.36	44.00
CCI	333.33	−333.33	12.81	72.41

Layer". Each technical indicator has its own inherent property through which traders generally predict the stock' s up or down movement. The job of this new layer is to convert this continuous values to '+1' or '−1' by considering this property during the discretization process. This way, the input data to each of the predictor models is converted to '+1' and '−1', where '+1' indicates up movement and '−1' shows down movement. Details about how the opinion of each of the technical indicators is derived is mentioned below.

First two technical indicators are moving averages. The moving average (MA) is simple technical analyses tool that smoothies out price data by creating a constantly updated average price. In this paper, 10 days' Simple Moving Average (SMA) and Weighted Moving Average (WMA) are used as we are predicting short term future. As a general guideline, if the price is above the moving average then the trend is up. If the price is below a moving average the trend is down <http://www.investopedia.com>, <http://www.stockcharts.com>. So, according to these, we have derived the opinion of both SMA and WMA indicators for each day from the value of SMA and WMA against the current price. If current price is above the moving average values then the trend is 'up' and represented as '+1', and if current price is below the moving average values then the trend is 'down' and represented as '−1'.

STCK%, STCD% and Williams R% are stochastic oscillators. These oscillators are clear trend indicators for any stock. When stochastic oscillators are increasing, the stock prices are likely to go up and vice-a-versa. This implies that if the value of stochastic oscillators at time 't' is greater than the value at time '$t-1$' then the opinion of trend is 'up' and represented as '+1' and vice-a-versa.

MACD follows the trend of the stock, i.e. if MACD goes up then stock price also goes up and vice-a-versa. So, if the value of MACD at time 't' is greater than the value at time '$t-1$', opinion on trend is 'up' and represented as '+1' and if the value of MACD at time 't' is less than value at time '$t-1$', opinion on trend is 'down' and represented as '−1'.

RSI is generally used for identifying the overbought and oversold points <http://www.stockcharts.com>. It ranges between 0 and 100. If the value of RSI exceeds 70 level, it means that the stock is overbought, so, it may go down in near future (indicating opinion '−1') and if the value of RSI goes below 30 level, it means that the stock is oversold, so, it may go up in near future (indicating opinion '+1'). For the values between (30, 70), if RSI at time 't' is greater than RSI at time '$t-1$', the opinion on trend is represented as '+1' and vice-a-versa.

CCI measures the difference between stock's price change and its average price change. High positive readings indicate that prices

are well above their average, which is a show of strength. Low negative readings indicate that prices are well below their average, which is a show of weakness. CCI is also used for identifying overbought and oversold levels. In this paper we have set 200 as overbought level and −200 as oversold level as 200 is more representative of a true extreme <http://www.stockcharts.com>. This means that if CCI value exceeds 200 level, the opinion for the trend is '−1' and if it is below −200 level then the opinion for the trend is '+1'. For the values between (−200, 200), if CCI at time 't' is greater than CCI at time '$t-1$', the opinion on the trend is '+1' and vice-a-versa.

A/D oscillator also follows the stock trend meaning that if its value at time 't' is greater than that at time '$t-1$', the opinion on trend is '+1' and vice-a-versa.

Momentum measures the rate of rise and fall of stock prices. Positive value of momentum indicates up trend and is represented by '+1' while negative value indicates down trend and is represented as '−1'.

In nutshell, trend deterministic data is prepared by exploiting the fact that each of the technical indicators has its own inherent opinion about the stock price movement. When we give these data as inputs to the model as opposed to their actual continuous value, we are already inputting trend information as perceived by each of the individual technical indicators. This is a step forward in a sense that now prediction models have to determine correlation between the input trends and the output trend.

Using these indicator values, the trend deterministic input set is prepared and given to the predictor models. Performance of all the models under study is evaluated also for this representation of inputs.

3. Prediction models

3.1. ANN model

Inspired by functioning of biological neural networks, Artificial Neural Networks are a dense network of inter-connected neurons which get activated based on inputs. A three layer feed-forward neural network is employed in our study. Inputs for the network are ten technical indicators which are represented by ten neurons in the input layer. Output layer has a single neuron with log sigmoid as the transfer function. This results in a continuous value output between 0 and 1. A threshold of 0.5 is used to determine the up or down movement prediction. For the output value greater than or equal to 0.5, prediction is considered to be the up movement else the down movement. Each of the hidden layer's neurons employed tan sigmoid as the transfer function. The architecture of the three-layered feed-forward ANN is illustrated in Fig. 1. Gradient descent with momentum is used to adjust the weights, in which at each epochs, weights are adjusted so that a global minimum can be reached. We have performed comprehensive parameter setting experiments to determine parameters for each stock and index. The ANN model parameters are number of hidden layer neurons (n), value of learning rate (lr), momentum constant (mc) and number of epochs (ep). To determine them efficiently, ten levels of n, nine levels of mc and ten levels of ep are tested in the parameter setting experiments. Initially, value of lr is fixed to 0.1. These parameters and their levels which are tested are summarized in Table 6. These settings of parameters yield a total of $10 \times 10 \times 9 = 900$ treatments for ANN for one stock. Considering two indices and two stocks, total of 3600 treatments for ANN are carried out. The top three parameter combinations that resulted in the best average of training and holdout performances are selected as the top three ANN models for comparison experiments on comparison data set. For these top performing models learning rate lr is varied in the interval of [0.1, 0.9].

3.2. SVM model

Support vector machine (SVM) were first introduced by Vapnik (1999). There are two main categories for support vector machines: support vector classification (SVC) and support vector regression (SVR). SVM is a learning system using a high dimensional feature space. Khemchandani and Chandra (2009) stated that in SVM, points are classified by means of assigning them to one of two disjoint half spaces, either in the pattern space or in a higher-dimensional feature space.

The main objective of support vector machine is to identify maximum margin hyper plane. The idea is that the margin of separation between positive and negative examples is maximized (Xu, Zhou, & Wang, 2009).

It finds maximum margin hyper plane as the final decision boundary. Assume that $x_i \in R^d,\ i = 1, 2, \ldots, N$ forms a set of input vectors with corresponding class labels $y_i \in \{+1, -1\},\ i = 1, 2, \ldots, N$. SVM can map the input vectors $x_i \in R^d$ into a high dimensional feature space $\Phi(x_i) \in H$. A kernel function $K(x_i, x_j)$ performs the mapping $\phi(\cdot)$. The resulting decision boundary is defined in Eq. (1).

$$f(x) = sgn\left(\sum_{i=1}^{N} y_i \alpha_i \cdot K(x, x_i) + b\right) \tag{1}$$

Quadratic programming problem shown in Eqs. (2)–(4) is solved to get the values of α_i.

$$Maximize \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j \cdot y_i y_j \cdot K(x_i, x_j) \tag{2}$$

$$Subject\ to\ 0 \leqslant \alpha_i \leqslant c \tag{3}$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0, \quad i = 1, 2, \ldots, N \tag{4}$$

The trade-off between margin and misclassification error is controlled by the regularization parameter c. The polynomial and radial basis kernel functions are used by us and they are shown in Eqs. (5) and (6) respectively.

$$Polynomial\ Function: \ K(x_i, x_j) = (x_i \cdot x_j + 1)^d \tag{5}$$

$$Radial\ Basis\ Function: \ K(x_i, x_j) = exp\left(-\gamma ||x_i - x_j||^2\right) \tag{6}$$

where d is the degree of polynomial function and γ is the constant of radial basis function.

Choice of kernel function, degree of kernel function (d) in case of polynomial kernel, gamma in kernel function (γ) in case of radial basis kernel and regularization constant c are the parameters of SVM. To determine them efficiently, four levels on d, ten levels of γ and 4 to 5 levels of c are tested in the parameter setting experiments. These parameters and their levels which are tested are summarized in Table 7. For one stock, these settings of parameters yield a total of 20 and 40 treatments for SVM employing polynomial and radial basis kernel functions respectively. Considering two indices and two stocks, total of 240 treatments for SVM are carried out. One parameter combination for each of the polynomial kernel SVM and radial basis kernel SVM that resulted in the best average of training and holdout performances is selected as the top two SVM models for comparison experiments.

3.3. Random forest

Decision tree learning is one of the most popular techniques for classification. Its classification accuracy is comparable with other classification methods, and it is very efficient. The classification

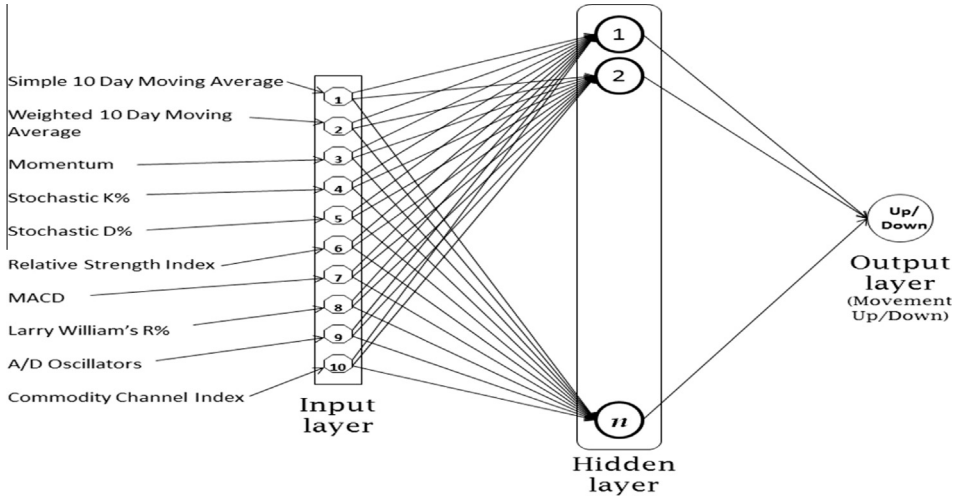

Fig. 1. Architecture of ANN model Kara et al. (2011).

model learnt through these techniques is represented as a tree and called as a decision tree. ID3 Quinlan (1986), C4.5 Quinlan (1993) and CART Breiman, Friedman, Stone, and Olshen (1984) are decision tree learning algorithms. Details can be found in Han, Kamber, and Pei (2006).

Random forest belongs to the category of ensemble learning algorithms. It uses decision tree as the base learner of the ensemble. The idea of ensemble learning is that a single classifier is not sufficient for determining class of test data. Reason being, based on sample data, classifier is not able to distinguish between noise and pattern. So it performs sampling with replacement such that given n trees to be learnt are based on these data set samples. Also in our experiments, each tree is learnt using 3 features selected randomly. After creation of n trees, when testing data is used, the decision which majority of trees come up with is considered as the final output. This also avoids problem of over-fitting. Our implementation of random forest algorithm is summarized in the Algorithm 1.

Algorithm 1. Our implementation of random forest

Input: training set D, number of trees in the ensemble k
Output: a composite model $M*$
1: **for** $i = 1$ to k **do**
2: Create bootstrap sample D_i by sampling D with replacement.
3: Select 3 features randomly.
4: Use D_i and randomly selected three features to derive tree M_i.
5: **end for**
6: return $M*$.

Number of trees in the ensemble *ntrees* is considered as the parameter of random forest. To determine it efficiently, it is varied from 10 to 200 with increment of 10 each time during the parameter setting experiments. For one stock, these settings of parameter yield a total of 20 treatments. Considering two indices and two stocks, total of 80 treatments are carried out. The top three parameter values that resulted in the best average of training and holdout performances are selected as the top three random forest models for the comparison experiments.

3.4. Naive-Bayes classifier

Naive-Bayes classifier assumes class conditional independence. Given test data Bayesian classifier predicts the probability of data belonging to a particular class. To predict probability it uses concept of Bayes' theorem. Bayes' theorem is useful in that it provides a way of calculating the posterior probability, $P(C|X)$, from $P(C), P(X|C)$, and $P(X)$. Bayes' theorem states that

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)} \tag{7}$$

Here $P(C|X)$ is the posterior probability which tells us the probability of hypothesis C being true given that event X has occurred. In our case hypothesis C is the probability of belonging to class *Up*/*Down* and event X is our test data. $P(X|C)$ is a conditional probability of occurrence of event X given hypothesis C is true. It can be estimated from the training data. The working of naive Bayesian classifier, or simple Bayesian classifier, is summarized as follows.

Assume that, m classes $C_1, C_2, \ldots, C_m$ and event of occurrence of test data, X, is given. Bayesian classifier classifies the test data into a class with highest probability. By Bayes' theorem (Eq. (7)),

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)} \tag{8}$$

Given data sets with many attributes $(A_1, A_2, \ldots, A_n)$, it would be extremely computationally expensive to compute $P(X|C_i)$. In order to reduce computation in evaluating $P(X|C_i)$, the naive assumption

Table 6
ANN parameters and their levels tested in parameter setting.

Parameters	Level(s)
Number of hidden layer neurons (n)	10,20,. . .,100
Epochs (ep)	1000, 2000, . . ., 10,000
Momentum constant (mc)	0.1, 0.2, . . ., 0.9
Learning rate (lr)	0.1

Table 7
SVM parameters and their levels tested in parameter setting.

Parameters	Levels (polynomial)	Levels (radial basis)
Degree of kernel function (d)	1, 2, 3, 4	–
Gamma in kernel function (γ)	–	0.5, 1.0, 1.5, . . . , 5.0, 10.0
Regularization parameter (c)	0.5, 1, 5, 10, 100	0.5, 1, 5, 10

of class conditional independence is made. This presumes that the values of the attributes are conditionally independent of one another, given the class label of the tuple (i.e. that there are no dependence relationships among the attributes). Therefore,

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times \ldots \times P(x_n|C_i) \quad (9)$$

Here x_k denotes to the value of attribute A_k for tuple X. Computation of $P(x_k|C_i)$ depends on whether it is categorical or continuous. If A_k is categorical, then $P(x_k|C_i)$ is the number of observations of class C_i in training set having the value x_k for A_k, divided by the number of observations of class C_i in the training set. If A_k is continuous-valued, then Gaussian distribution is fitted to the data and the value of $P(x_k|C_i)$ is calculated based on Eq. (10).

$$f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$
$$\text{so that,} \quad (10)$$
$$P(x_k|C_i) = f(x_k, \mu_{C_i}, \sigma_{C_i})$$

Here μ_{C_i} and σ_{C_i} are the mean (i.e., average) and standard deviation, respectively, of the values of attribute A_k for training tuples of class C_i. These two quantities are then plugged into Eq. (10) together with x_k, in order to estimate $P(x_k|C_i)$. $P(X|C_i)P(C_i)$ is evaluated for each class C_i in order to predict the class label of X. The class label of observation X is predicted as class C_i, if and only if

$$P(X|C_i)P(C_i) > P(X|C_j)P(C_j) \quad \text{for } 1 \leqslant j \leqslant m; \quad j \neq i. \quad (11)$$

Bayesian classifiers also serve as a theoretical justification for other classifiers that do not explicitly use Bayes' theorem. For example, under specific assumptions, it can be demonstrated that many neural networks and curve-fitting algorithms output the maximum posteriori hypothesis, as does the naive Bayesian classifier.

4. Experimental results

Accuracy and f-measure are used to evaluate the performance of proposed models. Computation of these evaluation measures requires estimating Precision and Recall which are evaluated from True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN). These parameters are defined in Eqs. (12)–(15).

$$Precision_{positive} = \frac{TP}{TP + FP} \quad (12)$$

$$Precision_{negative} = \frac{TN}{TN + FN} \quad (13)$$

$$Recall_{positive} = \frac{TP}{TP + FN} \quad (14)$$

$$Recall_{negative} = \frac{TN}{TN + FP} \quad (15)$$

Precision is the weighted average of precision positive and negative while Recall is the weighted average of recall positive and negative. Accuracy and F-measure are estimated using Eqs. (16) and (17) respectively.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (16)$$

$$F\text{-}measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (17)$$

First phase of experimentation considers input as the continuous valued data. The best parameter combinations are identified by means of experiments on parameter setting data set for each of the prediction models. These parameter combinations with corresponding accuracy and f-measure during parameter setting experiments are reported in Tables 8–10. It is to be noted that there are no parameters to be tuned for naive-Bayes classifier.

Table 8
Best three parameter combinations of ann model and their performance on continuous-valued parameter setting data set.

epochs : *neurons* : *mc* & *lr* = 0.1			
Nifty			
	10,000:20:0.6	7000:10:0.7	7000:10:0.9
Accuracy	0.8434	0.8450	0.8558
F-measure	0.8614	0.8606	0.8686
BSE-Sensex			
	1000:80:0.1	2000:40:0.2	10,000:100:0.1
Accuracy	0.7968	0.7827	0.7723
F-measure	0.7743	0.7982	0.7862
Infosys			
	1000:70:0.7	8000:150:0.7	3000:10:0.3
Accuracy	0.7417	0.7023	0.6949
F-measure	0.7581	0.7098	0.7412
Reliance			
	8000:50:0.6	6000:40:0.4	9000:20:0.5
Accuracy	0.6356	0.6326	0.6898
F-measure	0.6505	0.6116	0.7067

The purpose of experiments on comparison data set is to compare the prediction performance of these models for best parameter combinations reported during parameter setting experiments. During this comparison experiment, each of the prediction models is learnt based on best parameters reported by parameter setting experiments. Table 11 reports average accuracy and f-measure of each of the models during comparison experiment. Average accuracy and f-measure reported are averaged over the top performing models. It can be seen that naive-Bayes with Gaussian process is the least accurate while random forest is the most accurate with average accuracy of nearly 84%. Fig. 2 depicts the prediction process when data is continuous-valued.

Second phase of experimentation is identical to the first one except that the input to the models is trend deterministic data. The idea is depicted in Fig. 3. Tables 12–14 show result of best performing combinations for ANN, SVM and random forest respectively during parameter setting experiments. It is to be noted that when data is represented as trend deterministic data, naive-Bayes classifier is learnt by fitting multivariate Bernoulli distribution to the data. Results on comparison data set for all the proposed models is reported in Table 15. Final comparison shows that all the models perform well with discrete data input but SVM, random forest and naive-Bayes perform better than ANN. The accuracy of SVM, random forest and naive-Bayes is nearly 90%.

Table 9
Best two parameter combinations (one for each type of kernel) of SVM model and their performance on continuous-valued parameter setting data set.

	Kernel:Polynomial	Kernel:RBF
Nifty		
	c:100,degree:1	c:0.5,gamma:5
Accuracy	0.8427	0.8057
F-measure	0.8600	0.8275
BSE-Sensex		
	c:100,degree:1	c:1,gamma:5
Accuracy	0.8136	0.7823
F-measure	0.8321	0.8015
Infosys		
	c:0.5,degree:1	c:0.5,gamma:5
Accuracy	0.8139	0.7836
F-measure	0.8255	0.7983
Reliance		
	c:0.5,degree:1	c:1,gamma:5
Accuracy	0.7669	0.6881
F-measure	0.7761	0.7023

Table 10
Best three parameter combinations of random forest model and their performance on continuous-valued parameter setting data set.

	ntrees		
Nifty			
	140	20	30
Accuracy	0.9148	0.9146	0.9099
F-measure	0.9186	0.9185	0.9162
BSE-Sensex			
	80	50	70
Accuracy	0.8819	0.8719	0.8786
F-measure	0.8838	0.8742	0.8802
Infosys			
	50	110	200
Accuracy	0.8138	0.8059	0.8132
F-measure	0.8202	0.8135	0.8190
Reliance			
	160	60	150
Accuracy	0.7368	0.7441	0.7450
F-measure	0.7389	0.7474	0.7478

Table 11
Performance of prediction models on continuous-valued comparison data set.

Stock/Index	Prediction Models			
	ANN Kara et al. (2011)		SVM	
	Accuracy	F-measure	Accuracy	F-measure
S&P BSE SENSEX	0.7839	0.7849	0.7979	0.8168
NIFTY 50	0.8481	0.8635	0.8242	0.8438
Reliance Industries	0.6527	0.6786	0.7275	0.7392
Infosys Ltd.	0.7130	0.7364	0.7988	0.8119
Average	0.7494	0.7659	0.7871	0.8029
	Random forest		Naive-Bayes (Gaussian)	
	Accuracy	F-measure	Accuracy	F-measure
S&P BSE SENSEX	0.8775	0.8794	0.7354	0.7547
NIFTY 50	0.9131	0.9178	0.8097	0.8193
Reliance Industries	0.7420	0.7447	0.6565	0.6658
Infosys Ltd.	0.8110	0.8176	0.7307	0.7446
Average	0.8359	0.8399	0.7331	0.7461

5. Discussions

Stock market data is an example of non-stationary data. At particular time there can be trends, cycles, random walks or combinations of the three. It is desired that if a particular year is part of a cycle say a bullish one then our model should follow this pattern for trend prediction. Same can be considered for a trending year. However, usually stock values of a particular year are not isolated and there are days with random walks. Stock values are also affected by external factors creating trends and state of the country' s economy. Political scenarios are also the influencing factors which may result in cycles.

It can be seen from the results that all the models perform well when they are learnt from continuous-valued inputs but the performance of each of the models is further improved when they are learnt using trend deterministic data. The reason behind the improved performance is justified in the remainder of this section. Trend deterministic data is prepared by discretizing the continuous-valued data. The idea is based on the fact that each continuous-valued parameters when compared with its previous day's value indicates the future up or down trend. The data is discretized based on these heuristics. When this data is given as the input to the model, we are already inputting the trend based on each input parameters. It is actually the situation where each of the input parameters signify about the probable future trend and we have the actual future trend to identify the transformation from probable trends to the correct trend. This is a step forward from converting our dataset from a non-stationary dataset to trend deterministic data-set. Now our models have to determine co-relation between the input trends and output trend. Though it is non-linear, it is easier to create a model which can transform input trends to the output trend.

When we give continuous-valued technical indicators as an input to the models, we are depriving the models, the inherent trend that each technical indicator shows. This causes prediction models to classify based on values of these technical indicators but the information from the transition of values of stocks is lost and not utilized by the prediction models. We also argue that continuous-valued data is more suitable when one wants to predict the future price of the stock but in this paper, as the objective is to predict the direction of movement or trend, trend deterministic data is more suitable.

Also for any stock or indices there are scenarios when they are trading at some values say 200, then due to some external factors, they may start trading at higher price say 400 and then stabilize at that higher value. If our model is given direct continuous-valued input, then it is possible that it tries to establish relations between the values in 200 and that in 400 which is not required as far as

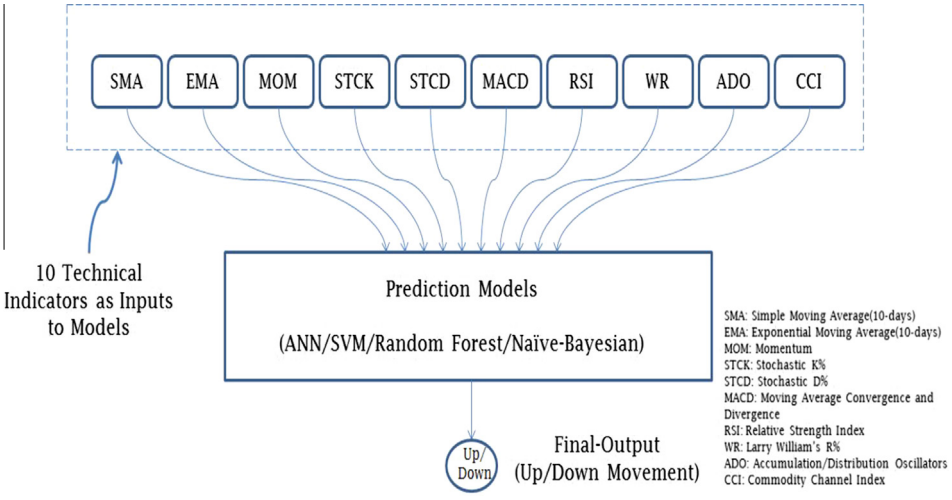

Fig. 2. Predicting with continuous-valued data.

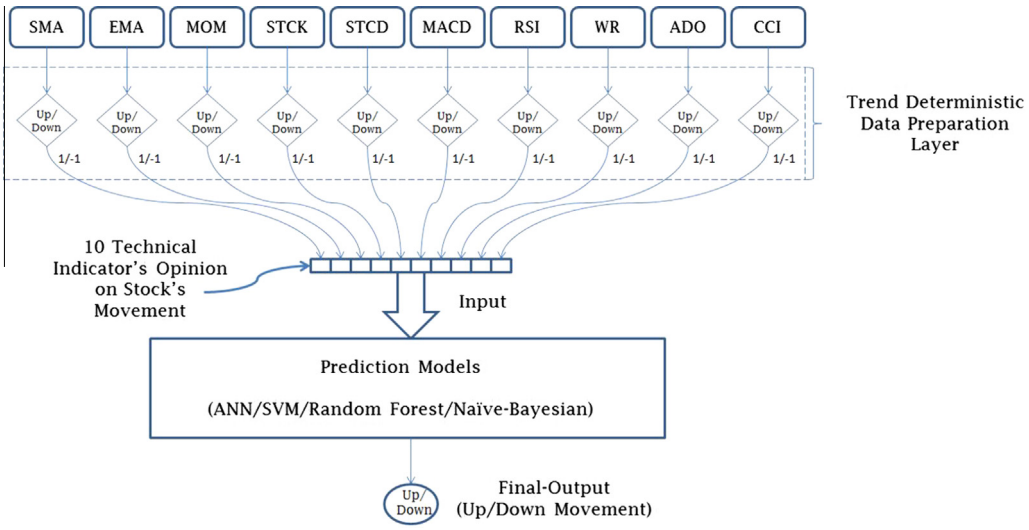

Fig. 3. Predicting with trend deterministic data.

Table 12
Best three parameter combinations of ann model and their performance on discrete-valued parameter setting data set.

Nifty			
	epochs : *neurons* : *mc* & *lr* = 0.2		
	4000:50:0.8	1000:100:0.6	3000:70:0.3
Accuracy	0.8703	0.8740	0.8729
F-measure	0.8740	0.8768	0.8801
BSE-Sensex			
	epochs : *neurons* : *mc* & *lr* = 0.1		
	6000:100:0.4	2000:30:0.3	4000:90:0.1
Accuracy	0.8563	0.8728	0.8717
F-measure	0.8632	0.8771	0.8759
Infosys			
	epochs : *neurons* : *mc* & *lr* = 0.1		
	6000:50:0.1	4000:70:0.2	9000:80:0.4
Accuracy	0.8531	0.8717	0.8468
F-measure	0.8600	0.8742	0.8503
Reliance			
	epochs : *neurons* : *mc* & *lr* = 0.2		
	1000:100:0.1	4000:90:0.9	8000:100:0.5
Accuracy	0.8573	0.8747	0.8808
F-measure	0.8620	0.8799	0.8826

Table 13
Best two parameter combinations (one for each type of kernel) of svm model and their performance on discrete-valued parameter setting data set.

	Kernel:Polynomial	Kernel:RBF
Nifty		
	c:1,degree:1	c:1,gamma:4
Accuracy	0.9010	0.8808
F-measure	0.9033	0.8838
BSE-Sensex		
	c:1,degree:1	c:5,gamma:1.5
Accuracy	0.8959	0.8780
F-measure	0.8980	0.8810
Infosys		
	c:0.5,degree:1	cc:1,gamma:3
Accuracy	0.8895	0.8865
F-measure	0.8916	0.8880
Reliance		
	c:1,degree:1	c:0.5,gamma:4
Accuracy	0.9221	0.8923
F-measure	0.9229	0.8932

Table 14
Best three parameter combinations of random forest model and their performance on discrete-valued parameter setting data set.

	ntrees		
Nifty			
	30	120	20
Accuracy	0.8913	0.8973	0.8969
F-measure	0.8934	0.8990	0.9005
BSE-Sensex			
	20	90	110
Accuracy	0.8886	0.8981	0.9011
F-measure	0.8914	0.9012	0.9028
Infosys			
	50	60	70
Accuracy	0.9035	0.8964	0.9004
F-measure	0.9051	0.8980	0.9019
Reliance			
	30	10	40
Accuracy	0.9079	0.9088	0.9070
F-measure	0.9085	0.9098	0.9078

Table 15
Performance of prediction models on discrete-valued comparison data set.

Stock/Index	Prediction Models			
	ANN		SVM	
	Accuracy	F-measure	Accuracy	F-measure
S&P BSE SENSEX	0.8669	0.8721	0.8869	0.8895
NIFTY 50	0.8724	0.8770	0.8909	0.8935
Reliance Industries	0.8709	0.8748	0.9072	0.9080
Infosys Ltd.	0.8572	0.8615	0.8880	0.8898
Average	0.8669	0.8714	0.8933	0.8952
	Random forest		Naive-Bayes	
	Accuracy	F-measure	Accuracy	F-measure
S&P BSE SENSEX	0.8959	0.8985	0.8984	0.9026
NIFTY 50	0.8952	0.8977	0.8952	0.8990
Reliance	0.9079	0.9087	0.9222	0.9234
Infosys	0.9001	0.9017	0.8919	0.8950
Average	0.8998	0.9017	0.9019	0.9050

predicting future trend is considered. Each parameters while signifying future trend is relative. It means that the important thing is how its value has changed with respect to previous days rather

than the absolute value of change. Therefore, our trend deterministic data which are discrete in nature are basically the statistical indication of whether the shares are over-bought or over-sold and are value independent. Thereby, these input parameters, when represented as probable future trends serve as a better measure of stocks condition rather than the scenario when they are represented as continuous values.

6. Conclusions

The task focused in this paper is to predict direction of movement for stocks and stock price indices. Prediction performance of four models namely ANN, SVM, random forest and naive-Bayes is compared based on ten years (2003–2012) of historical data of CNX Nifty, S&P BSE Sensex, Infosys Ltd. and Reliance Industries from Indian stock markets. Ten technical parameters reflecting the condition of stock and stock price index are used to learn each of these models. A Trend Deterministic Data Preparation Layer is employed to convert each of the technical indicator's continuous value to $+1$ or -1 indicating probable future up or down movement respectively.

Experiments with continuous-valued data show that naive-Bayes (Gaussian process) model exhibits least performance with 73.3% accuracy and random forest with highest performance of 83.56% accuracy. Performance of all these models is improved significantly when they are learnt through trend deterministic data. ANN is slightly less accurate in terms of prediction accuracy compare to other three models which perform almost identically. The accuracy of 86.69%, 89.33%, 89.98% and 90.19% is achieved by ANN, SVM, random forest and naive-Bayes (Multivariate Bernoulli Process) respectively.

Trend Deterministic Data Preparation Layer proposed in this paper exploits inherent opinion of each of the technical indicators about stock price movement. The layer exploits these opinions in the same way as the stock market's experts. In earlier researches, the technical indicators were used directly for prediction while this study first extracts trend related information from each of the technical indicators and then utilizes the same for prediction, resulting in significant improvement in accuracy. The proposal of this Trend Deterministic Data Preparation Layer is a distinct contribution to the research.

Owing to the noteworthy improvement in the prediction accuracy, the proposed system can be deployed in real time for stocks' trend prediction, making investments more profitable and secure. Improvement of accuracy with the help of this approach that is based on common investor's methods for stock investing, also promotes the idea of pre-processing the data based on the domain in which machine learning algorithms are used. This idea can be further extended not only in stock domain by incorporating other human approaches of investing but also in various other domains where expert systems and machine learning techniques are used.

Ten technical indicators are used in this paper to construct the knowledge base, however, other macro-economic variables like currency exchange rates, inflation, government policies, interest rates etc. that affect stock market can also be used as the inputs to the models or in construction of the knowledge base of an expert system. Average volume of a stock is also a potential candidate that may be useful in deciding the trend.

In this paper, at Trend Deterministic Data Preparation Layer, technical indicators' opinion about stock price movement is categorized as either 'up' or 'down'. Multiple categories like 'highly possible to go up', 'highly possible to go down', 'less possible to go up', 'less possible to go down' and 'neutral signal' are worth exploring. This may give more accurate input to inference engine of an expert system i.e. prediction algorithms in this paper.

Also, focus of this paper is short term prediction. Long term prediction can also be thought as one of the future directions which may involve analysis of stock's quarterly performance, revenue, profit returns, companies organizational stability etc. In this paper, technical indicators are derived based on the period of last 10 days (e.g. SMA, WMA, etc.). It is worth exploring the significance of the length of this period, particularly, when the objective is long term prediction.

Above all, success of proposed approach which is based on human approach of investing, encourages to emulate human approaches of decision making while developing expert systems and using machine learning algorithms for the problems in various other domains.

References

Abraham, A., Nath, B., & Mahanti, P. K. (2001). Hybrid intelligent systems for stock market analysis. In *Computational science-ICCS 2001* (pp. 337–345). Springer.

Ahmed, S. (2008). Aggregate economic variables and stock markets in india. *International Research Journal of Finance and Economics*, 141–164.

Araújo, R. d. A., & Ferreira, T. A. (2013). A morphological-rank-linear evolutionary method for stock market prediction. *Information Sciences, 237*, 3–17.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Chen, A.-S., Leung, M. T., & Daouk, H. (2003). Application of neural networks to an emerging financial market: Forecasting and trading the taiwan stock index. *Computers & Operations Research, 30*, 901–923.

Garg, A., Sriram, S., & Tai, K. (2013). Empirical analysis of model selection criteria for genetic programming in modeling of time series system. In *2013 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)* (pp. 90–94). IEEE.

Han, J., Kamber, M., & Pei, J. (2006). *Data mining: Concepts and techniques*. Morgan kaufmann.

Hassan, M. R., Nath, B., & Kirley, M. (2007). A fusion model of hmm, ann and ga for stock market forecasting. *Expert Systems with Applications, 33*, 171–180.

Hsu, S.-H., Hsieh, J., Chih, T.-C., & Hsu, K.-C. (2009). A two-stage architecture for stock price forecasting by integrating self-organizing map and support vector regression. *Expert Systems with Applications, 36*, 7947–7951.

Huang, W., Nakamori, Y., & Wang, S.-Y. (2005). Forecasting stock market movement direction with support vector machine. *Computers & Operations Research, 32*, 2513–2522.

Kara, Y., Acar Boyacioglu, M., & 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*, 5311–5319.

Khemchandani, R., Chandra, S., et al. (2009). Knowledge based proximal support vector machines. *European Journal of Operational Research, 195*, 914–923.

Kim, K.-j. (2003). Financial time series forecasting using support vector machines. *Neurocomputing, 55*, 307–319.

Liu, F., & Wang, J. (2012). Fluctuation prediction of stock market index by Legendre neural network with random time strength function. *Neurocomputing, 83*, 12–21.

Malkiel, B. G., & Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work*. *The Journal of Finance, 25*, 383–417.

Mantri, J. K., Gahan, P., & Nayak, B. (2010). Artificial neural networks-an application to stock market volatility. *International Journal of Engineering Science and Technology, 2*, 1451–1460.

Mishra, R. K., Sehgal, S., & Bhanumurthy, N. (2011). A search for long-range dependence and chaotic structure in Indian stock market. *Review of Financial Economics, 20*, 96–104.

Nair, B. B., Sai, S. G., Naveen, A., Lakshmi, A., Venkatesh, G., & Mohandas, V. (2011). A ga-artificial neural network hybrid system for financial time series forecasting. In *Information Technology and Mobile Communication* (pp. 499–506). Springer.

Ou, P., & Wang, H. (2009). Prediction of stock market index movement by ten data mining techniques. *Modern Applied Science, 3*, P28.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*, 81–106.

Quinlan, J. R. (1993). *C4. 5: Programs for machine learning* (Vol. 1). Morgan kaufmann.

Sun, J., & Li, H. (2012). Financial distress prediction using support vector machines: Ensemble vs. individual. *Applied Soft Computing, 12*, 2254–2265.

Tsai, C.-F., Lin, Y.-C., Yen, D. C., & Chen, Y.-M. (2011). Predicting stock returns by classifier ensembles. *Applied Soft Computing, 11*, 2452–2459.

Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks, 10*, 988–999.

Wang, J.-H., & Leu, J.-Y. (1996). Stock market trend prediction using arima-based neural networks. *IEEE International Conference on Neural Networks, 1996* (Vol. 4, pp. 2160–2165). IEEE.

Xu, X., Zhou, C., & Wang, Z. (2009). Credit scoring algorithm based on link analysis ranking with support vector machine. *Expert Systems with Applications, 36*, 2625–2632.