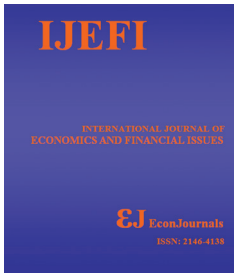


International Journal of Economics and Financial Issues

ISSN: 2146-4138

available at http: www.econjournals.com

International Journal of Economics and Financial Issues, 2019, 9(2), 189-201.

Machine Learning on Stock Price Movement Forecast: The Sample of the Taiwan Stock Exchange

Chin-Sheng Huang, Yi-Sheng Liu*

Department of Finance, National Yunlin University of Science and Technology, Taiwan. *Email: g9924810@yuntech.edu.tw

Received: 12 December 2018 **Accepted:** 27 February 2019 **DOI:** https://doi.org/10.32479/ijefi.7560

ABSTRACT

This paper addresses problem of predicting direction of movement of stock price index for Taiwan stock markets. The study compares four prediction models, artificial neural network (ANN), support vector machine, random forest and naive-Bayes with two approaches for input to these models. The first data preprocess approach involves computation of ten technical parameters using stock trading data 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 19 years of historical data from 2000 to 2018 of Taiwan stock market index. The experimental results suggest that for the first approach of input data where ten technical parameters are represented as continuous values, ANN 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 binary trend deterministic data.

Keywords: Naive-Bayes Classification, Artificial Neural Networks, Support Vector Machine, Random Forest, Machine Learning, Forecast
JEL Classifications: C11, C15, C53, G17

1. INTRODUCTION

Predicting stock price movement has long been regarded as both intriguing and challenging task in academic finance and financial industry as well. With the advance of information technology, it still has been considered as one of the most challenging applications of time series prediction. There have been plenty of empirical works devoted in sophisticated stock market data in developed markets, such as North American and European markets. However, the extant evidences of this area still lack of sufficient experiments of understanding on most developing markets, which have gained more and more attentions recently. This current research intends to fill the gap and conduct detailed study on Taiwan stock market, which has been proved typical and successful among emerging markets.

Stock market price movement prediction has to confront the strongest rejection from the academic paradigm of efficient market hypothesis states that prices of stocks are informationally efficient which means that it is impossible to predict stock prices based on the trading data (Malkiel and Fama, 1970). However, more recent results show that, 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 predictable (Patel et al., 2015). The new discovery can greatly benefit market practitioners because accurate predictions of movement of stock price indexes are very important for developing effective market trading strategies (Leung et al., 2000).

The core objective of this paper is to predict the direction of movement in the daily Taiwan stock exchange (TWSE) Composite Index using 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. The second approach focuses

on representing these technical parameters as trend deterministic data. The major contributions of this study are to demonstrate and verify the predictability of stock price index direction by four machine learning techniques, including ANN, SVM, random forest and naive-bayes.

The remainder of this paper is organized as follows. Section 2 provides a brief overview of the theoretical literature. Section 3 describes the research data. Section 4 provides the prediction models used in this study. Section 5 reports the empirical results from the comparative analysis. Finally, Section 6 contains the concluding remarks.

2. LITERATURE REVIEW

In last few years, there are more and more studies looking at the direction of movements of various kinds of financial instruments. Both academic researchers and practitioners have made tremendous efforts to predict the future movements of stock market index or its return and devise financial trading strategies to translate the forecasts into profits (Chen et al., 2003; Kara et al., 2011). In the following section, we focus the review of previous studies on ANN, SVM, random forest and naive-Bayes applied to stock market prediction.

There are many studies which concentrate on the predictability of the stock market. These literatures used various types of ANN to predict accurately the stock price return and the direction of its movement. ANN has been demonstrated to provide promising results in predict the stock price return (Avcı, 2007; Chen et al., 2003; Kara et al., 2011; Karaatli et al., 2005; Olson and Mossman, 2003; Patel et al., 2015). Chen et al. (2003) attempt to model and predict the direction of return on market index of the TWSE. The probabilistic neural network (PNN) is used to forecast the direction of index return after it is trained by historical data. Statistical performance of the PNN forecasts are measured and compared with that of the generalized methods of moments with Kalman filter. Empirical results show that the PNN-based investment strategies obtain higher returns than other investment strategies examined. Hassan et al. (2007) propose and implement a fusion model by combining the hidden markov model (HMM), ANN and genetic algorithms to forecast financial market behavior. Using ANN, the daily stock prices are transformed to independent sets of values that become input to HMM. Forecasts are obtained for a number of securities in the IT sector and are compared with a conventional forecast method. Cao et al. (2005) uses ANN to predict stock price movement (i.e., price returns) for firms traded on the Shanghai stock exchange. They compare the predictive power using linear models from financial forecasting literature to the predictive power of the univariate and multivariate neural network models. Their results show that neural networks outperform the linear models compared.

In recent years the SVM, has been successfully applied to predict stock price index and its movements. Fenghua et al. (2014), using the singular spectrum analysis (SSA), decomposes the stock price into terms of the trend, the market fluctuation, and the noise with different economic features over different time horizons, and then introduce these features into the SVM to make price predictions. The empirical evidence shows that, compared with the SVM without these price features, the combination predictive methods-the EEMD-SVM and the SSA-SVM, which combine the price features into the SVMs perform better, with the best prediction to the SSA-SVM. Hsu et al. (2009) employs a two-stage architecture for better stock price prediction. Specifically, the self-organizing map is first used to decompose the whole input space into regions where data points with similar statistical distributions are 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) is applied to forecast financial indices. The proposed technique is empirically tested using stock price series from seven major financial markets. The results show that the performance of stock price prediction can be significantly enhanced by using the two-stage architecture in comparison with a single SVR model. Kara et al. (2011) attempted to develop two efficient models and compared their performances in predicting the direction of movement in the daily Istanbul stock exchange National 100 index. The models are based on two classification techniques, ANNs and SVMs. Ten technical indicators were selected as inputs of the proposed models. Two comprehensive parameter setting experiments for both models were performed to improve their prediction performances.

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 (Patel et al., 2015). Basak et al. (2018) develop an experimental framework for the classification problem which predicts whether stock prices will increase or decrease with respect to the price prevailing n days earlier. Two algorithms, random forests, and gradient boosted decision trees facilitate this connection by using ensembles of decision trees. Empirical results show that a novelty of the current work is about the selection of technical indicators and their use as features, with high accuracy for medium to long-run prediction of stock price direction. Gupta et al. (2018) contribute to research on the predictability of stock returns in two ways. First, they use quantile random forests to study the predictive value of various consumption-based and income-based inequality measures across the quantiles of the conditional distribution of stock returns. Second, they examine whether the inequality measures, measured at a quarterly frequency, have out-of-sample predictive value for stock returns at three different forecast horizons. Their results suggest that the inequality measures have predictive value for stock returns in sample.

Khan et al. (2016) have applied machine learning classifiers that was based on SVM, Naive-Bayes and K Nearest Neighbor before and after applying principle component analysis (PCA) and

reported errors and accuracy of the algorithms before and after applying PCA. The performance of the selected algorithms has been compared using accuracy measure over the selected datasets. Chatzis et al. (2018) leverage the merits of a series of techniques including classification trees, SVMs, random forests, neural networks, extreme gradient boosting, and deep neural networks and find significant evidence of interdependence and cross-contagion effects among stock, bond and currency markets. Consequently, several algorithms have been used in stock prediction such as SVM, ANNs, linear discriminant analysis, linear regression, K-NN, and naïve Bayesian Classifier (Khan et al., 2016) to approach the subject of predictability with greater accuracy.

3. RESEARCH DATA

The data used in this paper all comes from the Taiwan database of the Taiwan economic journal (TEJ). We collect 4661 TWSE Index samples from the TEJ over the January 2000-September 2018 period. These data form our entire data set. Percentage wise increase and decrease cases of each year in the entire data set are shown in Table 1.

Some subsets were derived from the entire data set. The first subset was used to determine efficient parameter values for evaluated models. This data set is called "parameter setting data set" and used in the preliminary experiments. The parameter setting data set is consisted of approximately 20% of the entire data set and is proportional to the number of increases and decreases for each year in the entire data set. For instance, the number of cases with increasing direction in the parameter setting data for 2006 is 28 and that of decreasing direction is 22. Using this sampling method, the parameter setting data set becomes more capable of representing the entire data set. This parameter setting data set was also divided into two equal-sized training (~10% of the entire) and holdout (~10% of the entire) sets. The training data was used to determine the specifications of the models and parameters while the holdout data was reserved for out-of-sample evaluation and comparison of performances among the two prediction models. The parameter setting data set yielded a total of 938 cases. The number of cases for each year in the parameter setting data set is given in Table 2.

Once the efficient parameter values are specified, prediction performances of ANN, SVM, random forest and naive-Bayes models can be compared to each other. This performance comparison was performed on the entire data set considering the parameter values specified using the parameter setting data set. That is, the prediction models must be re-trained using a new training data set which must be a new part of the entire data set and must be larger than the training subset of parameter setting data set. After re-training, out-of-sample evaluation of models must be carried out using a new holdout data set, which is the remaining part of entire data set. Therefore, the entire data set was re-divided into the training data set (~50% of entire) and the holdout data set (~50% of entire) for comparison experiments. This was also realized by considering the dispersion of increases and decreases in the entire data set. The number of cases in the resulting comparison data sets is given in Table 3. These experimental settings are same as in Kara et al. (2011) and Patel et al. (2015).

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) and Patel et al. (2015) are used. These indicators are shown in Table 4. Table 5 shows summary statistics for the selected indicators of index. Table 6 shows correlation coefficient for the selected indicators of index.

In this study, two approaches for the representation of the input data are employed. These settings are same as in Patel et al. (2015). The first approach uses continuous value representation, i.e., the actual time series while the second one uses trend deterministic

Table 1: The number of increase and decrease cases percentage in each year in the entire data set of TWSE

Year	Increase	%	Decrease	%	Total
2000	121	45	150	55	271
2001	117	48	127	52	244
2002	108	44	140	56	248
2003	130	52	119	48	249
2004	131	52	119	48	250
2005	123	50	124	50	247
2006	137	55	111	45	248
2007	139	56	108	44	247
2008	113	45	136	55	249
2009	157	63	94	37	251
2010	136	54	115	46	251
2011	120	49	127	51	247
2012	132	53	118	47	250
2013	134	54	112	46	246
2014	136	55	112	45	248
2015	119	49	125	51	244
2016	139	57	105	43	244
2017	140	57	106	43	246
2018	97	54	84	46	181
Total	2429	52	2232	48	4661

TWSE: Taiwan stock exchange

Table 2: The number of increase and decrease cases in each year in the parameter setting data set of TWSE

Year	Training			Holdout		
	Increase	Decrease	Total	Increase	Decrease	Total
2000	12	15	27	12	15	27
2001	12	13	25	12	13	25
2002	11	14	25	11	14	25
2003	13	12	25	13	12	25
2004	13	12	25	13	12	25
2005	12	13	25	12	13	25
2006	14	11	25	14	11	25
2007	14	11	25	14	11	25
2008	11	14	25	11	14	25
2009	16	9	25	16	9	25
2010	14	11	25	14	11	25
2011	12	13	25	12	13	25
2012	13	12	25	13	12	25
2013	13	11	24	13	11	24
2014	14	11	25	14	11	25
2015	12	13	25	12	13	25
2016	14	11	25	14	11	25
2017	14	11	25	14	11	25
2018	10	8	18	10	8	18
Total	244	225	469	244	225	469

TWSE: Taiwan stock exchange

Table 3: The number of increase and decrease cases in each year in the comparison data set of TWSE

Year	Training			Holdout		
	Increase	Decrease	Total	Increase	Decrease	Total
2000	60	75	135	61	75	136
2001	58	63	121	59	64	123
2002	54	70	124	54	70	124
2003	65	59	124	65	60	125
2004	65	59	124	66	60	126
2005	61	62	123	62	62	124
2006	68	55	123	69	56	125
2007	69	54	123	70	54	124
2008	56	68	124	57	68	125
2009	78	47	125	79	47	126
2010	68	57	125	68	58	126
2011	60	63	123	60	64	124
2012	66	59	125	66	59	125
2013	67	56	123	67	56	123
2014	68	56	124	68	56	124
2015	59	62	121	60	63	123
2016	69	52	121	70	53	123
2017	70	53	123	70	53	123
2018	48	42	90	49	42	91
Total	1209	1112	2321	1220	1120	2340

TWSE: Taiwan stock exchange

representation (which is discrete in nature) for the inputs. Both the representations are discussed here.

Using the historical data, summary statistics and correlation coefficient for the selected indicators were calculated and given in Tables 5 and 6.

3.1. 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.

3.2. Discrete Representation – Trend Prediction Data

We convert continuous valued technical parameters to discrete value, representing the trend. We call this layer "Trend Deterministic Data Preparation Layer". The job of this new layer is to convert continuous values to "+1" or "−1" by considering this property during

Table 4: Selected technical indicators and their formulas

Name of indicators	Formulas
Simple 10-day moving average	$\frac{C_t + C_{t-1} + \cdots + C_{t-9}}{10}$
Weighted 10-day moving average	$\frac{((n)\times C_t + (n-1)\times C_{t-1} + \cdots + C_{t-9})}{(n + (n-1) + \cdots + 1)}$
Momentum	$C_t - C_{t-n}$
Stochastic K%	$\frac{C_t - LL_{t-n}}{HH_{t-n} - LL_{t-n}} \times 100$
Stochastic D%	$\frac{\sum_{i=0}^{n-1} K_{t-i}\%}{n}$
RSI	$100 - \frac{100}{1 + (\sum_{i=0}^{n-1} Up_{t-i}/n) / (\sum_{i=0}^{n-1} Dw_{t-i}/n)}$
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 oscillator	$\frac{H_t - C_{t-1}}{H_t - L_t}$
CCI	$\frac{M_t - SM_t}{0.015 D_t}$

Ct is the closing price, Lt is the low price, Ht is the high price at time t, DIFFt=EMA (12) t−EMA (26) t, EMA is exponential moving average, EMA (k) t=EMA (k) $_{t-1}$+α×(Ct−EMA (k) $_{t-1}$), α is a smoothing factor, $\alpha = \frac{2}{k+1}$, k is time period of k day exponential moving average, LLt and HHt mean lowest low and highest high in the last t days, respectively. $M_t = \frac{H_t + L_t + C_t}{3}$, $SM_t = \frac{(\sum_{i=1}^{n} M_{t-i+1})}{n}$, $D_t = \frac{(\sum_{i=1}^{n} |M_{t-i+1} - SM_t|)}{n}$ UPt means upward price change while DWt is the downward price change at time t, RSI: Relative strength index, MACD: Moving average convergence divergence, A/D: Accumulation/distribution, CCI: Commodity channel index

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. These settings are same as in Patel et al. (2015). Details about how the opinion of each of the technical indicators is derived is mentioned below.

Table 5: Summary statistics for the selected indicators

Indicator	Max	Min	Mean	Standard deviation
SMA10	11166.33	3539.52	7507.86	1728.88
WMA10	11171.35	3533.46	7508.86	1730.65
MOM	1318.90	−1324.37	6.06	283.32
STOCK%	100.00	0.60	53.37	27.38
STOCD%	99.92	3.80	53.37	25.30
RSI	90.35	5.91	52.46	15.79
MACD	387.64	−432.40	4.40	110.39
WILLR%	−0.00	−100.00	−44.28	32.12
A/D Osc	12364992.07	−8945569.83	−55369.08	2826748.97
CCI	370.80	−361.36	10.80	108.24

SMA10: Simple 10-day moving average, WMA10: Weighted 10-day moving average, MOM: Momentum, STOCK%: Stochastic K%, STOCD%: Stochastic D%, RSI: Relative strength index, MACD: Moving average convergence divergence, WILLR%: Larry William's R%, A/D Osc: Accumulation/distribution oscillator, CCI: Commodity channel index

The moving average (MA) is simple technical analyses tool. In this paper, 10 days' simple MA (SMA) and weighted MA (WMA) are used as we are predicting short term future. If current price is above the MA values then the trend is "up" and represented as "+1", and if current price is below the MA values then the trend is "down" and represented as "−1."

When MA convergence divergence, Stochastic K%, Stochastic D% and Williams R% are increasing, the stock prices are likely to go up and vice-a-versa. This implies that if the value 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.

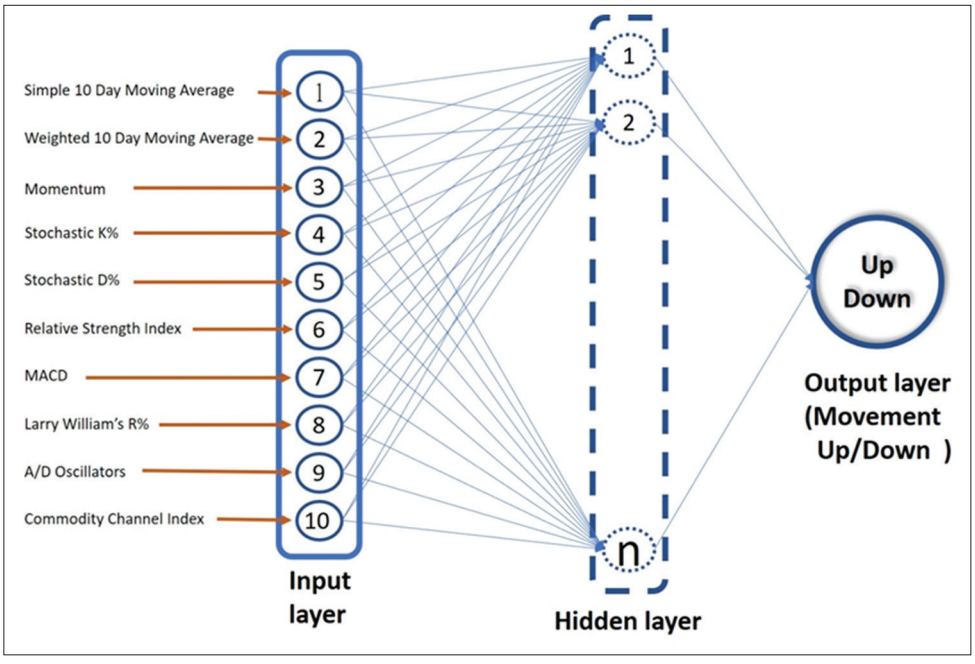
Relative strength index (RSI) ranges between 0 and 100. It is generally used for identifying the overbought and oversold points. If the value of RSI >70, it means that the stock is overbought, so, it may go down in near future (indicating opinion "−1") and if the value of RSI <30, it means that the stock 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.

Table 6: Correlation coefficient for the selected indicators

Indicator	SMA10	WMA10	MOM	STOCK%	STOCD%	RSI	MACD	WILLR%	A/D Osc	CCI
SMA10	1									
WMA10	0.9996	1								
MOM	0.0223	0.0499	1							
STOCK%	0.0397	0.0585	0.6387	1						
STOCD%	0.0539	0.0768	0.7076	0.8921	1					
RSI	0.0819	0.1043	0.8255	0.7521	0.7337	1				
MACD	0.1743	0.1909	0.5890	0.2852	0.3686	0.6903	1			
WILLR%	0.0550	0.0764	0.7998	0.7905	0.7498	0.9157	0.5014	1		
A/D Osc	0.1272	0.1417	0.5163	0.5818	0.5981	0.5564	0.2744	0.5930	1	
CCI	0.0295	0.0509	0.7880	0.8160	0.7655	0.8992	0.4717	0.9397	0.5476	1

SMA10: Simple 10-day moving average, WMA10: Weighted 10-day moving average, MOM: Momentum, STOCK%: Stochastic K%, STOCD%: Stochastic D%, RSI: Relative strength index, MACD: Moving average convergence divergence, WILLR%: Larry William's R%, A/D Osc: Accumulation/distribution oscillator, CCI: Commodity channel index

Figure 1: Architecture of artificial neural network model (Kara et al., 2011; Patel et al., 2015)



Commodity channel index (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. 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.

Accumulation/distribution 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 short, 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. 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. Prediction models must determine correlation between the input trends and the output trend. Using the trend deterministic input set is prepared and given to the predictor models. In this study, performance of all the models is evaluated also for this representation of inputs.

4. PREDICTION MODELS

4.1. ANN Model

ANN represents one widely used soft computing technique for stock market forecasting. ANN has demonstrated capability in financial modeling and prediction. A three-layered feed forward ANN model was structured to predict stock price index movement in this study. This ANN model consists of an input layer, a hidden layer and an output layer, each of which is connected to the other. Inputs for the network were ten technical indicators which were represented by ten neurons in the input layer. The architecture of the three-layered feedforward ANN is illustrated in Figure 1.

The neurons of a layer are linked to the neurons of the neighboring layers with connectivity coefficients (weights). The outputs of the model will vary between 0 and 1. If the output value is smaller than 0.5, then the corresponding case is classified as a decreasing direction; otherwise, it is classified as an increasing direction in movement. The number of neurons (n) in the hidden layer, value of learning rate (lr), momentum constant (mc) and number of iterations (ep) are ANN model parameters that must be efficiently determined. Ten levels of n, nine levels of mc and ten levels of ep were tested in the parameter setting experiments. Initially, value of lr was selected as 0.1. The parameter levels evaluated in parameter setting yield a total of 10 × 10 × 9=900 treatments for ANN. Each parameter combination was applied to the training and holdout data sets and prediction accuracy of the models were evaluated. A training performance and a holdout performance were calculated for each parameter combination. The parameter combination that resulted in the best average of training and holdout performances was selected as the best one for the corresponding model. The ANN parameters and their levels are summarized in Table 7.

4.2. SVM Model

SVMs emerged from research in statistical learning theory on how to regulate generalization and find an optimal tradeoff between structural complexity and empirical risk. SVMs classify points by assigning them to one of two disjoint half spaces, either in the pattern space or in a higher-dimensional feature space. One of the most popular SVM classifiers is the "maximum margin" one, which aims to minimize an upper bound on the generalization error through maximizing the margin between two disjoint half planes (Burges, 1998; Cortes and Vapnik, 1995). The main idea of SVM is to construct a hyperplane as the decision surface such that the margin of separation between positive and negative examples is maximized (Xu et al., 2009).

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

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

Quadratic programming problem shown in Eq. (2), (3), (4) is solved to get the values of α_i

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

$$\text{Subject to } 0 \le \alpha_i \le c \quad (3)$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0, i = 1, 2, \cdots, N \quad (4)$$

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 Equation (5), (6) respectively.

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

Table 7: ANN parameters and their levels tested in parameter setting

Parameters	Level (s)
Number of hidden layer neurons (n)	10,20,……,90,100
Epochs (ep)	1000,2000,……,9000,10000
Momentum constant (mc)	0.1,0.2,……,0.8,0.9
Learning rate (lr)	0.1

ANN: Artificial neural network

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

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 8.

4.3. 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 model learnt through these techniques is represented as a tree and called as a decision tree. Details can be found in Han et al. (2011).

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classification. 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.

Choice of criterion function to measure the quality of a split. Supported criteria are "gini" for the Gini impurity and "entropy" for the information gain. Number of trees in the ensemble ntrees and the maximum depth of the tree are considered as the parameter of random forest. To determine it efficiently, Number of trees is varied from 10 to 200 with increment of 10 each time during the parameter setting experiments. Maximum depth of the tree is varied from 2 to 10 during the parameter setting experiments. For one stock, these settings of parameter yield a total of 360 treatments. These parameters and their levels which are tested are summarized in Table 9.

4.4. Naïve-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

$$P(C \mid X) = \frac{P(X \mid C)P(C)}{P(X)} \quad (7)$$

Table 8: 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,……,4.5,5.0
Regularization parameter (c)	0.5,1,5,10,100	0.5,1,5,10,100

SVM: Support vector machine

Table 9: Random forest parameters and their levels tested in parameter setting

Parameters	Level (s)
Criterion function (ct)	Gini, entropy
Maximum depth of the tree (md)	2,3,……,9,10
Number of trees (n trees)	10,20,……,190,200

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, 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 (Equation 7),

$$P(C_i \mid X) = \frac{P(X \mid C_i)P(C_i)}{P(X)} \quad (8)$$

Given data sets with many attributes (A_1, A_2, 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 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. Therefore,

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

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 Equation (10).

$$f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \quad (10)$$

So that,

$$P(x_k|C_i) = f(x_k, \mu C_i, \sigma C_i) \quad (11)$$

Here μC_i and σC_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

$$P(X|C_i)\,P(C_i) > P(X|C_j)\,P(C_j) \text{ for } 1 \le j \le m; j \ne I \quad (12)$$

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.

Table 10: Ann model and their performance on continuous-valued parameter setting data set

Parameters	Output	Precision	Recall	F-measure	Accuracy	Selected
ep=6000, n=10, mc=0.7	−1	0.6842	0.7123	0.6980	0.7122	***
	1	0.7386	0.7120	0.7251		
	Avg/total	0.7132	0.7122	0.7124		
ep=10000, n=40, mc=0.8	−1	0.7108	0.6621	0.6856	0.7164	
	1	0.7208	0.7640	0.7417		
	Avg/total	0.7161	0.7164	0.7155		
ep=4000, n=10, mc=0.5	−1	0.6974	0.7260	0.7114	0.7249	
	1	0.7510	0.7240	0.7373		
	Avg/total	0.7260	0.7249	0.7252		

n: The number of hidden layer neurons, ep: The epochs, mc: The momentum constant, lr: Learning rate was selected as 0.1

Table 11: SVM model and their performance on continuous-valued parameter setting data set

Parameters	Output	Precision	Recall	F-measure	Accuracy	Selected
k=polynomial, c=0.5, d=1	−1	0.6439	0.6027	0.6226	0.6588	***
	1	0.6705	0.7080	0.6887		
	Avg/total	0.6581	0.6588	0.6579		
k=polynomial, c=1, d=1	−1	0.6414	0.5799	0.6091	0.6525	
	1	0.6605	0.7160	0.6871		
	Avg/total	0.6516	0.6525	0.6507		
k=polynomial, c=100, d=1	−1	0.6381	0.6119	0.6247	0.6567	
	1	0.6718	0.6960	0.6837		
	Avg/total	0.6561	0.6567	0.6562		
k=radial basis, c=5, g=0.5	−1	0.6613	0.7489	0.7024	0.7036	
	1	0.7511	0.6640	0.7049		
	Avg/total	0.7092	0.7036	0.7037		
k=radial basis, c=10, g=1.5	−1	0.6872	0.6621	0.6744	0.7015	
	1	0.7132	0.7360	0.7244		
	Avg/total	0.7010	0.7015	0.7011		
k=radial basis, c=10, g=0.5	−1	0.6585	0.7397	0.6968	0.6993	
	1	0.7444	0.6640	0.7019		
	Avg/total	0.7043	0.6994	0.6995		

k: The kernel function, c: The regularization parameter, d: The degree of kernel function, g: The gamma in kernel function, SVM: Support vector machine

Table 12: Random forest model and their performance on continuous-valued parameter setting data set

Parameters	Output	Precision	Recall	F-measure	Accuracy	Selected
ct=gini, md=4, ntrees=100	−1	0.6344	0.6575	0.6457	0.6631	***
	1	0.6901	0.6680	0.6789		
	Avg/total	0.6641	0.6631	0.6634		
ct=gini, md=4, ntrees=150	−1	0.6344	0.6575	0.6457	0.6631	
	1	0.6901	0.6680	0.6789		
	Avg/total	0.6641	0.6631	0.6634		
ct=gini, md=7, ntrees=170	−1	0.6169	0.7352	0.6708	0.6631	
	1	0.7212	0.6000	0.6550		
	Avg/total	0.6725	0.6631	0.6624		
ct=entropy, md=4, ntrees=10	−1	0.6432	0.6256	0.6343	0.6631	
	1	0.6797	0.6960	0.6877		
	Avg/total	0.6626	0.6631	0.6628		
ct=entropy, md=4, ntrees=20	−1	0.6484	0.6484	0.6484	0.6716	
	1	0.6920	0.6920	0.6920		
	Avg/total	0.6716	0.6716	0.6716		
ct=entropy, md=4, ntrees=30	−1	0.6455	0.6484	0.6469	0.6695	
	1	0.6908	0.6880	0.6894		
	Avg/total	0.6696	0.6695	0.6696		

ct: The criterion function to measure the quality of a split. Supported criteria are "gini" for the Gini impurity and "entropy" for the information gain. md: The maximum depth of the tree. Grow a tree with ntrees in best-first fashion

Table 13: Performance of prediction models on continuous-valued comparison data set

Parameters	Prediction models	Output	F-measure	Accuracy
ep=4000, n=10, mc=0.5	ANN	−1	0.6815	0.7020
		1	0.7200	
		Avg/total	0.7021	
k=radial basis, c=5, g=0.5	SVM	−1	0.6132	0.6460
		1	0.6737	
		Avg/total	0.6456	
ct=entropy, md=4, ntrees=20	Random forest	−1	0.6039	0.6342
		1	0.6601	
		Avg/total	0.6340	
None	Naïve-Bayes	−1	0.5480	0.5846
		1	0.6157	
		Avg/total	0.5842	

ct: The criterion function to measure the quality of a split, md: The maximum depth of the tree. Grow a tree with ntrees in best-first fashion, k: The kernel function, c: The regularization parameter, g: The gamma in kernel function, SVM: Support vector machine, n: The number of hidden layer neurons, ep: The epochs, mc: The momentum constant, ANN: Artificial neural network

Figure 2: Random forest classification rule map on continuous-valued comparison data set

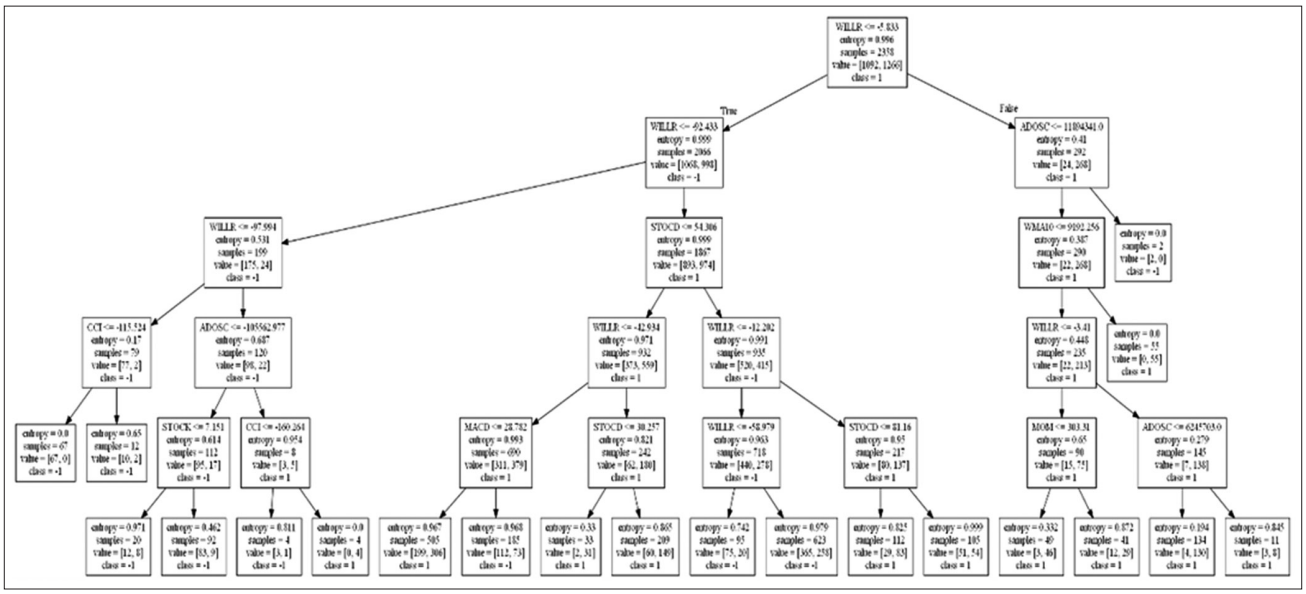
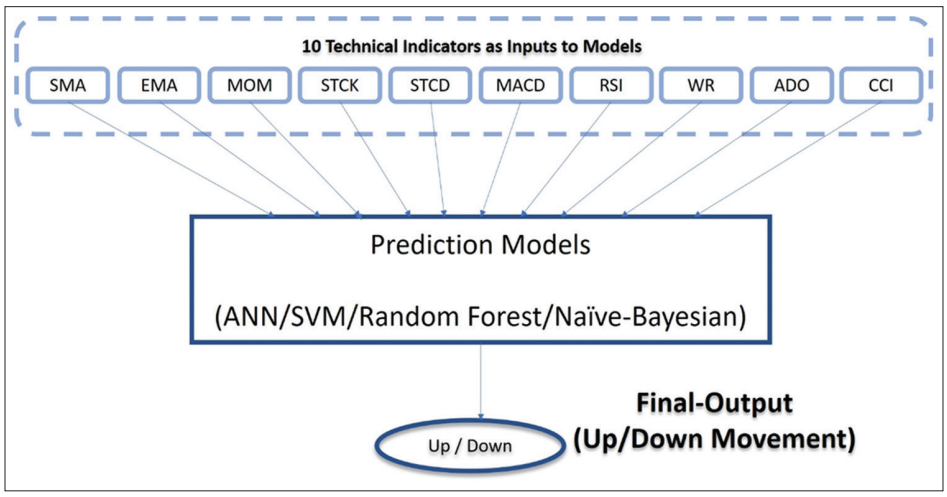


Figure 3: Predicting with continuous-valued data



5. 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 Equation (13), (14), (15), (16).

$$\text{Precision}_{\text{positive}} = \frac{TP}{TP + FP} \quad (13)$$

$$\text{Precision}_{\text{negative}} = \frac{TN}{TN + FN} \quad (14)$$

$$\text{Recall}_{\text{positive}} = \frac{TP}{TP + FN} \quad (15)$$

$$\text{Recall}_{\text{negative}} = \frac{TN}{TN + FP} \quad (16)$$

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 Equation (17), (18) respectively.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad (17)$$

$$\text{F-measure} = \frac{TP + TN}{TP + FP + TN + FN} \quad (18)$$

First phase of purpose is to compare the prediction performance of these models when data is continuous-valued. Tables 10-12 show result of best performing combinations for ANN, SVM and random forest. Table 13 reports average accuracy and f-measure of each of

Figure 4: Predicting with trend deterministic data (Patel et al., 2015)

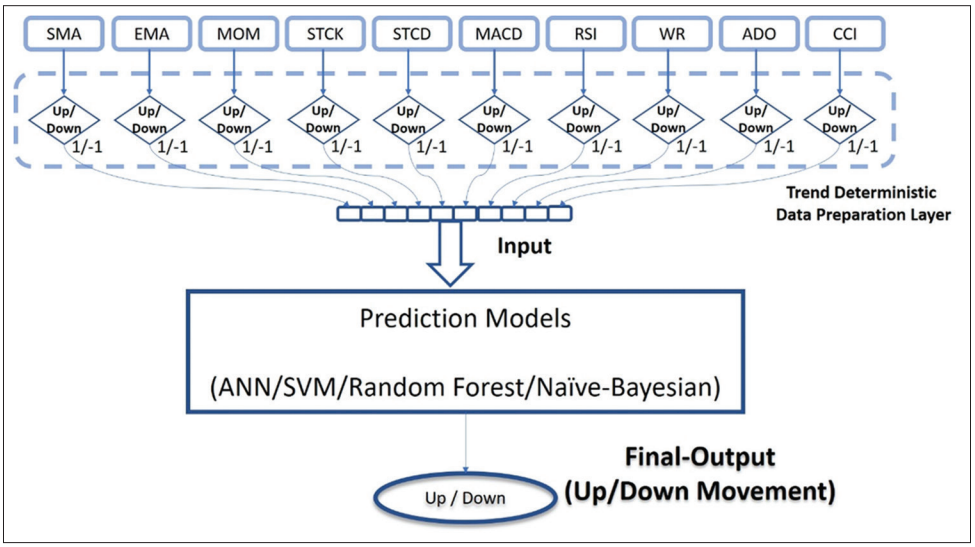
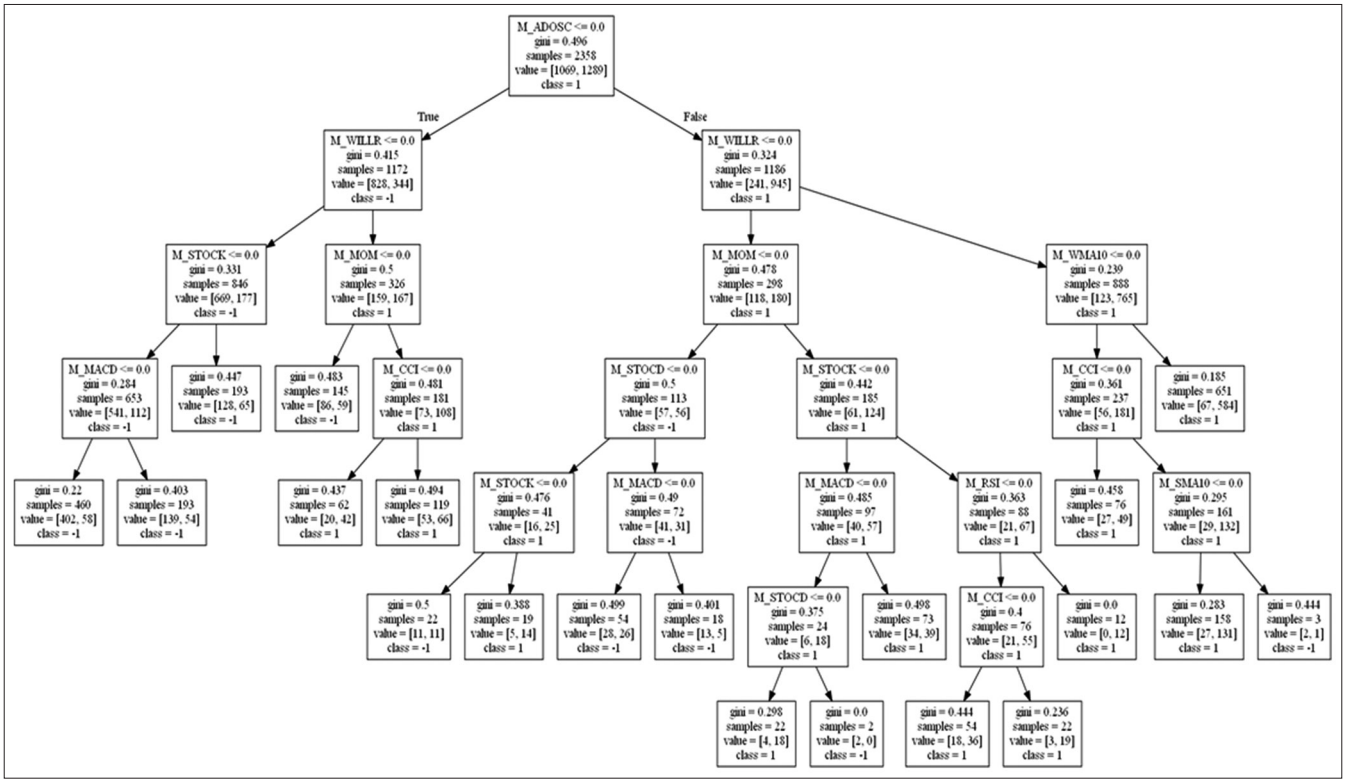


Figure 5: Random forest classification rule map on discrete-valued comparison data



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 is the least accurate while ANN is the most accurate with average accuracy 70.2%. Figure 2 is the random forest classification rule map on continuous-valued comparison data set. Figure 3 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. Figure 4

Table 14: ANN model and their performance on discrete-valued parameter setting data set

Parameters	Output	Precision	Recall	F-measure	Accuracy	Selected
ep=8000, n=10, mc=0.1	−1	0.7069	0.7421	0.7241	0.7335	***
	1	0.7595	0.7258	0.7423		
	Avg/total	0.7347	0.7335	0.7337		
ep=9000, n=10, mc=0.6	−1	0.7302	0.7104	0.7202	0.7399	
	1	0.7480	0.7661	0.7570		
	Avg/total	0.7396	0.7399	0.7396		
ep=7000, n=50, mc=0.4	−1	0.7051	0.7466	0.7253	0.7335	
	1	0.7617	0.7218	0.7412		
	Avg/total	0.7350	0.7335	0.7337		

n is the number of hidden layer neurons, ep is the epochs, mc is the momentum constant, learning rate (lr) was selected as 0.1, ANN: Artificial neural network

Table 15: SVM model and their performance on discrete-valued parameter setting data set

Parameters	Output	Precision	recall	F-measure	Accuracy	Selected
k=polynomial	−1	0.7368	0.7602	0.7483	0.7590	***
c=5, d=1	1	0.7801	0.7581	0.7689		
	Avg/total	0.7597	0.7591	0.7592		
k=polynomial,	−1	0.7409	0.7376	0.7392	0.7548	
c=0.5, d=3	1	0.7671	0.7702	0.7686		
	Avg/total	0.7547	0.7548	0.7548		
k=polynomial,	−1	0.7500	0.7466	0.7483	0.7633	
c=1, d=3	1	0.7751	0.7782	0.7767		
	Avg/total	0.7633	0.7633	0.7633		
k=radial basis, c=1, g=0.5	−1	0.7093	0.7285	0.7187	0.7313	
	1	0.7521	0.7339	0.7429		
	Avg/total	0.7319	0.7313	0.7315		
k=radial basis, c=5, g=0.5	−1	0.7074	0.7330	0.7200	0.7313	
	1	0.7542	0.7298	0.7418		
	Avg/total	0.7321	0.7313	0.7315		
k=radial basis, c=10, g=0.5	−1	0.7074	0.7330	0.7200	0.7313	
	1	0.7542	0.7298	0.7418		
	Avg/total	0.7321	0.7313	0.7315		

k is the kernel function, c is the regularization parameter, d is the degree of kernel function, g is the gamma in kernel function, SVM: Support vector machine

Table 16: Random forest model and their performance on discrete-valued parameter setting data set

Parameters	Output	Precision	Recall	F-measure	Accuracy	Selected
ct=gini, md=4, ntrees=10	−1	0.7747	0.6380	0.6998	0.7420	***
	1	0.7213	0.8347	0.7738		
	Avg/total	0.7465	0.7420	0.7389		
ct=gini, md=5, ntrees=20	−1	0.7409	0.7376	0.7392	0.7548	
	1	0.7671	0.7702	0.7686		
	Avg/total	0.7547	0.7548	0.7548		
ct=gini, md=7, ntrees=30	−1	0.7442	0.7240	0.7339	0.7527	
	1	0.7598	0.7782	0.7689		
	Avg/total	0.7525	0.7527	0.7524		
ct=entropy, md=8, ntrees=50	−1	0.7453	0.7149	0.7298	0.7505	
	1	0.7549	0.7823	0.7683		
	Avg/total	0.7503	0.7505	0.7502		
ct=entropy, md=5, ntrees=20	−1	0.7198	0.7557	0.7373	0.7463	
	1	0.7722	0.7379	0.7546		
	Avg/total	0.7475	0.7463	0.7465		
ct=entropy, md=5, ntrees=30	−1	0.7289	0.7421	0.7354	0.7484	
	1	0.7664	0.7540	0.7602		
	Avg/total	0.7487	0.7484	0.7485		

ct is the criterion function to measure the quality of a split. Supported criteria are “gini” for the Gini impurity and “entropy” for the information gain. md is the maximum depth of the tree. Grow a tree with ntrees in best-first fashion.

Table 17: Performance of prediction models on discrete-valued comparison data set

Parameters	Prediction models	Output	F-measure	Accuracy	Increase
ep=9000, n=10, mc=0.6	ANN	−1	0.7422	0.7558	0.0538
		1	0.7681		
		Avg/total	0.7558		
k=polynomial c=1, d=3	SVM	−1	0.7580	0.7626	0.1166
		1	0.7671		
		Avg/total	0.7628		
ct=gini, md=5, ntrees=20	Random forest	−1	0.7617	0.7681	0.1339
		1	0.7742		
		Avg/total	0.7683		
None	Naïve-Bayes	−1	0.7359	0.7465	0.1619
		1	0.7563		
		Avg/total	0.7466		

ct: The criterion function to measure the quality of a split, md: The maximum depth of the tree. Grow a tree with ntrees in best-first fashion, k: The kernel function, c: The regularization parameter, g: The gamma in kernel function, SVM: Support vector machine, n: The number of hidden layer neurons, ep: The epochs, mc: The momentum constant, ANN: Artificial neural network

depicts the predicting with trend deterministic data. Figure 5 is the random forest classification rule map on discrete-valued comparison data set. Tables 14-16 show result of best performing combinations for ANN, SVM and random forest. 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 17. Final comparison shows that all the models perform well with discrete data input but SVM, random forest and ANN perform better than naive-Bayes. The accuracy of SVM and random forest is nearly 77%.

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 19 years (2000-2018) of historical data of TWSE Index samples. 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 model exhibits least performance with 58.46% accuracy and ANN with highest performance of 70.2% accuracy. Experiments with discrete-valued data show that naive-Bayes model exhibits least performance with 74.65% accuracy and Random forest with highest performance of 76.81% accuracy. Performance of all these models is improved significantly when they are learnt through trend deterministic data. SVM, random forest and ANN perform better than naive-Bayes. The accuracy of SVM, random forest and ANN is nearly 77%.

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, resulting in significant improvement in accuracy. The proposal of this trend deterministic data preparation layer is a distinct contribution to the research. Improvement in the prediction accuracy makes investments more profitable and secure.

In this study, 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" and "neutral signal" are worth exploring. Also, focus of this paper is short term prediction. Technical indicators are derived based on the period of last 10 days (e.g., SMA, WMA, etc.). Long term prediction can also be thought as one of the future directions.

Above all things, encourages to emulate human approaches of decision making while using machine learning algorithms for the problems in various other domains.

REFERENCES

Avcı, E. (2007), Forecasting daily and sessional returns of the ISE-100 index with neural network models. Journal of Dogus University, 8(2), 128-142.

Basak, S., Kar, S., Saha, S., Khaidem, L., Dey, S.R. (2018), Predicting the direction of stock market prices using tree-based classifiers. The North American Journal of Economics and Finance, 47, 552-567.

Burges, C.J. (1998), A tutorial on support vector machines for pattern recognition. Data Mining Knowledge Discovery, 2(2), 121-167.

Cao, Q., Leggio, K.B., Schniederjans, M.J. (2005), A comparison between Fama and French's model and artificial neural networks in predicting the Chinese stock market. Computers and Operations Research, 32(10), 2499-2512.

Chatzis, S.P., Siakoulis, V., Petropoulos, A., Stavroulakis, E., Vlachogiannakis, N. (2018), Forecasting stock market crisis events using deep and statistical machine learning techniques. Expert Systems with Applications, 112, 353-371.

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 and Operations Research, 30(6), 901-923.

Cortes, C., Vapnik, V. (1995), Support-vector networks. Machine Learning, 20(3), 273-297.

Fenghua, W.E.N., Jihong, X., Zhifang, H.E., Xu, G. (2014), Stock price prediction based on SSA and SVM. Procedia Computer Science,

31, 625-631.
Gupta, R., Pierdzioch, C., Vivian, A.J., Wohar, M.E. (2018), The predictive value of inequality measures for stock returns: An analysis of long-span UK data using quantile random forests. Finance Research Letters, 1-8. Doi: 10.1016/j.frl.2018.08.013.
Han, J., Pei, J., Kamber, M. (2011), Data Mining: Concepts and Techniques. Amsterdam: Elsevier.
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(1), 171-180.
Hsu, S.H., Hsieh, J.J.P., 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(4), 7947-7951.
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.
Karaatli, M., Gungor, I., Demir, Y., Kalayci, S. (2005), Estimating stock market movements with neural network approach. Journal of Balikesir University, 2(1), 22-48.
Khan, W., Ghazanfar, M., Asam, M., Iqbal, A., Ahmad, S., Khan, J.A. (2016), Predicting trend in stock market exchange using machine learning classifiers. Science International, 28(2), 1363-1367.
Leung, M.T., Daouk, H., Chen, A.S. (2000), Forecasting stock indices: A comparison of classification and level estimation models. International Journal of Forecasting, 16(2), 173-190.
Malkiel, B.G., Fama, E.F. (1970), Efficient capital markets: A review of theory and empirical work. The Journal of Finance, 25(2), 383-417.
Olson, D., Mossman, C. (2003), Neural network forecasts of Canadian stock returns using accounting ratios. International Journal of Forecasting, 19(3), 453-465.
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.
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.