

Critical Transformation of Trend Deterministic Technical Indicators for Short-Term Stock Market Trading Strategy Prediction

Nitin Sakhare^{1,*}, Divya Midhun², Dharmesh Dhabliya³

¹Department of Computer Engineering, Vishwakarma Institute of Technology, Pune, India

¹Lincoln University College, Malaysia.

²Lincoln University College, Malaysia.

³Department of Information Technology, Vishwakarma Institute Technology, Pune, India.
nitinsakhare4@gmail.com, divya@lincoln.edu.my, dharmeshdhabliya@gmail.com

Abstract: A stock market is a market where company stocks or share offers are issued and exchanged through trades. Stock prices are defined by five price characteristics: open, high, low, close, and volume over a particular period. By analyzing the historical prices of the stocks, one can get better predictions of stock prices. Technical analysis of stocks is often performed using various statistical measures known as Technical indicators which can be used to predict the future price trends and patterns. A large number of machine learning techniques are available for technical analysis of the stock market with the help of indicators. This paper addresses in-depth exploration of various technical indicators which can be effectively utilized for machine learning-based technical analysis and thereby contributing to the prediction of stock prices over a short time frame. We have used a 10 years historical dataset of NIFTY-50 index of Indian stock market. This dataset is then transformed to indicator wise buy/sell dataset and which then applied on machine learning algorithms: MP-ANN, SMO, and RF. Based on indication or trading signal given by indicators, traders can formulate the strategy to earn maximum profit.

Keywords: Stock market; machine learning; technical analysis; MLP-ANN; Random Forest; SMO;

1. Introduction

A stock market is the accumulation of buyers and sellers responsible for the financial give-and-take of stocks which indicates the ownership of certain amount of shares. In India, there is considerable number of intraday traders who trade in companies listed on National Stock Exchange (NSE) and Bombay Stock Exchange (BSE). As per NSE statistics, there are 15 Million intraday traders in India. Understanding that "right" exchange of stocks implies that one understands the market situation well. Utilizing both social, financial aspects and quantitative examination, technical analysts mean to use past execution to foresee future market conduct. The two most basic types of technical analysis are chart analysis and analysis using statistics based technical indicators (Tsai & Quan 2014). Technical Analysis is a mathematical analysis of stock prices and understanding the right price of the stock at the right time. Technical analysts use various statistical-based measures, called as technical indicators to determine entry and exit points in stock markets (Li, Xie, Wang, et al. 2016). There are several technical indicators available for technical analysis. These indicators are typically classified as: Trend, Momentum, Volatility, and Volume (Patel, Shah,

Thakkar, & Kotecha 2015). Technical indicators are simply statistical measures that can be computed using open, high, low prices and volume of stocks exchanged. Technical analysts are the people who compute technical indicators by observing the past data and predict the future price movements. List of technical indicators is given in table 1. Different indicators have different significance for technical analysis (Patel & Parikh 2016). Based on various indicators and types, investors can define suitable strategies. Strategies are predefined conditions that determine entry, exit, and/or trade settlement. This research paper does not focus on any developing specific trading strategies; however, it gives direction explanation of how indicators can be used for technical analysis to predict stock market direction using machine learning techniques and thereby helping investors whether to buy or sell stocks (Patel, Shah, Thakkar, & Kotecha 2015 ; Feng, He, Wang, Luo, Liu & Chua 2019). It is commonly advised most technical analysts to not to use technical indicators alone for technical analysis (Calzon, Harold, Garzas & Herrera 2018). Technical indicators should be used in a group for analysis. Such trading strategies can be formulated using a machine learning-based technical analysis approach. This approach helps analysts/traders to analyze a massive amount of stock market data and give prediction results with more accuracy and within less time so that traders can identify profitable opportunities, eliminate risks and adjust or update investment strategy if required (Sheikh, Alam, & Agarwal, 2019). Machine learning techniques have the ability to learn automatically and improve the analysis/prediction performance with learning experiences of data/recorded observations. It is based on computer programs – learning algorithms which start the learning process with historical data, look for hidden information or patterns and help analysts to predict and make the better decisions ahead based on past data (Sakhare, & Shaik 2019). There are various machine learning algorithms available; however, we have used most widely used algorithms: Random Forest (RF), Multilayer Perceptron Artificial Neural Networks (MP-ANN) and Sequential Minimal Optimization (SMO) from research experience and as recommended by researchers and stock market analysts for our research work (Patel, Shah, Thakkar, & Kotecha 2015 ; Sezer, Ozbayoglu, & Dogdu 2017; Chen & Hao 2017; Lei & Lin 2015). Since a long time, machine learning techniques have been extensively used for prediction of stock price movements. Feng et al. (2014) proposed a stock prediction system by identifying similarities using candlestick charts. These charts were prepared using technical indicators. They didn't explore the significance of indicators. Prediction results can be significantly improved if indicators of different types were a part of the prediction process. Feng et al. (2019) proposed deep learning- neural modeling with temporal graph convolution approach. They developed a model based on deep learning for identifying stocks that would yield the highest expected profit. For better results, other learning algorithms can be studied, and their performance can be evaluated to suggest the best strategy or decision making. Li, Xie, Wang, et al. (2016) used extreme learning machine based on combination of Support Vector Machine and Back Propagation Neural Networks (SVV-BPNN) for prediction. They used two types of dataset for prediction: Historical stock prices and news related to the market. This work can be extended with fundamental analysis of the stock market for long term investments. Calzon et al. (2018) developed a group-based decision making for selecting optimized indicators. They used an expert selection-based approach to preventing market decision. This work can be extended to a combination of decision from a group of market experts with the trading signal from technical indicators to get more accurate prediction results. Sheikh, Alam, & Agarwal (2019) did time series analysis of Indian Stock Market. They developed different models based on Auto Regression, Moving Averages for prediction of stock market volatility. This work would have created more significance with the use of the combination

of technical indicators based on moving averages, volatility, oscillations, etc. Pehlivanli & Guzhan (2016) had done significant work on indicator selection. They used features selection technique with information gain for prediction of one day ahead stock price. Some indicators are simply duplicates of each other. An optimal subset of indicators is selected by discarding irrelevant and unwanted indicators. Other attribute evaluation techniques like principal component analysis (PCA), Pearson correlation coefficient, information gain, gain ratio can be used to validate an optimal subset of indicators (Waqar, Dawood, Bilal, Ali, & Guo 2017 ; Pehlivanli & Guzhan 2016). Cailee, Itani, Gueriot, & Rakotondratsimba (2017) had focused on performance comparison of machine algorithms, and for analysis, they used different combinations of indicators. This work can create more sense when combinations of indicators are based on one combination containing indicators belonging to different types. The same kind of indicators could create more ambiguity in predictions. For example, a combination of all moving average based indicators could lead to false signals. Barmish & Primbs (2016) developed a stock trading model based on classical feedback controllers. This work can be extended with statistics based technical indicators for analysis based on back-tracking. Wen, Li, Zhang & Chen (2019) developed a novel methodology to analyze noisy financial time series via sequence reconstruction using frequent patterns. Different machine learning algorithms can be experimented to generate frequent patterns. Chong, Han, and Park (2017) performed a detailed study of deep learning method to understand the impact of feature reduction techniques on prediction of stock market. For this work Chong, Han, and Park (2017) used principle component analysis, autoencoder and restricted Boltzmann machine techniques. They improved the performance of the autoregressive model with feature extractors. Feature extractors applied in a broader dimension of technical indicators can generate enough number of subsets with optimal indicators. Nayak, Mishra and Kumar (2015) developed support vector machine and k-nearest neighbors based combined approach for technical analysis of the Indian stock market. They used different kernel functions to understand its impact on prediction of profit or loss. Other machine learning algorithms can be used, and the performance of different models can be analyzed. Analysis of the prediction performance of random forest, multilayer perceptron- artificial neural network, and sequential minimal optimization has been done. This paper proposes a transformed trading decision data layer, which converts numerical values of technical indicators to trade decision calls- buy or sell. Then we have used Wisdom of Crowd-collective decision intelligence-based approach to fit prediction problem into a multiclass problem with five classes- strong buy, strong sell, buy, sell, and hold. Since this work is well suited for short term prediction- we have used a dataset of NIFTY 50 index of NSE India for a period of 10 years (July 2008-June 2019).

The remaining of the paper contains the following sections:

Section 2 describes the research data used for analysis work, pre-processing of data using a statistical approach, interpretation of 75 technical indicators to generate buy/sell calls. It also gives details on the preparation of transformed dataset. Section 3 shows experimental work with three prediction models. The results of experiments and performance analysis and comparison of machine learning models is done in section 4. Section 5 concludes the research work with future work to extend the study.

2. Research Data

We have used NIFTY 50 index dataset of National Stock Exchange: India for technical analysis using 75 technical indicators. We can use the dataset of any company listed on stock market indices like BSE and

NSE for technical analysis provided that it contains values: date, open, high, low, close prices and volume traded for the day [24]. The work done in this paper focuses on understanding the technical indicators and how they can be helpful for investors to take a decision on buying or selling of stocks. The time frame considered is ten years (July 2009- June 2019). Technical indicators discussed in this paper expect dataset to be available in Open, High, Low, Close, and Volume (OHLCV) format [35] (primary source of data) which is directly available on [24].

2.1. Implementing Technical Indicators

Almost all the indicators used historical information of stocks prices: open price, high price, low, close price, and volume. Based on these values, it is possible to calculate technical indicators, which in turn help investors to decide on stock under consideration.

Table 1. List of Indicators

Simple Moving Average	Stochastic Oscillator %D	Triple Exponential Moving Average	Double Exponential Moving Average
Simple Moving Median	Stochastic RSI	Elastic Volume Moving Average	Elastic-Volume weighted MACD
Smoothed Simple Moving Average	Williams %R	Bollinger Bands Width	Market Momentum 'MOM'
Exponential Moving Average	Ultimate Oscillator	Pivot Points	Rate-of-Change
Weighted Moving Average	Awesome Oscillator	Fibonacci Pivot Points	Relative Strength Index RSI
Hull Moving Average	Mass Index	Volume Flow Indicator ‘	Inverse Fisher Transform
Triangular Moving Average	Vortex Indicator	Average Directional Index	True Range
Triple Exponential Moving Average Oscillator	Know Sure Thing	Volume Weighted Average Price	Average True Range
Volume Adjusted Moving Average	True Strength Index	Smoothed Moving Average	Stop-and-Reverse
Kaufman Efficiency Indicator	Typical Price	Convergence Divergence	Money Flow Index
Kaufman's Adaptive Moving Average	Accumulation-	Percentage Price	On Balance Volume (OBV)
Zero Lag Exponential Moving Average	Chaikin Oscillator	Volume-Weighted	Weighted OBV
Wave Trend Oscillator	Volume Price Trend	Bull power and Bear Power	Buy and Sell Pressure (BASP)
Moving Standard Deviation	Volume Zone Oscillator	Ease of Movement	Normalized BASP
Ichimoku Cloud	Price Zone Oscillator	Commodity Channel Index 'CCI'	Chande Momentum Oscillator
Vector Size Indicator	Elder's Force Index	Coppock Curve	Chandelier Exit
Squeeze Momentum Indicator	Cumulative Force Index	Twiggs Money Index	Qstick
Directional Movement Indicator	Stochastic Oscillator %K	Adaptive Price Zone	Finite Volume Element
Fisher Transform	Bollinger Bands		

In this research paper, we have considered 75 technical indicators to understand their significance for technical analysis. For calculating values of technical indicators, we have used FinTA (Financial Technical Analysis) library [35]. FinTA is Pandas based library implemented in Python. It is most widely used library for data analysis. Stock market data is time series in nature. We can easily work with it using Pandas. We need properly formatted ohlc (open, high, low, close) DataFrame to work with FinTA. It uses comma separated datasheet with column names ["open", "high", "low", "close", "volume"] for indicators that expect ohlc input.

- Steps to prepare ohlc DataFrame required for FinTA [35].
 - Standardize column names of your input file.
- `Dataframe.columns = ["date", "close", "volume"]`
 - Set index on the date column, to sort it by time periods
- `Dataframe.set_index('date', inplace=True)`
 - Selecting 'close' column and sampling values day wise
`Ohlc = dataframe["close"].resample("24h").ohlc()`

`ohlc()` method prepares DataFrame in the format required by the FinTA library so resulting ohlc series data is ready to use.

Technical indicators are calculated using statistical formula are numeric data. FinTA supports 75 trading indicators given in Table 1.

2.2 Dataset Transformation Using Wisdom of Crowd Approach

After calculating numeric values for all technical indicators using various mathematical formulae and interpreting these values in accordance with attributes present in the primary dataset (open, high, low, close, volume), we can generate the buy/sell call for each indicator. Generating final dataset for analysis is a three-fold approach. In this research work, we have used 75 indicators to generate buy/sell call.

In the first phase, we calculate numeric values of technical indicators using mathematical formulae given. In the second phase, we generate the buy/sell call for next trading day using the value of the corresponding technical indicator. There are only two trading decision calls in the second phase for the dataset generation. The novelty of this three-fold approach is in using Wisdom of crowd approach to take a final decision on trading call. Wisdom of Crowd is a collective prediction of an approach based on majority or voting based collective intelligence mechanism. The crowd here corresponds to indicators we have considered, and their trading decision call is the opinion given by them. Also, we have used five decision making categories useful for trading: strong buy, buy, hold, sell, strong sell. Most of the researchers have used only two categories for trading decision: buy or sell. However, crowdsourcing approach makes it more practical to take a trading decision and boosts trader's confidence in the trading decision as it is subjected to less error-prone and helpful in making an unbiased decision. The captivating, accurate trading decision is very important as the wrong decision by the trader may cause considerable loss of wealth. To improve the decision accuracy, we have used Wisdom of Crowd technique in the following manner:

Algorithm: Preparation of Transformed Dataset

Input: dataset containing buy/sell call for each of the 75 indicators

Output: A transformed dataset with the class label- strong buy, strong sell, buy, sell, and hold

- For each trading day
 - if more than 75% indicators generate buy/sell signal then strong buy/strong sell
 - if more than 55% and less than 75% indicators generate buy/sell signal then buy/sell
 - if more than 45% indicators and less than 55% indicators generate buy/sell signal then hold.

This transforms prediction problem to be a multiclass problem (Strong buy, buy, hold, sell, strong sell) rather than binary (buy, sell) or ternary (buy, sell, hold) classification problem which will improve trading decision making ability and confidence of investors.

3. Experimental Work

For experimental work, we will be using Multilayer Perceptron Artificial Neural Network, Random Forest, and Sequential Minimal Optimization machine learning algorithms. Transformed trend deterministic data with class labels strong buy, strong sell, buy, sell, and hold will be inputted to these algorithms and prediction performance of all these models under exploration will be evaluated.

3.1. Multilayer Perceptron – Artificial Neural Networks

The multilayered structure of the neural networks leads to extrapolative proficiency of neural networks. These multiple layers learn to represent features at lower orders and then combine them into higher-order features (David, Nelson, Adriano, Pereira and Renato 2017; Yetis, Kaplan and Jamshidi 2014). An Artificial Neural Network has three types of layers also known as nodes - an input layer, a hidden layer, and an output layer (Minh, Niaraki, Huy, Min and Moon 2018). These nodes in the layers are connected by means of a directed graph. Every node is a neuron and it is associated with a nonlinear activation function (Zhang, Zhang, Dong, Wang, and Wang 2019; Yetis, Kaplan and Jamshidi 2014). Neurons are basic computational units having weighted input signals, and with the help of an activation function, they produce an output signal. MLP is a supervised learning technique. Correlating the training data with the output variable or prediction variable is the major advantage of MLP-ANN. They are basically mapping functions, and thus can be interpreted as universal approximation algorithms (Minh, Niaraki, Huy, Min and Moon 2018; Huynh, Dang and Duong 2017; Yetis, Kaplan and Jamshidi 2014).

In MLP linear activation function does mapping of the weighted input features to the output of each neuron. Using linear algebraic equations a learning problem with multiple layers can be optimized to a input-output model with two layers (Huynh, Dang and Duong 2017).

The two most popular common activation function used in MLP is based on sigmoid functions (Patel, Shah, Thakkar, & Kotecha 2015).

$$y(v_i) = \tanh(v_i) \quad (1)$$

$$y(v_i) = \frac{1}{1 + e^{-v_i}} \quad (2)$$

Equation 1 here indicates hyperbolic tangent function with value ranging from -1 to 1. Equation 2 is a logistic function, having values 0 to 1. The output of 'i'th node is y_i and v_i is the weighted sum of the inputs is indicated as v_i . Every neuron is assigned with weight for each input signal w_i (Minh, Niaraki, Huy, Min and Moon 2018; David, Nelson, Adriano, Pereira and Renato 2017). Learning in MLP is done using backpropagation approach where when one unit of data is processed. Based on aggregated error values in the output, input weights are updated using back propagation (Yetis, Kaplan and Jamshidi 2014). For the 'k' output node in 'n'th sample of input data, degree of error is given as:

$$e_k(n) = d_k(n) - y_k(n) \quad (3)$$

Where d is the expected output, and y is the value calculated by perceptron. The weights assigned to the input nodes are adjusted based on minimization of error in overall output (David, Nelson, Adriano, Pereira and Renato 2017).

$$E(n) = \frac{1}{2} \sum_k e_k^2(n) \quad (4)$$

Stochastic gradient descent is a learning algorithm for MLP. In this step, each row of our dataset (buy/sell call from each indicator) will be passed as input to the network. The network processes this input using an activation function and produces an output variable. This is called feed-forward or feed pass in MLP (Zhang, Zhang, Dong, Wang, and Wang 2019). The output variable produced is then compared with the expected output, and error between them is calculated using equation (3). This error is then backpropagated in the network, layer by layer and weights of each input neurons are updated as per their influence in error, hence the name backpropagation (Yetis, Kaplan and Jamshidi 2014). The same process is iterated for all samples in training dataset. This process is called an epoch. MLP network can be trained using thousands of epochs. Once the MLP network is trained, it can be used for prediction.

Change in weight for each input (David, Nelson, Adriano, Pereira and Renato 2017),

$$\Delta w_{ki}(n) = -\eta \frac{\partial E(n)}{\partial v_k(n)} y_i(n) \quad (5)$$

Here,

y_k is output of last neuron, η is learning rate which is typically .1. Learning rate makes sure that weights respond to an output variable quickly without any disturbances. The simplified equation of derivative for an output node is

$$-\frac{\partial E(n)}{\partial v_k(n)} = e_k(n) \phi'(v_k(n)) \quad (6)$$

ϕ' is derivative for an activation function and it remains unchanged. As the weights are updated, modified equation for derivative becomes

$$-\frac{\partial E(n)}{\partial v_k(n)} = \phi'(v_k(n)) \sum_m -\frac{\partial E(n)}{\partial v_m(n)} w_{mk}(n) \quad (7)$$

The equation (7) is for changing weight of mth output layer. Derivative of an activation function is used to change weights of hidden layers (Minh, Niaraki, Huy, Min and Moon 2018).

Table 2. MLP parameters and settings in testing (Patel, Shah, Thakkar, & Kotecha 2015)

Parameters	Levels
Epochs	1000, 2000,.....,10000
Number of Hidden Layers- neurons	10, 20, 30,.....,100
Learning rate	.1

3.2. Random Forest

Random forest as the name indicates is an ensemble version of decision tree algorithm as a base learner for prediction, classification, or regression (Krauss, Anh, & Huck 2017) and hence also known as random decision forest. It constructs the crowd of decision trees in the training phase, and output of each tree is gathered, and the final output is the class that is the aggregated outcome of the classification result from all trees (Patel, Shah, Thakkar, & Kotecha 2015). The major advantage of Random decision over decision trees is that random forest easily overcomes the problem of overfitting. Detail description about decision tree algorithm is available with (Sakhare and Joshi 2014).

The construction of a forest is merged with randomized node optimization and bagging or bootstrap aggregating (Sharma & Juneja 2017).

Generalization error is estimated using strength of trees and using their correlation in the forest and is known as out-of-bag error.

Permutation based computation of variable importance

The construction of the forest starts with the training of individual trees using bootstrap aggregating or bagging (Sharma & Juneja 2017). Given a training set $X = (x_1, x_2, \dots, x_n)$ represents 75 technical indicators and an output set $Y = (y_1, y_2, \dots, y_m)$ represents class labels. Y repeatedly starts bagging (n times) and selects a random instance from training dataset and tries to fit the individual decision tree to these instances (Patel, Shah, Thakkar, & Kotecha 2015).

for $i = 1, \dots, n$

for m training examples $X \rightarrow X_i$

train a classification tree $f_i X_i$ on, Y_i

Once the model is trained, prediction of unknown instances X' can be done by averaging the prediction output by individual decision trees on X' or by majority based voting (Sharma & Juneja 2017).

$$\hat{f} = \frac{1}{n} \sum_{i=1}^n f_n(X') \quad (8)$$

As bagging reduces the variance of the prediction model, it leads to better performance of the model. As long as there is no correlation between individual trees, averaging the output from multiple trees is not sensitive to noise present in the training dataset. However, there would be a strong correlation between individual trees trained on a single dataset. Bootstrap aggregating anti-correlates these trees by training on different datasets (Sharma & Juneja 2017).

The standard deviation for the prediction output from all individual trees gives the estimation of uncertainty (Patel, Shah, Thakkar, & Kotecha 2015).

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (f_i(X') - \hat{f})^2}{(n-1)}} \quad (9)$$

n = number of instances per tree ($n=100, \dots, 10000$)

An optimal number of trees can be found using cross-validation or by calculating mean prediction error X_i (Patel, Shah, Thakkar, & Kotecha 2015).

3.3. Sequential Minimal Optimization

Many researchers use support vector machine (SVM) for prediction of stock market movement (Sakhare and Shaik 2019; Li, Feng, and Tseng 2017; Patel, Shah, Thakkar, & Kotecha 2015; Zhang, Li, and Pan 2016). SVM is well suited for binary classification problem (Patel, Shah, Thakkar, & Kotecha 2015; Wu and Diao 2015). Since the stock prediction problem in our research work is multiclass classification problem, we are using optimization of SVM- known as Sequential Minimal Optimization (SMO) algorithm. It uses a

heuristic-based approach by dividing the input training problem into a smaller subset of problems that can be analytically solved.

Given a training dataset-

$$\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_m)\}$$

Where,

(x_1, x_2, \dots, x_n) represents input dataset containing 75 technical indicators

(y_1, y_2, \dots, y_m) represents output classes

Support vector machine is trained using the following quadratic programming approach (Patel, Shah, Thakkar, & Kotecha 2015).

$$\max_{\alpha} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^m y_i y_j k(x_i, x_j) \alpha_i \alpha_j \tag{10}$$

Where,

$$0 \leq \alpha_i \leq C \quad \text{for } i=1, 2, \dots, n$$

$$\sum_{i=1}^m y_i \alpha_i = 0$$

C is hyper-parameter for SVM, $k(x_i, x_j)$ is kernel function and α_i represents Lagrange multiplier.

The error between actual output and the expected output is reduced by controlling parameter C. The polynomial function, and radial basis functions are two popular kernel functions which can be used as C (Patel, Shah, Thakkar, & Kotecha 2015 ; Zhang, Li, and Pan 2016).

As SMO breaks training problem into sub-problems, it uses the iterative approach for solving the above-mentioned optimization problem.

Lagrange multipliers induce constraints in the above problem such that,

$$0 \leq \alpha_1, \alpha_2 \leq C$$

$$y_1 \alpha_1 + y_1 \alpha_1 + \dots + y_m \alpha_m = k \tag{11}$$

Here we need to find a min of a 1-dimensional quadratic equation.

Steps of SMO algorithm:

For the given optimization problem

1. Lagrange multiplier α_1 is computed to disagree the Karush–Kuhn–Tucker (KKT) conditions
2. Find Lagrange multiplier α_2 and optimize the pair (α_1, α_2) .
3. Repeat steps 1,2 till convergence

The problem gets solved when both α_1, α_2 satisfy the KKT conditions the problem. Although SMO continues till convergence, rate of convergence is increased by selecting pair (α_1, α_2) using heuristics. Identifying optimal pair (α_1, α_2) very difficult as there are $\frac{n(n-1)}{2}$ possible choices for α_i and α_j .

4. Results and Discussion

Prediction results and performance of MLP-ANN, Random Forest, and SMO models are evaluated by confusion matrix. The confusion matrix is a matrix containing entries with actual prediction entries and expected output entries, and accuracy of the model can be computed using the difference of actual prediction entries and expected output entries using different derivations (Patel, Shah, Thakkar, & Kotecha 2015; Joshi and Sakhare 2015). Since we have five output class labels, the size of the confusion matrix for our prediction problem will be 5 X 5. For performance evaluation, we have used cross-validation with ten-folds. Here ten folds mean the entire dataset is randomly partitioned into ten equal training dataset to train the model with nine datasets, and one test set to evaluate it at any given point of time. Ten-fold cross validation means out of 10 subsamples, a single subsample is kept aside, and the remaining nine subsamples are used as the training dataset. For better validation each of the ten subsamples are used exactly once as the training dataset. The results from all ten folds are averaged to produce the final result. Dataset contains buy/sell call for 2479 trading days with class labels: hold, Strong_sell, sell, buy, Strong_buy.

Table 3. Confusion Matrix for MLP-ANN

Actual vs. predicted	a	b	c	d	e
a=hold	104	41	48	2	1
b=buy	43	568	0	151	1
c=sell	30	3	453	0	91
d= strong_buy	0	137	0	482	0
e=strong_sell	0	1	81	0	234

Table 4. Confusion Matrix for Random Forest

Actual vs. Predicted	a	b	c	d	e
a=hold	89	57	58	0	0
b=buy	18	634	0	111	0

c=sell	23	1	481	0	72
d= strong_buy	0	102	0	517	0
e=strong_sell	0	0	78	0	238

Table 5. Confusion Matrix for SMO

Actual vs. predicted	a	b	c	d	e
a=hold	116	47	41	0	0
b=buy	24	617	1	121	0
c=sell	33	0	459	0	85
d= strong_buy	0	108	0	511	0
e=strong_sell	0	0	78	0	238

A confusion matrix is useful for evaluating the performance of machine learning model using derivations like- 'True Positive Rate' (TPR), 'False Positive Rate' (FPR), 'Precision' (P), 'Recall' (R), 'F-measure', 'Matthew's Correlation Coefficient' (MCC), 'ROC area', 'PRC area' (Joshi and Sakhare 2015) .. Calculations and interpretation of evaluators are given as follows:

True Positives and False Positives (Joshi and Sakhare 2015):

Instances for which predicted value and actual value matches they are known to be true positives (TP) and the instances for which there is a mismatch between predicted value and actual value are false positives (FP).

$$TPR/FPR = \frac{\text{Corresponding diagonal element}}{\text{Sum of relevant row}} \quad (12)$$

The number of instances extracted from the number of relevant instances is known as precision. The number of relevant instances retrieved is known as recall. By definition TPR and recall are the same terms. (Joshi and Sakhare 2015).

$$F - \text{measure} = \frac{2 * P * TPR}{P + TPR} \quad (13)$$

Matthew's correlation coefficient (MCC) measures the strength of the classification/prediction problem. MCC value lies from -1 to +1. -1 shows a comprehensive divergence between predicted and actual value, 0 shows arbitrary prediction, and +1 shows the ideal prediction (Joshi and Sakhare 2015). The mathematical formula to calculate MCC is:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (14)$$

TN stands for true negatives and FN stands for false negatives, respectively.

ROC area measures how well the classifier performs. PRC is the precision-recall curve area. Other statistical measures used for performance analysis are mean absolute error (MAE), root mean squared error (RMSE), relative absolute error (RAE) and root relative squared error (RRSE).

Table 6. Performance Evaluators for MLP-ANN

Evaluator vs. class label	TPR	FPR	P	R	F-measure	MCC	ROC Area	PRC Area
hold	.510	.032	.588	.510	.546	.510	.914	.515
buy	.744	.111	.749	.744	.747	.635	.918	.820
sell	.785	.068	.788	.785	.782	.715	.953	.833
strong_buy	.779	.082	.759	.779	.769	.691	.949	.817
strong_sell	.741	.043	.716	.741	.728	.687	.964	.786
Weighted average	.743	.079	.741	.743	.741	.664	.939	.793

Table 7. Performance Evaluators for Random Forest

Evaluator vs. class label	TPR	FPR	P	R	F-measure	MCC	ROC Area	PRC Area
hold	.436	.018	.685	.436	.533	.516	.956	.613
buy	.831	.093	.798	.831	.814	.730	.944	.879
sell	.834	.072	.780	.834	.806	.745	.965	.880
strong_buy	.835	.060	.823	.835	.829	.772	.967	.877
strong_sell	.753	.033	.768	.753	.760	.726	.977	.832
Weighted average	.790	.066	.787	.790	.786	.726	.960	.851

Table 8. Performance Evaluators for SMO

Evaluator vs. class label	TPR	FPR	P	R	F-measure	MCC	ROC Area	PRC Area
hold	.569	.025	.671	.569	.615	.586	.930	.493
buy	.809	.090	.799	.809	.804	.716	.913	.743
sell	.795	.063	.793	.795	.794	.732	.944	.748

strong_buy	.826	.065	.809	.826	.817	.755	.945	.765
strong_sell	.753	.039	.737	.753	.745	.707	.962	.677
Weighted average	.783	.066	.782	.783	.782	.718	.936	.721

Table 9. Detailed statistics of accuracy for MLP-ANN

Correctly predicted instances	1841
Incorrectly predicted instances	638
Prediction accuracy	74.26%
Kappa Statistics	.6634
MAE	.1062
RMSE	.3006
RAE	34.6589
RRSE	76.8069

Table 10. Detailed statistics of accuracy for Random Forest

Correctly predicted instances	1959
Incorrectly predicted instances	520
Prediction accuracy	79.02%
Kappa Statistics	.724
MAE	.1363
RMSE	.2473
RAE	44.4909
RRSE	63.1921

Table 11. Detailed statistics of accuracy for SMO

Correctly predicted instances	1941
Incorrectly predicted instances	538
Prediction accuracy	78.29%
Kappa Statistics	.7159
MAE	.2489
RMSE	.33
RAE	81.2622
RRSE	84.3324

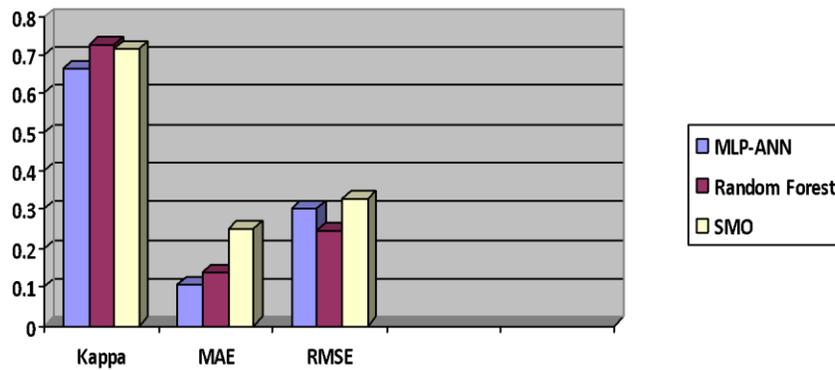


Figure 1. Kappa, MAE, MSE comparison

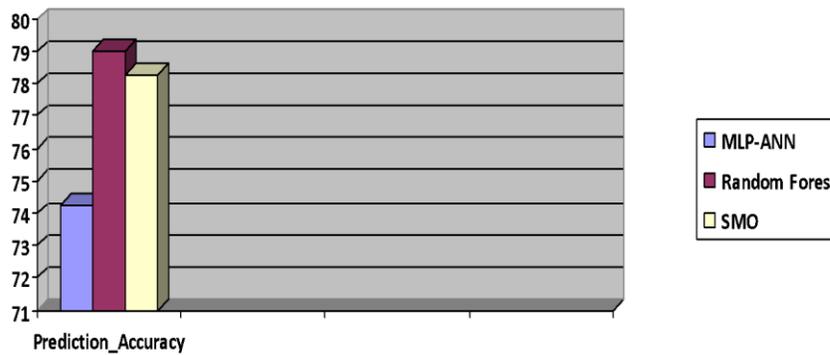


Figure 2. Prediction accuracy comparison

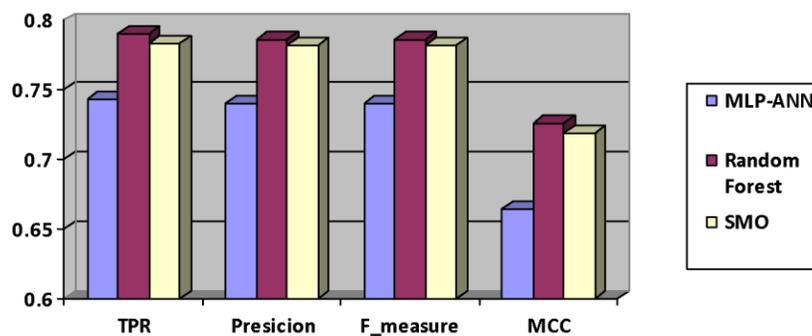


Figure 3. Performance comparison using various evaluators

The novelty in this research work is the preparation of transformed dataset, which is prepared by interpreting technical indicators and transforming their values into buy and sell calls for the corresponding day. From the statistics, it is clear that the Sequential Minimal Optimization model outperforms the Multilayer Perceptron- ANN and Random Forest by a small margin. For the analysis, we have considered

75 indicators. Block diagram of the research work is shown in figure 5. It will be interesting to perform the analysis on a redundant set of indicators where an irrelevant set of indicators can be eliminated using principal component analysis, Pearson correlation coefficient, information gain based approaches. Experimental setup mentioned in this paper can also be performed for the prediction of 5-day, 15-day, and 30-day stock price.

Conclusions

Technical Indicators themselves do not generate any trading signals. Each trader is responsible for formulating the exact methodology/strategy in which the indicators will be used for trading opportunities and to generate strategies. Traders need at least one technical indicator for creating the best suited technical trading strategy. The business logic-set of rules used by traders allow them to analyze historical data, backtrack and identify practicability of a particular strategy. Indicators and technical analysis help traders to interpret the mathematical as well as statistical anticipation of the trading rules, or the strength of the strategy to execute in the future. However, traders must explore the performance of the strategy and modify rules as and when needed. There is no as such Holy Grail (secret strategy leading to instantaneous strategy) and ensure traders with potential profit. Each trader has own elegance of trading, confidence, risk-taking ability, personality, and based on these parameters. Traders interpret technical indicators required for technical analysis. The trader also needs to do in-depth research on how indicators perform and develop strategies based on the output of these indicators. To predict the stock market movements with the help of analysis based technical indicators machine learning models can be employed. We have designed a transformed dataset based on the Wisdom of Crowd approach. Its machine learning-based analysis will help traders in decision making with multiple categories useful for trading: Strong buy, buy, hold, sell, strong sell. This is novel work carried out in this research. We have used Random Forest (RF), Multilayer Perceptron Artificial Neural Networks (MP-ANN) and Sequential Minimal Optimization (SMO) machine learning algorithms to analyze the transformed dataset and also compared their prediction accuracy and performance. SMO beats MLP-ANN and Random Forest only by a small margin. In this paper more focus is given on intra-day trading. Long term prediction of stock market price movements can be considered as a future scope. It is also important to experiment machine learning-based fundamental analysis of company's stocks using fundamental parameters like quarter's performance, P/E ratio and revenue, debt to equity ratio, security, and stability. News related to global trade wars, economy, takeover, the merger of companies, announcements directly or indirectly associated with stock market impact its movement to a great extent. News analytics can also be done to study the correlation between news and stock market movement which helps traders to take the decision on buy/sell of stocks.

References

1. Tsai, C., F., & Quan, Z., Y. (2014). Stock Prediction By Searching for Similarities in Candlestick Charts. ACM Transactions on Management Information Systems Article No.: 9 <https://doi.org/10.1145/2591672>
2. Li, X., Xie, H., Wang, R. et al. (2016) Empirical analysis: stock market prediction via extreme learning machine. Neural Comput & Applic 27, 67–78. <https://doi.org/10.1007/s00521-014-1550-z>
3. Sakhare, N., & Shaik I. (2019). Performance Analysis of Regression-Based Machine Learning Techniques for Prediction of Stock Market Movement. International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-7, Issue-6S4, April 2019.

4. Feng, F., He, X., Wang, X., Luo, C., Liu, Y., & Chua, T., S. (2019). Temporal Relational Ranking for Stock Exploration. Transactions on Information Systems (TOIS). ACM Trans. Inf. Syst. 37, 2, Article 27 (March 2019), 30 pages [10.1145/3309547](https://doi.org/10.1145/3309547)
5. Calzon, C., B., Harold, M., B., Garzas, J., J., & Herrera, F., J. (2018). Expert Selection in Prediction Market With Homological Invariants. INSPEC Accession Number: 17905708, IEEE ACCESS, Vol. 6., pp. 32226-32239. [10.1109/ACCESS.2018.2846878](https://doi.org/10.1109/ACCESS.2018.2846878)
6. Sheikh, M., I., Alam, M., A., & Agarwal, P. (2019). A Prediction Approach for Stock Market Volatility Based on Time Series Data. INSPEC Accession Number: 18453700, IEEE ACCESS, Vol. 7, pp. 17287-17298, [10.1109/ACCESS.2019.2895252](https://doi.org/10.1109/ACCESS.2019.2895252)
7. Cailee, J., M., Itani, A., Gueriot, G., & Rakotondratsimba, Y. (2017). Stock Picking by Probability-Possibility Approaches. INSPEC Accession Number: 16776016, IEEE Transaction on Fuzzy Systems, Vol. 25, Issue 2, 333 – 349. [10.1109/TFUZZ.2016.2574921](https://doi.org/10.1109/TFUZZ.2016.2574921)
8. Barmish, B., R., & Primbs, J., A. (2016). On a New Paradigm For Stock Trading Via a Model- Free Feedback Controller. INSPEC Accession Number: 15806153, IEEE Transaction on Automatic Control, Vol. 6, Issue 3, 662 – 676. [10.1109/TAC.2015.2444078](https://doi.org/10.1109/TAC.2015.2444078)
9. Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock market index using fusion of machine learning techniques. Expert Systems with Applications. Volume 42, Issue 4, Pages 2162-2172. <https://doi.org/10.1016/j.eswa.2014.10.031>
10. 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. Volume 42, Issue 1, January 2015, Pages 259-268. <https://doi.org/10.1016/j.eswa.2014.07.040>
11. Patel, H., & Parikh, S. (2016). Comparative Analysis of Different Statistical and Neural Network Based Forecasting Tools for Prediction of Stock Data. ICTCS '16: Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies March 2016 Article No.: 126 Pages 1–6. <https://doi.org/10.1145/2905055.2905186>
12. Wen, M., Li, P., Zhang, L., & Chen, Y. (2019). Stock Market Trend Prediction Using High-Order Information of Time Series. INSPEC Accession Number: 18521046, IEEE Access Vol. 7, pp.28299-28308. DOI:1.1109/ACCESS.2019.2901842
13. Chong, E. and Han, C. and Park, F.C. (2017) 'Deep learning networks for stock market analysis and prediction : methodology, data representations, and case studies.', Expert systems with applications., 83 . pp. 187-205. <https://doi.org/10.1016/j.eswa.2017.04.030>
14. Li, Y. F., Feng, C., and Tseng, V. (2017). A Novel methodology for Stock Investment Using High Utility Episode Mining and Genetic Algorithms. Elsevier, Applied Soft Computing, Volume 59, October 2017, Pages 303-315. <https://doi.org/10.1016/j.asoc.2017.05.032>
15. Nayak, R., Mishra, D., and Kumar, A. (2015). A Naïve SVM-KNN Based Stock Market Trend Reversal Analysis For Indian Benchmark Indices. Applied Soft Computing. Volume 35, October 2015, Pages 670-680. <https://doi.org/10.1016/j.asoc.2015.06.040>
16. Zhang, X., D., Li, A., and Pan, R. (2016). Stock Trend Prediction Based on a New Status Box Method and Adaboost probabilistic Support Vector Machine. Applied Soft Computing. Volume 49, December 2016, Pages 385-398. <https://doi.org/10.1016/j.asoc.2016.08.026>
17. Sakhare, N., Shaik, I., Kagad, S., Kapadwanjwala, T., Malekar, H., Dalal M. (2020). Stock Market Prediction Using Sentiment Analysis. International Journal of Advanced Science and Technology, 29(4s), pp.1126 – 1133
18. Minh, D., L., Niaraki, A., S., Huy, H., D., Min, K., Moon, H., (2018). Deep Learning Approach For Short Term Stock Trends Prediction Based On Two-Stream Gated Recurrent Unit Network. INSPEC Accession Number: 18159342, IEEE Access, vol. 6, pp. 55392-55404. [10.1109/ACCESS.2018.2868970](https://doi.org/10.1109/ACCESS.2018.2868970)
19. Lin, G., and Wang, X. (2018). A Numerical based Attention Method for Stock Market Prediction with Dual Information. IEEE ACCESS, vol. 7, pp.7357-7367. [10.1109/ACCESS.2018.2886367](https://doi.org/10.1109/ACCESS.2018.2886367).
20. Sezer, B., A. Ozbayoglu, A., M., & Dogdu, E. (2017). An Artificial Neural Network Based Stock Trading System Using Technical Analysis and Big Data Framework, ACM SE '17: Proceedings of the SouthEast Conference. Pages 223–226 <https://doi.org/10.1145/3077286.3077294>
21. Chen, Y., & Hao, Y. A (2017). Feature Weighted Support Vector Machine and K- nearest neighbor Algorithm For Stock Market Indices Prediction. Expert Systems with Applications. Volume 80, 1 September 2017, Pages 340-35. <https://doi.org/10.1016/j.eswa.2017.02.044>
22. David M., Nelson, Q., Adriano C., Pereira, M., Renato A., (2017). Stock Market's Price Movement Prediction with LSTM Neural Networks. INSPEC Accession Number: 17070691, IEEE International Conference on Neural Networks (IJCNN), [10.1109/IJCNN.2017.7966019](https://doi.org/10.1109/IJCNN.2017.7966019).
23. Lei, Z., & Lin, W., (2015). Price Trend Prediction of Stock Market Using Outlier Data Mining Algorithm", INSPEC Accession Number: 15570996, 2015 IEEE Fifth International Conference on Big Data and Cloud Computing. [10.1109/BDCLOUD.2015.19](https://doi.org/10.1109/BDCLOUD.2015.19)
24. <https://www.bseindia.com/> <https://www.nseindia.com/> .
25. Zhang, K., Zhang, G., Dong, J., Wang, S., and Wang, Y. (2019). Stock Market Prediction Based on Generative Adversarial Network. Procedia Computer Science. Volume 147, 2019, Pages 400-406. <https://doi.org/10.1016/j.procs.2019.01.256>

26. Huynh, H., Dang, L., Duong, D. (2017). A New Model for Stock Price Movement Prediction Using Deep Neural Network. SolCT 2017: Proceedings of the Eighth International Symposium on Information and Communication Technology December 2017 Pages 57–62. <https://doi.org/10.1145/3155133.3155202>.
27. Wu, M., and Diao, X. (2015). Technical Analysis of Three Stock Oscillators- Testing MACD, RSI and KDJ Rules in SH and SZ stock markets. INSPEC Accession Number: 16090318, 2015 4th International Conference on Computer Science and Network Technology (ICCSNT), [10.1109/ICCSNT.2015.7490760](https://doi.org/10.1109/ICCSNT.2015.7490760)
28. Krauss, C., Anh, X., and Huck, N. (2017). Deep Neural Networks, Gradient-Boosted Trees, Random Forests: Statistical Arbitrage on S&P 500. European Journal of Operational Research. Volume 259, Issue 2, 1 June 2017, Pages 689-702. <https://doi.org/10.1016/j.ejor.2016.10.031>
29. Yetis, Y., Kaplan, H., Jamshidi, M. (2014). Stock Market Prediction by Using Artificial Neural Network. INSPEC Accession Number: 14700939 IEEE, World Automation Congress. [10.1109/WAC.2014.6936118](https://doi.org/10.1109/WAC.2014.6936118)
30. Sharma, N., and Juneja, A., (2017). Combining of Random Forest Estimates Using LSboost for Stock Market Index Prediction. INSPEC Accession Number: 17449287, 2nd International Conference for Convergence in Technology (I2CT), [10.1109/I2CT.2017.8226316](https://doi.org/10.1109/I2CT.2017.8226316)
31. Pehlivanli, A., C., & Guzman, A. (2016). Indicator Selection with Committee Decision of Filter Methods for Stock Market Price Trend in ISE. Applied Soft Computing Volume 49, December 2016, Pages 792-800. <https://doi.org/10.1016/j.asoc.2016.09.004>
32. Waqar, M., Dawood, H., Bilal, M., Ali, M., Guo, P. (2017). Prediction of Stock Market by Principle Component Analysis. INSPEC Accession Number: 17578286, 2017 13th International Conference on Computational Intelligence and Security (CIS), [10.1109/CIS.2017.00139](https://doi.org/10.1109/CIS.2017.00139).
33. Joshi, S., and Sakhare, N. (2015). History Bits based novel algorithm for classification of structured data. INSPEC Accession Number: 15292807, IEEE International Advance Computing Conference (IACC), Bangalore, pp. 609-612. [10.1109/IADCC.2015.7154779](https://doi.org/10.1109/IADCC.2015.7154779)
34. Sakhare, N., and Joshi, S. (2014). Classification of Criminal Data Using J48 Algorithm”, International Journal of Datawarehousing and Mining. Vol. 4. pp. 167-171.
35. <https://github.com/leomrocha/finta>
36. <https://www.investopedia.com/terms>