Pune Institute of Computer Technology
Dhankawadi, Pune

SEMINAR REPORT
ON

Predicting financial index movement using trend indicators and machine learning

SUBMITTED BY

Sooraj V S
Roll No.: 31165
Class: TE-I

Under the guidance of
Prof. P.R. Rajmane

DEPARTMENT OF COMPUTER ENGINEERING
Academic Year 2020-21

DEPARTMENT OF COMPUTER ENGINEERING

Pune Institute of Computer Technology

Dhankawadi, Pune-43

CERTIFICATE

This is to certify that the Seminar report entitled

"PREDICTING FINANCIAL INDEX MOVEMENT USING TREND INDICATORS AND MACHINE LEARNING"

Submitted by

Sooraj V S Roll No. 31165

has satisfactorily completed a seminar report under the guidance of Prof. P.R. Rajmane towards the partial fulfillment of third year Computer Engineering Semester II, Academic Year 2020-21 of Savitribai Phule Pune University.

Prof. P.R. Rajmane
Internal Guide

Prof. M.S.Takalikar
Head
Department of Computer Engineering

Place:
Date:

ACKNOWLEDGEMENT

I sincerely thank our Seminar Coordinator Prof. B.D.Zope and Head of Department Prof. M.S.Takalikar for their support.

I also sincerely convey my gratitude to my guide Prof. P.R. Rajmane, Department of Computer Engineering for her constant support, providing all the help, motivation and encouragement from beginning till end to make this seminar a grand success.

Contents

List of Tables

List of Figures

Abstract

Stock price prediction has always been a challenging task for researchers in the financial domain. While the Efficient Market Hypothesis claims that it is impossible to predict stock prices accurately, there are studies in finance literature that have demonstrated the index price movements can be forecasted with a reasonable degree of accuracy, if appropriate variables are chosen and suitable predictive models are built using those variables.

This paper compares prediction models based on machine learning algorithms that can be used to forecast the index movement (trend) for a coming short-term interval, with the inputs being trend indicators, computed on trading data (open, high, low & close prices). Extensive results have been presented on the performance of these models. Evaluation is carried out on 10 years of historical data from 2010 to 2020 of National Stock Exchange (NSE) of India indices at daily intervals..

Keywords

Trend indicators, Naive-Bayes classification, Logistic Regression, Support vector machine, Random forest, Artificial neural networks, Long short-term memory

1 INTRODUCTION

Prediction of future movement of stock prices has been the subject matter of many research work. On one hand, we have proponents of the Efficient Market Hypothesis who claim that stock prices cannot be predicted. On the other hand, there are work that have shown that, if correctly modeled, stock prices can be predicted with a fairly reasonable degree of accuracy. The latter have focused on choice of variables, appropriate functional forms and techniques of forecasting. In this regard, Sen and Datta Chaudhauri propose a novel approach of stock price forecasting based on a time series decomposition approach of the stock prices time series.

There is also an extent of literature on technical analysis of stock prices where the objective is to identify patterns in stock movements and profit from it. The literature is geared towards making money from stock price movements, and various indicators like Bollinger Band, Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI), Moving Average, Momentum Stochastics, Meta Sine Wave etc., have been devised towards this end. There are also patterns like Head and Shoulders, Triangle, Flag, Fibonacci Fan, Andrew's Pitchfork etc., which are extensively used by traders for gain. These approaches provide the user with visual manifestations of the indicators which helps the stors to understand which way stock prices may move.

In this paper, we compare different machine learning algorithms for prediction models that uses a granular approach for estimating the movemnt for next short-term interval by combining statistical learning and trend deterministic data preperation. We have presented several approaches for short-term price movement forecasting using various classification, regression techniques and compared their performance in terms of accuracy, recall and f1-score with visualisation of the results as confusion matrix heat map and plotting actual index movement against predicted movement.

2 MOTIVATION

Financial Index price prediction is a classic and important problem. With a successful model for index prediction, we can gain insight about market behavior over time, spotting trends that would otherwise not have been noticed. With the increasingly computational power of the computer, machine learning will be an efficient method to solve this problem.

Many people are interested in the financial markets, need guidance and accurate predictions to invest wisely. Investors are always looking for the accurate future results. The most fundamental motivation for trying to predict the market prices is financial gain. The ability to uncover a mathematical model that can consistently predict the direction of the future index prices would make the owner of the model maximize the gains from the same. Thus, researchers, investors and investment professionals are always attempting to find a stock market model that would yield them higher returns than their counterparts.

In this paper, we discuss the Machine Learning techniques which have been applied for trading to predict the rise and fall of index prices before the actual event of an increase or decrease in the stock price occurs. In particular, the paper discusses the application of Support Vector Machines, Logistic Regression, Deep Neural Networks, Random Forest Classifier in detail along with the benefits and pitfalls of each method. The paper introduces the parameters and variables that can be used in order to recognize the patterns in index prices which can be helpful in future prediction of the index movement.

3 LITERATURE SURVEY

Time Series Prediction: Predicting Stock Price

Ref: [1] The aim of this research is to develop a predictive model to forecast financial time series data. This study examined 5 and developed 4 predictive models. The mean and linear regression analysis imply that the predictive values and the real values are deviating from the mean. Then the GLM and RNN model compared with Mean and ordinary linear model. Empirical examinations of predicting precision for the price time series show that the proposed models (GLM, LSTM-RNN) fail to improve on the precision of forecasting 1 dimensional time series.

Using Machine Learning Algorithms on Prediction of Stock Price

Ref: [2] This paper investigate analysis and prediction of the time-dependent data. We focus our attentionon four different stocks are selected from Yahoo Finance historical database. To build up models and predict thefuture stock price, we consider three different machine learning techniques including Long Short-Term Memory(LSTM)and Support Vector Regression (SVR). By treating close price,open price, daily low, daily high, adjusted close price, and volume of trades as predictors in machine learning methods.

Predicting the Trend of Stock Market Index Using the Hybrid Neural Network Based on Multiple Feature Learning

Ref: [3] Hybrid neural network based on multiple time scale feature learningfor stock market index trend prediction. Because there are multi-scale features in financial time series,it makes sense to combine them to predict future trends. First, the proposed model only utilizes one CNN to extract multiple time scale features, instead of using multiple networks like other models. It simplifies the model and makes more accurate predictions. Second, time dependences in the multiple time scale features are learned by three LSTMs. Finally, the information learned by LSTMs is fused through fully connected layers to predict the price trend.

Stock market prediction using machine learning and deep learning techniques

Ref: [4] This survey concludes though various methods and approaches can be used for predicting the stock price, every method has its limitations and advantages. By considering both technical indicators and fundamental analysis the accuracy of the predictions can be made more accurate and reliable. Despite many such algorithms available, there is always room for improvement. It is observed that use of different parameters of the data set in different algorithms results in various accuracy rates.

Stock Price Prediction Using Machine Learning and Deep Learning Frameworks

Ref: [5] Proposed a framework of stock price movement prediction in the short-term time period using eight classification and eight regression models. These models are based on machine learning and deep learning approaches. We built, fine-tuned and tested these models using the data of two stocks listed in the National Stock Exchange (NSE). Observed that while among the classification techniques, Artificial Neural Networks (ANNs) has, on the average, produced the highest level of accuracy.

The Following table shows the literature survey by comparing techniques propose in various references:

Table 1: Literature survey

Model/Algorithm	Overview	Positive Aspects	Limitations
Naive Bayes Classifier [4]	Previous interval Close, Low and High price are taken as input, Output predicts the direction of movement in next interval.	This approach can be used for making real time predictions as it doesn't require as much training data making it ideal for short term predictions.	Naive Bayes assumes that all predictors are independent, this limits the applicability of this algorithm in predicting index movements which are continuous and highly correlated to each other.
Logistic Regression [4]	This model uses technical indicators calculated from the previous rates and predicts output class corresponding to upward or downward movement for future interval.	It can interpret model coefficients as indicators of feature importance. Good accuracy for many simple data sets and it performs well when the dataset is linearly separable.	Exact value of the movement can't be predicted using logistic regression as the output has to be in form of classes.
Random Forest Classifier [5]	Continuous Close, Low and High price are taken as input to predict the direction of the movement.	It works well with both categorical and continuous values thus becomes flexible to both classification and regression problem.	Due to the ensemble of decision trees, it also suffers interpretability and fails to determine the significance of each variable.
Deep Neural Network [2]	Trend indicators are used as inputs along with fear/greed index estimated using sentiment analysis on social media platforms.	Investor's emotions derived from Twitter have impacts on stock indicators which is considered while making prediction.	Taking twitter posts into accord compromise the ability of the model in predicting real-time results.

4 PROBLEM DEFINITION AND SCOPE

4.1 Problem Definition

To compare the accuracy of well known machine learning algorithms, extracted from literature survey when used with trend indicators as inputs to predict financial indices.

4.2 Scope

This study focuses on comparing prediction performance of Naive-Bayes, Logistic regression, Random forest, SVM and ANN models for the task of predicting index price movement. Twelve technical parameters computed at different intervals are used as the inputs to these models *(wiz. SMA, WMA, EMA, Momentum, Stochastic K%, Stochastic D%, RSI, MACD and Commodity Channel Index).*

This paper uses Trend Deterministic Data Preparation Layer which converts continuous valued inputs to discrete ones. Each input parameters in its discrete form indicates a possible up or down trend determined based on its inherent property. Correlation between the technical parameters and discrete index movement is also compared using correlation heat-map and pairplots against every input feature.

All the experiments are carried out using 10 years historical data from 2010 to 2020 of National Stock Exchange (NSEI) indices NIFTY 50, ONGC, HDFC BANK and BOSCH LTD obtained from the Yahoo Finance Website . Both stocks and indices are highly voluminous and vehemently traded in and so they reflect Indian economy as a whole.

5 DATA DESCRIPTION

The datasets used in this project include NIFTY50, ONGC, HDFC BANK, and BOSCH LTD of National Stock Exchange India historical data obtained from Yahoo Finance. The data contain open price, close price, daily high price, daily low price, adjusted close price and the volume of trades for these indices from April 1st, 2010 until March 31st, 2020.

Table 2: Unprocessed Data NIFTY50

Date	Open	High	Low	Close	Adj. Close	Volume
2010-04-01	5249.200195	5298.600098	5249.200195	5290.500000	5290.500000	0.0
2010-04-05	5291.399902	5377.549805	5291.399902	5368.399902	5368.399902	0.0
2010-04-06	5369.649902	5388.649902	5351.700195	5366.000000	5366.000000	0.0
2010-04-07	5365.700195	5399.649902	5345.049805	5374.649902	5374.649902	0.0
2010-04-08	5376.299805	5383.649902	5290.250000	5304.450195	5304.450195	0.0

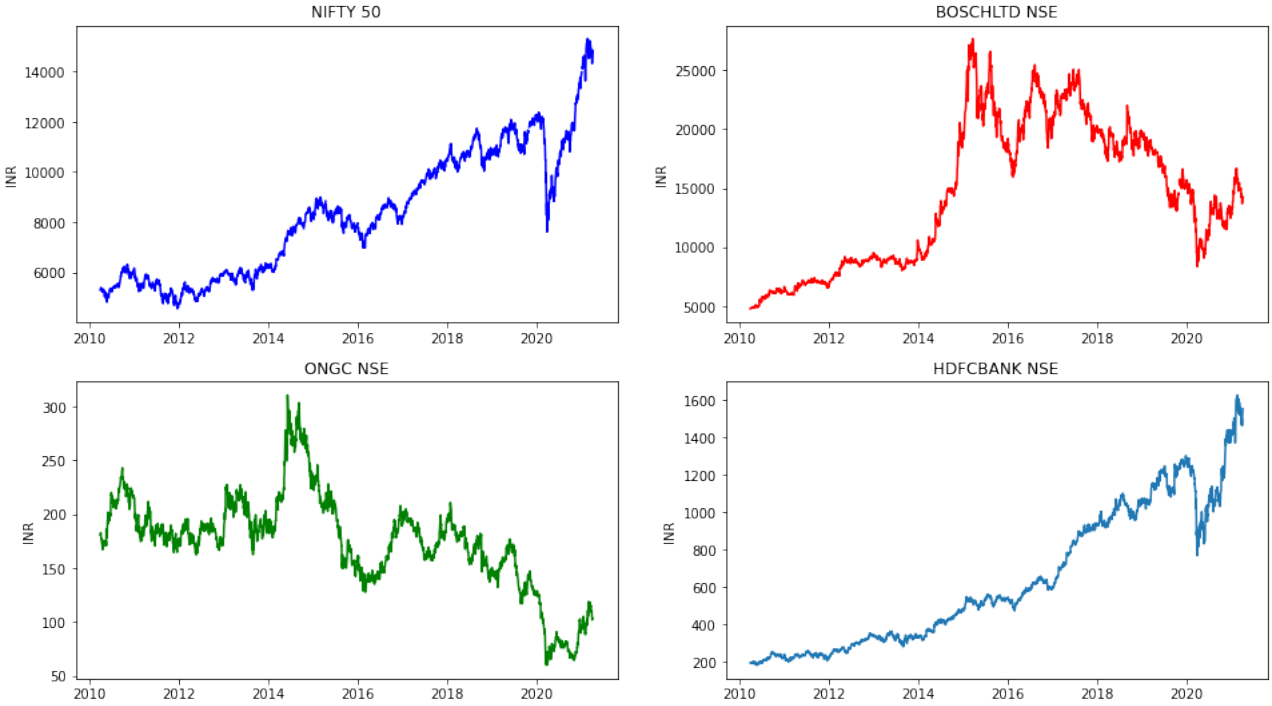

Figure 1: Line plots: Date vs Close Price (Raw Data)

5.1 Processing Close Price

Continuous 'Close Price' attribute is converted into 'Trend' with two categorical values 'UP' and 'DOWN' denoted by +1,-1 respectively.

$$Trend_t = \frac{Close_t - Close_{t-1}}{|Close_t - Close_{t-1}|}$$

Table 3: UP/DOWN Trends Annually

Year	NIFTY 50		ONCG		HDFC BANK		BOSCH LTD	
	+VE	-VE	+VE	-VE	+VE	-VE	+VE	-VE
2010	132	121	123	130	125	128	122	131
2011	110	132	118	128	126	120	123	123
2012	127	115	112	133	127	118	124	121
2013	132	114	123	125	122	126	115	133
2014	131	109	113	130	123	120	123	120
2015	119	125	122	124	117	129	116	130
2016	135	111	140	107	134	113	121	126
2017	129	116	111	135	135	111	108	138
2018	137	108	121	125	128	118	110	136
2019	121	124	110	135	121	125	113	132
2020	150	96	136	111	132	115	119	128
Total	1423	1271	1329	1383	1390	1322	1294	1418

5.2 Calculating Trend Indicators

Indicators, such as moving average and Relative Strenght index, are mathematically-based technical analysis tools that traders and investors use to analyze the past and anticipate future price trends and patterns. Where fundamentalists may track economic data, annual reports, or various other measures of corporate profitability, technical traders rely on charts and indicators to help interpret price moves.

A growing number of technical indicators are available for traders to study, including those in the public domain, such as a moving average or the stochastic oscillator, as well as commercially available proprietary indicators. In addition, many traders develop their own unique indicators, sometimes with the assistance of a qualified programmer. Most indicators have user-defined variables that allow traders to adapt key inputs such as the "look-back period" (how much historical data will be used to form the calculations) to suit their needs. The indicators used in this study are referred from Technical Analysis section of Investopedia Website.

6.2.1 Simple Moving Average (SMA)

A simple moving average (SMA) is an arithmetic moving average calculated by adding recent prices and then dividing that figure by the number of time periods in the calculation average. For example, one could add the closing price of a security for a number of time periods and then divide this total by that same number of periods. Short-term averages respond quickly to changes in the price of the underlying security, while long-term averages are slower to react.

$$SMA(n) = \frac{Close_t + Close_{t-1} + Close_{t-2} + ... + Close_{t-n+1}}{n}$$

where:

$n = no.\ of\ lookback\ interval$

6.2.2 Weighted Moving Average (WMA)

A linearly weighted moving average (WMA) is a moving average calculation that more heavily weights recent price data. The most recent price has the highest weighting, and each prior price has progressively less weight. The weights drop in a linear fashion. LWMAs are quicker to react to price changes than simple moving averages (SMA) and exponential moving averages (EMA).

$$WMA(n) = \frac{nC_t + (n-1)C_{t-1} + (n-2)C_{t-2} + ... + C_{t-n+1}}{n + (n-1) + (n-2) + ... + 1}$$

where:

$C_t = Closing\ Price\ at\ T^{th}\ Day$

$n = no.\ of\ lookback\ interval$

6.2.3 Exponential Moving Average (EMA)

An exponential moving average (EMA) is a type of moving average (MA) that places a greater weight and significance on the most recent data points. The exponential moving average is also referred to as the exponentially weighted moving average. An exponentially weighted moving average reacts more significantly to recent price changes than a simple moving average (SMA), which applies an equal weight to all observations in the period.

$$EMA(n) = C_t k + EMA_{t-1}(1-k)$$

where:

$$k = \frac{2}{n+1}$$

$C_t = Closing\ Price\ at\ T^{th}\ Day$

$n = no.\ of\ lookback\ interval$

6.2.4 Momentum

Momentum is the rate of acceleration of a security's price—that is, the speed at which the price is changing. Momentum trading is a strategy that seeks to capitalize on momentum to enter a trend as it is picking up steam. Simply put, momentum refers to the inertia of a price trend to continue either rising or falling for a particular length of time, usually taking into account both price and volume information. In technical analysis, momentum is often measured via an oscillator and is used to help identify trends.

$$Momentum(n) = C_t k - C_{t-n+1}$$

where:

$C_t = Closing\ Price\ at\ T^{th}\ Day$

$n = no.\ of\ lookback\ interval$

6.2.5 Stochastic Oscillator

A stochastic oscillator is a momentum indicator comparing a particular closing price of a security to a range of its prices over a certain period of time. The sensitivity of the oscillator to market movements is reducible by adjusting that time period or by taking a moving average of the result. It is used to generate overbought and oversold trading signals, utilizing a 0–100 bounded range of values. Notably, %K is referred to sometimes as the fast stochastic indicator. The "slow" stochastic indicator is taken as %D = 3-period moving average of %K.

$$\%K_t = \frac{C_t - LL_{14}}{HH_{14} - LL_{14}}, \qquad \%D_t = \frac{\%K_t + \%K_{t-1} + \%K_{t-2}}{3}$$

where:

$C_t = Closing\ Price\ at\ T^{th}\ Day$

$LL_{14} = Lowest\ Low\ value\ in\ 14\ day\ period$

$HH_{14} = Higher\ High\ value\ in\ 14\ day\ period$

$n = no.\ of\ lookback\ interval$

6.2.6 Relative Strength Index(RSI)

The relative strength index (RSI) is a momentum indicator used in technical analysis that measures the magnitude of recent price changes to evaluate overbought or oversold conditions in the price of a stock or other asset. The RSI is displayed as an oscillator and can have a reading from 0 to 100. Traditional interpretation and usage of the RSI are that values of 70 or above indicate that a security is becoming overbought or overvalued and may be primed for a trend reversal or corrective pullback in price. An RSI reading of 30 or below indicates an oversold or undervalued condition.

$$RSI(n) = 100 - \frac{100}{1 + RS}$$

where:

$$RS = Relative\ Strength = \frac{\sum_{i=0}^{n-1} UP_{t-i}}{\sum_{i=0}^{n-1} DW_{t-i}}$$

$UP_t = UP\ movement\ at\ t^{th}\ day$

$DW_t = DOWN\ movement\ at\ t^{th}\ day$

$n = no.\ of\ lookback\ interval$

6.2.7 Moving Average Convergence Divergence (MACD)

Moving average convergence divergence (MACD) is a trend-following momentum indicator that shows the relationship between two moving averages of a security's price. The MACD is calculated by subtracting the 26-period exponential moving average (EMA) from the 12-period EMA. The result of that calculation is the MACD line. A nine-day EMA of the MACD called the "signal line," is then plotted on top of the MACD line, which can function as a trigger for buy and sell signals.

$$MACD = EMA(12) - EMA(26), \qquad Single\ Line = EMA(9)_{MACD}$$

where:

$$EMA(n) = exponantial\ moving\ average\ over\ 'n'\ intervals$$

$$EMA(n)_{MACD} = EMA\ calculated\ on\ MACD\ over\ 'n'\ intervals$$

6.2.8 Commodity Channel Index (CCI)

The Commodity Channel Index (CCI) is a technical indicator that measures the difference between the current price and the historical average price. The CCI is an unbounded oscillator, meaning it can go higher or lower indefinitely. For this reason, overbought and oversold levels are typically determined for each individual asset by looking at historical extreme CCI levels where the price reversed from.

$$CCI(n) = \frac{M_t - SM_t}{0.015 D_t}$$

where:

$$M_t = \frac{H_t + L_t + C_t}{n}$$

$$SM_t = \frac{\sum_{i=1}^{n} M_{t-i+1}}{n}$$

$$D_t = \frac{\sum_{i=1}^{n} |M_{t-i+1} - SM_t|}{n}$$

$$H_t, L_t, C_t = Highest,\ Lowest,\ Close\ Price\ at\ T^{th}\ Day$$

$$n = no.\ of\ lookback\ interval$$

Table 4: Summary statistics for the selected indicators: NIFTY 50

Indicator	Max	Min	Mean	Std Deviation
SMA(10)	15160.595020	4667.675000	8321.948450	2485.256663
WMA(10)	15251.270052	4625.610124	8333.409202	2494.077298
MOMENTUM(10)	1538.700196	-3005.549805	32.657335	296.673529
%K	100.000000	0.000000	59.415354	37.561276
%D	100.000000	0.000000	59.425484	34.944314
RSI(14)	82.169304	12.941799	54.154687	12.488198
EMA(20)	14957.464768	4753.103289	8303.347910	2465.464784
EMA(50)	14626.442215	4855.949583	8250.264506	2413.729929
MACD	357.070016	-1005.837460	25.765568	117.551862
EXP	324.517714	-772.093159	25.619369	106.515835
CCI(14)	360.650710	-314.380437	19.350198	106.068600
CCI(21)	352.533134	-377.993237	23.230316	108.655335
CCI(50)	344.544991	-403.735743	34.897643	112.935668

Table 5: Summary statistics for the selected indicators: ONGC

Indicator	Max	Min	Mean	Std Deviation
SMA(10)	294.433331	63.455000	174.566731	44.452353
WMA(10)	298.428884	62.333334	174.458385	44.619389
MOMENTUM(10)	553.033341	-35.650009	-0.304290	9.423572
%K	100.000000	0.000000	47.446372	35.943790
%D	100.000000	0.000000	47.475261	33.082159
RSI(14)	87.214551	12.543849	49.278377	11.568327
EMA(20)	288.392843	67.475441	174.700765	44.011077
EMA(50)	279.709266	70.281982	175.131335	42.817106
MACD	15.864080	-12.063560	-0.194555	3.526729
EXP	13.080537	-10.030195	-0.188547	3.196583
CCI(14)	327.253289	-325.514335	-5.346346	107.359441
CCI(21)	365.963226	-331.714383	-4.746114	109.983279
CCI(50)	388.004552	-382.778839	-7.866427	114.618867

Table 6: Summary statistics for the selected indicators: HDFC BANK

Indicator	Max	Min	Mean	Std Deviation
SMA(10)	1595.9599865	188.706000	639.923981	368.096277
WMA(10)	1604.796655	188.706000	641.470688	369.093165
MOMENTUM(10)	210.300049	-346.100037	4.4115175	33.510232
%K	100.000000	0.000000	57.994529	36.040605
%D	100.000000	0.000000	58.002700	33.188843
RSI(14)	86.664908	11.428095	54.526852	11.428095
EMA(20)	1559.2241055	365.980622	637.397820	365.980622
EMA(50)	1513.980251	360.192262	630.015943	360.192262
MACD	61.258115	512.568404	3.519025	12.568404
EXP	54.235703	11.345468	3.528668	11.345468
CCI(14)	378.636379	107.874169	21.089384	107.874169
CCI(21)	460.375431	110.070731	27.538794	110.070731
CCI(50)	404.662856	111.413076	41.999341	111.413076

Table 7: Summary statistics for the selected indicators: BOSCH LTD

Indicator	Max	Min	Mean	Std Deviation
SMA(10)	26896.795117	4997.209961	14530.000362	6055.364749
WMA(10)	27256.423828	5049.589876	14540.838493	6053.468321
MOMENTUM(10)	3759.500000	-5510.199218	31.109331	878.794065
%K	100.000000	0.000000	49.415657	36.337594
%D	100.000000	0.000000	49.440404	33.581013
RSI(14)	89.676688	18.092195	51.554647	12.110104
EMA(20)	26431.200721	4987.672578	14511.181251	6045.834634
EMA(50)	25141.734832	4938.967353	14455.420699	6032.516494
MACD	1381.414808	-1339.512459	26.227523	335.235400
EXP	1222.004645	-1043.214107	26.774525	303.278645
CCI(14)	426.160517	-345.158646	2.395266	110.791910
CCI(21)	473.643897	-340.474997	4.730810	114.535708
CCI(50)	692.619048	-383.343305	12.645849	121.044874

6 PREDICTION MODELS

6.1 Naive Bayes Classifier

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

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

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 C1, C2, ..., Cn 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

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

Gaussian Naive Bayes classifier

In Gaussian Naive Bayes, continuous values associated with each feature are assumed to be distributed according to a Gaussian distribution. A Gaussian distribution is also called Normal distribution. When plotted, it gives a bell shaped curve which is symmetric about the mean of the feature values as shown below:

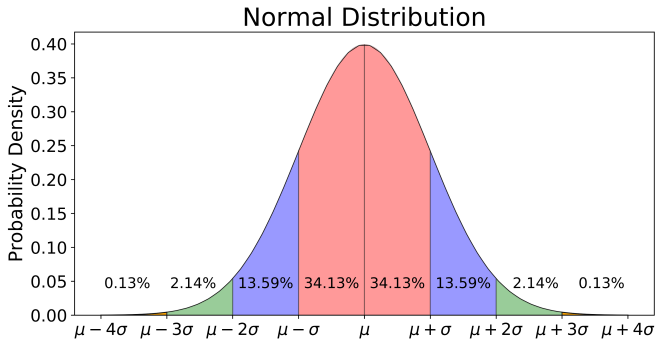

Figure 2: Normal Distribution

6.2 Logistic Regression

Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Unlike linear regression which outputs continuous number values, logistic regression transforms its output using the logistic sigmoid function to return a probability value which can then be mapped to two or more discrete classes.

Sigmoid Activation

In order to map predicted values to probabilities, we use the sigmoid function. The function maps any real value into another value between 0 and 1. In machine learning, we use sigmoid to map predictions to probabilities.

$$s(z) = \frac{1}{1 + e^{-z}}$$

where:

$$s(z) = output\ between\ 0\ and\ 1\ (probability\ estimate)$$

$$z = input\ to\ the\ function\ (your\ algorithm's\ prediction\ e.g.\ mx + b)$$

Our current prediction function returns a probability score between 0 and 1. In order to map this to a discrete class (UP/DOWN or +1/-1), we select a threshold value or tipping point above which we will classify values into class 1 and below which we classify values into class 2.

$$p \geq 0.5,\ \ class = +1$$

$$p \leq 0.5,\ \ class = -1$$

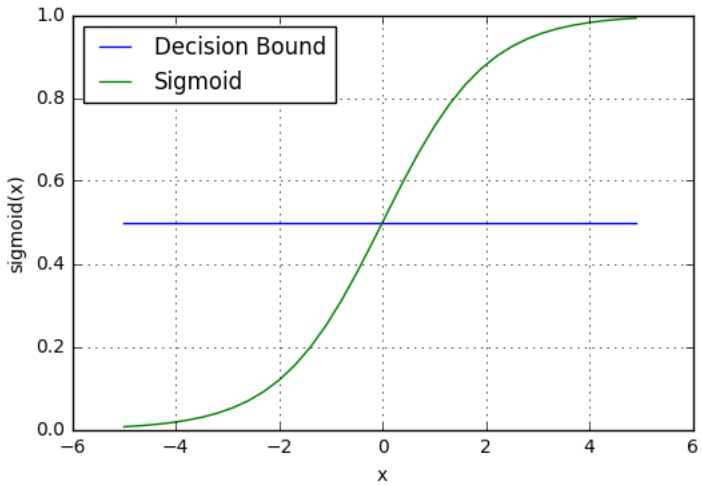

Figure 3: Sigmoid Decision

6.3 Random Forest Classifier

RF classifier is an ensemble method that trains several decision trees in parallel with bootstrapping followed by aggregation, jointly referred as bagging. Bootstrapping indicates that several individual decision trees are trained in parallel on various subsets of the training dataset using different subsets of available features. Bootstrapping ensures that each individual decision tree in the random forest is unique, which reduces the overall variance of the RF classifier. For the final decision, RF classifier aggregates the decisions of individual trees; consequently, RF classifier exhibits good generalization. RF classifier tends to outperform most other classification methods in terms of accuracy without issues of overfitting. Like DT classifier, RF classifier does not need feature scaling. Unlike DT classifier, RF classifier is more robust to the selection of training samples and noise in training dataset. RF classifier is harder to interpret but easier to tune the hyperparameter as compared with DT classifier.

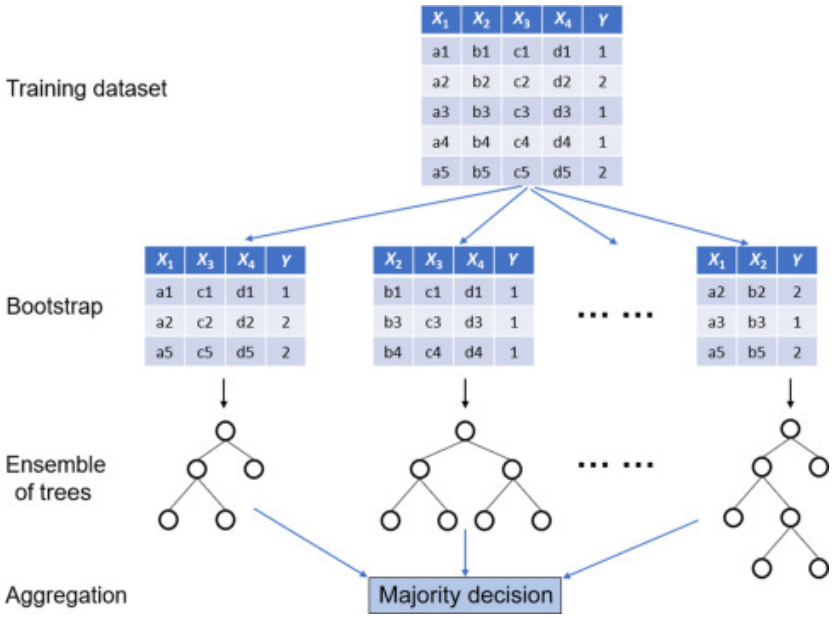

Figure 4: Random Forest Ensemble

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

6.4 Support Vector Machine

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points.

To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. Also, the dimension of the hyperplane depends upon the number of features. If the number of input features is 2, then the hyperplane is just a line. If the number of input features is 3, then the hyperplane becomes a two-dimensional plane. It becomes difficult to imagine when the number of features exceeds 3.

Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane. These are the points that help us build our SVM.

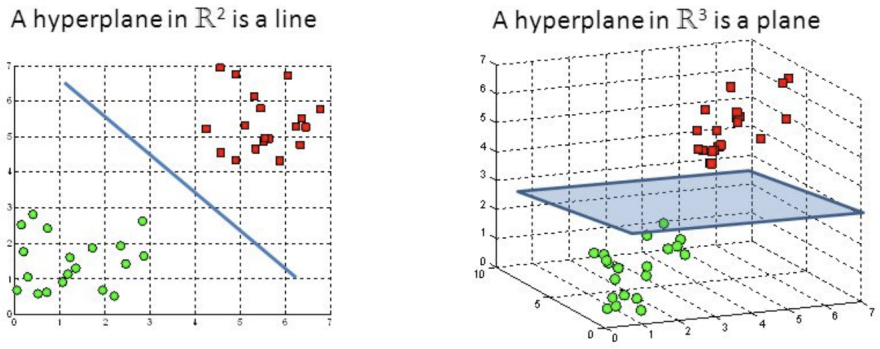

Figure 5: Hyperplanes in 2D and 3D feature space

Hyperparameters: Gamma & C

The gamma parameter defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'. The gamma parameters can be seen as the inverse of the radius of influence of samples selected by the model as support vectors.

The C parameter trades off correct classification of training examples against maximization of the decision function's margin. For larger values of C, a smaller margin will be accepted if the decision function is better at classifying all training points correctly. A lower C will encourage a larger margin, therefore a simpler decision function, at the cost of training accuracy. In other words C behaves as a regularization parameter in the SVM.

6.5 Neural Networks

An ANN is based on a collection of connected units or nodes called artificial neurons. Each connection, can transmit a signal to other neurons. An artificial neuron that receives a signal then processes it and can signal neurons connected to it. The "signal" at a connection is a real number, and the output of each neuron is computed by some non-linear function of the sum of its inputs. The connections are called edges. Neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Neurons may have a threshold such that a signal is sent only if the aggregate signal crosses that threshold. Typically, neurons are aggregated into layers. Different layers may perform different transformations on their inputs. Signals travel from the first layer (the input layer), to the last layer (the output layer), possibly after traversing the layers multiple times.

Connections and weights

The network consists of connections, each connection providing the output of one neuron as an input to another neuron. Each connection is assigned a weight that represents its relative importance. A given neuron can have multiple input and output connections.

Propagation function

The propagation function computes the input to a neuron from the outputs of its predecessor neurons and their connections as a weighted sum. A bias term can be added to the result of the propagation.
A neuron first computes the weighted sum of the inputs.

$$y = \sum_{i=1}^{n} (w_i x_i) + bias$$

where:

n = no. of features

Learning rate

The learning rate defines the size of the corrective steps that the model takes to adjust for errors in each observation. A high learning rate shortens the training time, but with lower ultimate accuracy, while a lower learning rate takes longer, but with the potential for greater accuracy. Optimizations such as Quickprop are primarily aimed at speeding up error minimization, while other improvements mainly try to increase reliability. In order to avoid oscillation inside the network such as alternating connection weights, and to improve the rate of convergence, refinements use an adaptive learning rate that increases or decreases as appropriate. The concept of momentum allows the balance between the gradient and the previous change to be weighted such that the weight adjustment depends to some degree on the previous change. A momentum close to 0 emphasizes the gradient, while a value close to 1 emphasizes the last change.

7 METHODOLOGY

7.1 Block Diagram

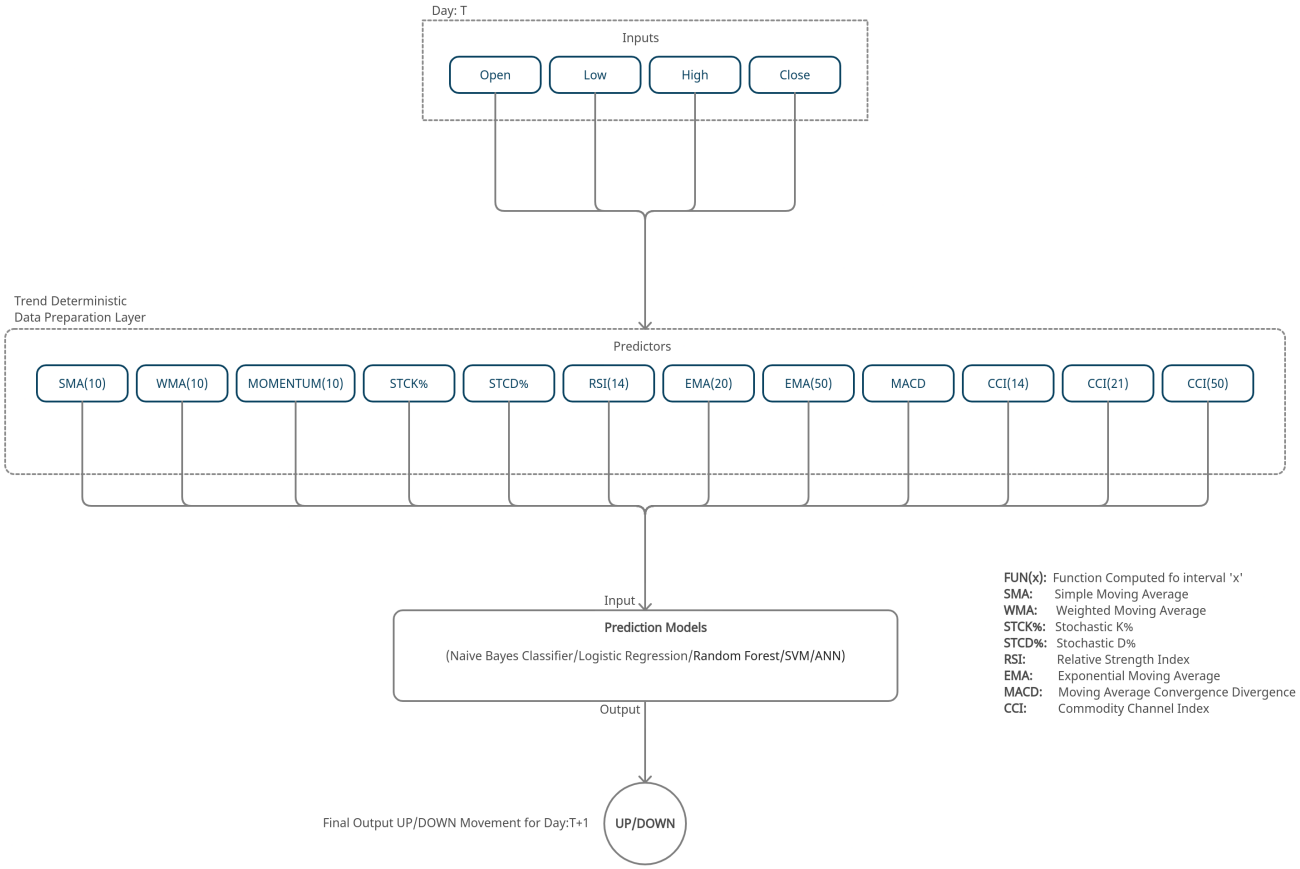

Figure 6: Predicting Movement with Trend Indicators.

8 Results

8.1 Data

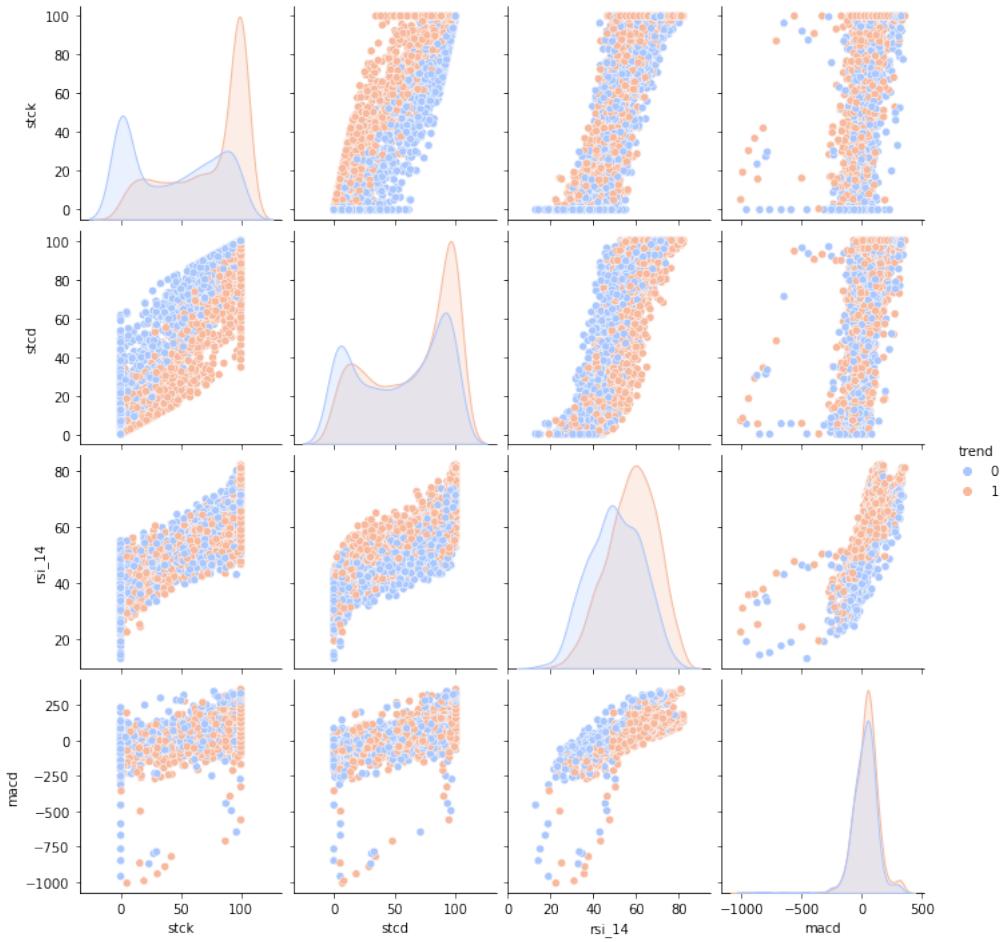

Figure 7: Pair plots of trend indicators for NSEI

8.2 Implementation 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).

$$Precision_{positive} = \frac{TP}{TP + FP}, \qquad Precision_{negative} = \frac{TN}{TN + FN}$$

$$Recall_{positive} = \frac{TP}{TP + FN}, \qquad Recall_{negative} = \frac{TN}{TN + FP}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}, \qquad F\ measure = \frac{2 * Precision * Recall}{Precision + Recall}$$

8.2.1 Naive Bayes Classifier

Table 8: Performance of Naive Bayes Classifier

Class	Precision	Recall	F1 Score	Support
NIFTY 50				
-1	0.60	0.51	0.55	404
+1	0.62	0.70	0.66	469
accuracy			0.61	873
avg	0.61	0.61	0.61	873
ONGC				
-1	0.63	0.64	0.63	457
+1	0.60	0.59	0.65	422
accuracy			0.61	879
avg	0.61	0.61	0.61	879
HDFC BANK				
-1	0.63	0.52	0.57	445
+1	0.58	0.69	0.63	434
accuracy			0.60	879
avg	0.61	0.60	0.60	879
BOSCH LTD				
-1	0.59	0.58	0.58	450
+1	0.57	0.57	0.57	429
accuracy			0.58	879
avg	0.58	0.58	0.58	879

Figure 8: Naive Bayes: Line plot for NSEI

8.2.2 Logistic Regression

Table 9: Performance of Logistic Regression

Class	Precision	Recall	F1 Score	Support
NIFTY 50				
-1	0.86	0.78	0.82	404
+1	0.83	0.89	0.86	469
accuracy			0.84	873
avg	0.84	0.84	0.84	873
ONGC				
-1	0.85	0.84	0.85	457
+1	0.84	0.84	0.84	422
accuracy			0.85	879
avg	0.85	0.85	0.85	879
HDFC BANK				
-1	0.87	0.86	0.87	445
+1	0.86	0.87	0.87	434
accuracy			0.87	879
avg	0.87	0.87	0.87	879
BOSCH LTD				
-1	0.85	0.85	0.85	450
+1	0.85	0.85	0.85	429
accuracy			0.85	879
avg	0.85	0.85	0.85	879

Figure 9: Logistic Regression: Line plot for NSEI

8.2.3 Random Forest Classifier

Table 10: Performance of Random Forest Classifier

Class	Precision	Recall	F1 Score	Support
NIFTY 50				
-1	0.84	0.87	0.86	404
+1	0.89	0.86	0.87	469
accuracy			0.86	873
avg	0.86	0.87	0.86	873
ONGC				
-1	0.85	0.84	0.85	457
+1	0.83	0.5984	0.84	422
accuracy			0.84	879
avg	0.84	0.84	0.84	879
HDFC BANK				
-1	0.87	0.86	0.87	445
+1	0.86	0.87	0.87	434
accuracy			0.87	879
avg	0.87	0.87	0.87	879
BOSCH LTD				
-1	0.81	0.88	0.84	450
+1	0.86	0.78	0.82	429
accuracy			0.83	879
avg	0.83	0.83	0.83	879

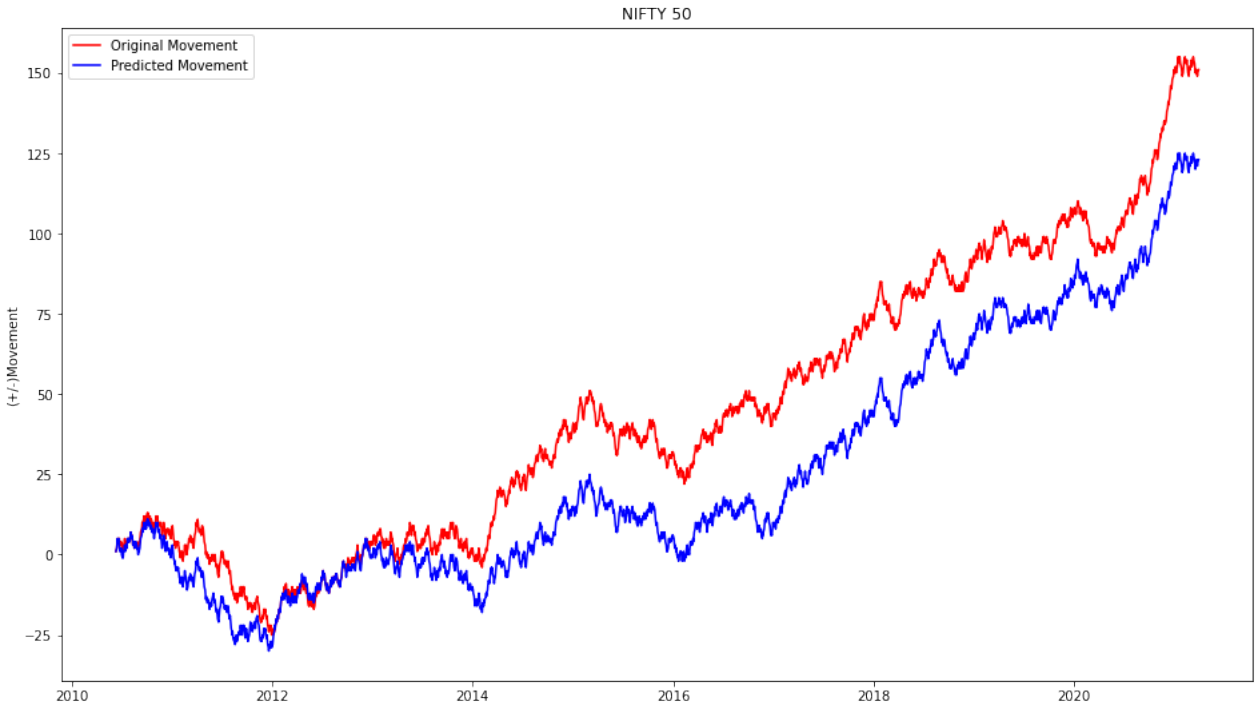

Figure 10: Random Forest Classifier: Line plot for NSEI

8.2.4 Support Vector Machine

Table 11: Performance of Support Vector Machine

Class	Precision	Recall	F1 Score	Support
NIFTY 50				
-1	0.86	0.79	0.82	404
+1	0.83	0.88	0.86	469
accuracy			0.84	873
avg	0.84	0.84	0.84	873
ONGC				
-1	0.85	0.86	0.86	457
+1	0.85	0.84	0.84	422
accuracy			0.85	879
avg	0.85	0.85	0.85	879
HDFC BANK				
-1	0.88	0.86	0.87	445
+1	0.86	0.88	0.87	434
accuracy			0.87	879
avg	0.87	0.87	0.87	879
BOSCH LTD				
-1	0.84	0.83	0.84	450
+1	0.83	0.83	0.83	429
accuracy			0.83	879
avg	0.83	0.83	0.83	879

Figure 11: SVM: Line plot for NSEI

8.2.5 Neural Network

Table 12: Performance of Artificial Neural Network

Class	Precision	Recall	F1 Score	Support
NIFTY 50				
-1	0.84	0.83	0.84	404
+1	0.86	0.86	0.86	469
accuracy			0.85	873
avg	0.85	0.85	0.85	873
ONGC				
-1	0.88	0.85	0.87	457
+1	0.85	0.87	0.86	422
accuracy			0.86	879
avg	0.86	0.86	0.86	879
HDFC BANK				
-1	0.86	0.87	0.86	445
+1	0.86	0.85	0.86	434
accuracy			0.86	879
avg	0.86	0.86	0.86	879
BOSCH LTD				
-1	0.80	0.90	0.84	450
+1	0.87	0.76	0.81	429
accuracy			0.83	879
avg	0.84	0.83	0.83	879

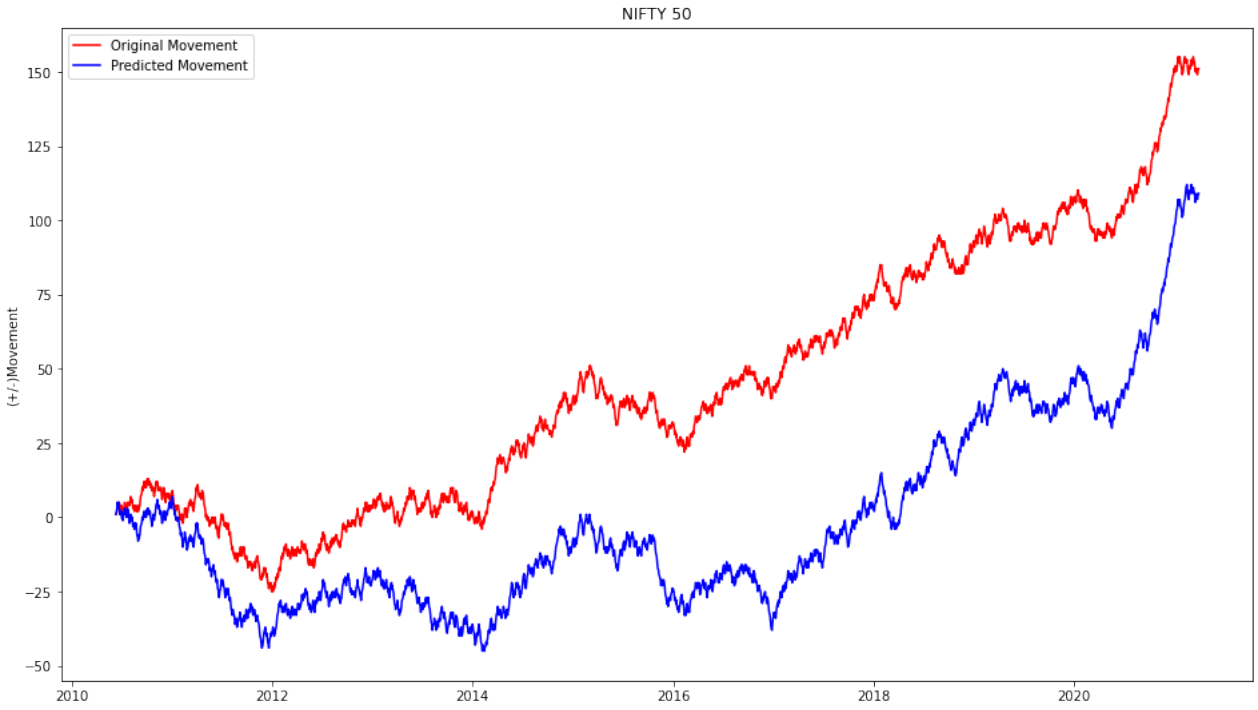

Figure 12: Neural Network: Line plot for NSEI

9 CONCLUSION

In this survey paper we conclude that though various methods and approaches can be used for predicting the stock price, every method has its limitations and advantages. By considering both technical indicators and fundamental analysis the accuracy of the predictions can be made more accurate and reliable. Despite many such algorithms available, there is always room for improvement. It is observed that use of different parameters of the data set in different algorithms results in various accuracy rates. We conclude that the use of Trend deterministic forecasting with the Support Vector Machines and Random Forest Classifier yield in more accurate prediction.

References

[1] A. Elliot, C. Hsu, "Time Series Prediction : Predicting Stock Price, 2018 " *arXiv: Machine Learning, 2018.*

[2] Li-Pang Chen, "Using Machine Learning Algorithms on Prediction of Stock Price" Journal of Modeling and Optimization , vol. 12, no. 2 (2020), ISSN: 1759-7676

[3] Yaping Hao * and Qiang Gao, "Predicting the Trend of Stock Market Index Using the Hybrid Neural Network Based on Multiple Time Scale Feature Learning" MPDI: Applied Science Vol. 10, issue. 11, 2020, April 2020.

[4] Prerana C , Pratheeksha Mahishi J. "Stock market prediction using machine learning an deep learning techniques" International Research Journal of Engineering and Technology (IRJET), vol. 07, issue 04, April 2020

[5] Jaydip Sen, "Stock Price Prediction Using Machine Learning and Deep Learning Frameworks," 6th International Conference on Business Analytics and Intelligence (ICBAI), at Indian Institute of Science, Bangalore, INDIA, December 2018.

[6] Yahoo Finance Website (Dataset): `https://in.finance.yahoo.com`

[7] Investopedia Website: `https://www.investopedia.com/technical-analysis-4689657`

[8] Implementation: `https://github.com/HeptaDecane/IndexPrediction`

attach your review and visit log here.....

attach plagiarism report here.....