Performance Evaluation of Machine Learning Algorithms for Stock Price and Stock Index Movement Prediction Using Trend Deterministic Data Prediction

Munish Khanna, Hindustan College of Science and Technology, India*

Mohak Kulshrestha, Hindustan College of Science and Technology, India

Law K. Singh, Hindustan College of Science and Technology, India

https://orcid.org/0000-0002-7073-6852

Shankar Thawkar, Hindustan College of Science and Technology, India

https://orcid.org/0000-0002-0118-9605

Kapil Shrivastava, Hindustan College of Science and Technology, India

ABSTRACT

This experimental study addresses the problem of predicting the direction of stocks and the movement of stock price indices for three major stocks and stock indices. The proposed approach for processing input data involves the computation of 10 technical indicators using stock trading data. The dataset used for the evaluation of all the prediction models consists of 11 years of historical data from January 2007 to December 2017. The study comprises four prediction models which are long short-term memory, XGBoost, support vector machine, and random forests. Accuracy scores and F1 scores for each of the prediction models have been evaluated using this input approach. Experimental results reveal that a continuous data approach using 10 technical indicators gives the best performance in the case of the random forest classifier model with the highest accuracy of 84.89% (average wise 83.74%) and highest F1 score of 89.33% (average wise 83.74%). The experiments also give us an insight into why a Naïve Bayes classification model is not a suitable prediction model for the above task.

KEYWORDS

Machine Learning Algorithms, Random Forest, Stock Market Prediction, Support Vector Machine, XGBoost Model

1. INTRODUCTION

Everyone wants to earn money in the shortest possible time; stock market can be one of the instruments to fulfill such dreams. The stock market of a developing country like India has grown multi-folds during the last 30 years which is why it attracts plenty of investors. A major chunk of the community assumes that making money from the stock market is child's play. Prediction of the stock price index along with its movement is an exigent task in time series prediction. Many macroeconomic factors

*Corresponding Author

such as firms' policies, investors' expectations, political scenarios, institutional investor's choices, economic conditions, etc. have a significant impact on the way the stock market behaves. Meanwhile, smart investors always consult fundamental analysts or technical analysts before purchasing any script. In the case of fundamental analysis, various factors are taken into consideration before investing such as economic growth, demand and supply, political stability in the country. For a technical analyst, it is not a simple task to forecast script price index and movement due to the uncertainties involved. They focus on factors based on statistics such as volume per day, price movement above/below the daily moving average(DMA), trends and patterns and founded on these they suggest future movement. A study by (Malkiel & Fama,1970) suggests that data prediction of the stock price is possible based on trading. Various factors such as political circumstances, the image of the management of the company and purchasing/selling of the own script by management are generally reflected in the prices. Thus, if the information of previous stock prices can be proficiently pre-processed and if suitable algorithms are applied, and then it is possible to predict the trend of a stock or stock price index.

Several studies have already been conducted in the past by various researchers regarding the pricing of different stock market and stock index financial instruments. However, one factor that all such studies seem to exempt from their research is an examination of the predictability of the direction of the stock market movement. Predicting the direction of the stock market in the near future is of great importance and value to several types of stock market researchers and investors. Predicting the track of the stock market movement in the near future accurately gives numerous insights about shares to invest in, current market conditions, the economy of a region etc.

There are various studies already conducted dealing with stock direction prediction with the help of Artificial Neural Networks (ANN), specifically with the use of Backpropagation ANN referred to as BNN. But these studies were not able to predict the direction of market movement with much accuracy. One significant reason for it was that the BNN model does not take into account past trends of the stock market while making predictions. To overcome this shortcoming, the presented study makes use of an improvement in ANN models that is LSTM. This is a type of Recurrent Neural Network which has a memory element that helps in taking into account previous trends while making future predictions, thus improving the accuracy score.

XGBoost is a relatively new technique for ensemble machine learning. However, because of its performance and speed, it is gaining popularity among researchers and data scientists. It improves on the basic gradient boosting method.

Machine learning techniques such as Support Vector Machine (SVM) and Random Forest have proven to be extremely effective in time series forecasting, particularly in stock market prediction. These techniques are focused on developing accurate predictions for some variables given other variables. On the other hand, solutions generated by neural network models may be local optimal, whereas solutions generated by SVM and random forest may be global optimal.

As a case study for evaluating the performances of various shortlisted machine learning algorithms, standard data from reputable stocks and indexes was selected. Apple, Amazon, and Google are among the FAANG Stocks, an acronym for the world's five technology behemoths that trade publicly on the market. The S&P 500, Dow Jones Industrial Average, and Nasdaq Composite are the three most widely followed indices in the United States, accounting for a significant portion of the US equity market.

We used classical, effective and latest machine-learning algorithms to validate the originality of this work for stock market prediction. We focused on three well-known shares of world-renowned businesses and three US stock market indices for 11 years. The input data involves the computation of ten technical indicators using stock trading data. As we note, to date very little work has been done on this combination (11 years of data of 3 reputed stocks and 3 reputed indices, 10 technical indicators and 4 machine learning algorithms), which justifies the uniqueness of the work we have done. Furthermore, one factor that most previous studies appear to overlook is the predictability of stock market movement direction, which we also worked on, which justifies the novelty.

The paper is divided into the sections listed below. Section 2 contains a report on a review of prior published studies' literature. This section also includes a detailed description of five similar studies. Section 3 discusses the research data as well as preprocessing. This Section also discusses the ten shortlisted technical parameters. The prediction models are described in detail in Section 4. The results of the experiments are presented in Section 5. Finally, in Section 6, the paper concludes by highlighting some of the limitations encountered and future directions.

2. LITERATURE REVIEW

Several techniques for predicting stock trends have been proposed by the research community over the last decade or so, beginning with classical regression methods. We are aware that stock data can also be classified as non-stationary time series data; thus, non-linear machine learning techniques have been used extensively for this task. To date, many techniques have been developed to predict stock trends as accurately as possible. Later, SVM and ANN were widely accepted and used by practitioners to forecast stock and stock price index movement.

Many recent studies have justified the effectiveness of machine (and deep) learning approaches to solve problems in different domains and have produced stupendous results (Hu & Razmjooy, 2020;. Yu, Wang, Liu, Jermsittiparsert & Razmjooy,2019 ; Razmjooy, Ashourian, Karimifard, Estrela, Loschi, do Nascimento, ... & Vishnevski, 2020; Hu, & Razmjooy, 2020; Deshpande, Patavardhan, Estrela, Razmjooy, & Hemanth,2020; Singh, Garg, & Khanna, 2021; Singh, Garg, Khanna, & Bhadoria,2021).

(Hassan, Nath, & Kirley, 2007) presented financial market behavior forecasting using a model based on the Hidden Markov Model, ANN, and Genetic Algorithm.(Wang & Leu, 1996) proposed a system, based on recurrent neural network, for mid-term price trend forecasting of the Taiwan stock market.(Abraham, Nath & Mahanti,2001) selected the Nasdaq-100 index as the subject for the next day's stock forecasting, using a neural network and neuro-fuzzy system for conducting trend analysis of predicted stock values. (Chen, Leung, & Daouk, 2003) trained Probabilistic neural network using historical data index direction forecasting and the results were promising as compared to other investment strategies like buy-and-hold.

Another algorithm that was widely accepted by practitioners is the SVM algorithm proposed by (Vapnik, 1999), a learning algorithm which searches for the hyper plane to categorize classes, (Huang, Nakamori & Wang,2005) predicted the weekly movement direction of the NIKKEI 225 index with the support of SVM. The performance of SVM was promising when compared with other classification methodologies. In another study by (Kim,2003), the authors proved that SVM outperforms back-propagation neural networks and case-based reasoning while predicting daily stock price change in the case of the KOSPI stock exchange.

In this paper (Hiransha, Gopalakrishnan, Menon, & Soman, 2018), four categories were used to forecast an existing business market price using deep learning models such as Multilayer Perceptron (MLP), recurring neural networks (RNN), long short term memory (LSTM) and revolutionary neural network (CNN). Here, two competing financial exchanges-the NSE of India and the New York Stock Exchange-use day-to-day closing prices (NYSE). It was trained on the market price of a single NSE firm and was forecast for five NSE and NYSE firms. CNN outperforms the other models, it was noted. Since NSE data was learned, the Network could predict the NYSE.

This article (Moghar & Hamiche, 2020) seeks to develop a model for prediction of potential stock market prices by using Recurrent Neural Networks, in particular the Long-Short Memory model (LSTM). This paper is mostly aimed at seeing the accuracy of the algorithm of machine learning, and how much the epochs might enhance their model. This paper suggests RNN based on the LSTM, which is intended to predict potential prices, both for GOOGL and NKE assets. The results of the tests adhere to the model proposed to monitor opening prices for both assets.

In this study (Vijh, Chandola, Tikkiwal, & Kumar, 2020) the Artificial Neural Network and Random Forest techniques were used to estimate the closing price next day for 5 firms from various

operating sectors. The financial figures: Open, high, low and close market values were used to create new variables that were used as model inputs. Models were assessed by means of common strategic indicators: MAPE and RMSE. The lower values of these two metrics indicate that the models predicted market closing prices effectively.

In this report (Nti, Adekoya, & Weyori, 2020), a new homogenous ensemble classification named GASVM based on Genetic Algorithm (GA) enhanced supporting vector machine for the selection of functions and SVM kernel optimization for stock market prediction is proposed.. In this analysis, the GA was implemented in order to optimise the various design variables of the SVM at the same time. Experiments conducted with Ghana Stock Exchange (GSE) data for eleven (11) years generated impressive results. The findings indicate that, by forecasting a 10-day inventory flow, other classical ML algorithms (Decision Tree (DT), Random Forest (RF) and Neural Network (NN)) were outperformed in the proposed novel "homogenous" ensemble classification (named GASVM). Increased predictability was achieved with the suggested (GASVM) of 93.7% relative to 82.3% (RF), 75.3% (DT), and 80.1%. (NN).

The goal of this article (Pang, Zhou, Wang, Lin, & Chang, 2020) was to create a groundbreaking approach to neural networks to help forecast stocks. Authors explain the idea of "stock vector," based on the evolution of word vector in deep learning. The input is no longer a single index or stock index, but multi-store historical records. In order to forecast the stock market, it proposed the deep long-term, integrated layer neural memory network (LSTM) and the long-term stock neural memory network with automated encoder system. In both models, practitioners used the embedded layer and the automated encoder to vectorise the input, respectively, to predict the inventory via the long short-term memory neural network. The experimental findings have shown that the LSTM is deep with a built-in layer. For the Shanghai A-share composite index, the accuracy of both versions is respectively 57.2 and 56.9%.

They (Mondal, Dutta & Chatterjee,2020) have demonstrated both machine learning techniques and the concept of Long Short-Term Memory (LSTM) memory in this script. They have used daily data from 2013 to prepare a training model, and apply it to all future months up to 2019 on the set of upcoming NSE listed stocks to make the resulting closing values. Logistic, artificial neural networks, and machine learning are assisting in diagnosis improvement (Bagging, Boosting).Using an LSTM, they built the model for the closing prices of the stocks, and then applied the accuracy on top of the models. It has been shown that deep learning and machine learning models combined with the whole machine learning algorithms have fantastic stock-picking results that support investments. The small training problem of gradient movement towards zero (also known as a gradient phenomenon) with back-propagitation (called exploding gradient). The system reduces this difficulty by only reducing LSTM computational time.

In this study, authors(Nti, Adekoya, & Weyori, 2020) have tried to compare ensemble techniques such as boost, bagging, mixing, and super learning extensively (stacking). They have built 25 regressors and classificators using Decision Trees (DT), Support Vector Machine (SVM) and Neural Network (NN). From January to December 2018, they compared the implementation time, accuracy and bug measurements of Ghana Stock Exchange (GSE), Johannesburg Stock Exchange (JSE), Bombay Stock Exchange (BSE-SENSEX) and New York Stock Exchange (NYSE). The results of the study reveal that the stacks and mixings combinational techniques are more accurate (90–100%) and (85.7–100%) compared with baggage techniques (53–97.78%) and boosting techniques (52.7–96.32%).

In order to make prediction easier and true, machine learning itself uses different models. The paper focuses on the use of machinery based on regression and LSTM to predict stock values. Open, close, low, high and volume factors are considered. In this paper (Parmar, Agarwal, Saxena, Arora, Gupta, Dhiman & Chouhan, 2018) two techniques were used: LSTM and Regression on the Yahoo financial information dataset. Both techniques have demonstrated an increase in predictive accuracy and have produced positive results.

In the empirical study, investigators (Shen, Jiang, & Zhang, 2012) suggested using data from various global financial markets with algorithms for machine learning to predict movements in stock indexes. In order to forecast the future stock trend with help from the SVM, they proposed a new prediction algorithm using the temporal link between the world stock markets and various financial products. A prediction accuracy of 74.4% for NASDAQ, 76% for S & P500 and 77.6% for DJIA is indicated by numerical results. The same algorithm was used for tracking the actual market increase with different regression algorithms. In this report (Nandhini, Bari, & Pradip, 2020), the authors discuss the development and implementation of an algorithm for the prediction of stock market prices. To solve this problem, they follow three different approaches: Fundamental analysis, technical analysis and machine learning application.

The subsequent shortlisted algorithm is a random forest algorithm which creates classification trees on the basis of a provided sample and then predicts the class on the basis of the prediction of the majority of trees. (Tsai, Lin, Yen & Chen, 2011) proposed prediction presentation of stock returns using classifier ensembles. (Sun & Li, 2012) in their study presented a novel SVM ensemble based on Financial Distress Prediction (FDP); it was also empirically justified that the SVM ensemble was promising when compared with the individual SVM classifier. (Ou & Wang, 2009) compared the performance of ten techniques for predicting the price movement of the Hang Seng index (Hong Kong stock market); authors experimentally justify that the performance of SVM and Least square support –SVM (LS-SVM) was most promising.(Ghosh, Sanyal & Jana,2018) proposed a framework to accomplish experimental evaluation and carry out predictive modeling on daily index prices of various stock exchanges. They inspect the equity markets thoroughly to verify whether they follow pure random walk models or not.

It can be said that every algorithm, along with its constraints, has its own way of solving the problem in hand. The final outcome, in the form of prediction, is dependent on the applied algorithm as well as representation of the input.

Along with the above discussion on prior published studies, we have also shortlisted five studies whose key-point description is as follows.

The first selected study (Patel, Thakkar & Kotecha,2015), published in a reputed journal, also works on the prediction of stock and stock price index movement using machine learning techniques. This paper throws light on the problems and techniques used in predicting stock prices and index movement for Indian Stock Markets. The study makes use of four prediction models, namely, Artificial Neural Network (ANN), support vector machine (SVM), Random Forest and Naive-Bayes. The major approach proposed by this paper for stock and stock price index movement is the method of passing input data to the classifier models used. The first approach for input data involves computation of ten technical parameters using stock trading data (open, high, low & close prices). In this approach, the dataset used is first converted into the form of technical indicators such as simple n-day moving average, weighted n-day moving average etc. The Architecture of the ANN model implemented in the study (Patel, Thakkar & Kotecha,2015) can be referred to with the help of Figure 1. The input data thus obtained from this approach is continuous valued and is normalized over the range of [-1,1]. After this, the technical indicators form the input for the classifier models used. The second approach for input takes the previous approach one step forward. In this approach, a new layer of decision is employed that converts continuous valued technical parameters to discrete values, representing the trend. This layer is referred to as the "Trend Deterministic Data Preparation Layer". This layer's main task is to convert the continuous value indicators obtained from the first input approach to discrete values of '+1' or '-1'. This conversion is done on the basis of predefined criteria and conditions which vary from indicator to indicator. In this input approach, input data is passed in the form of discrete data instead of continuous data. The researchers achieved the highest average accuracy of 83.59% using Random Forest classifier on continuous data. An interesting point to be noted here is that the dataset they used (as mentioned) consisted of Reliance Industries, Infosys Ltd, CNX Nifty and BSE Sensex, for the period of 2003-2012. However, if we look at this data, which can be obtained from

the source they mentioned, it consists of many Null and NAN values which are a sign of possible data leakage. Another point is, nowhere in their paper have they mentioned the duration for which the prediction is made, i.e., the number of days ahead for which the stock and stock price index movement direction is predicted. The Result Analysis section discusses the important role that prediction time period plays in such cases.

The second shortlisted study (Kara, Boyacioglu & Baykan, 2011) throws light on the prediction of the direction of stock price movement using ANN and SVM. This work focuses on the study of the Istanbul Stock Exchange (ISE) for the purpose of predicting the direction of stock price movement. It was an emerging market and can give several insights about prediction techniques to use for such markets as it is characterized by high volatility in market returns. This study makes an attempt to create two models which can be efficient for stock market direction prediction and compares their performance in predicting the direction of movement in the daily ISE National 100 Index. The models applied in this work are based on two accepted classification techniques, ANN and SVM (Figure 2). The research data used in this paper is the direction of daily closing price movement in the ISE National 100 Index. It covers the time period from January 2, 1997 to December 31, 2007 and covers a total of 2733 days. In this study, the input data used comprises of 10 technical indicators calculated using the ISE stock exchange data. The target vector for training is prepared in the form of 0 & 1. They achieved an average accuracy of 71.52% using the SVM model, with the parameter settings as d = 3,/2.5/3.2, c = 100. The highest accuracy achieved in their study was 75.74% using Back Propagation Network of ANN. The parameter settings they used for this model were ep = 6000, mc = 0.4, n = 90.

Figure 1. Architecture of ANN Model (Kara, Boyacioglu & Baykan, 2011)
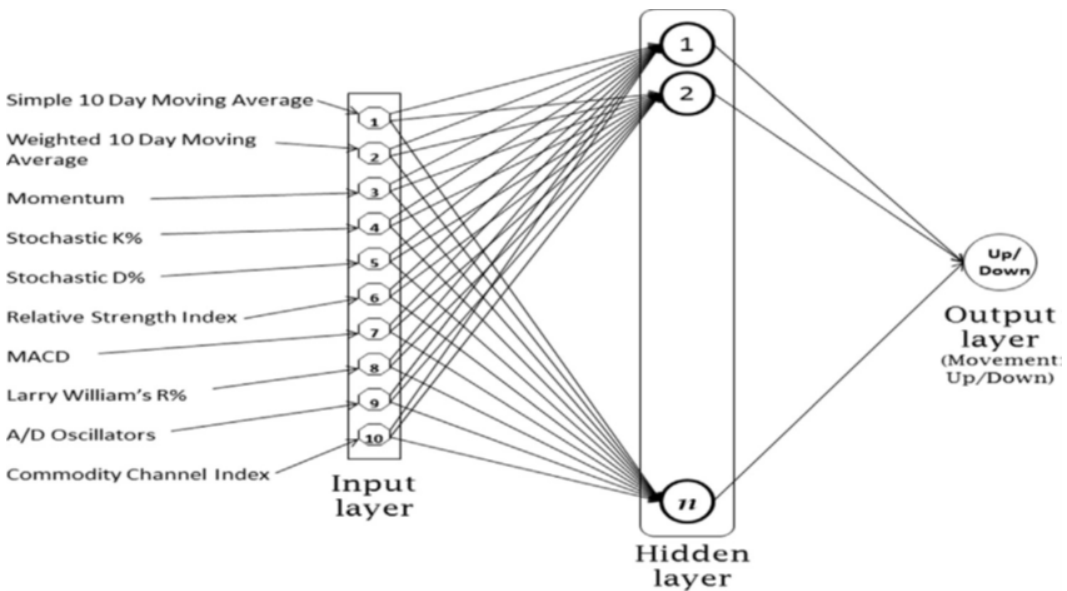
The third selected study (Diler, 2003) focused on predicting the direction of the ISE national-100 index with back propagation trained neural network. The paper's main aim was to forecast the direction of the ISE National 100 Index for the following day or 1-day ahead using ANN Back propagation Network (BNN). The research used six technical indicators as input data to the BNN. The dataset used for the research ranged over the period from Jan 1990 to Nov 2004. A total of 16 experiments were conducted in this research using different values of parameters such as number of neurons in the

Figure 2. A Separating Hyperplane in the feature space of SVM (Hua & Sun, 2001).
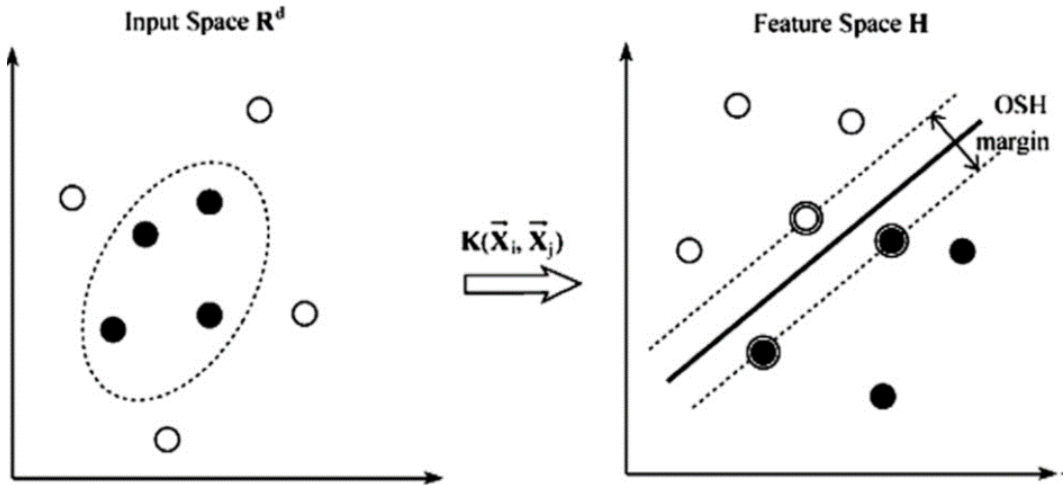
input & output layer and the learning rate. The best result of 60.81% was obtained using the number of neurons in the input layer = 10, number of neurons in the output layer = 1 and learning rate = 0.08.

The fourth shortlisted study (Di, 2014) deals with prediction of the stock price trend of a company or an organization for the very near future. The study was concerned with the potential movement of past prices using feature space derived from the time series of the stock itself rather than some other approaches which are dependent on the company's fundamental analysis. This study takes the approach of analyzing the time series indicators as features for classifying market trends (Up or Down), by focusing on short term (1-10 days) prediction of stock trends. RBF-kernelized SVM parameters were passed through Cross Validated Grid Search in order to fit the training data to balance bias and variances. Data used in this study consisted of three stocks (AAPL (Apple), MSFT (Microsoft), AMZN (Amazon)) and two stock indices (NASDAQ and S & P 500). The data ranged over a time period of 2010-01-04 to 2014-12-10. A total of 12 technical indicators were used to form the input data from the given dataset. These indicators were selected in such a way that they covered various market trend aspects in order to make the predictions more generalized and dependable. The results showed that for Apple, the prediction was around 70% on average across all days, for Amazon, better accuracies were achieved when a longer period (10-days) prediction was taken into account and came in at 71.25%, and for Microsoft, accuracies reached 77% when 7-days and 10-days ahead prediction was done.

Already, there is much research related to stock market prediction as well as prediction of pricing of stock index financial movements. But most of these aim mainly to forecast levels of the underlying stock index. However, not many studies have been conducted regarding predictability of the direction of stock index movement. The last shortlisted paper (Kumar & Thenmozhi, 2006) makes an attempt to predict the direction of the S & P CNX NIFTY Market Index of NSE. Techniques used to accomplish this task are Random Forest and Support Vector Machines. The research data used in this study ranged from January 2000 to May 2005, totaling 1360 trading days. The study then selected 12 technical indicators calculated from the research data, as the input to all training models used. This study achieved the best result of 68.44% accuracy using an SVM model with c=1 and δ^2=70. The next best result was obtained using Random Forest Classifier with an accuracy score of 67.40%.

3. RESEARCH DATA

This section describes the research data collected for this study along with shortlisted technical indicators.

In this empirical study, we have used 11 years of historical data of a total six different stocks and stock indices. All the data used in the project have been collected from https://in.finance.yahoo.com/. The data used is initially in the format of (open, high, low, close, volume). Then it is pre-processed in order to remove any rows that may contain NULL or NAN values. Table 1and Table 2 show how data looks after the pre-processing step.

Table 1. Apple Stock Data

```
In [12]: stocks['AAPL'].head()

Out[12]:        Date      Open        High       Low       Close     Adj Close   Volume

         0  2007-01-03  12.327143  12.368571  11.700000  11.971429  10.609749  309579900

         1  2007-01-04  12.007143  12.278571  11.974286  12.237143  10.845238  211815100

         2  2007-01-05  12.252857  12.314285  12.057143  12.150000  10.768006  208685400

         3  2007-01-08  12.280000  12.361428  12.182858  12.210000  10.821184  199276700

         4  2007-01-09  12.350000  13.282857  12.164286  13.224286  11.720100  837324600
```

Table 2. S&P 500 Stock Index Data

```
In [10]: indices['GSPC'].head()

Out[10]:        Date        Open          High          Low          Close      Adj Close      Volume

         0  2007-01-03  1418.030029  1429.420044  1407.859985  1416.599976  1416.599976  3429160000

         1  2007-01-04  1416.599976  1421.839966  1408.430054  1418.339966  1418.339966  3004460000

         2  2007-01-05  1418.339966  1418.339966  1405.750000  1409.709961  1409.709961  2919400000

         3  2007-01-08  1409.260010  1414.979980  1403.969971  1412.839966  1412.839966  2763340000

         4  2007-01-09  1412.839966  1415.609985  1405.420044  1412.109985  1412.109985  3038380000
```

But before proceeding with training the data, ensuring whether the data are balanced is an important issue. Figure 3 and Figure 4 show the percentage of positive returns instances for each day and for each stock. Fortunately, the data did not need to be balanced since they were almost evenly split for all the stocks.

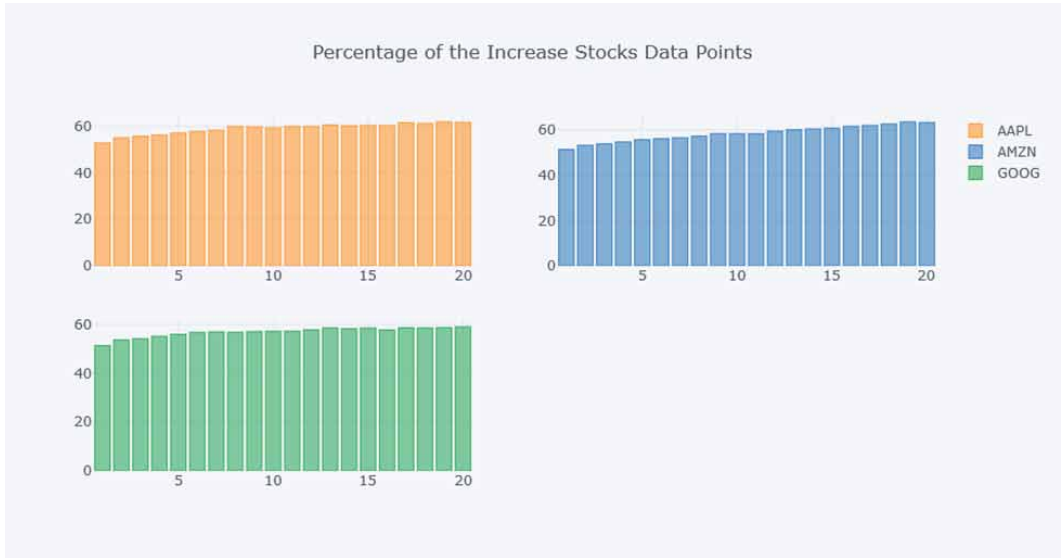**Figure 3. Percentage of Increase Stock Data Points**
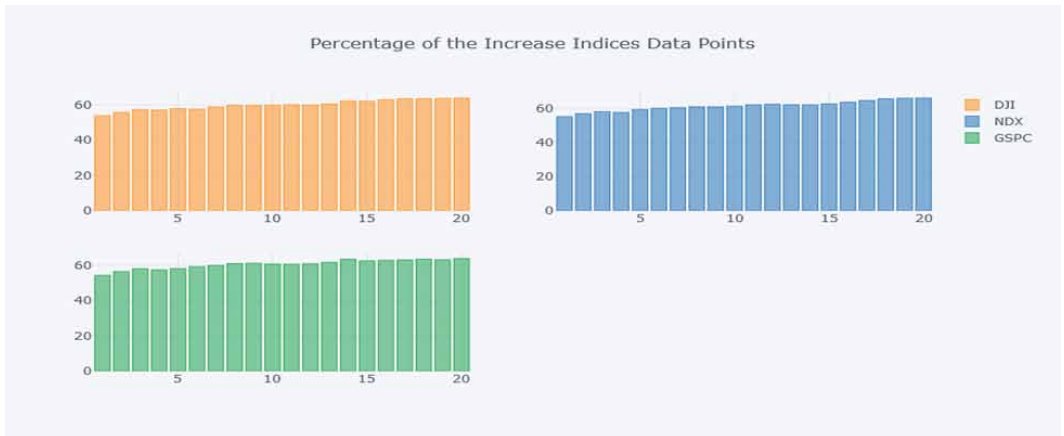

Figure 4. Percentage of Increase Indices Data Points

Table 3. Abbreviations of various stocks and indices used in this paper.

Serial Number	Short form used in figure	Actual name
1.	AAPL	Apple
2.	NDX	Nasdaq
3.	GSPC	S&P 500
4.	DJI	Dow Jones Index
5.	AMZN	Amazon
6.	GOOG	Google

In Figure 3 and Figure 4, the X-axis represents number of days while Y-axis represents increase in stock prices. Table 3providesthe expanded form of abbreviations used in the above as well as upcoming Figures.

The data obtained after the pre-processing step was then subjected to a series of mathematical calculations to yield ten technical indicators, which were then used as input data or training data for the four prediction models.

However, before we proceed, we must first understand what Technical Indicators are and which technical indicators have been selected for this experimental study. Technical indicators are mathematical or heuristic calculations based on a security's or contract's price, volume, or open interest that are used by traders who use technical analysis. Investment decisions can be made based on these parameters, and it helps to forecast price changes. In this study, ten technical indicators covering various aspects of stock market prediction were chosen. Table 4 shows the technical indicators used in this experimental study, as well as the calculation method. Figure 5 is drawn to represent the complete layout of the work.

4. PREDICTION MODELS

This section is divided into five subsections. Each subsection describes one of the implemented prediction models in detail, along with its parameters tuning, if any.

We first reviewed several machine learning-based prediction models as potential candidates for short listing, and then we carefully checked Google scholar for the most recent and best performing classifiers used by researchers in the last four years. Our preference for the best models was not restricted to the best stock market prediction models; they had to have done well in other realms prediction problems as well, so that we could shortlist scalable prediction models. We reviewed 100 recent studies and chose 20 (five for each model) where we observed and found justifications for short listing these four models. For selecting XGboost we have reviewed(Bhattacharya, Maddikunta, Kaluri, Singh, Gadekallu, Alazab, & Tariq, 2020; Parsa, Movahedi, Taghipour,Derrible, Mohammadian,, 2020; Gumelar, Setyorini, Adi, Nilowardono, Widodo, Wibowo,, ... & Christine, 2020; Zolotareva, 2021; Podasca, 2021), and for short listing LSTM we have studied (Bukhari, Raja, Sulaiman, Islam, Shoaib, & Kumam, 2020;Chimmula, & Zhang, 2020; Shahid, Zameer, & Muneeb, 2020;Wang, Xuan, Zhen, Li, Wang, & Shi,2020; Ding, & Qin, 2020).

Similarly for selecting remaining 2 machine learning based prediction models, SVM and Random Forests, we have gone through 10 more recently published studies(Liu, Dang, & Yu, 2020; Nti, Adekoya, & Weyori, 2020;Vignesh,2020;Xiao, Zhu, Huang, Yang, Wen, & Zhong, 2019;NAHIL, & Lyhyaoui, 2017; Sadorsky, 2021;Loke, 2017; Zhang, Cui, Xu, Li, & Li, 2018;Nikou, Mansourfar, Bagherzadeh, 2019; Basak, Kar, Saha, Khaidem, & Dey, 2019).

4.1 Random Forest

Ensemble learning is the technique where multiple models are used together and decision is made by taking into consideration the results of all the models, and applying some suitable operation such as majority or mean, depending upon whether classification or regression is being performed.

All ensemble learning methods are based on Weak Law of Large Numbers (WLLN), which means that if each participating member is able to predict the correct answer with certain accuracy slightly more than 50%, and if each of them can be made independent, then as we take more and more members, accuracy of the prediction made with majority of decisions reaches towards 100%.

Some popular approaches are Bagging, Boosting, Stacking and Pasting. One of most popular approaches is Random Forest, which comes under the category of Bagging, a bootstrap aggregation i.e., repeatedly creating samples over data points or features.

An ensemble of multiple decision trees, which does sampling over feature space rather than all available samples, is one of the best techniques considered right now in machine learning literature.

Table 4. Calculation of Technical Indicators

Indicator Name	Formula	Parameters
5-days and 10-days Simple Moving Average(SMA(5), SMA(10))	$$\dfrac{P_t + P_{t-1} + ... + P_{t-(n-1)}}{n}$$	$n = 5,$ $n = 10$
5-days and 10-days Weighted Moving Average(WMA(5), WMA(10))	$$\dfrac{nP_t + (n-1)P_{t-1} + ... + P_{t-(n-1)}}{n + (n-1) + ... + 1}$$	$n = 5,$ $n = 10$
Stocastic % k	$$\dfrac{P_t - LL_n}{HH_n - LLn} \times 100$$	$n = 5$
Stocastic % D	$$\dfrac{\sum_{i=0}^{n-1} \%k_{t-1}}{n}$$	$n = 3$
Moving Average Convergence Divergence(MACD)	$EMA(n_{fast})_t - EMA(n_{slow})_t$	$n_{fast} = 12,$ $n_{slow} = 26$
Commodity Channel Index(CCI)	$$\dfrac{\dfrac{P + H + C}{3} - SMA(n)_t}{0.015 \times \sigma(n)_t}$$	$n = 14$
10-days Momentum	$P_t + P_n$	$n = 10$
Relative Strength Index(RSI)	$$100 - \dfrac{100}{1 + \dfrac{sum \quad of \quad gains \quad over \quad the \, past \quad n \quad days}{sum \quad of \quad losses \quad over \quad the \quad past \quad n \quad days}}$$	$n = 14$
Williams %R	$$\dfrac{H_n - H_t}{H_n - L_n} \times 100$$	$n = 14$
Chalkin A/D Oscillator	$EMA(n_{fast})_t \; of \;\; A/D \;\; Line - EMA(n_{slow})_t \;\; of \;\; A/D \;\; Line$	$n_{fast} = 3,$ $n_{slow} = 10$

This technique aids in reducing overfitting caused by feature selection. Because each component of Random forest is a simple decision tree, complex decision boundaries cannot be formed with the individual decision tree, but ensemble can assist in determining them.

Because we already have a subset of features during best feature selection, this behaviour trades higher bias for lower variance. Random forests are not only the best machine learning algorithms in their class; they can also be used to feature select by calculating feature importance, which is nothing more than the average height of features across all decision trees created within the ensemble of random forests.

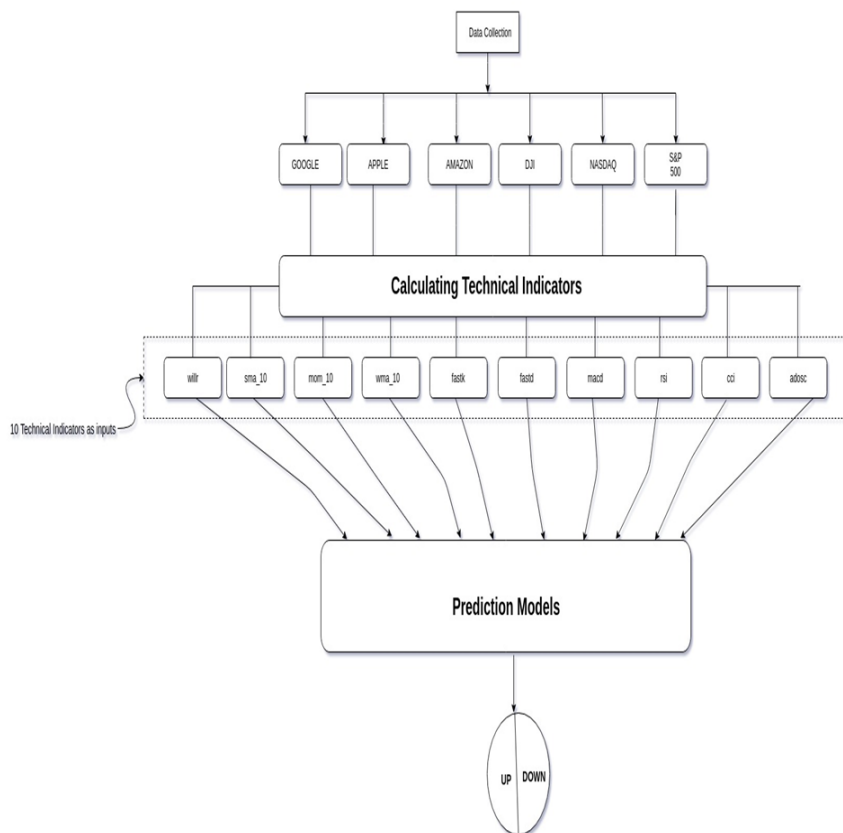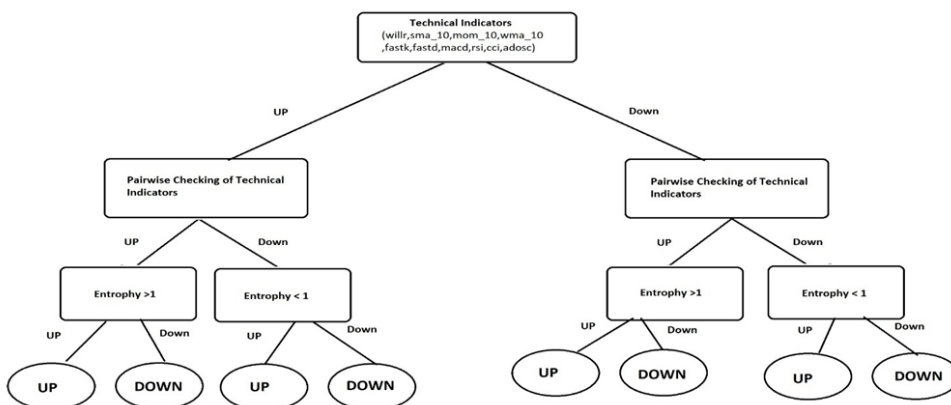**Figure 5. Complete Layout of the implemented work.**

Figure 6. Random Forest Classifier Flow-Chart

Random Forest pseudocode:

1. Randomly select **m** features from total **n** features, where **m << n.**
2. Among the **m** features, calculate node **"d"** using the best split point.
3. Split the node into **daughter nodes** using **best split**.
4. Repeat steps **1 to 3** until one number of nodes has been reached.
5. Build forest by repeating steps **1 to 4** for **k** times to create **k-number of trees**.

The algorithm begins with selection of **m** features from **n** features. Later on, the algorithm uses the randomly selected **m** features to find the root node by using the best split approach. In the next stage, the algorithm calculates daughter nodes using the same best split approach. The third stage is repeated until the algorithm forms the tree with a root node and having the target as the leaf node(Figure 6).

Finally, 1 to 4 stages are repeated to create **k** randomly created trees. The randomly created trees form the **random forest.**

4.1.1 Splitting Criteria

For classification tasks, Random Forest uses **Gini criterion**(Equation 1)

$$Gini = nL\sum k = 1,..,kpkL(1-pkL) + nR\sum k = 1,...,kpkR(1-pkR) \tag{1}$$

where,
pkL = proportion of class k in left node.
pkR = proportion of class k in right node.
In this classification model, the most important step is selecting the **number of trees 'k'**. The selection of k largely depends on the type of data used and the size of data. In this study, experiments were conducted taking the value of n from 350-500, increasing the value by 10 in each iteration. It was found out that best results were achieved when *k=450 for stocks and k=500 for stock indices*.

4.2 Support Vector Machines (SVM)

Support vector machines were developed by Vladimir Vapnik(Vapnik, 1999), and several other researchers in the 1960s. The SVM model is fundamentally a linear classifier whose idea is to select the best linear classification boundary possible, when multiple boundaries exist (which is not so uncommon). SVM was initially developed for linear classification, but the same general ideas with slight modifications can now be extended to non-linear classification and regression. Sometimes people refer to it as "Large Margin Classifier", due to the fact that the best linear decision boundary obtained in SVM provides maximum separation between two classes. SVM models were popular due to the quality of results, performance on medium sized datasets and support on CPU based architectures. Support vectors, as the name suggests, are instances which lie closest to the decision boundary/boundaries. SVM is generally applied in classification, recognition, regression and time series. Support Vector Machines work on the concept of constructing a hyperplane as a decision surface, in order to maximize the margin of separation between positive and negative examples.

There are certain extensions to the basic idea of SVM, known as Hard (strictly separable classes) & Soft margins (Error tolerance in separability), Linear & Kernelized versions. The core idea of SVM, which is popular nowadays, is that it relies on the data being linearly separable when projected in some higher dimensional space, which is controlled by the Kernel.

There are certain issues with SVMs; some of them are:

1. SVMs are sensitive to feature scaling; hence data must be pre-processed.

2. Time complexity is Quadratic to Cubic, depending upon whether kernel trick is used or not, in terms of number of samples, which is difficult to execute for very large datasets.
3. Choice of suitable kernel and experimentations needs to be performed if we have no prior information about datasets.
4. Curse of dimensionality affects SVMs, as it is based on distance based metrics.

In this classification model, the main criteria on which performance of the algorithm depends is selection of hyper plane (or kernel).

4.2.1 Tuning Parameters Used in the Algorithm

4.2.1.1 Kernel

SVM algorithm makes use of a set of mathematical functions which helps in setting up the hyper plane. This set of mathematical functions is known as Kernel. Its main task is to take the input data and convert it into required form. There are various type of kernel functions available today which can be used with various type of SVM algorithms. Some of the most commonly used Kernel functions are:

4.2.1.2 Polynomial Kernel

It is popular in image processing tasks and is defined as (Equation 2):

$$k(x_{i,}x_{j}) = (x_{i}.x_{j} + 1)^{d} \tag{2}$$

where d is degree of the polynomial.

4.2.1.3 Gaussian Kernel

It is a general purpose kernel and is defined as(Equation 3):

$$k(x,y) = e^{-\frac{(x-y)^2}{2\sigma^2}} \tag{3}$$

4.2.1.4 Gaussian Radial Basis Function (RBF) Kernel

It is another general purpose kernel and is used when no prior knowledge about the data is available. It is defined as(Equation 4):

$$k(x,y) = e^{-y(x-y)^2} \tag{4}$$

4.2.1.5 Regularization and Gamma

The Regularization parameter is often represented as 'c'. It is used in SVM optimization and indicates how much we want to avoid misclassifying each training sample. It serves as a degree of importance that is given to misclassifications.

Gamma parameter in SVM defines the range of influence that a single training example will cover. In this, low values mean 'far' and high values mean 'close'. It indicates that with low gamma values, points far away from the plausible separation line are considered in calculation for separation line. On the other hand, high gamma values suggest that points close to the plausible line are considered in the calculation.

The SVM classifier flow chart is represented with the help of Figure 6.Configurations of the parameters project used for SVM model are as follows:

Kernel: RBF (Radial Basis Function)

Regularization Constant (c): 2**10

Gamma = 'auto'

Degree = 1

Random State = 0

4.3 XGBoost

XGBoost is a fairly new hybrid algorithm in the field of Machine Learning having introduced recently in the year 2014. Despite being fairly new XGBoost has proved its mettle in terms of performance and speed.

XGBoost is another example of ensemble learning, just like Random Forest Technique discussed earlier. Ensemble learning is powerful as it gives the benefit of combining various learning models in cases where using just one machine learning model might not be sufficient. It combines the predictive power of multiple learners.

Figure 7. SVM Classifier Flow Chart
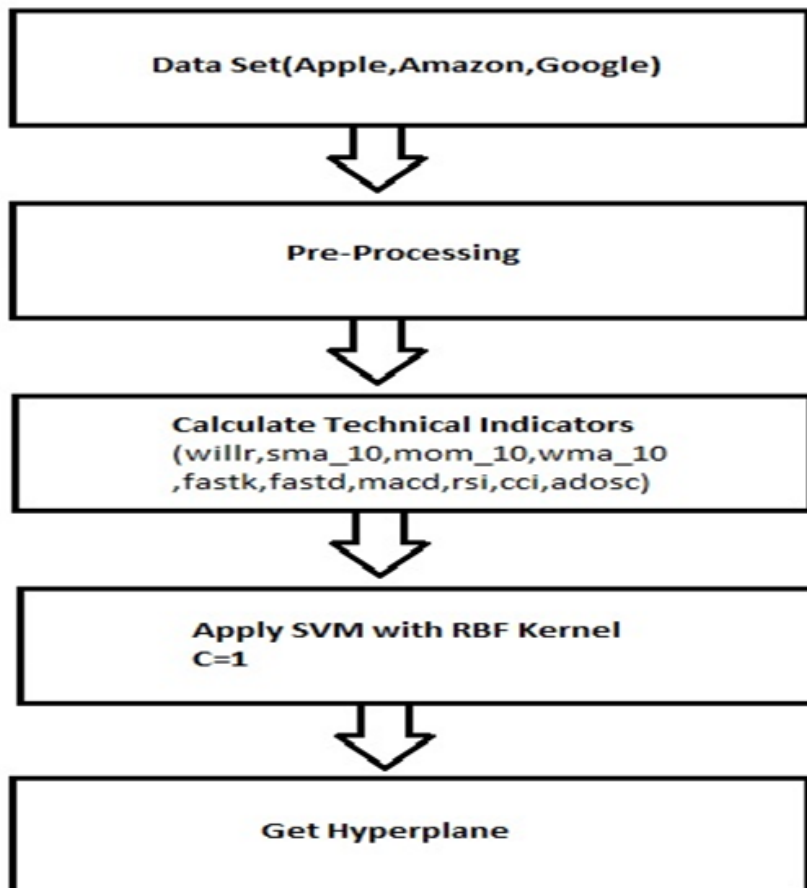
XGBoost is also called a gradient boosting algorithm because it uses gradient descent technique to minimize loss when adding new models. It is a supervised learning algorithm, used for developing high performance and fast gradient boosting tree models. The entire process of XGBoost can be defined in two techniques: **Bagging** and **Boosting**. Bagging and Boosting are two widely used ensemble learners.

Bagging- Although decision trees are among the most easily interpreted models, there is a problem with their highly variable behaviour. Bagging is an abbreviation for Bootstrap Aggregation. It is a meta-algorithm that takes N samples from the initial dataset and trains the predictive model on the samples.

It is typically used when we want to reduce variance while retaining the bias in any learners such as decision trees. Base learners of the bagging technique are generally several decision trees that are generated in parallel. Data sampled with replacement is fed to these learners for training. The final prediction of the model is evaluated by averaging the outputs from all the learners.

Boosting-In this technique decision trees are built in a sequential fashion. This is done in a way that each tree looks to reduce errors of the previous trees. Each tree learns from its predecessors and then helps in updating the residual errors. Because of this the next tree that grows in the sequence will learn from an updated version of the errors and thus will be a better predictor.

The base learners used in Boosting are very weak learners and their prediction power is just a little more than random guessing. But all these weak learners contribute some vital information useful for prediction to the trees next generated in the sequence, thus making them an improvement over the weak learners. Boosting technique thus makes use of the weak learners by combining them and producing a final strong learner that brings down both bias and variance.

Boosting can be defined as an iterative combination of weak learning algorithms to form an algorithm with strong predictive power. The selected sample is obtained more intelligently in boosting.

In contrast to bagging technique, boosting uses trees with comparatively fewer number of splits. We use validation techniques such as **k-fold cross validation,** in order to optimally select **parameters like number of trees, depth of tree and the rate at which gradient boosting will learn**. While selecting these parameters, it should be kept in mind that selecting a large number of tress might lead to overfitting. Hence, the stopping criteria for boosting should be selected carefully, based on the size and distribution of data.

Boosting is carried out as follows:

i.   Fit a simple decision tree on the input data used. We call 'x' as input and 'y' as output.
ii. Next step is to calculate the error residuals 'e'. It is calculated by subtracting the predicted target value from the actual target value (Equation 5).

$$e = y - y_{pred1} \tag{5}$$

iii. Next, a new model is fitted on error residuals as target variables, using the same input variables. Letus call it e_{pred}.
iv. Add the predicted residuals to the previous predictions (Equation 6).

$$y_{pred2} = y_{pred2} + e_{pred} \tag{6}$$

v.   Next, fit another model on the residuals that still remain $e_{new} = y - y_{pred2}$, and repeat steps (ii) to (v) until it starts overfitting or the sum of residuals becomes constant.

Over fitting can be controlled by checking accuracy on the validation data consistently.

4.3.1 Reason for Applying XGBoost

XGBoost provides excellent performance and speed. It is capable of solving both linear and tree learning models. The gradient boosting trees algorithm is efficiently implemented in XGBoost. Gradient boosting attempts to predict target value accurately by estimating sets of similar weak models. Regression trees are the learners in this case, with input data points mapped to one of its leaves containing a continuous score. The training is done iteratively by adding new trees that predict the error of the previous tree, which is then combined with the previous tree to give a final prediction.

In this classification model, the most important step is selecting the number of estimators or trees. This selection largely depends on the type of data used and size of data. In this project, experiments were done taking the value of n from 400-500, increasing the value of n by 10 in each iteration. We found that we achieved best results when *n-estimators = 475* and *cross-validation generator(cv) = 10*. Flowchart of the XGBoost Classifier can be understood using Figure 8

Figure 8. XGBoost Classifier Flow-Chart
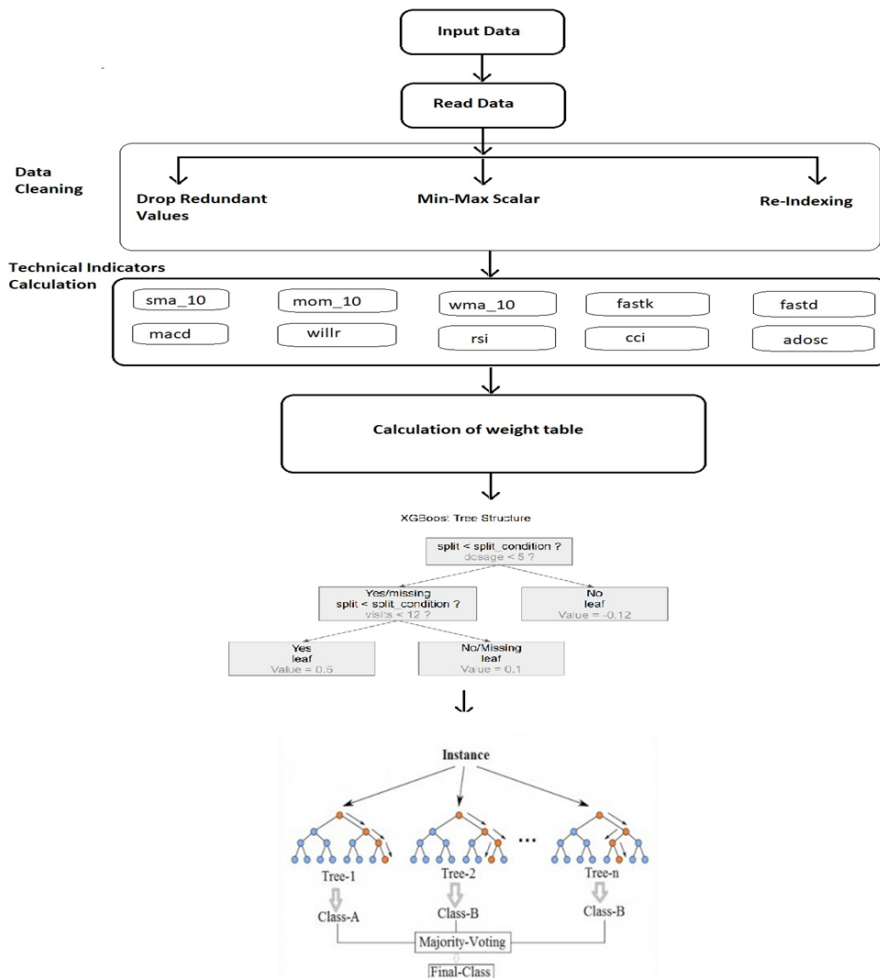
4.4 Long Short Term Memory Networks (LSTM)

Long Short Term Memory Networks are recurrent neural networks. They can be thought of as neural networks enclosed or looped within a neural network. LSTMs are capable of solving complex problems that encompass sequentiality within them. Consider the example below to understand the type of problems that neural networks can solve.

When we, human beings, read a text, we do not always attempt to understand the text from the start. We connect previous word or words to understand, or more specifically predict the next word that would come up.

Less or more, LSTMs do the same thing. They try to predict the next sequence of data, based on their knowledge of the previous sequence. They do this with help of the memory cells that they have. Memory cells in Long Short Term Memory Networks are capable of storing previous outputs so that they can be used to predict the next sequence.

Xt is the input and A is the LSTM. After 'n' iterations, the neural network outputs ht, input data x_0 is fed to the LSTM, it outputs h_0, which is then used as an input to the next network in the consecutive series. After the input data x_0 has passed through 'n' networks in the series, the output is ht.

The structure of an LSTM cell is as follows:

4.4.1 The Input Gate

First, the input is squashed between -1 and 1 using a *tanh* activation function. This can be expressed by (Equation 7):

$$g = \tanh(b^g + x_t U^g + h_{t-1} v^g) \tag{7}$$

where U^g and V^g are weights for the input and previous cell output respectively, and b^g is the input bias. The expression for the input gate is(Equation 8):

$$i = \sigma(b^i + x_t U^i + h_{t-1} V^i) \tag{8}$$

4.4.2 The Internal State and the Forget Gate

The forget gate is expressed as (Equation 9):

$$f = \sigma(b^f + x_t U^f + h_{t-1} V^f) \tag{9}$$

The output from this stage, s_t is expressed by (Equation 10):

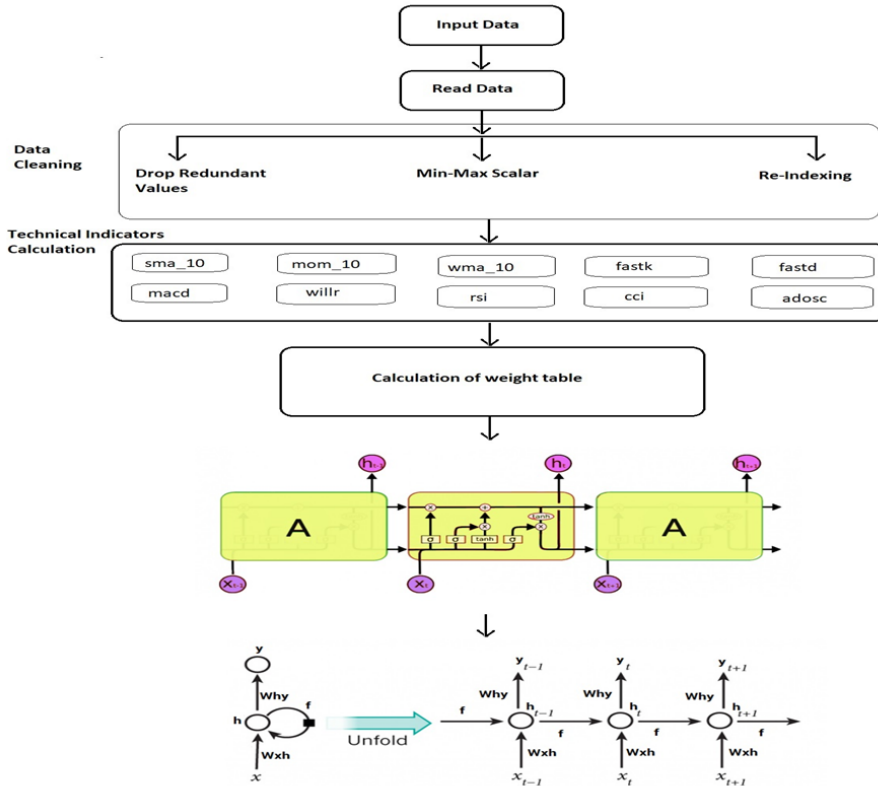$$s_t = s_{t-1} \circ f + g \circ i \tag{10}$$

4.4.3 The Output Gate

The output gate is expressed as (Equation 11):

$$o = \sigma(b^o + x_t U^o + h_{t-1} V^o) \tag{11}$$

So the final output of the cell can be expressed as(Equation 12):

$$h_t = \tanh(s_t) \circ o \tag{12}$$

Figure 9. LSTM Classifier Flow-Chart

Above presented Figure 9 represents the flow chart of the LSTM Classifier. The parameter settings used in this study are as follows:

Epochs = 10 n_classes = 1 n_units = 200 n_features = 9 batch_size = 40 .

4.5 Feasibility of Naive Bayes

Naive Bayes is a really simple classification algorithm that assumes strong independence between the features of a dataset. Independence denotes the assumption of Naive Bayes algorithm that the effect of one feature of a dataset over another does not affect the result of the model. It works on the Bayes theorem from Bayesian Statistics. Naive Bayes uses Bayes theorem to update degrees of belief of the features, and thus returns the class with the highest degree of belief.

The formula for Naive Bayes is given by (Equation 13):

$$P(C \mid X) = P(X \mid C) * P(C) / P(X) \tag{13}$$

where P(C|X)-- The posterior probability. It means that given X features (x1,x2,x3,x4......xn), what is the probability of the data point belonging to class C,

P(X|C) -- The likelihood of a feature in a class C,

P(C) -- The probability of class C without the evidence of features X,

P(X) -- The probability of each feature X.

4.5.1 Why the Naive Bayes Algorithm is Naive

Bayes theorem assumes that there is no dependence between features. It assumes that the presence of one feature does not affect the presence of another feature.

In this work of Stock Price Direction prediction, there is a strong relation and dependence between various stock indicators and features such as Open, High, Low and Close. This relation is further strengthened by the dependence of target variable of a day on to the previous or next day.

In Naïve Bayes Classification, if categorical data has a category, which was not observed in training the dataset, then the Naïve Bayes model assigns a probability of 0 to that category. Because of this, the model will be unable to make a prediction for that category. This is known as "Zero Frequency problem". Naïve Bayes is referred to as a bad estimator, based on its dependence on probability based output.

On conducting experiments, an accuracy of 59.5% was obtained using Naïve Bayes Classifier. On the other hand, least average accuracy obtained in this project was 67.82% using SVM classifier. The reason we achieve such poor accuracy using Naive Bayes is because the algorithm assumes independence between features, whereas the problem of stock price trend determination is highly dependent on features. Naïve Bayes classifiers do not take feature interactions into consideration at all, whereas this study requires an algorithm that takes feature interactions into prior consideration before making predictions.

5. RESULT ANALYSIS

This section is split into four subsections. The first subsection discusses various performance metrics that are used to evaluate the performance of prediction models. The second subsection presents the computed results in both tabular and graphical form. The third subsection depicts an examination of the generated results. Finally, in the fourth subsection, a detailed comparison of the results of the shortlisted five prior published studies and the current study is made.

5.1 Performance Metrics

The most formal way for evaluation of performance of this model is Accuracy and f-measure (F1-score). These performance measures are computed by estimating Precision(Equations 14 and 15) and Recall values(Equations 16 and 17) using True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN).

$$\Pr ecision+ = \frac{TP}{TP + FP} \tag{14}$$

$$\Pr ecision- = \frac{TN}{TN + FN} \tag{15}$$

$$\mathrm{Re}\,call+ = \frac{TP}{TP + FN} \tag{16}$$

$$\mathrm{Re}\,call- = \frac{TN}{TN + FP} \tag{17}$$

Precision is the weighted average of precision positive and negative, while Recall is the weighted average of recall positive and negative.

Accuracy (Equation 18) and F1-score (Equation 19) are then calculated as:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{18}$$

$$F1 - score = \frac{2 * \mathrm{Pr}\,ecision * \mathrm{Re}\,call}{\mathrm{Pr}\,ecision + \mathrm{Re}\,call} \tag{19}$$

5.2 Experimental Results

Experiments were conducted using the technical indicators as continuous input data for the prediction models, derived from market research data, or stocks and stock price indices. Prediction models used for conducting these experiments were Support Vector Machines (SVM), Long Short Term Memory (LSTM) Networks, XGBoost and Random Forest.

Each of the prediction models is compared on the basis of the performance achieved after optimal parameter setting from parameter setting experiments. Final selected optimal parameters for each prediction model have been mentioned in previous section.

Summary of results obtained on each of these models is listed using Table 5.

Table 5. Computed Results from various prediction models

Prediction Model			Results Achieved	
	Best Accuracy	Best F1-Score	Average Accuracy	Average F1-Score
SVM	70.55	82.78	67.82	80.78
LSTM	77.68	87.44	72.49	83.87
XGBoost	79.04	86.38	77.25	83.87
Random Forest	84.80	89.33	83.74	88.02

We observe that the best results were obtained using Random Forest Classifier as the prediction model for the project, with an average Accuracy and F1 score of 84.80 and 89.33 respectively.

Figure 10a-d. Graphical representation of computed results from every selected prediction model on data of stocks as well as indices

(a) Accuracy graph of stocks when SVM is applied

(b) F1 graph of stocks when SVM is applied

(c) Accuracy graph of indices when SVM is applied

(d) F1 graph of indices when SVM is applied
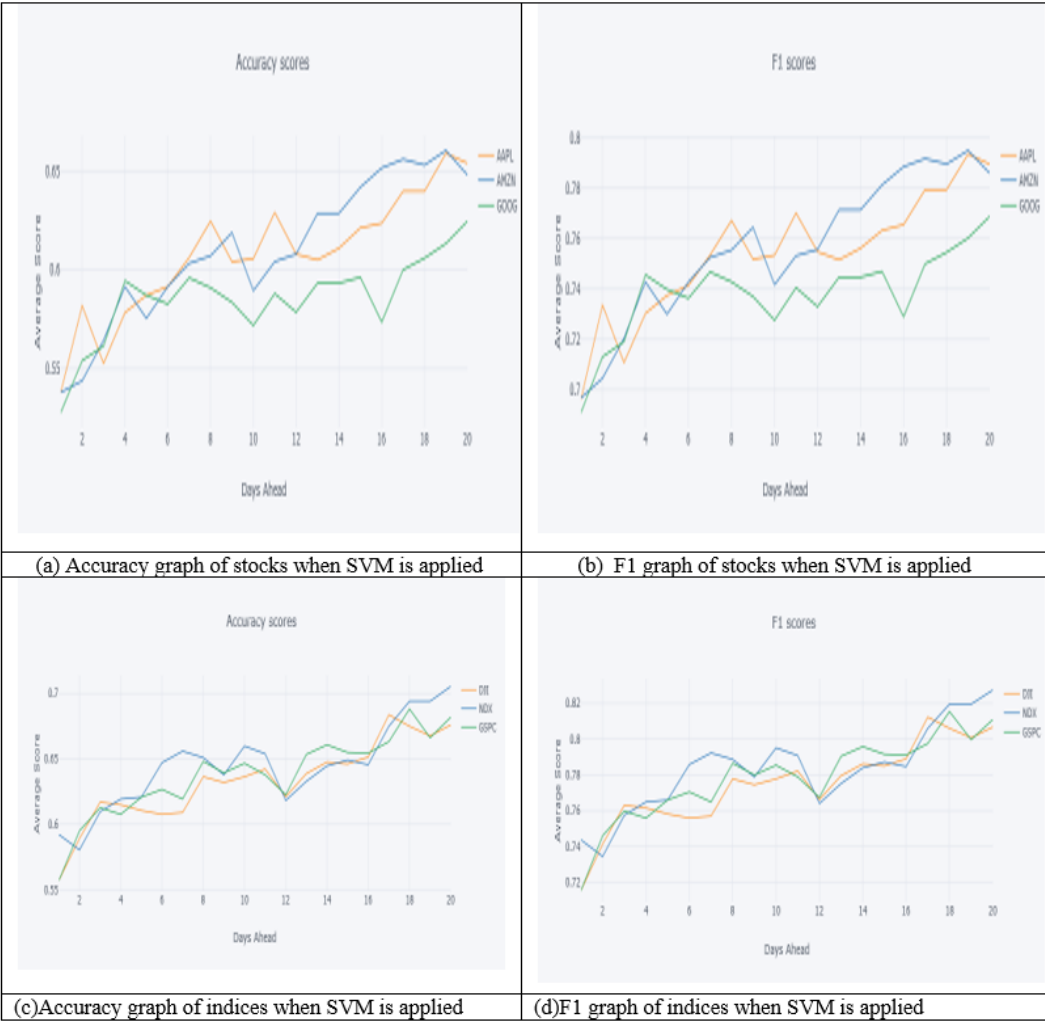
Graphical representation of the results for every prediction model is given below Figures 10a to 10p.

5.3 Result Observations

In a graphical representation of all of its outputs, this study shows that the accuracy and F1-score for predicting stock and stock index movement for one day ahead is the lowest in every scenario. This is because a single stock price is prone to being influenced by large amounts of noise, whereas a long-term trend inherently converges to the company's market performance. It occurs because the next 1-day price in the stock market is highly volatile and should be close to random. The study can consistently outperform the random walk with this model.

By closely examining the study's output, it is clear that as time or days-ahead prediction increases, so does accuracy and F1-score for each model. This is due to the fact that as the time period for prediction increases, the model's noise decreases and the results begin to converge.

Figure 10(e-h). Graphical representation of computed results from every selected prediction model on data of stocks as well as indices

(e)Accuracy graph of stocks when LSTM is applied

(f)F1 graph of stocks when LSTM is applied

(g)Accuracy graph of indices when LSTM is applied

(h)F1 graph of indices when LSTM is applied

It is clear from the results that the best results, in terms of both accuracy and F1-score for each classification model used, are obtained for long term predictions, i.e., in the range of 17-20 days.
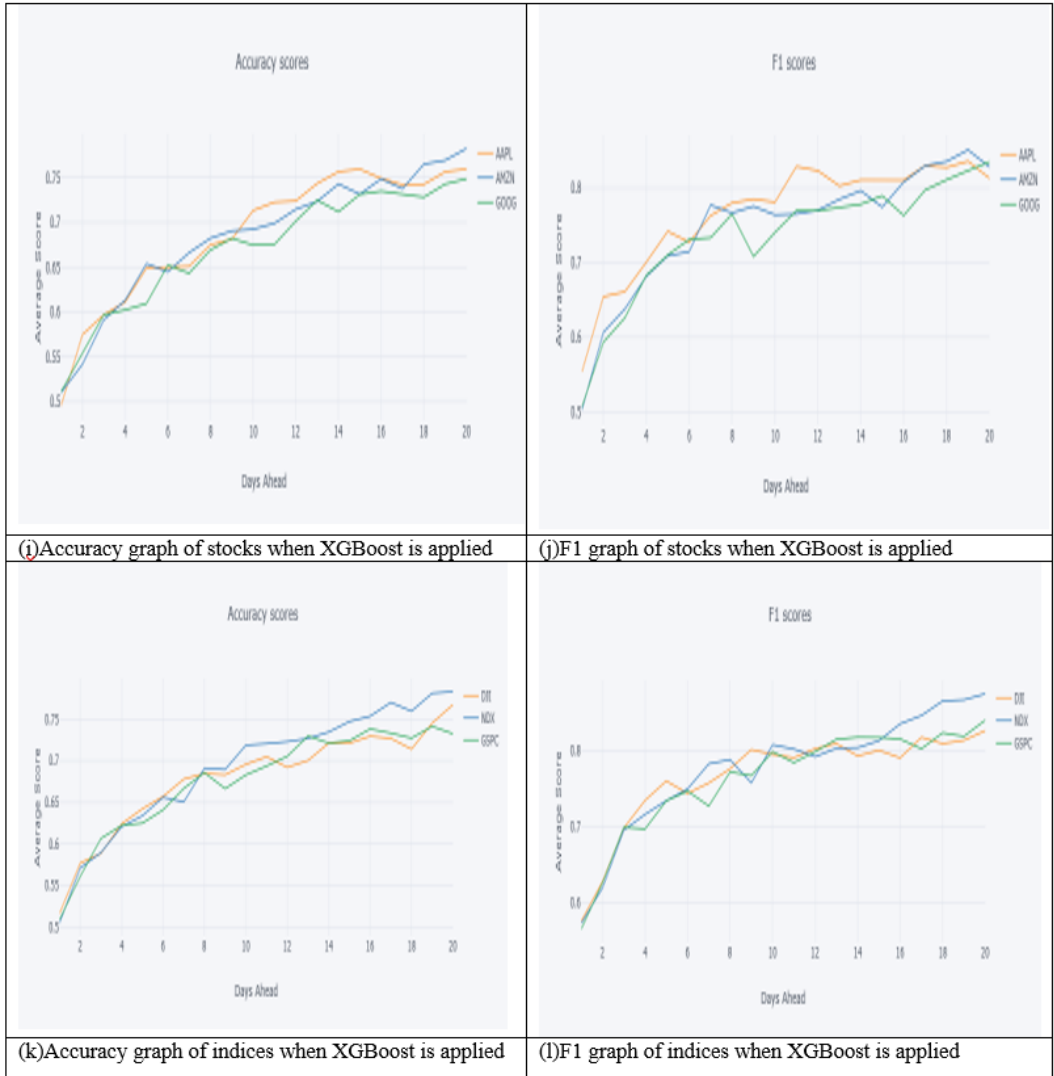
Another important point to note here is that we cannot keep increasing the prediction time period in the hope of getting better results. This is due to the fact that these predictions have a threshold point above which accuracy and F1-score increase; once that point is reached, both accuracy and F1-score begin to fall sharply. Another reason is that very long-term predictions would cause overfitting in the classification model.

5.4 Discussion on Random Forest's Structure, Nature, and Performance

We have analyzed the nature, behavior and the performance of the random forest to observe why the random forest performs auspiciously and the following text of this sub-section is dedicated to it.

Decision trees are the basics of Random Forest; it's a method of predictive decision-making. An info graphic can be either tree-based or simplified for accurate inferences. It takes into consideration
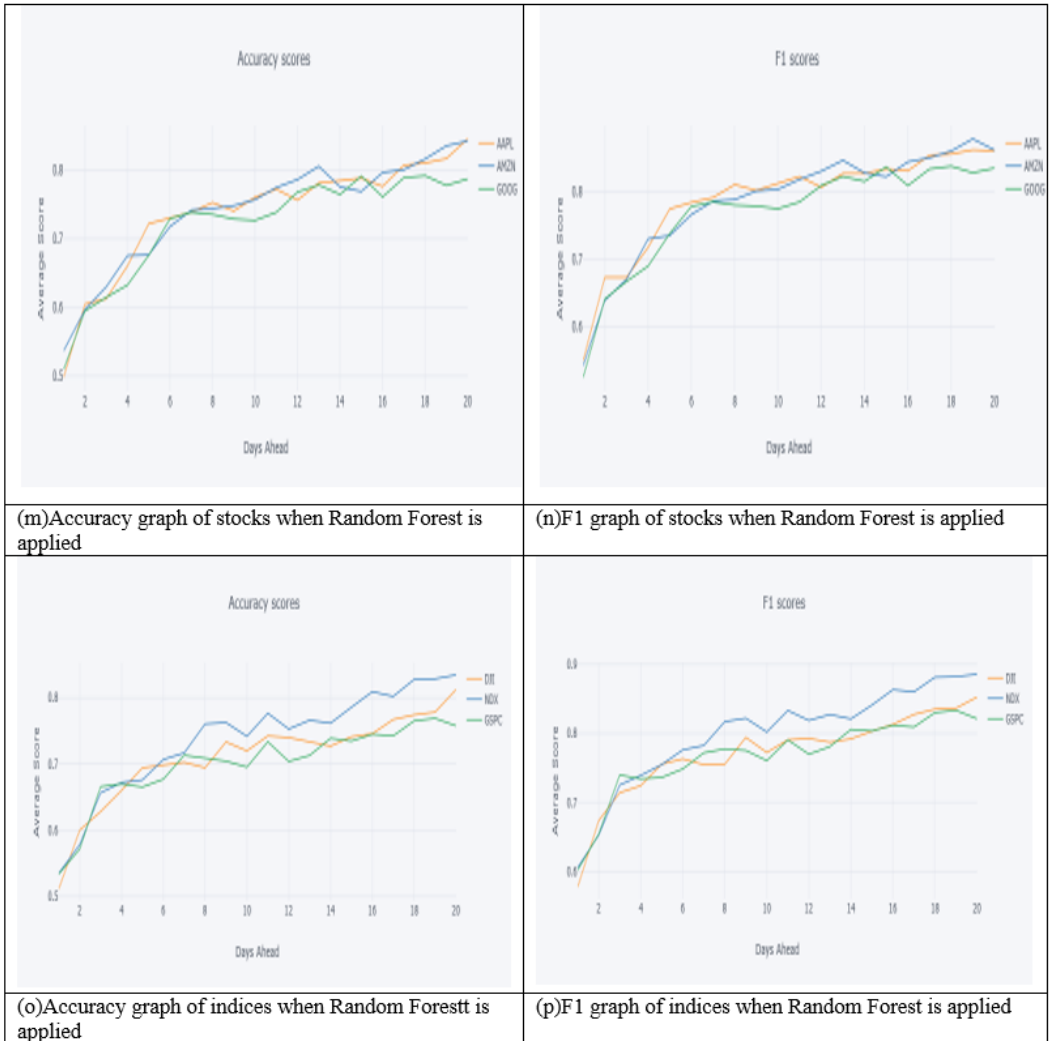
Figure 10(i-l). Graphical representation of computed results from every selected prediction model on data of stocks as well as indices

(i)Accuracy graph of stocks when XGBoost is applied

(j)F1 graph of stocks when XGBoost is applied

(k)Accuracy graph of indices when XGBoost is applied

(l)F1 graph of indices when XGBoost is applied

an additional variable to estimate the results of a dependent variable. It is the most versatile machine learning approach available today, which is then used to fine-tune a generalized model. Decision forest is similar to a decision tree algorithm in that it uses an ensemble method to consolidate weak practice (trees) ensembling.The random forest condenses thousands of trees and assumes a certain number of predefined features or attributes in each division. The predictions of each individual tree are pooled to arrive at the final result. When using a dataset, the random forest algorithm creates a decision tree for each and applies that tree to a sample. Then, estimated results will be tallied. In the last step, the most voted answer is declared to be the final. At the heart of all ensemble machine learning algorithms is Bootstrap aggregating, also known as bagging. This method improves the stability and accuracy of learning algorithms. At the same time, it also reduces variance and overfitting, which is a common problem while constructing decision trees.

No other algorithm can do as well on huge datasets as it can quantify important classifications and proposes experimentally important variables for quantification. Data outliers are determined

Figure 10(m-p). Graphical representation of computed results from every selected prediction model on data of stocks as well as indices

(m)Accuracy graph of stocks when Random Forest is applied

(n)F1 graph of stocks when Random Forest is applied

(o)Accuracy graph of indices when Random Forestt is applied

(p)F1 graph of indices when Random Forest is applied

by the concurrence function with datasets. Many records have missing data, but it can still remain precise. Models can be used to obtain data about the relationships between variables as well as to detect outliers.Using Random Forests, decision trees remove the overfitting problem. The growth of trees occurs in feature subsets in a random forest. Random forests are built by randomly selecting M features or samples, and then predicting their likelihoods. Using Random Forests, decision trees remove the overfitting problem. Every tree is grown in a subset of space. A tree-Forest method uses as many features (M) and data samples (N) to predict values. Since the probability of each feature being used in a random forest depends on the number of trees, it is possible to assert that the probability of the feature being included in the ensemble is binomial distributed. Additionally, it works well for both regression and classification tasks. It's quite reliable because the default settings work well most of the time.

The market analyst uses a random forest algorithm because it's regarded as one of the simplest to apply and best for machine learning. It's very reliable and commonly utilized in classifying algorithms.

Stock markets fluctuate widely, when their predictions are especially hard to make. A tree classifier uses the same hyperparameters as a random forest. It emulates a tree. The strategy takes into account different factors like cost, impact, impact, and utility. The random forest algorithm chooses various observations and features and then amalgamates the combined all of the trees together. Labels or attributes are used to determine partitions based on the data. Random Forest overcomes this problem by training multiple decision trees on different subspaces of the feature space at the cost of slightly increased bias. This means none of the trees in the forest see the entire training data. The data is recursively split into partitions. At a particular node, the split is done by asking a question about an attribute.

We can apply our model to devising new trading strategies or to manage portfolios according to predictions. The graph that was constructed confirms our theory about the efficacy of our model. When we have more random forests added, the classifier converges The approach taken is new to this type of approach, and only slight modifications will likely be necessary for different cases. From our analysis, we can conclude that our model outshines the various models presented in our study as well as various papers (studies) discussed in the studies.

5.5 Comparison With Similar Prior Published Works

(Patel, Thakkar & Kotecha, 2015) in a similar type of study depict that best results for continuous valued input data have been achieved using Random Forest classifier with an average Accuracy Score of 83.59 and average F1-Score of 83.99. Accuracy measure and F1-Score of the Random Forest model presented in this study is evaluated at 83.74(Average) and 88.02(Average) respectively.

(Kara, Boyacioglu & Baykan 2011) conducted a similar study where authors also selected Istanbul stock exchange as the subject. Similar technical indicators based prediction was conducted in this paper for predicting stock and stock index movement using Artificial Neural Network (ANN) and Support Vector Machines (SVM). Experimental results showed that average performance of ANN model (75.74%) was significantly better than that of SVM model (71.52%). The current experimental study exceeds maximum accuracy achieved by them by 8% with even higher F1 score.

(Diler, 2003) in 2003 experimented on Istanbul stock exchange and published a similar study, conducted using Back-Propagation, an improvement over Artificial Neural Networks. He achieved an accuracy of just 60.81%. The lowest accuracy achieved in our project is 67.82% using SVM prediction model which is better than the results achieved in his study.

(Di, 2014), Apple Scholar at Stanford University, did a similar study using Technical Indicators in 2014. He achieved an accuracy of 77.25%, but he did not take into consideration the effect of stock prediction over long time period. The results achieved in the presented study using XGBoost& Random Forest Classifier still comes out to be better, even when the experiments are conducted over a moderate time period.

(Kumar & Thenmozhi, 2006) in 2005, used SVM and Random Forest classifier to perform stock movement direction prediction. The highest accuracy achieved by him was using SVM classifier and was recorded at 68.44%. This presented study again performs better with all the four classifiers used in that work.

6. CONCLUSIONS, LIMITATIONS, AND FUTURE DIRECTION

The stock market is very unpredictable, although in recent years, machine learning methods have shown great accuracy in the stock market forecasting. The primary goal of this research was to forecast the direction of movement of stock and stock price indices. It assesses and compares the performance of four algorithms for the aforementioned task: Random Forest, Support Vector Machines, Long Short Term Memory Network, and XGBoost. The models are tested using data from 11 years of historical stocks and stock indices, ranging from January 2007 to December 2017.This study employs trend deterministic data in the form of ten technical indicators as training data, with stock prices in the

form of (Open, High, Low, Close, and Volume) as input data to the model. For the evaluation of various subject models, the project used accuracy scores and F1 scores as performance metrics. This study shows that the Random Forest classifier outperforms all other algorithms, with the highest accuracy of 84.80% and F1 score of 89.33%. From the comparative analysis we have done in the previous section, we can confidently say that our model outperforms the models as seen in many prior published and discussed papers.

Several constraints were encountered during these experiments. Experiments in this study necessitate the use of a powerful GPU to perform certain computations. The SVM algorithm could have performed even better if a Poly kernel was used instead of RBF and c=100 was used. However, it was not possible due to the high speed of computation and processing required, which only a GPU can provide. Additionally, the use of a GPU would have aided in the implementation of the LSTM algorithm because it is a deep learning algorithm that performs better on a GPU.

However, this work meant that all services were used to the maximum of their ability, and it was a positive. The efficiency of artificial neural networks, and the recently introduced models focused on them, for solving the problem at hand should be explored as a potential path for researchers. The use of a bigger dataset would allow the stock market forecast framework to be more precise in the future. This will help our forecasting models. Besides, there are several models of machine learning that can also be analysed for the error rate of precision. A shortage of historical details remains. The market's text content, like news, is largely underutilized. So in the next phase, we will use text knowledge element to boost efficiency. We are not using the Chinese, European, and Japanese capital markets in reality; next, we're expanding our datasets and predicting whether our models are valid in the Chinese, Japanese, and European markets as well.

REFERENCES

Abraham, A., Nath, B., & Mahanti, P. K. (2001). Hybrid intelligent systems for stock market analysis. In *Computational science-ICCS 2001* (pp. 337–345). Springer. doi:10.1007/3-540-45718-6_38

Basak, S., Kar, S., Saha, S., Khaidem, L., & Dey, S. R. (2019). Predicting the direction of stock market prices using tree-based classifiers. *The North American Journal of Economics and Finance*, *47*, 552–567. doi:10.1016/j.najef.2018.06.013

Bhattacharya, S., Maddikunta, P. K. R., Kaluri, R., Singh, S., Gadekallu, T. R., Alazab, M., & Tariq, U. (2020). A novel PCA-firefly based XGBoost classification model for intrusion detection in networks using GPU. *Electronics (Basel)*, *9*(2), 219. doi:10.3390/electronics9020219

Bukhari, A. H., Raja, M. A. Z., Sulaiman, M., Islam, S., Shoaib, M., & Kumam, P. (2020). Fractional neuro-sequential ARFIMA-LSTM for financial market forecasting. *IEEE Access: Practical Innovations, Open Solutions*, *8*, 71326–71338. doi:10.1109/ACCESS.2020.2985763

Chen, A.-S., Leung, M. T., & Daouk, H. (2003). Application of neural networks to an emerging financial market: Forecasting and trading the taiwan stock index. *Computers & Operations Research*, *30*(6), 901–923. doi:10.1016/S0305-0548(02)00037-0

Chimmula, V. K. R., & Zhang, L. (2020). Time series forecasting of COVID-19 transmission in Canada using LSTM networks. *Chaos, Solitons, and Fractals*, *135*, 109864. doi:10.1016/j.chaos.2020.109864 PMID:32390691

Deshpande, A., Patavardhan, P., Estrela, V. V., Razmjooy, N., & Hemanth, J. (2020). Deep learning as an alternative to super-resolution imaging in UAV systems. *Imaging and Sensing for Unmanned Aircraft Systems, 2*, 9.

Di, X. (2014). *Stock Trend Prediction with Technical Indicators using SVM. Independent Work Report*. Stanford Univ.

Diler, A. I. (2003). Predicting direction of ISE national-100 index with back propagation trained neural network. *Journal of Istanbul Stock Exchange*, *7*(25-26), 65–81.

Ding, G., & Qin, L. (2020). Study on the prediction of stock price based on the associated network model of LSTM. *International Journal of Machine Learning and Cybernetics*, *11*(6), 1307–1317. doi:10.1007/s13042-019-01041-1

Ghosh, I., Sanyal, M. K., & Jana, R. K. (2018, August). Fractal Inspection and Machine Learning-Based Predictive Modelling Framework for Financial Markets. *Arabian Journal for Science and Engineering*, *43*(8), 4273–4287. doi:10.1007/s13369-017-2922-3

Gumelar, A. B., Setyorini, H., Adi, D. P., Nilowardono, S., Widodo, A., Wibowo, A. T., . . . Christine, E. (2020, September). Boosting the Accuracy of Stock Market Prediction using XGBoost and Long Short-Term Memory. In *2020 International Seminar on Application for Technology of Information and Communication (iSemantic)* (pp. 609-613). IEEE. doi:10.1109/iSemantic50169.2020.9234256

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. doi:10.1016/j.eswa.2006.04.007

Hiransha, M., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2018). NSE stock market prediction using deep-learning models. *Procedia Computer Science*, *132*, 1351–1362. doi:10.1016/j.procs.2018.05.050

Hu, A., & Razmjooy, N. (2020). Brain tumor diagnosis based on metaheuristics and deep learning. *International Journal of Imaging Systems and Technology*.

Hua, S., & Sun, Z. (2001). Support vector machine approach for protein subcellular localization prediction. *Bioinformatics (Oxford, England)*, *17*(8), 721–728. doi:10.1093/bioinformatics/17.8.721 PMID:11524373

Huang, W., Nakamori, Y., & Wang, S.-Y. (2005). Forecasting stock market movement direction with support vector machine. *Computers & Operations Research*, *32*(10), 2513–2522. doi:10.1016/j.cor.2004.03.016

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. doi:10.1016/j.eswa.2010.10.027

Kim, K. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, *55*(1-2), 307–319. doi:10.1016/S0925-2312(03)00372-2

Kumar, M., & Thenmozhi, M. (2006, January). Forecasting stock index movement: A comparison of support vector machines and random forest. In *Indian institute of capital markets 9th capital markets conference paper*. doi:10.2139/ssrn.876544

Liu, Z., Dang, Z., & Yu, J. (2020, November). Stock Price Prediction Model Based on RBF-SVM Algorithm. In *2020 International Conference on Computer Engineering and Intelligent Control (ICCEIC)* (pp. 124-127). IEEE doi:10.1109/ICCEIC51584.2020.00032

Loke, K. S. (2017, November). Impact of financial ratios and technical analysis on stock price prediction using random forests. In *2017 International Conference on Computer and Drone Applications (IConDA)* (pp. 38-42). IEEE. doi:10.1109/ICONDA.2017.8270396

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. doi:10.1111/j.1540-6261.1970.tb00518.x

Moghar, A., & Hamiche, M. (2020). Stock market prediction using lstm recurrent neural network. *Procedia Computer Science*, *170*, 1168–1173. doi:10.1016/j.procs.2020.03.049

Mondal, S., Dutta, A., & Chatterjee, P. (2020, February). Application of Deep Learning Techniques for Precise Stock Market Prediction. *Proceedings Of The 3 rd National Conference On Machine Learning And Artificial Intelligence (NCMLAID'20)*.

Nahil, A., & Lyhyaoui, A. (2017). Stock price prediction based on SVM: The impact of the stock market indices on the model performance. *Proceedings of the Engineering and Technology–PET*, *21*, 91-95.

Nikou, M., Mansourfar, G., & Bagherzadeh, J. (2019). Stock price prediction using DEEP learning algorithm and its comparison with machine learning algorithms. *Intelligent Systems in Accounting, Finance & Management*, *26*(4), 164–174. doi:10.1002/isaf.1459

Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2020). Efficient stock-market prediction using ensemble support vector machine. *Open Computer Science*, *10*(1), 153–163. doi:10.1515/comp-2020-0199

Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2020). A comprehensive evaluation of ensemble learning for stock-market prediction. *Journal of Big Data*, *7*(1), 1–40. doi:10.1186/s40537-020-00299-5

Ou, P., & Wang, H. (2009). Prediction of stock market index movement by ten data mining techniques. *Modern Applied Science*, *3*(12), 28. doi:10.5539/mas.v3n12p28

Pang, X., Zhou, Y., Wang, P., Lin, W., & Chang, V. (2020). An innovative neural network approach for stock market prediction. *The Journal of Supercomputing*, *76*(3), 2098–2118. doi:10.1007/s11227-017-2228-y

Parmar, I., Agarwal, N., Saxena, S., Arora, R., Gupta, S., Dhiman, H., & Chouhan, L. (2018, December). Stock market prediction using Machine Learning. In *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)* (pp. 574-576). IEEE. doi:10.1109/ICSCCC.2018.8703332

Parsa, A. B., Movahedi, A., Taghipour, H., Derrible, S., & Mohammadian, A. K. (2020). Toward safer highways, application of XGBoost and SHAP for real-time accident detection and feature analysis. *Accident; Analysis and Prevention*, *136*, 105405. doi:10.1016/j.aap.2019.105405 PMID:31864931

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. doi:10.1016/j.eswa.2014.07.040

Podasca, E. (2021). *Predicting the Movement Direction of OMXS30 Stock Index Using XGBoost and Sentiment Analysis*. Academic Press.

Qiu, M., & Song, Y. (2016). Predicting the direction of stock market index movement using an optimized artificial neural network model. *PLoS One*, *11*(5). doi:10.1371/journal.pone.0155133 PMID:27196055

Razmjooy, N., Ashourian, M., Karimifard, M., Estrela, V. V., Loschi, H. J., do Nascimento, D., ... Vishnevski, M. (2020). Computer-aided diagnosis of skin cancer: a review. *Current Medical Imaging, 16*(7), 781-793.

Sadorsky, P. (2021). A Random Forests Approach to Predicting Clean Energy Stock Prices. *Journal of Risk and Financial Management*, *14*(2), 48. doi:10.3390/jrfm14020048

Shahid, F., Zameer, A., & Muneeb, M. (2020). Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM. *Chaos, Solitons, and Fractals*, *140*, 110212. doi:10.1016/j.chaos.2020.110212 PMID:32839642

Shen, S., Jiang, H., & Zhang, T. (2012). *Stock market forecasting using machine learning algorithms*. Department of Electrical Engineering, Stanford University.

Shiva Nandhini, J., Bari, C., & Pradip, G. (2020). Stock Market Prediction Using Machine Learning. *Journal of Computational and Theoretical Nanoscience*, *17*(4), 1584–1589. doi:10.1166/jctn.2020.8405

Singh, L. K., Garg, H., & Khanna, M. (2021). An Artificial Intelligence-Based Smart System for Early Glaucoma Recognition Using OCT Images. *International Journal of E-Health and Medical Communications*, *12*(4), 32–59. doi:10.4018/IJEHMC.20210701.oa3

Singh, L. K., Garg, H., Khanna, M., & Bhadoria, R. S. (2021). An enhanced deep image model for glaucoma diagnosis using feature-based detection in retinal fundus. *Medical & Biological Engineering & Computing*, *59*(2), 333–353. doi:10.1007/s11517-020-02307-5 PMID:33439453

Sun, J., & Li, H. (2012). Financial distress prediction using support vector machines: Ensemble vs. individual. *Applied Soft Computing*, *12*(8), 2254–2265. doi:10.1016/j.asoc.2012.03.028

Tsai, C.-F., Lin, Y.-C., Yen, D. C., & Chen, Y.-M. (2011). Predicting stock returns by classifier ensembles. *Applied Soft Computing*, *11*(2), 2452–2459. doi:10.1016/j.asoc.2010.10.001

Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, *10*(5), 988–999. doi:10.1109/72.788640 PMID:18252602

Vignesh, C. K. (2020). Applying machine learning models in stock market prediction. *EPRA International Journal of Research & Development*, 395-398.

Vijh, M., Chandola, D., Tikkiwal, V. A., & Kumar, A. (2020). Stock closing price prediction using machine learning techniques. *Procedia Computer Science*, *167*, 599–606. doi:10.1016/j.procs.2020.03.326

Wang, F., Xuan, Z., Zhen, Z., Li, K., Wang, T., & Shi, M. (2020). A day-ahead PV power forecasting method based on LSTM-RNN model and time correlation modification under partial daily pattern prediction framework. *Energy Conversion and Management*, *212*, 112766. doi:10.1016/j.enconman.2020.112766

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

Xiao, J., Zhu, X., Huang, C., Yang, X., Wen, F., & Zhong, M. (2019). A new approach for stock price analysis and prediction based on SSA and SVM. *International Journal of Information Technology & Decision Making*, *18*(01), 287–310. doi:10.1142/S021962201841002X

Yu, D., Wang, Y., Liu, H., Jermsittiparsert, K., & Razmjooy, N. (2019). System identification of PEM fuel cells using an improved Elman neural network and a new hybrid optimization algorithm. *Energy Reports*, *5*, 1365–1374. doi:10.1016/j.egyr.2019.09.039

Zhang, J., Cui, S., Xu, Y., Li, Q., & Li, T. (2018). A novel data-driven stock price trend prediction system. *Expert Systems with Applications*, *97*, 60–69. doi:10.1016/j.eswa.2017.12.026

Zolotareva, E. (2021). Aiding Long-Term Investment Decisions with XGBoost Machine Learning Model. doi:10.1007/978-3-030-87897-9_37

Mohak Kulshrestha is a software engineer working in Tata Consultancy Services. He has a bachelor's degree in Computer Science Engineering. His fields of interest include Machine Learning, Neural Networks, and NLP.