APPLICATION OF MACHINE LEARNING IN HIGH FREQUENCY TRADING OF STOCKS

Obi Bertrand Obi

WORLDQUANT UNIVERSITY

201 St. Charles Avenue, Suite 2500
New Orleans, LA 70170, USA

obibertrand@gmail.com

ABSTRACT

Algorithmic trading strategies have traditionally been centered on follwing the market trends and the use of technical indicators. Over the years High Frequency algorithmic Trading has been left only in the hands of institutional players with deep pockets and lots of assets under management, despite huge returns involved. In this project we built trading strategies by applying Machine Learning models to technical indicators based on High Frequency Stock data. The result is an automated trading system which when applied to any stock could generate returns which are ten times higher than the market returns without significant increase in volatility.

TABLE OF CONTENTS

Contents

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE: INTRODUCTION

1.0: INTRODUCTION

Not too long ago, Algorithmic Trading was only available for institutional players with deep pockets and lots of assets under management. Recent developments in the areas of open source, open data, cloud computing and storage as well as online trading platforms have leveled the playing field for smaller institutions and individual traders, making it possible to venture in this fascinating discipline with only a modern notebook and an Internet connection. Nowadays, Python and its eco-system of powerful packages is the technology platform of choice for algorithmic trading. Among others, Python allows you to do efficient data analytics (with e.g. numpy, pandas), to apply machine learning to stock market prediction (with e.g. scikit-learn) or even make use of Google's deep learning technology (with tensorflow) and Microsoft's CNTK.

Algorithmic trading basically refers to the trading of financial instruments based on some formal algorithm. An algorithm is a set of operations (mathematical, technical) to be conducted in a certain sequence to achieve a certain goal. For example, there are mathematical algorithms to solve a Rubik's cube (The Mathematics of the Rubik's Cube or Algorithms for Solving Rubik's Cube). Such an algorithm can perfectly solve the problem at hand via a step-by-step procedure. Another example is algorithms for finding the root(s) of an equation (if it (they) exist(s) at all). In that sense, the objective of a mathematical algorithm is often well specified and an optimal solution is often expected

High-frequency trading is a type of algorithmic trading characterized by complex computer algorithms that trade in and out of positions in fractions of seconds, leveraging arbitrage strategies in order to profit from the public markets. Commonly, traders take advantage of the penny spread between the bids-ask on equities. For the typical retail trader, this would seem redundant and the pay-off would be minuscule. For HFTs, the profit from the spread accumulates and as thousands of trades are executed, there are millions of dollars to be made.

1.1: PROBLEM STATEMENT

Traditionally, financial markets operated on a quote-driven process where a few market makers provided the sole liquidity and prices for Financial Assets. Recently, major developments have been made to automate the Financial Markets which have led to many trading firms using computer algorithms to trade the Assets. High Frequency Trading (HFT), in particular, has been a major topic due to the features that distinguishes it from electronic and manual trading. This includes the extremely high speed of execution (microseconds), multiple executions per session, and very short holding periods (usually less than a day).

Time series data in financial markets are highly nonlinear, nonstationary and noisy in nature. Traditional models based on statistical methods, such as the Autoregressive Moving Average (ARMA) model, Autoregressive Integrated Moving Average (ARIMA) model, and General Autoregressive Conditional Heteroskedasticity (GARCH) model, suffer from limitations due to their linearity assumption. Predicting

how the stock market will perform is one of the most difficult things to do. There are so many factors involved in the prediction such as; physical factors, psycological, rational and irrational behaviour, etc. All these aspects combine to make share prices volatile and very difficult to predict with a high degree of accuracy. Waren Buffet states that: “Forecasts may tell you a great deal about the forecaster; they tell you nothing about the future.” Hence finding the right algorithm to automatically and successfully predict and trade in financial markets is the Holy Grail in finance.

1.2: PROJECT OBJECTIVES

The main objective of this project is to develop a High Frequency Trading System which uses Machine learning to predict the the movements of stock market prices with reasonable level of accuracy, and to trade the stock with simple trading strategy to generate adequate performance.

Other obvjectives include the following:

- Comparative analyses of Machine learning Algorithims on High Frequency Stock data to determine algorithms with high predictive power of stock price movements
- Perform technical analyses as features to the Machine Learnming models in the High frequency Trading System
- Generate and track adequate performance from the High frequency Trading System.
- Add to the ellaborate body of literature on application of Machine learning to Finance and High Frequency Trading

1.3: HYPOTHESIS

1. Machine Learning Algorithms cannot be used to predict stock price movement with reasonable amount of certainty
2. Market regimes cannot be predicted with reasoble ammount of certainty

CHAPTER TWO: LITERATURE REVIEW

2.0: STOCK MARKET PREDICTION

According to Wikipedia, Stock market prediction is the act of trying to determine the future value of a company stock or other financial instrument traded on an exchange. The successful prediction of a stock's future price could yield significant profit.

The Efficient Market Hypothesis (EMH)

The EMH hypothesizes that the future stock price is completely unpredictable given the past trading history of the stock. There are 3 types of EMHs: strong, semi-strong, and weak form. In the weak EMH, any information acquired from examining the stock's history is immediately reflected in the price of the stock.

While the efficient market hypothesis finds favor among financial academics, its critics point to instances in which actual market experience differs from the prediction-of-unpredictability the hypothesis implies. A large industry has grown up around the implication proposition that some analysts can predict stocks price movements better than others; ironically that would be impossible under the Efficient Markets Hypothesis if the stock prediction industry did not offer something its customers believed to be of value.

The Random Walk Hypothesis

Burton Malkiel, in his influential 1973 work "A Random Walk down Wall Street", claimed that stock prices could therefore not be accurately predicted by looking at price history. As a result, Malkiel argued, stock prices are best described by a statistical process called a "random walk" meaning each day's deviations from the central value are random and unpredictable. This led Malkiel to conclude that paying financial services persons to predict the market actually hurt, rather than helped, net portfolio return. A number of empirical tests support the notion that the theory applies generally, as most portfolios managed by professional stock predictors do not outperform the market average return after accounting for the managers' fees.

In practice, there are three Stock market Prediction Methodologies:

1. **Fundamental Analysis:** Performed by the Fundamental Analysts, this method is concerned more with the company rather than the actual stock. The analysts make their decisions based on the past performance of the company, the earnings forecast etc.
2. **Technical Analysis:** Performed by the Technical Analysts, this method deals with the determination of the stock price based on the past patterns of the stock (using time-series analysis).
3. **Machine learning:** With the advent of the digital computer, stock market prediction has since moved into the technological realm**.** When applying Machine Learning to Stock Data, we are more interested in doing a Technical Analysis to see if our algorithm can accurately learn the underlying patterns in the stock time series. Machine Learning can also play a major role in

evaluating and forecasting the performance of the company and other similar parameters helpful in Fundamental Analysis. In fact, the most successful automated stock prediction and recommendation systems use some sort of a hybrid analysis model involving both Fundamental and Technical Analysis.

2.1: MACHINE LEARNING

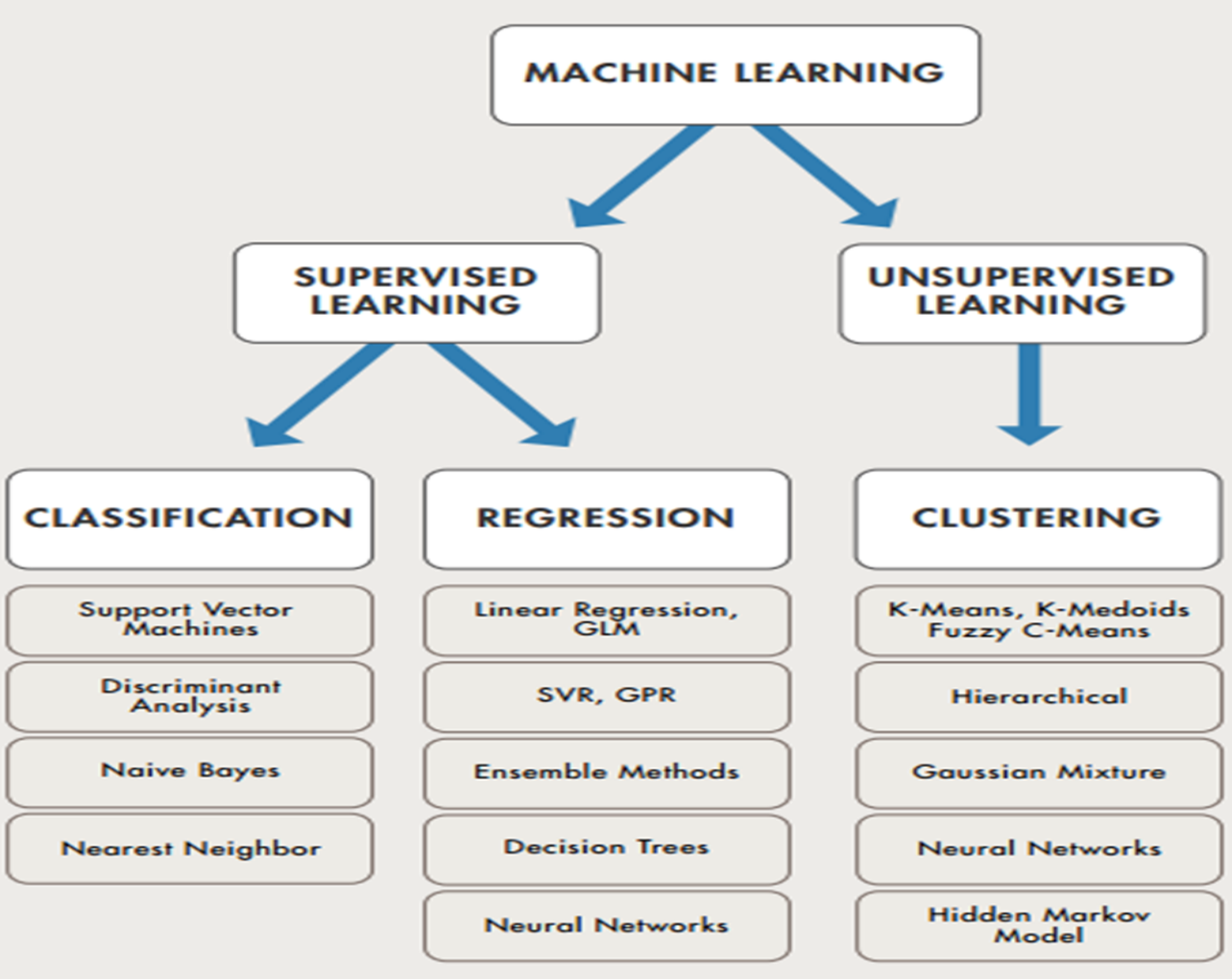
According to Wikipedia [3]; Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to effectively perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model of sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. The name Machine Learning was coined by Arthur Samuel in 1959. Tom M. Mitchell provided a widely quoted, more formal definition of the algorithms studied in the Machine Learning field: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E. This definition of the tasks in which machine learning is concerned offers a fundamentally operational definition rather than defining the field in cognitive terms. This follows Alan Turing's proposal in his paper "Computing Machinery and Intelligence", in which the question "Can machines think?" is replaced with the question "Can machines do what humans can do?". In Turing's proposal the various characteristics that could be possessed by a thinking machine and the various implications in constructing one are exposed.Machine learning tasks are classified into several broad categories: supervised learning, unsupervised learning and reinforcement learning.

- **Supervised and semi-supervised learning:** Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs. The data is known as training data, and consists of a set of training examples. Each training example has one or more inputs and a desired output, also known as a supervisory signal. In the case of semi-supervised learning algorithms, some of the training examples are missing the desired output. In the mathematical model, each training example is represented by an array or vector, and the training data by a matrix. Through iterative optimization of an objective function, supervised learning algorithms learn a function that can be used to predict the output associated with new inputs. An optimal function will allow the algorithm to correctly determine the output for inputs that were not a part of the training data. An algorithm that improves the accuracy of its outputs or predictions over time is said to have learned to perform that task.
- **Unsupervised learning:** Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms therefore learn from test data that has not been labeled, classified or categorized. Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data. A central application of unsupervised learning is in the field of density estimation in statistics

- **Reinforcement learning:** Reinforcement learning is an area of machine learning concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward. Due to its generality, the field is studied in many other disciplines, such as game theory, control theory, operations research, information theory, simulation-based optimization, multi-agent systems, swarm intelligence, statistics and genetic algorithms.[22][23] In machine learning, the environment is typically represented as a Markov Decision Process (MDP). Many reinforcement learning algorithms use dynamic programming techniques. Reinforcement learning algorithms do not assume knowledge of an exact mathematical model of the MDP, and are used when exact models are infeasible. Reinforcement learning algorithms are used in autonomous vehicles or in learning to play a game against a human opponent.

According to MathWorks [4]; common Machine Learning Algoritms grouped under the two broad categories are listed in the schema below:

Figure 1.1: Representation of Machine Learning



Choosing the right algorithm can seem overwhelming. There are dozens of supervised and unsupervised machine learning algorithms, and each takes a different approach to learning. There is no best method or one size fits all. Finding the right algorithm is partly just trial and error, even highly experienced data scientists can't tell whether an algorithm will work without trying it out. But algorithm selection also depends on the size and type of data you're working with, the insights you want to get from the data, and how those insights will be used.

Here are some guidelines on choosing between supervised and unsupervised machine learning:

- Choose supervised learning if you need to train a model to make a prediction. For example, the future value of a continuous variable, such as temperature or a stock price, or a classification. For example, identify makes of cars from webcam video footage.
- Choose unsupervised learning if you need to explore your data and want to train a model to find a good internal representation, such as splitting data up into clusters.

2.1.1: SOME APPLICATIONS OF MACHINE LEARNING

With the rise in big data, machine learning has become a key technique for solving problems in areas, such as:

- Computational finance, for credit scoring and algorithmic trading
- Image processing and computer vision, for face recognition, motion detection, and object detection
- Computational biology, for tumor detection, drug discovery, and DNA sequencing
- Energy production, for price and load forecasting
- Automotive, aerospace, and manufacturing, for predictive maintenance
- Natural language processing, for voice recognition applications

2.2: REVIEW OF RELATED LITERATURE

Several authors have emplyed Machine learning technologies in predicting and trading stock markets. The follwing Algorithms have been employed in the follwing situations:

Because of their ability to model nonlinear relationships without pre-specification during the modeling process, neural networks (NNs) have become a popular method in financial time-series forecasting. NNs also offer huge flexibility in the type of architecture of the model, in terms of number of hidden nodes and layers. Indeed, Pekkaya and Hamzacebi compare the results from using a linear regression versus a NN model to forecast macro variables and show that the NN gives much better results.

Many studies have used NNs and shown promising results in the financial markets. Grudnitski and Osburn implemented NNs to forecast S&P500 and Gold futures price directions and found they were able to correctly prediction the direction of monthly price changes 75% and 61% respectively. Another study showed that a NN-based model leads to higher arbitrage profits compared to cost of carry models [5]. Phua, Ming and Lin implement a NN using Singapore's stock market index and show a forecasting

accuracy of 81% [36]. Similarly, NN models applied to weekly forecasting of Germany's FAZ index find favorable predictive results compared to conventional statistical approaches

More recently, NNs have been augmented or adapted to improve performance on financial time series forecasting. Shaoo et al. show that cascaded functional link artificial neural networks (CFLANN) perform the best in FX markets [6]. Egrioglu et al. introduce a new method based on feed forward artificial neural networks to analyze multivariate high order fuzzy time series forecasting models [7]. Liao and Wang used a stochastic time effective neural network model to show predictive results on the global stock indices [8]. Bildirici and Ersin combined NNs with ARCH/GARCH and other volatility based models to produce a model that out performed ANNs or GARCH based models alone [9]. Moreover, Yudong and Lenan used bactrial chemotaxis optimization (BCO) and back-propagation NN on S&P500 index and conclude that their hybrid model (IBCO-BP) offers less computational complexity, better prediction accuracy and less training time [10]

Another popular machine learning classification technique that does not require any domain knowledge or parameter setting is the decision tree. It also often offers a better visually interpretable model compared to NN, as the nodes in the tree can be easily understood. The simplest type of decision tree model is the classification and regression tree (CART). Sorensen et al. show that CART decision trees perform better than single-factor models based on the same variables in picking stock portfolios [11]. Wang and Chan use a two-layer bias decision tree to predict the daily stock prices of Microsoft, Intel and IBM, finding excess returns compared to a buy-andhold method [12]. Another study found that a boosted alternating decision tree with expert weighing generated abnormal returns for the S&P500 index during the test period [13]. To improve accuracy, some studies used the random forest algorithm for classification, which will be further discussed in chapter 4. Namely, Booth et al. show that a regency-weighted ensemble of random forests produce superior results when analyzed on a large sample of stocks from the DAX in terms of both profitability and prediction accuracy compared with other ensemble techniques [14]. Similarly, a gradient boosted random forest model applied to Singapore's stock market was able to generate excess returns compared with a buy-and-hold strategy [15]. Some recent researches combine decision tree analysis with evolutionary algorithms to allow the model to adapt to changing market conditions. Hsu et al. present constraintbased evolutionary classification trees (CECT) and show strong predictability of a company's financial performance [16].

Support Vector Machines (SVM) is also often used in predicting market behaviors. Huang et al. compare SVM with other classification methods (random Walk, linear discriminant analysis, quadratic discriminant analysis and elman backpropagation neural networks) and finds that SVM performs the best in forecasting weekly movements of the Nikkei 225 index [17]. Similarly, Kim compares SVM with NN and case-based reasoning (CBR) and finds that SVM outperforms both in forecasting the daily direction of change in the Korea composite stock price index (KOSPI) [18]. Likewise, Yang et al. use a margin-varying Support Vector Regression model and show empirical results that have good predictive value for the Hang Seng Index [19]. Nair et al. propose a system that is a genetic algorithm optimized decision tree support vector machine hybrid and validate its performance on the BSE-Sensex and found that its predictive accuracy is better than that of both a NN and Naive bayes based model [20]

While some studies have tried to compare various machine learning algorithms against each other, the results have been inconsistent. Patel et al. compares four prediction models, NN, SVM, random forest and naive-Bayes and find that over a ten year period of various indices, the random forest model performed the best [21]. However, Ou and Wang examine the performance of ten Machine learning classification techniques on the Hang Seng Index and found that the SVM outperformed the other models [22]. Kara et al. compared the performance of NN versus SVM on the daily Instabul Stock Exchange National 100 Index and found that the average performance of the NN model (75.74%) was significantly better than that of the SVM model (71.52%) [23]

CHAPTER THREE: METHODOLOGY

3.1: BACKGROUND TO STUDY AREA

The Dow Jones Industrial Average (DJIA) [24], or simply the Dow, is a stock market index that indicates the value of 30 large, publicly owned companies based in the United States, and how they have traded in the stock market during various periods of time. The value of the Dow is not a weighted arithmetic mean and does not represent its component companies' market capitalization, but rather the sum of the price of one share of stock for each component company. The sum is corrected by a factor which changes whenever one of the component stocks has a stock split or stock dividend, so as to generate a consistent value for the index.

It is the second-oldest U.S. market index after the Dow Jones Transportation Average, created by Wall Street Journal editor and Dow Jones & Company co-founder Charles Dow. Currently owned by S&P Dow Jones Indices, which is majority owned by S&P Global, it is the best known of the Dow Averages, of which the first (non-industrial) was originally published on February 16, 1885. The averages are named after Dow and one of his business associates, statistician Edward Jones. The industrial average was first calculated on May 26, 1896. As at the 31st of December 2018; the Market capitalisation of the Dow Jones Industrial Average is $6.56 trillion. The components are traded in the New York Stock Exchange (NYSE) and NASDAQ.

The choice of this index is due to the availability of high-frequency financial data with high order-to-trade ratios. Alternative Indices that could be used are: S&P 500, NIFTY, HANSENG, CAC 40, etc.

3.2: DATA COLLECTION

One of the 30 Stocks of the Dow Jones Industrial Average (DJIA) based on their historical Sharp Ratios. High Frequency Historical (Minute by minute) Stock Data will be scrapped from Yahoo Finance using a Data Mining Function in Python. Stock prices dataset downloaded include the following features: Date/Time, Open, High, Low, Close, Volume, and Adj. Close. This dataset is dowloaded for the last 2700 trading periods (Minute) consisting of 7 Trading Days.

3.3: DATA ANALYSES

Three stages of Data analyses are conducted: Feature engineering through Technical Analyses, Machine Learning and choice of high performance learning algorithm, forcasts of market trends and application of simple trading strategy.

3.3.1: FEATURE ENGINEERING

Several features are calculated and added to the features listed above on Data collection. These features will be computed using the following Technical Analysis on the stock data downloaded (Open, High, Low, Close, Volume, and Adj. Close). The features are as follows:

- **Trend Indicators:** Average directional index (A.D.X.), Commodity channel index (CCI), Detrended price oscillator (DPO), Know sure thing oscillator (KST), Ichimoku Kinkō Hyō, Moving average convergence/divergence (MACD), Mass index, Moving average (MA), Parabolic SAR (SAR), Smart money index (SMI), Trend line, Trix, Vortex indicator (VI)
- **Momentum Indicators:** Money flow index (MFI), Relative strength index (RSI), Stochastic oscillator, True strength index (TSI), Ultimate oscillator, Williams %R (%R)
- **Volume Indicators:** Accumulation/distribution line, Ease of movement (EMV), Force index (FI) , Negative volume index (NVI), On-balance volume (OBV), Put/call ratio (PCR), Volume–price trend (VPT)
- **Volatility Indicators :** Average true range (ATR), Bollinger Bands (BB), Donchian channel, Keltner channel, CBOE Market Volatility Index (VIX), Standard deviation (σ)

These indicators (features) are computed and included on the data set based on the degree of relaationship (coorelation) or the effects of these features with the movement in stock prices.

3.3.2: MACHINE LEARNING

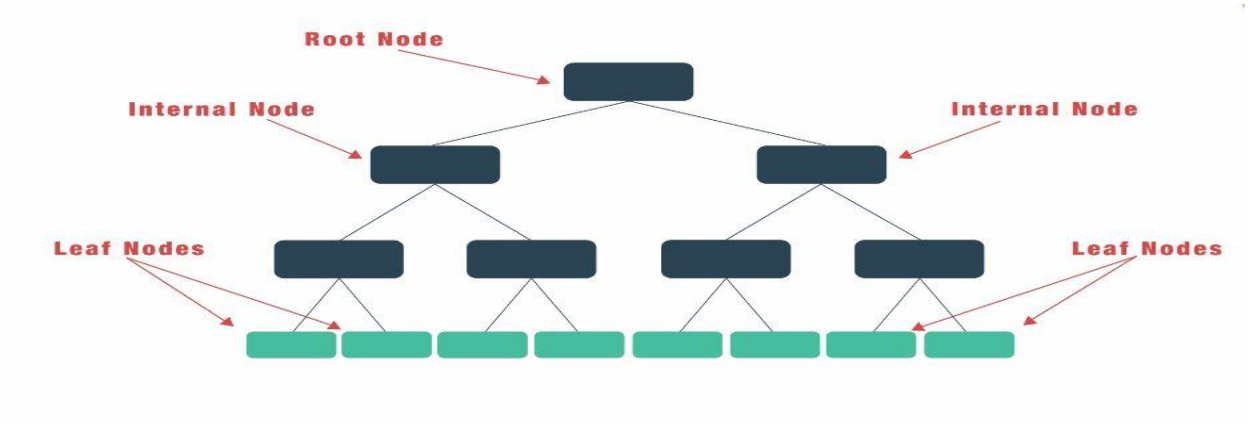
The following Supervised Learning Algorithms (As discussed in quantinsti) [25] will be employed in the forecasting of stock markets:

1. Decision Trees (CART)
2. Naïve Bayes (NB)
3. Support Vector Machines (SVM)
4. Kneighbours(KNN)
5. Random Forest(RF)
6. Linear Decriminant Analyses(LDA)
7. Boosting with Extreme Gradient Boosting(XGBOOST)

Decision trees:

Decision trees are basically a tree-like support tool which can be used to represent a cause and its effect. Since one cause can have multiple effects, we list them down (quite like a tree with its branches).

Figure 1.2: Decision Trees



We can build the decision tree by organising the input data and predictor variables, and according to some criteria that we will specify.

The main steps to build a decision tree are:

1. Retrieve market data for a financial instrument.
2. Introduce the Predictor variables (i.e. Technical indicators, Sentiment indicators, Breadth indicators, etc.)
3. Setup the Target variable or the desired output.
4. Split data between training and test data.
5. Generate the decision tree training the model.
6. Testing and analyzing the model.

The disadvantage of decision trees is that they are prone to overfitting due to their inherent design structure.

Naïve Bayes (NB)

From probability, Bayes theorem was formulated in a way where we assume we have prior knowledge of any event that related to the former event. For example, to check the probability that you will be late to the office, one would like to know if you face any traffic on the way.

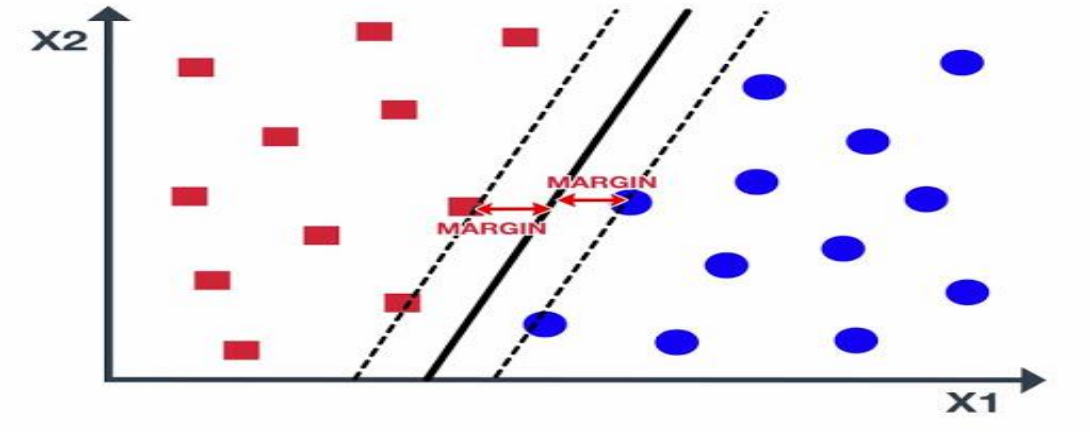
However, Naive Bayes classifier algorithm assumes that two events are independent of each other and thus, this simplifies the calculations to a large extent. Initially thought of nothing more than an academic exercise, Naive Bayes has shown that it works remarkably well in the real world as well.

Naive Bayes algorithm can be used to find simple relationships between different parameters without having complete data

Support Vector Machines (SVM)

Support Vector Machine was initially used for data analysis. Initially, a set of training examples is fed into the SVM algorithm, belonging to one or the other category. The algorithm then builds a model that starts assigning new data to one of the categories that it has learned in the training phase.
In the SVM algorithm, a hyperplane is created which serves as a demarcation between the categories. When the SVM algorithm processes a new data point and depending on the side on which it appears it will be classified into one of the classes.

Figure 1.3: Support Vector Machines Representation

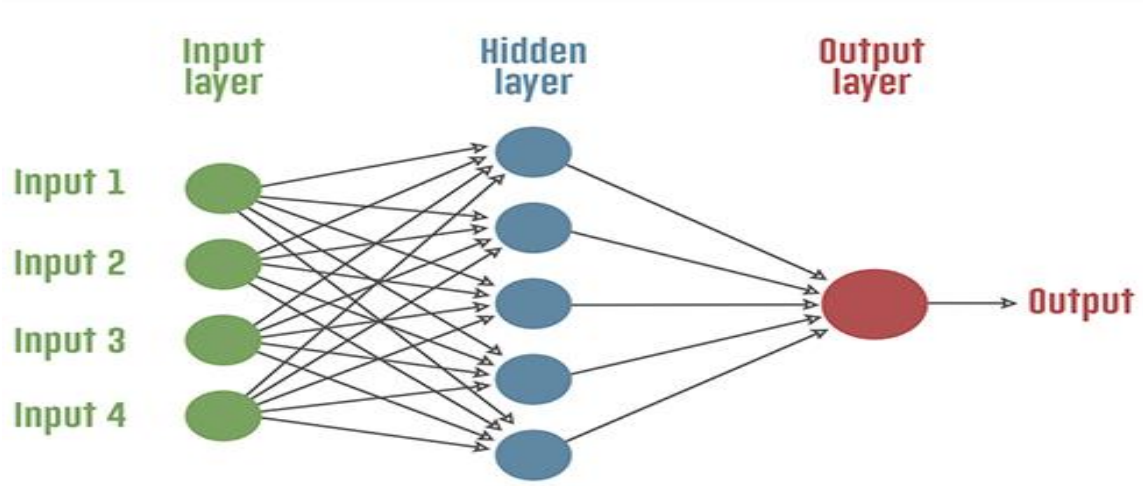


In trading, an SVM algorithm can be built which categorises the equity data as a favourable buy, sell or neutral classes and then classifies the test data according to the rules.

Artificial Neural Network (NN)

In our quest to play God, an artificial neural network is one of our crowning achievements. We have created multiple nodes which are interconnected to each other, as shown in the image, which mimics the neurons in our brain. In simple terms, each neuron takes in information through another neuron, performs work on it, and transfers it to another neuron as output

Figure 1.4: Representation Artificial Neural Network (NN)



Each circular node represents an artificial neuron and an arrow represents a connection from the output of one neuron to the input of another.Neural networks can be more useful if we use it to find interdependencies between various asset classes, rather than trying to predict a buy or sell choice

Random Forest:

A random forest algorithm was designed to address some of the limitations of decision trees.

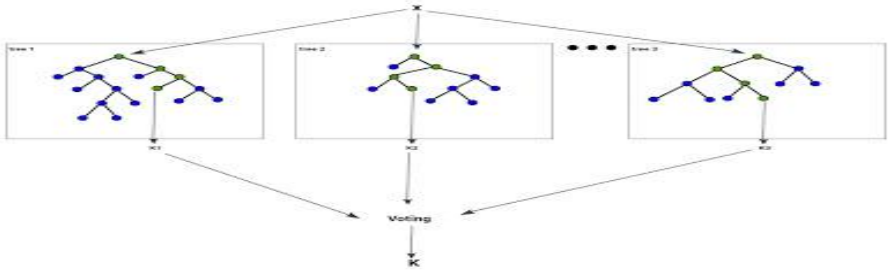
Random Forest comprises of decision trees which are graphs of decisions representing their course of action or statistical probability. These multiple trees are mapped to a single tree which is called Classification and Regression (CART) Model.

To classify an object based on its attributes, each tree gives a classification which is said to “vote” for that class. The forest then chooses the classification with the greatest number of votes. For regression, it considers the average of the outputs of different trees. Random Forest works in the following way:

1. Assume the number of cases as N. A sample of these N cases is taken as the training set.

2. Consider M to be the number of input variables, a number m is selected such that m < M. The best split between m and M is used to split the node. The value of m is held constant as the trees are grown.
3. Each tree is grown as large as possible.
4. By aggregating the predictions of n trees (i.e., majority votes for classification, average for regression), predict the new data.

Figure 1.5: Random Forest Representation:



The Machine Learning Algorithms presented are applied on the High Frequency Stock Data. The performances pf the algorithms in predicting the stock markets will be computed by calculating the F-Score. The F-Score will be ranked and the algorithm with the highest F-Score will be retained for the implementation of the project.

3.3.3: TRADING STRATEGY (BACKTESTING)

1. Buy and hold – the stock is purchased at the opening price on the first minute of the test period and then sold at the closing price of the last minute of the test period.

2. Buy only - since always buying has a slight (52-54%) accuracy advantage over selling, the stock is bought each minute at the opening price and then sold at the closing price. This strategy is repeated each minute, in contrast to the 'buy and hold' approach, which involves a single buy and a single sell event.

3. The model itself is evaluated as follows: if the model predicts the price will close higher, then the stock is bought at the open and sold at the close. If the model predicts the price will close lower, then the stock is sold at the open and bought at the close

3.4: PROJECT IMPLEMENTATION TOOLS

The High Frequency Trading system is implemented in Python 2.7, Anaconda and Jupiter Notebook using the Following Liabraries:

- Numpy for Data analyses
- Pandas for Data Analyses

- Scipy for statistical analyses
- Scikit learn for implementation of Machine learning Algorithms
- Matplotlib and seaborn for graphical representation of results.

3.5: PRESENTATION OF RESULTS

Results are presented on the following figures:

- Heat maps for feature engineering
- Table showing performance matrix of different Machine Learning Algorithms
- Table showing the classification report of the Machine Learning Algorithm retained for the project
- Line graphs showing evolution in performance of different trading strategies
- Performance Matrix showing anualised performance ratios of the Machine Learning Trading system
- A SWOT(Streghth, Weakness, Opportunities and Threats) Analyses of the trading system will also be conducted

CHAPTER FOUR: RESULTS

4.1: HEATMAP OF RELATIONSHIP OF FEATURES OR INDICATORS USED FOR MODELLING

Figure 1.6: Heatmap showing the relationship of various Features or technical indicators for the model.

From the heatmap above it is clear that all the features contribute to the prediction of the target variable (y). The most outstanding features are: The rate of change (ROC), returns (RET), Relative

Strength Index (RSI), Commodity Channel Index (CCI). The target variable (y); represents the increae (1) or the decrease (-1) in stock prices.

4.2: PERFORMANCE OF MACHINE LEARNING ALGORITHMS IN PREDICTION OF STOCK PRICE MOVEMENTS

Eight classification algorithms were used. The accuracy score of the different Machine Learning Models were computed. The results are shown on table 1.1 below:

TABLE I.: ACURACY SCORE OF MACHINE LEARNING MODELS:

	Model	Description	Acuracy_Score
0	LR	Logistic Regression Classifier	0.994307
1	LDA	Linear Decriminant Analysis Classifier	0.888046
2	KNN	K Nearest Neighbours Clasifier	0.827324
3	CART	Decision Trees Classifier	1.000000
4	NB	Gausian naïve Bayes Classifier	0.810247
5	SVM	Support Vector Machines Classifier	0.954459
6	RF	Random Forest Classifier	1.000000
7	XGBoost	Extreme Gradient Boosting Classifier	1.000000

From the result above; all the models achieve considerable level of performance in predicting movements in stock prices. Outstanding models include: Decision Trees (CART), Random Forest (RF), and Extreme Gradient Boosting (XGBoost) with an accuracy score of 100%.

4.2.1: HYPOTHESIS TESTING

The Null hypothesis stipulates that Machine Learning Algorithms cannot predict movements in high frequency stock prices with reasonable level of accuracy:

The Hypothesis was tested in python by applying eight Machine learning models to forecast the direction of movement of high frequency (One minute) stock prices obtained from Yahoo Finance. Theminumum accuracy score of the models is 80% justifying the hypothesis that Machine Learning algorthms could be used to predict stock prices movements in High Frequency Trading setting.

Decision Trees Classifier is thus retained to predict stock prices movements for the purpose of this project.

4.2.2: DETAIL PERFORMANCE OF MACHINE LEARNING ALGORITHM RETAINED

Decision Trees Classifier was used to train 80 Percent of the data set consisting of 2105 data point with 27 features. The model was then use to classify or predict the target(y) consisting of 527 data points. The following classification performance report was produced in table 1.2.

TABLE II.: CLASSIFICATION REPORT OF MACHINE LEARNING ALGORITHM:

Class(y)	precision	recall	f1-score	support
-1	1.00	1.00	1.00	245
1	1.00	1.00	1.00	282
avg / total	**1.00**	**1.00**	**1.00**	**527**

From the table, the model predicts two classes of data points "y"; a predicted decrease in stock price is denoted by "-1" while a predicted increase is denoted by "1". The model predicted 245 decreases and 282 increases for the stock price of IBM tested. This is inline with the target variable (yTest) giving an accuracy score of 100% as denoted by the f1-score. This also justifies the hypothesis that Machine Learning Algorithms can predict the stock market with reasonable accuracy.

4.3: PERFORMANCE OF THE TRADING STRATEGY

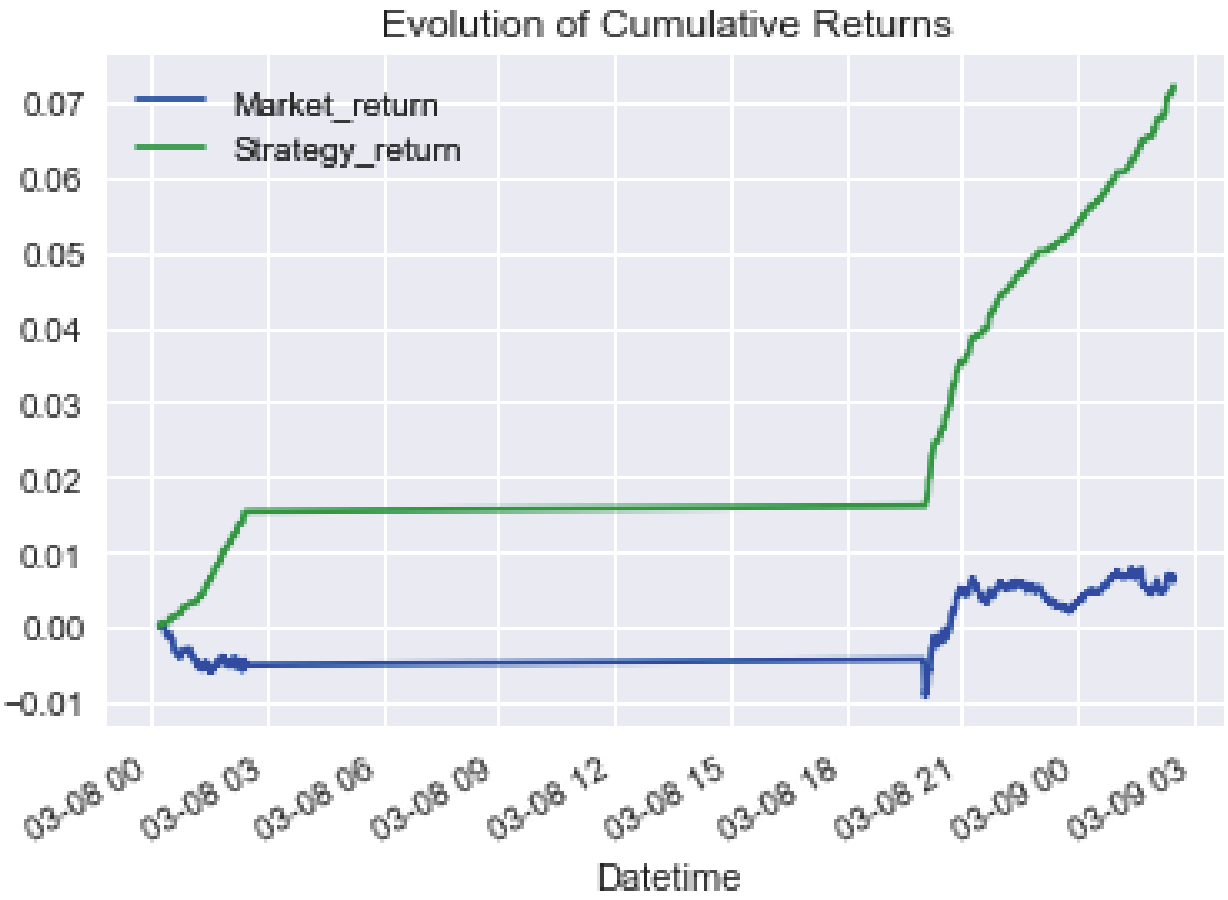
The performance of the trading strategy is measured against the market returns (Buy and Hold strategy). The trading strategy is as follows:

- If the Model predicts an increase in price; we buy at the Open
- If the model predicts a decrease in stock price, we sell the stock.
- The asumption for this strategy is that short selling is allowwed, No transaction cost and there is equal investment.

The performance of this strategy and the market (''Buy and Hold) is presented in the following Line graphs and performance table as follows:

4.3.1: PERFORMANCE EVOLUTION

Figure 1.7: Evolutions of Cumulative Returns:



Evolution of returns shows that at the end of the trading period the cumulative strategy returns is 10 times higher than the market returns. This return is generated from 527 trading operations (282 Purchase orders and 245 sales orders of stocks). These transactions are all carried out within one day (The 8th of March 2019) representing the test set in the prediction model. This is highly diferent from low frequency trading system which could have conducted only one or two trades per stock in a day.

4.3.2: Quantitative performance of the strategy Vs Market

Table III.: Performance of strategy Vs Market:

Indicator	Market_return	Strategy_return
Annualized Return	0.5121	9.5550
Annualized Std Dev	0.0696	0.3508
Avg Loss Return	-0.0038	NaN
vg Win Return	0.0049	0.0379
Gain to Pain Ratio	2.0517	NaN
Lake Ratio	0.1407	0.0000
Loss Rate	0.3245	0.0000
Max Drawdown	-0.4824	0.0000
Sharpe Ratio	7.3529	27.2359
Trade Expectancy	0.0045	NaN
Win Rate	0.6755	1.0000

From the table above; the performance indicators of the strategy are highly superior to those of the market:

1. **The annualised return** of the strategy is almost 10 times higher than that of the market. Hence an investor using Machine Learning system could perform significantly higher than the market
2. **The annualised standard deviation** shows the risk or volatility of the system. Here the volatility of the strategy is significantly higher than the Market. This is due to the fact that several trade orders of thesame stock are carried out in minutes, coupled with orders received from other investors causes the market to be highly volatile. This justifies the fact that High fequency markets are highly volatile.
3. **The Average Loss Return for** the strategy is 0, compared to -0.0038 for the market. This is due to the fact that the strategy model could comfortably predict the stock movements in the period concerned
4. **The Average Win return** is also 0.0379, greater than 0.0049 for the market. This supports the increase in strategy returns morethan the market
5. **The Lake Ratio, Loss rate and the Maximum Draw Down** for the strategy is 0. This surports the positive evolution of cumulative returns as shown on Fig 1.2.
6. **The sharp Ratio** is the ratio of excess returns and volatitility (Annualised standard deviation). The strategy Sharp Ratio (27) is highly superior to that of the Market (7). This is due to the minimal risk (volatility) of the strategy of 0.3. This justifies the fact that the increased strategy returns is due to smart trading strategy, and not an increase in volatility.

7. **The Win Rate** of the strategy is 100% as compared to 65% for the market. This with other indicators shows clearly that Machine Learning Algorithms could effectively predict stock price movements, trade and produce superior return than the market in High Frequency Environment.

4.4: SWOT ANALYSIS OF THE SYSTEM

The strengths, weaknesses, opportunities and threats of the sytem are analysed below:

4.4.1: STRENGTHS

1. Ability the generate several trades(527) in a day using simple machine learning strategy on High Frequency data
2. The system can make use of trading opportunities immidietely as they present themselves in minutes.
3. Ability to generate superior returns about 10 times higher than the market
4. Simple trading strategy based on accurate prediction of market movements using simple Machine Learning Algorithms
5. High Win rate of 100% with a win return per trade of 3%
6. Increased anualised sharp ratio leading to high alpha generation.
7. It ensures "best execution" of trades because and it minimizes the human element in trading decision making.
8. Improves liquidity with lesser Drawdowns
9. The sytem also reduces transactions costs significantly due to limited human interferences
10. The sytem performs significantly well on all the stocks in Dow Jones Industrial Average index and even on stocks out of the index

4.4.2: WEAKNESSES

1. High increase in volatility (from 6% to 35%) due to large number of trades within a limited time frame.
2. The system is not very interative to the user, as the user has to enter functions to generate the data, model and return outputs. opportunities exiists to make the system fully functional and interactive
3. Require huge amount of time in designing the functions and optimising the algorithms.
4. Strict monitoring of the system to avoid to system overuns and failures
5. Deficulties in applying the system to several(morethan one) stock at a time due to deficulties in obtaining free High frequency data
6. Market sentiment indicators were not included as part of the features set. These indicators can greatly influence the market returns
7. Transaction cost and other expenses are not factored into the system further development will include the modules

4.4.3: OPPORTUNITIES

1. Availbility of performant computers, softwares and internet facilities which facilitates the implementation of High Frequency algorithmic trading

2. Availability of programming, application development tools and modules like python, pandas, scikit learn, statsmodels, CNTK, matplotlib, Technical analyses library, etc to facilitate the designing of this project
3. Availability of huge trading opportunities in minutes to take advantage of.
4. Availability of financial markets with regulatory mechanisms (Securities Exchange Commision IN USA) to curtail the effects and imperfections of high frequency algorithmic trading.

4.4.4: THREATS

1. Unavailability of free quality High Frequency data for longer period of times. For this project we could only get one minute data for the last seven trading days.
2. High Volatility could lead to frequent stock market breakdowns and imperfections
3. It requires high testing, monitoring and regulation as error in the system could lead to high lost of capital
4. It requires huge investment for the system implementation and trading.
5. Return margins are very tiny. Low investment and volatility will lead to very low profits and low cash flow.
6. High cost of acquisition of data for the trading sytem on longer time frames.

4.5: Further Research:

Further research will be based on building a fully functional interactive Algorithm trading sytem in the following areas:

- Continue and build a fully functional and interactive trading system with multiple stcoks and portfolios at a time.
- Build a database of historical high frquency stock data for major stock exchanges in the world and populate it with Data for the last five years of trading (I intend to purchase the data).
- Add other technical indicators to the system and enable the system to trade based on technical indicators and machine learning at a time.
- Add other Machine Learning, Learning and Reinforcement Learning algorithms to trade combination of technical indicators and machine learning on huge High frequency stock data
- Add options, commodities, Forex and cryto data on the system for effective High frequency algorithm trading
- Develop and add other backtest functions based on technical indicators.

Bibliography

[1]. High Frequency Trading : https://en.wikipedia.org/wiki/High-frequency_trading

[2]. STOCK MARKET PREDICTION: https://en.wikipedia.org/wiki/Stock-market_prediction

[3]. Machine Learning. https://en.wikipedia.org/wiki/Machine_learning

[4]. Machine Learning https://www.mathworks.com/discovery/machine-learning.html

[5]. Shang-Wu Yu. Forecasting and arbitrage of the nikkei stock index futures: An applicationofbackpropagationnetworks. Asia-Pacific Financial Markets, 6:341– 354, 1999.

[6]. D.K Sahoo, A. Patra, Mishra S.N., and M. R. Senapati. Techniques for time series prediction. International Journal of Research Science and Management, 2, 2015.

[7]. Erol Egrioglu, Aladag Cagda H., and Ufuk Yolcu. Fuzzy time series forecasting with a novel hybrid approach combining fuzzy c-means and neural networks. Expert Systems with Applications, 40:854–857, 2013.

[8]. Zhe Liao and Jun Wang. Forecasting model of global stock index by stochastic time effective neural network. Expert Systems with Applications, 37:834–841, 2010.

[9]. Melike Bildirici and Ozgar O. Ersin. Support vector machine garch and neural network garch models in modeling conditional volatility: An application to turkish financial markets. Expert Systems with Applications, 36:7355–7362, 2009.

[10]. Yudong Zhang and Lenan Wu. Stock market prediction of s&p 500 via combination of improved bco approach and bp neural network. Expert Systems with Applications, 36:8849–8854, 2009.

[11]. Miller-Keith L. Sorensen, Eric H. and Chee K. Ooi. The decision tree approach to stock selection. Journal of Portfolio Management, 27:42, 2000.

[12]. Miller-Keith L. Sorensen, Eric H. and Chee K. Ooi. Stock market trading rule discovery using two-layer bias decision tree. Expert Systems with Applications, 30:605–611, 2006.

[13]. GermanCreamerandYoavFreund. Automatedtradingwithboostingandexpert weighting. Quantitative Finance, 4:401–420, 2010.

[14]. Gerding Enrico Booth, Ash and Frank McGroarty. Automated trading with performance weighted random forests and seasonality. Expert Systems with Applications, 41:3651–3661, 2014.

[15]. Qing-Guo; Li Jin Qin, Qin; Wang and Shuzhi Sam Ge. Linear and nonlinear trading models with gradient boosted random forests and application to singapore stock market. Journal of Intelligent Learning Systems and Applications, 5:1–10, 2013.

[16]. Yuan Lin Hsu, Chi-I; Hsu and Pei Lun Hsu. Financial performance prediction using constraint-based evolutionary classification tree (cect) approach. Advances in Natural Computation, 3612:812–821, 2005.

[17]. Yoshiteru Huang Wei, Nakamori and Shou-Yang Wang. Forecasting stock market movement direction with support vector machine. Computers & Operations Research, 32:2513–2522, 2005.

[18]. Kyoung-jaeKim. Financial timeseriesforecasting using support vector machines. Neurocomputing, 55:307–319, 2003.

[19]. Chang Laiwan Yang, Haiqin and Irwin King. Support vector machine regression for volatile stock market prediction. IDEAL, 2412:391–396, 2002.

[20]. Modhandas V.P Nair, Binoy B. and N.R. Sakthivel. A genetic algorithm optimized decision tree-svm based stock market trend prediction system. International journal on computer science and engineering, 2:2981–2988, 2010

[21]. Shah Shail Thakkar Priyank Patel, Jigar and K. Kotecha. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. Expert Systems with Applications, 42:259–268, 2015.

[22]. PhichhaungouandHengshanWang. Predictionofstockmarketindexmovement by ten data mining techniques. Modern Applied Science, 3:28–42, 2009.

[23]. Boyacioglu Melek Acar Kara, Yakup and Omer Kaan Baykan. Predicting direction of stock price index movement using artificial neural networks and support vector machines: the sample of the Istanbul stock exchange. Expert Systems with Applications, 38:5311–5319, 2011.

[24]. Dow Jones Industrial Average. https://en.wikipedia.org/wiki/Dow_Jones_Industrial_Average

[25]. Machine Learning Algorithms in Finance. https://www.quantinsti.com/blog/top-10-machine-learning-algorithms-beginners

[26]. Trading Strategy: Technical Analysis with Python TA-Lib: https://towardsdatascience.com/trading-strategy-technical-analysis-with-python-ta-lib-3ce9d6ce5614