Investigating Algorithmic Stock Market Trading using Ensemble Machine Learning Methods

Khaled Sharif
University of Jordan *
kldsrf@gmail.com

Mohammad Abu-Ghazaleh
University of Jordan *
mohd.ag@live.com

Ramzi Saifan ǂ
University of Jordan *
r.saifan@ju.edu.jo

* Computer Engineering Department, School of Engineering, University of Jordan, Queen Rania Street, Amman, Jordan.

ǂ Corresponding Author (contact: r.saifan@ju.edu.jo)

ABSTRACT

Recent advances in the machine learning field have given rise to efficient ensemble methods that accurately forecast time-series. In this paper, we use the Quantopian algorithmic stock market trading simulator to assess ensemble method performance in daily prediction and trading. The ensemble methods used are Extremely Randomized Trees, Random Forest, and Gradient Boosting. All methods are trained using multiple technical indicators and automatic stock selection is used. Simulation results show significant returns relative to the benchmark and large values of alpha are produced from all methods. These results strengthen the role of ensemble method based machine learning in automated stock market trading.

JEL CLASSIFICATION CODES

- G170 (Financial Forecasting and Simulation)
- C150 (Statistical Simulation Methods: General)
- C630 (Computational Techniques; Simulation Modelling)

KEYWORDS

machine learning; stock price prediction; ensemble methods; gradient boosting; extremely randomized trees; random forest; stock market simulation; algorithmic trading; financial forecast; forecasting returns; risk analysis; volatility forecasting.

GLOSSARY OF TECHNICAL TERMS

1.	**Asset Ratios**	The ratio of company's total sales relative to the value of their assets.
2.	**Liquidity Ratio**	Determines company's ability to pay off its short-term debt obligations.
3.	**Debt Ratio**	Describes the financial health of the company. Determined by dividing total liabilities by total assets.
4.	**Algorithm**	Computer program consisting of a set of instructions to achieve a well-defined task in a finite number of steps.
5.	**Classifier**	Algorithm responsible of classifying new observed data into a set of categories.
6.	**Ensemble Methods**	Methods that use multiple learning algorithms to obtain a better performance than any of the constituent algorithms.
7.	**Exponential Moving Average**	The average of the stock prices over a defined number of time periods, giving more weight to more recent prices.
8.	**Clustering**	A method, algorithm, that assigns a set of observations into subsets 'clusters', in which observations in the same cluster are similar in some sense. Different method than classifiers but used for the same purposes.

1. INTRODUCTION

Predicting the stock market has been the ultimate goal of stock investors since its existence. Everyday billions of dollars are traded in stock markets around the world, and behind each dollar is an investor hoping to profit by correctly forecasting the rise or fall of the associated stock price.

If an investor somehow predicts that a stock price will rise, he will buy a certain amount of that stock, wait for a specified period of time, and then sell those stocks at their increased price; this method of trading is referred to as *longing*. It is also possible for the investor to profit from the decrease of a stock through a different process called *shorting*; this is when the investor predicts that a stock will fall, borrows a certain amount of that stock and sells them, buys the same amount of stocks after their price has decreased, then returns the stocks he has borrowed to the lender.

Longing and shorting stocks combined with an accurate way of stock market price forecasting makes it possible for an investor to profit from any change in the stock market. This creates a dire need for strong prediction methods. There are various ways for stock price prediction; they basically fall into two categories, either Fundamental Analysis (FA) or Technical Analysis (TA). Many experts use a combination of the two for finer predictions.

For decades, investors have been using a human-based prediction method called *fundamental analysis* (FA); this technique involves acquiring all the relevant information that a person can collect about a certain stock in order to determine its "*true value*". It goes into the economics of the company itself, such as sales and profit data. External factors are also taken into consideration, such as politics, regulations, and industry trends [1]. Methods that aid an investor in FA include financial statements, asset ratios {1}, liquidity ratios {2}, debt ratios {3}, market value ratios, and portfolio management [2]. Based on the determined true value, the investor will decide what sort of position to take with the stock; if it is overpriced the investor will *short* the stock, or *long* the stock if it is underpriced, under the belief of the investor that the price will eventually fall or rise respectively to meet its *true value*). One of the limitations of FA is that it has been practiced for decades without any unifying theoretical framework [3]. Since it lacks a solid mathematical foundation, there is an emotional factor that may cause the investor to make the wrong decisions.

The second method used is called *technical analysis* (TA). It is a method that does not take into account anything about the company, because the investor is interested only in short term movements in the stock price. It concentrates on the movement of stock prices; by examining past stock price movements, future stock price can be accurately predicted. Investors that use TA believe that all the information you need to know about a stock, and the stock prices future movement, is embedded in its historical data. Based on visual examination of the historical data such as price changes and volume of transactions, usually in graphical form and charts, trading advice can be provided [4]. The volume of a stock is the total number of shares that are traded in a security during a certain period of time, and a security with higher volume means that it is more active. TA can use the fluctuations in a stock's volume and price over a certain period of time to try to determine the future movement of the price.

With new advances in technology, and the emergence of high speed computing, computer programs can automatically run complex TA methods on big amounts of historical data and automatically trade stocks based on the program's inferred predictions. This entire workflow is known in the financial industry as *algorithmic trading* (AT) [32]. AT has revolutionized the market and the way financial assets are traded after it became popular in the early 2000s. Investors wanted to make sure to use all the tools that can be offered from the increasing technological advancements, which will place them in a better position to address the changing market environment [5]. Figure 1 shows the trend of using AT through years 2003 to 2012. These estimates include even investors that do not directly deal with the AT program, but deal with a stock broker who eventually will use an AT program to place the order on the stocks required.

The concept of automated prediction is known in the world of computer science as *machine learning* (ML), and is a term that relates to the construction of algorithms that can learn from and make predictions on data. The emergence of strong machine learning methods that can accurately identify stock market patterns and predict the future movement of a stock has led to a surge in research in AT based on ML methods.

The increasing usage of AT makes perfect sense, and this can be accredited to multiple reasons: firstly, the use of AT completely eliminates any emotional and psychological factors that might affect any trade undertaken by an investor. Secondly, placing orders through an AT system occurs instantly with precision and accuracy. Thirdly, AT allows the investor to monitor huge amounts of stock market and financial data in real-time, without the risk of manual human errors. Lastly, because of the programmatic nature of AT systems, simulating an algorithm on large amounts of historical data provides relatively accurate [1] indication of portfolio performance. A paper published by Hendershott et. al. (2007) studied the effect of AT on the New York Stock Exchange (NYSE). In it they concluded that AT most likely causes an improvement in market liquidity [6]. Another study on the foreign exchange market concluded through evidence that due to AT programs being highly correlated to each other, the use of AT had reduced volatility in the stock market [7].

The use of ML in AT has been met with resistance by economists due to three main reasons: firstly, the complexity of ML methods from the perspective of fields other than computer science; secondly, the random nature of a machine learning method and the inconsistency in its prediction results; thirdly, the insufficient amount of published academic work (in the area of stock market prediction) that include AT simulations showing the predictions being undertaken in live trading.

However, the ML methods currently being investigated rarely perform well enough (i.e.: make enough accurate predictions to be considered profitable) for them to be used in real trading situations. Existing methods also suffer from low returns over long trading periods, making them less attractive to traders when compared to existing algorithms reliant on human predictions.

The problem with currently published research that attempts to investigate AT that uses ML for prediction is either the results are undesirable, or that no simulation is included in the results, or both. This lack of research is, in the opinion of the authors, the main reason stopping the widespread use of machine learning prediction in stock market trading today. This paper will thoroughly investigate using efficient ML techniques to accurately predict the future movement of a stock, taking into account the three reasons for resistance mentioned above. It is therefore the chief goal of this paper to encourage the economic world to undertake AT using new ML methods, by providing them with solid, consistent, and repeatable simulations.

In this paper, we will focus on three new ML methods, namely Gradient Boosting, Random Forest, and Extremely Randomized Trees; we have chosen these methods because they have all been published recently in the ML world. Moreover, these methods have been tested before on time series prediction and have shown accurate prediction results even with noisy data (i.e.: data that fluctuates randomly) and very large datasets (i.e.: datasets that are too large for weaker ML methods to work on in sufficient time).

1 When compared to manual investing approaches, algorithmic trading is more likely to produce a similarly performing result given the same data, and this is because it depends on a series of steps rather than an investor's intuition.

To simulate these ML methods in AT, we will use Quantopian, a browser-based AT platform that can be used to write trading strategies in Python [33] and back-test them against 13 years of minute-level US stock price and fundamental data. In each simulation, the returns of the algorithm {4} are compared with a suitable benchmark, and performance is evaluated according to eight evaluation methods. Our simulation results will prove to the readers that using our suggested ML methods in AT will consistently provide better revenue than the benchmark.

In the Literature Review section, we will review the state of the art literature and academic research that revolves around AT, and the application of ML methods into AT. In the Trading Strategy section, we will discuss how the ML model is created and trained, how stocks are automatically selected during the AT process, and briefly go over some simulator settings. In the Methodology section, we will go over the performance indicators that will be used to judge how well the ML methods perform relative to the performance of known financial benchmarks. Finally, in the Results section we will compare and comment on the simulation results of the ML methods when using Quantopian.

2. LITERATURE REVIEW

While academic journals are filled with projects discussing stock trading techniques [28] [29] [30], the world of algorithmic trading is relatively new, and therefore the application of machine learning to algorithmic trading is the new trend of academic research [31] [32]. The three machine learning techniques, interchangeably referred to as classifiers {5}, we will be using are the Gradient Boosting [25], Random Forests [26], and Extremely Randomized Trees algorithms [27].

The Gradient Boosting algorithm produces a prediction model that is in the form of an ensemble {6} of weak decision tree prediction models, also known as *estimators*. [8] The Random Forest and Gradient Boosting algorithms are much related, because both of the algorithms are techniques for regression and classification problems by constructing a multitude of decision trees. [9] The Random Forest algorithm is easier to tune than the Gradient Boosting algorithm, although the Gradient Boosting algorithm will, in general, outperform Random Forests with proper tuning. This is because the Gradient Boosting algorithm attempts to add new trees that compliment the already built trees, and usually this produces better accuracy with fewer trees. The Extremely Randomized Trees algorithm is one step further than the Random Forests algorithm in the way it chooses to split each node in the decision tree during the construction of the decision tree and how the parameters for the node is computed. [10]

Algorithmic trading (AT) is using the computational power at our disposal in the stock market. Computers programmed with a specific set of instructions, large amounts of data, and mathematical models that decide how to trade in a speed and frequency that humans are not capable of achieving, in order to generate more profit ruling out human errors and emotions. Multiple studies show the effect of algorithmic trading on the stock market. A study was done from 2001 to 2011 on the stock market and how AT affects it and it showed that it improved liquidity, efficiency, but also increased volatility [24]. However, a paper showed that results were not uniform across different stocks and there were different outcomes under different conditions [11].

Machine learning (ML) has been a hot topic between researchers for its use in a lot of fields. We are concerned with it being used in the stock market to assist investors in trading, by trying to predict the behavior of the stock market through computations of large amounts of historical stock market data. A considerable amount of effort was also put into using Neural Networks as a prediction technique. One of the first papers that attempted to apply that to the stock market was used to predict the index of the Tokyo Stock Market [12]. A much recent paper about Neural Networks used two kinds of neural networks, namely a feed forward Multilayer Perception (MLP) and an Elman recurrent network [23]; the paper concluded that MLP has more potential in predicting stock value changes than Elman recurrent network and linear regression, although a simple linear regression model was better than the other two when it comes to predicting the direction of stock price changes one day ahead. [13]

Another paper proposed a model that combined the Support Vector Machine algorithm with other classification methods, in a way such that the weakness of a method will be balanced out by the strength of another (i.e., early attempts at ensemble methods in stock market prediction). [14] Papers that used technical indicators for their machine learning methods typically computed the Exponential Moving Average (EMA) {7} and compared it to the stock markets, specifically using the Google and Yahoo stocks (NYSE: GOOG and NASDAQ: YHOO); in one particular paper, the authors suggested using other indicators as they believe that might provide more accurate results instead of just using the EMA. [15]

Results from papers in the field have been both positive and negative towards the idea of using ML in AT. An example of a paper that was negative towards the idea used ML to facilitate *automated stock portfolio optimization*; the authors used the Dow Jones Industrial Average Index as a benchmark; they concluded that none of the techniques they used outperform the index, mainly because the index resulted in more returns at a lower risk than their proposed method. [16] An example of another paper that was positive towards the idea used a method that consisted of linear regression, generalized linear model, with the aid of the Support Vector Machine algorithm, to predict future stock market prices; results were desirable and they generated a higher profit than the selected benchmark. [17] Another positive paper proposed a stock price prediction system also based on the Support Vector Machine algorithm and was tested on the Taiwan stock market; the method performed better than conventional stock market prediction systems (in terms of accuracy). [18]

More advanced papers have used hybrid combinatorial methods of clustering {8} and classification. One of these papers first applies a clustering algorithm such as K-Nearest Neighbors and partitions the clustered values into number of parties, and then applies a horizontal partition based decision tree algorithm; the paper used the algorithm on data from the Shanghai Stock Exchange and their predicted results were very close to the actual values. [19]

In this project, we will compare our efficient ensemble methods with the K-Nearest Neighbors and the Support Vector Machine algorithms as a way of comparing our methods to those used in previous literature. Our simulation results will show that our efficient ensemble methods outperform those used in previous literature in predictive accuracy.

The use of Quantopian in academic research is rare; one of the few papers to use it begins with an explanation of the Efficient Market Hypothesis [2] and Self Defeating Strategies [3], and uses these two ideas to reason why there aren't enough academic papers showing positive results predicting the market using machine learning; in the author's opinion, if a model succeeds and is distributed to the public, it will not be successful for too long. The author also

2 In financial economics, the efficient-market hypothesis states that current stock prices fully reflect all available information. It is therefore, according to the hypothesis, impossible to find a pattern in stock price movement.
3 A self defeating strategy is a term used for a strategy that will eventually stop working (or reduce in effectiveness) after it is applied to the stock market.

used different methods of trading using Quantopian and machine learning, but showed that results were undesirable. [20]

As we can see from the aforementioned literature, there have been many different techniques tried and tested in an attempt to predict the stock market and automate stock market trading. All methods used different algorithms, factors, and parameters that could be tuned to deliver better results. In this paper, we will use what we consider to be the latest machine learning methods to try and produce positive results in prediction and simulation.

3. TRADING STRATEGY

3.1: MODEL CREATION

In this section, we will explain the trading strategy that we will simulate. It is coded entirely in the Python language and it runs on the Quantopian simulator. As mentioned earlier, we will use three machine learning methods for our daily predictions. The classifiers used are the Gradient Boosting, Extremely Randomized Trees, and Random Forest classifiers, and they are all part of the open source *scikit-learn* library. [21]

Creating the model is the first step in the algorithm, and the model creation is scheduled to happen at the beginning of every month throughout the simulation period. It is created by training the classifier [4] data based on the previous 1000 days (which we define as the history range) relative to the model creation date, and based on this data we generate features, namely the Average True Range (ATR) and the Bollinger Bands (BB). The ATR is a measure of the volatility for the stocks: it is calculated through a 14-day period by finding the moving average of the “true range”. Simply put, if stocks are experiencing high volatility, then they would have higher ATR, and they will have lower ATR at lower volatility, and the difference between the maximum and minimum moving average is deemed the true range. The BB is another popular method to measure volatility: the prices of the stock along with a ten-day period moving average are banded by an upper band and a lower band, and the bands keep changing according to the market conditions. A wider band from the moving average means that the stock price is becoming more volatile, whereas tighter bands mean that the volatility is decreasing. If stock price moves closer to the upper band, this means that the stock is being overbought, and the stock is being oversold if the prices are moving closer to the lower band.

4 In machine learning, creating a model by training a classifier means that we feed the classifier with historical data to ‘train’ on. The created model will decide which class to allocate the newly observed data based on previous data.

The following list outlines the organization of the features and the predicted target, before being used to train the classifiers. There is a total of 89 features [5], and the 90th column contains the target to be predicted by the classifier. The value of the prediction target is a function that is detailed after the following table, in Equation 1. The feature organization in the dataset used to train the classifiers is as follows:

- Price Changes
- ATR Upside Signal
- ATR Downside Signal
- Upper Bollinger Band
- Middle Bollinger Band
- Lower Bollinger Band

The following equation is used to determine the target for prediction.

(eq. 1)

$$\text{PCT}(p) = \begin{cases} +1 & \text{If p is greater than a certain percentage of the price change the day before, and is positive} \\ 0 & \text{If p is within a certain percentage of the price change the day before} \\ -1 & \text{If p is greater than a certain percentage of the price change the day before, and is negative} \end{cases}$$

(where **p** is the price change for tomorrow)

3.2: AUTOMATIC STOCK SELECTION

The algorithm is also able to choose certain stocks automatically every month, and therefore fully automates the trading process and keeps our simulations free of survivorship bias. The selection is based on fundamental data [6], and it does so by filtering according to a stock's Price-to-Earnings Ratio (PER) and Market Capitalization (MC). The PER of a stock is measured by dividing the current share price over its earnings per share, and this is used as an indication of the value of the company. MC is calculated by multiplying the current market price of one share with the company's total number of shares, and this shows the total market value of the shares in a company.

5 The selection of 89 features is arbitrary. The number 89 comes from six technical indicators, each of which has a 14-day period. The use of a 14-day period is also arbitrary.
6 The fundamental data of a stock is in the broadest terms any data, besides the trading patterns of the stock itself, which can be expected to impact the price or perceived value of a stock.

3.3: LONGING AND SHORTING STOCKS

The final stage of the trading strategy of our AT program is the longing and shorting of the selected stocks and the program is scheduled to long and short stocks daily (i.e.: every trading day in the NYSE during the selected time period). Our AT programs will use two different techniques to base our trading on: the first technique is using one classifier and the other is using two classifiers working simultaneously. If one classifier is used, the algorithm will long or short based on how sure the predictor is of its prediction, and we specify that it should be more than a certain value (defined as the minimum probability) for the AT program to take the appropriate action of longing or shorting. When two classifiers are used, the AT program takes action when both predictions are the same. The actions taken by either of the classification methods is outlined in detail in Table 2 below.

Table 2: This table outlines the workflow for each of the two trading strategies.

	Output of One Classifier	***Outputs of Two Classifiers***	***Action taken by the AT program***
Workflow of the trading strategy given the output of either one or two classifiers	Classifier predicts increasing price with strong probability.	Both classifiers agree on an increasing price prediction.	Begin longing the stock. If we are already shorting the stock (betting that it will decrease), stop trading the stock.
	Classifier predicts decreasing price with strong probability.	Both classifiers agree on an increasing price prediction.	Begin shorting the stock. If we are already longing the stock (betting that it will increase), stop trading the stock.
	Classifier either predicts no change with strong probability or predicts any outcome with weak probability.	The classifiers either agree on no change in stock price or disagree on a prediction.	Make no changes to our ongoing action with the stock.

3.4: SLIPPAGE AND COMMISSION

For all simulations in this project, we are using the default *slippage* and *commission* models that are being used on the Quantopian simulator. *Slippage* calculates and simulates the impact of our order on the market, and it is measured by assessing how large our order is in comparison with the current trading volume; this is used to check if

an order is too big (given that a trader cannot trade more than the market's volume at any given time); therefore, our algorithm will be limited to ordering up to 2.5% of the total available stocks, a percentage defined by the simulator to make the simulation results more realistic. The *commission* is set to $0.03 dollars per share, as is the default on the simulator.

4.: METHODOLOGY

4.1: TESTING THE CHOSEN MACHINE LEARNING METHODS IN PREDICTIVE ACCURACY

Before beginning to trade with the model predictions, it is better to first test the accuracy of the algorithms in predicting the future stock price movement. This would give us a better understanding of each algorithm's performance in prediction only, and lets us tune the algorithm's parameters to get better accuracy. Quantopian provides a research environment to experiment and try out trading strategies without running them through a simulator. We will assess the accuracy of each algorithm by using a confusion matrix (a table that counts the predictions that were classified and misclassified) and repeat each simulation multiple times to gain confidence in the results. Each algorithm has its own set of adjustable parameters, which we will try to fine-tune to attain the best accuracy from each algorithm.

4.2: PERFORMANCE INDICATORS

After finding the best parameters for an accurate prediction of stock price movement, we can move our algorithms from the research environment to the simulation. The algorithms will be part of the larger work-flow, which was discussed in detail in the previous section. We define a certain time-period (greater than two years) for the simulation to run through day-by-day, and a fixed starting capital of 1 million US dollars. At the end of each simulation, the algorithm's performance is assessed automatically through eight performance indicators that are usually used to assess and compare different trading strategies together; they are outlined in Table 3 below.

Table 3: The table below describes in details each of the eight performance indicators the simulator produces.

Name of the performance indicator	*Brief description of the indicator*
Algorithm Returns	Cumulative returns (as a percentage) of the algorithm relative to the starting capital at the beginning of the simulation
Alpha	The return on an investment that is not a result of general movement in the greater market.
Beta	The tendency of the algorithm's price movement to respond to swings in the market. A beta value of 0 means the algorithm is uncorrelated to the market, and in some sense is risk-free.
Sharpe Ratio	A measure for calculating risk-adjusted return; it is defined as the average return earned in excess of the risk-free rate per unit of volatility or total risk.
Sortino Ratio	A modification of the Sharpe ratio that differentiates harmful volatility from general volatility by taking into account the standard deviation of negative asset returns (downside deviation). A large Sortino ratio indicates that there is a low probability of a large loss.
Information Ratio	A ratio of portfolio returns above the returns of a benchmark (usually an index) to the volatility of those returns. The information ratio (IR) measures a portfolio manager's ability to generate excess returns relative to a benchmark, but also attempts to identify the consistency of the investor.
Volatility	An identification of price ranges and breakouts; the ratio uses a true price range to determine an algorithm's true trading range and is able to identify situations where the price has moved out of this true range.
Maximum Draw-down	The maximum draw-down experienced by the cumulative returns of the algorithm during a certain period of time defined by the simulator.

We will provide the mathematical equations that were used in determining six of the eight performance indicators below. The remaining two indicators (i.e., cumulative returns and maximum draw-down) are considered to be straight forward and will not be explained due to lack of space. We have chosen to consider *the market* to be reasonably approximated by the Standard and Poor 500 index (NYSE: SPY), and *the risk-free rate* to be reasonably approximated by the US Treasury Index (NYSE: BIL).

4.2.1: Alpha and Beta

The values for alpha and beta are found from an equation that is a part of the capital asset pricing model (CAPM), show below.

$$\alpha = R_p - [R_f + (R_m - R_f) \cdot \beta]$$

(eq. 2)

R_p is the realized return of portfolio (this is the portfolio that is being simulated).
R_m is the market return (this can be approximated by a portfolio with only the SPY Standard & Poor 500 stock longed with initial capital).
R_f is the risk-free rate (this can be approximated by a portfolio with only the BIL US Treasury Bill Index stock longed with initial capital).

We find beta first using the following equation, then we substitute it in the previous equation to get alpha:

$$\beta = \frac{Cov(R_p, R_m)}{Var(R_p)}$$

(eq. 3)

Cov(X, Y) is the covariance between the two variables X and Y.
Var(X) is the variance in the variable X.

4.2.2: Sharpe Ratio

$$Sharpe = \frac{Mean(R_p - R_f)}{StdDev(R_p - R_f)}$$

(eq. 4)

R_p is the realized return of portfolio (this is the portfolio that is being simulated).
R_f is the risk-free rate (this can be approximated by a portfolio with only the BIL US Treasury Bill Index stock longed with initial capital).
StdDev(x) is the standard deviation in x, and Mean(x) is the average value of x.

4.2.3: Sortino Ratio

$$Sortino = \frac{Mean(R_p - R_f)}{StdDev(F(R_p, R_f))}$$

(eq. 5)

R_p is the realized return of portfolio (this is the portfolio that is being simulated).
R_f is the risk-free rate (this can be approximated by a portfolio with only the BIL US Treasury Bill Index stock longed with initial capital).
StdDev(x) is the standard deviation in x, and Mean(x) is the average value of x.
F(x, y) is a set of values that only contain the value of x - y when y was greater than x.

4.2.4: Information Ratio

$$Information = \frac{Mean(R_p - R_m)}{StdDev(R_p - R_m)}$$

(eq. 6)

R_p is the realized return of portfolio (this is the portfolio that is being simulated).
R_m is the market return (this can be approximated by a portfolio with only the SPY Standard & Poor 500 stock longed with initial capital).
StdDev(x) is the standard deviation of x, and Mean(x) is the average value of x.

4.2.5: Volatility Ratio

$$Volatility = \frac{T}{EMA_n(T)}$$
$$\rightarrow T = Max(H_t - L_t, H_t - C_{t-1}, C_{t-1} - L_t)$$

(eq. 7, 8)

T is coined the "true range" and is determined by the previous equation.
$EMA_n(T)$ is the exponentially moving average of T over a time period of n days.
H is the highest price a stock reached during the day.
L is the lowest price a stock reached during the day.
C is the closing price of the day for a given stock.
The subscripts of the variables in the true range definition indicate which day the variables are taken from.

4.3: OTHER CONSIDERATIONS

It is a difficult task to compare all trading strategies with only one performance indicator, and during our experimentation we will find strategies that perform well through some indicators but poorly in others, so we will compare algorithms using all indicators and leave it to the investor to decide which algorithms are the most favorable. We will first show how changing the prediction algorithm affects the performance indicators, and then we will show how fine-tuning the different algorithm parameters affect the indicators too. In the conclusion of our simulations and analysis of all the different methods and trading strategies, we will try to find the strengths of each prediction algorithm when applied to a trading strategy and show their defining characteristics when that trading strategy is used in AT.

5. RESULTS

5.1: RESULTS FROM INITIAL TESTING OF PREDICTIVE ACCURACY

We precede the simulations with initial testing in the research environment provided by Quantopian. Using a gradient boosting classifier, we trained the classifier on a normalized SPY index during a certain period of time (between 2006 and 2010), and then used that trained classifier to predict three randomly selected stocks, as shown in

Table 4. The classifier outputs one of the three classes of prediction: the stock price one day from today will either increase by a certain percentage, decrease by that percentage, or stay within that percentage (which we considered to be negligible movement). The Table 4 contains a confusion matrix[7], and it counts the number of predictions and their outcomes, as a way of assessing the performance of the classifier. The result is the average of ten simulations. The matrix in Table 4 yields an accuracy of 57%, and the accuracy is calculated by summing the diagonal of the matrix.

Table 4: The confusion matrix from the best classifier (Gradient Boosting) with fine-tuned parameters (obtained through a grid-search) after predicting 970 stock price movements.

	Stock actually decreased in price	Stock price actually stayed almost the same	Stock actually increased in price
Stock predicted to decrease in price	36%	7%	6%
Stock price predicted to stay almost the same	12%	17%	15%
Stock predicted to increase in price	1%	2%	4%

Following that experiment, we can compare the accuracy of each classifier when used to predict each of the stocks separately. Because there are three classes to predict, we can consider a random guess to be a uniform distribution between the three classes (i.e.: 33%). This is a good reference to be used when comparing the classifiers, because any classifier that has a predictive accuracy below random guessing is not considered useful. Observing the values in Table 5, all classifiers achieve significantly higher accuracy than both the reference and classifiers from previous work (refer to Section 2); this leads us to believe *the market is not random* [8] and we can try to use these predictions in trading.

Table 5: This table provides a useful side by side comparison of the predictive performance each of the three classifiers with fine-tuned parameters, along with two classifiers from previous literature.

	Apple Inc. (NYSE: AAPL)	**JPMorgan Chase (NYSE: JPM)**	**Microsoft Corp. (NYSE: MSFT)**
K-Nearest Neighbors Classification	41.9%	42.6%	48.0%
Support Vector Machine	39.4%	41.2%	42.4%
Random Forest Classification	50.5%	56.6%	51.1%
Extremely Randomized Trees	50.1%	56.0%	50.5%

7 A confusion matrix is a commonly used tool in classification tasks to assess the accuracy of a classifier.
8 We refer to this because there is a popular hypothesis in financial literature named the “Efficient Market Hypothesis”, and it is an investment theory that states it is impossible to find a pattern in the stock market because stock market efficiency causes existing stock prices to always incorporate and reflect all relevant information.

Classification			
Gradient Boosting Classification	52.2%	57.0%	53.1%

In Table 5, all classifiers are trained on the Standard and Poor 500 index (NYSE: SPY) and then used to predict the stock indicated in each column. In most cases, the Gradient Boosting classifier reaches the highest accuracy amongst the three classifiers, reaching almost two times the accuracy of the reference.

5.2: SIMULATIONS USING PRE-SELECTED STOCKS

Following the results presented in Table 5, we have confidence in the prediction ability of our ensemble methods, and we can move these methods onto the trading simulation. The details of the inner workings of the trading algorithm are in the previous section (i.e.: Trading Strategy). As described before, there are two versions of the algorithm, each differing by either using one or two classifiers, and by either using preselected stocks or automatic stock selection. We will present the cumulative returns of each version of the algorithm and discuss them briefly. All methods are compared to the Standard and Poor 500 index (NYSE: SPY) as a benchmark for assessing performance. The time period throughout which the classifiers are simulated is selected based on simulation complexity, and we kept all periods to a minimum of two years, usually starting no earlier than 2010, and all algorithms traded daily. In Figure 1, we will show the cumulative returns of each of the three classifiers when using preselected stocks and the one classifier method.[9] The preselected stocks are a random selection of 36 stocks that were constituents of the Standard and Poor 500 index (NYSE: SPY) during the year of 2010, which is the starting year for all of the simulations in this project.

9 The Gradient Boosting Classifier uses fewer estimators than its counterparts, and this is due to the greater complexity of the algorithm compared to the other algorithms. In general, it takes significantly less processing time to train a Random Forest Classifier or Extremely Randomized Trees Classifier than a Gradient Boosting Classifier with an equal number of estimators for all three classifiers.

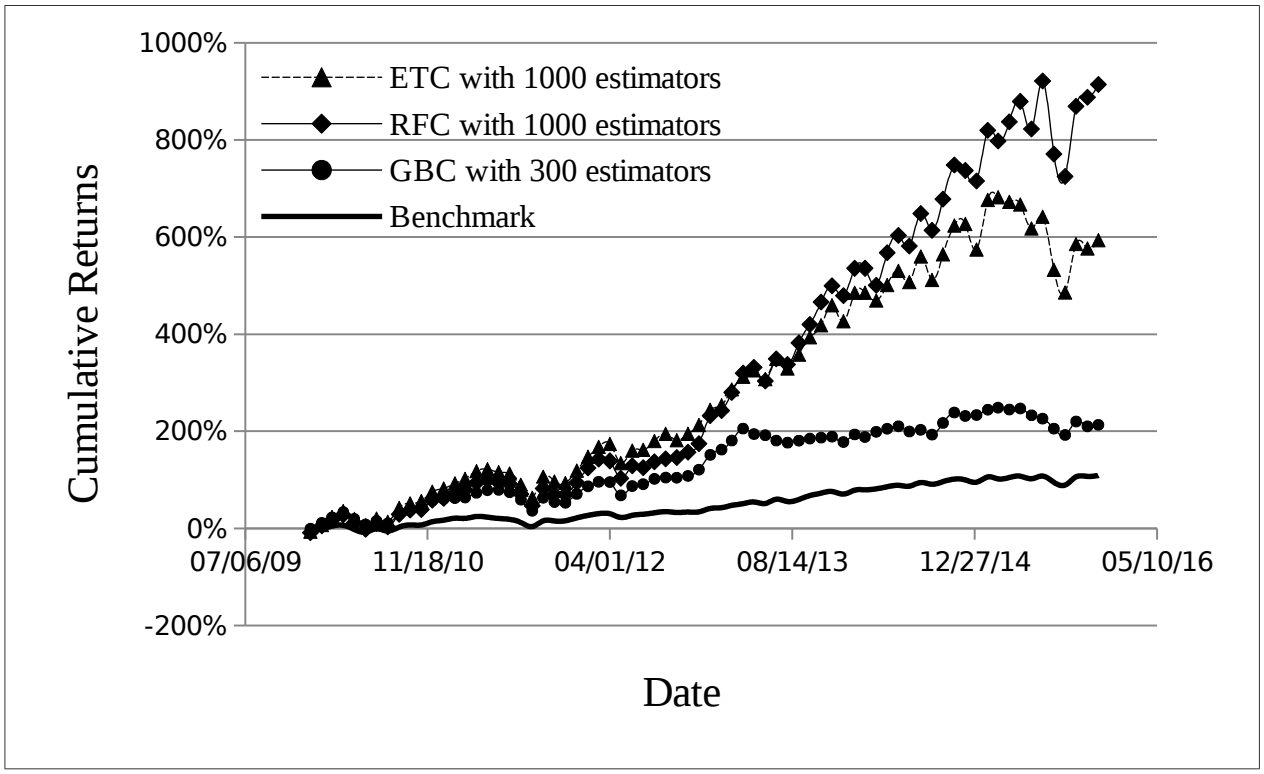

Figure 1: The graph shows the cumulative returns of each of the three algorithms when working with 36 preselected stocks and using one classifier to predict the trading stocks.

It can be deduced from Figure 1 that the Extremely Randomized Trees classifier and the Random Forest Classifier strongly outperform the benchmark, with the Random Forest Classifier having higher cumulative returns towards the end of the period. The Gradient Boosting Classifier under-performs compared to the other classifiers if we compare them using cumulative returns. Gradient Boosting outperforms the other two when it comes to stability and volatility of the simulation.

We will now discuss a problem that may arise when using only one classifier in a trading strategy for stock price prediction. The uncertainty a classifier has in its own prediction, as per the discussed trading strategy, can sometimes lead the AT program to not act upon the prediction. A more complex approach that solves this problem is to use two similar classifiers and only act upon their agreement; we will call this method hereafter the Two Classifier method, and the method that uses only one classifier and a probability threshold will be called hereafter the One Classifier method. The result of this alternative approach, namely the Two Classifier method, is shown in Figure 2 below with preselected stocks only.

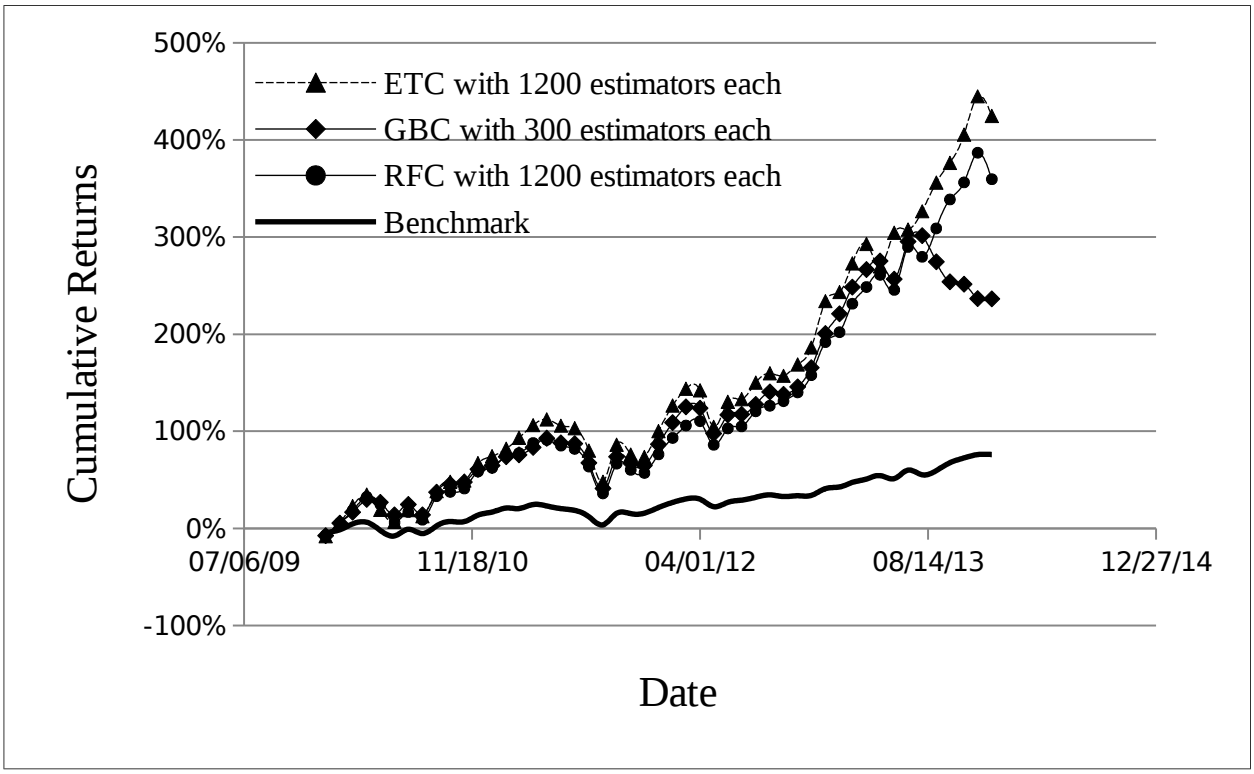

Figure 2: The graph shows the cumulative returns of each of the three algorithms when working with 36 preselected stocks and using the agreement of two classifiers to predict the trading stocks.

The reader can see from Figure 2 that all classifiers easily outperform the benchmark, with the ETC and the RFC classifiers having the best overall cumulative returns. It is worth noting that, compared to the one classifier method the two classifier methods usually have less volatility but also less cumulative returns as well.

5.3: SIMULATIONS USING AUTOMATICALY SELECTED STOCKS

The main problem with using preselected stocks is that we are prone to *survivorship bias*; this means that our selection of stocks manually uses our information of the future, relative to the time of the simulation. Our proposed solution to survivorship bias is to let our AT program automatically select stocks every month using basic fundamental analysis. The selection scheme for the 100 stocks we used was a simple one based on the stock's Price-to-Earnings Ratio (PER) and Market Capitalization (MC), and all stocks are selected from the NYSE and NASDAQ exchanges. Our selection used stocks the first 100 stocks that were above $100 million in MC, had a PER less than 10, and were sorted by their MC value in descending order. Our choice of filter values for the automatic stock selection was based on trial and error through multiple simulations.

At the beginning of each month, the available stocks are filtered and reselected, ensuring that the algorithm has the best selection of stocks to analyze and trade. The two graphs that follow, Figures 3 and 4, will show the cumulative

returns over time when using the one classifier and two classifier methods respectively with automatic selection of trading stocks.

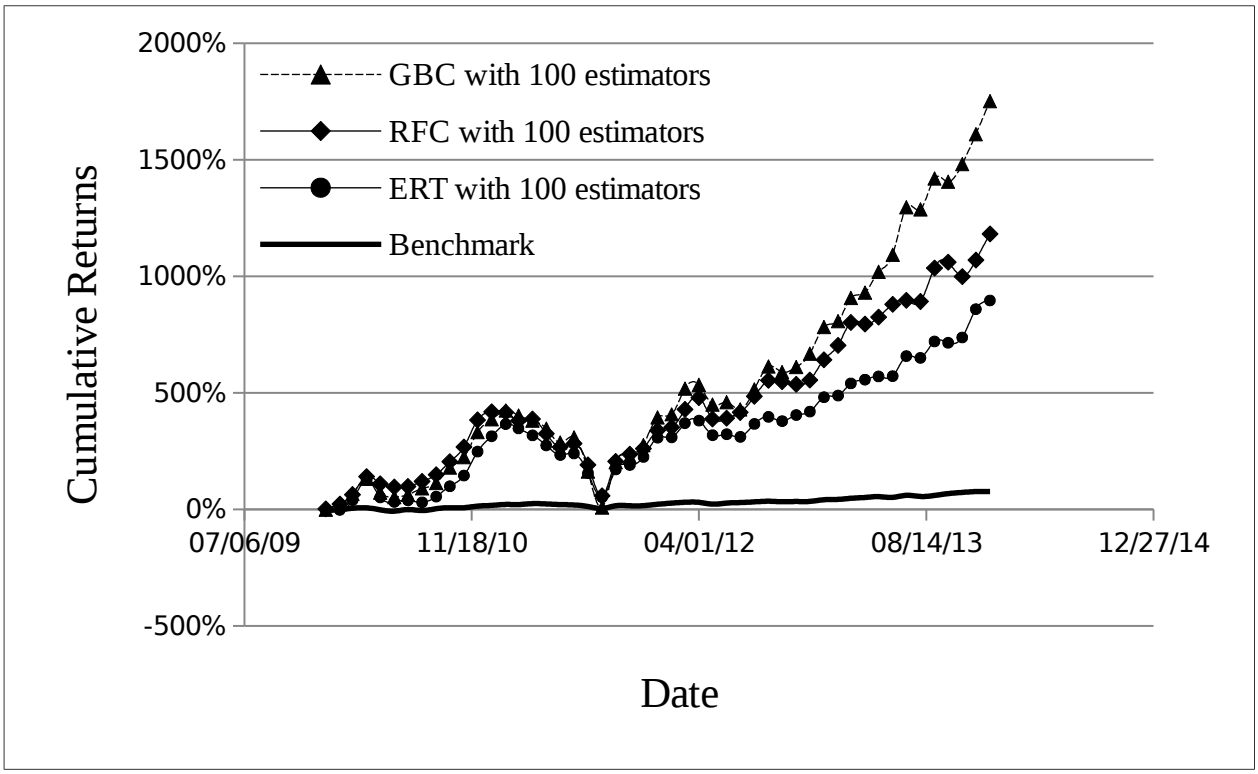

Figure 3: The graph shows the cumulative returns of each of the three algorithms when working with 100 automatically selected stocks (selected at the start of each month) and using one classifier to predict the trading stocks.

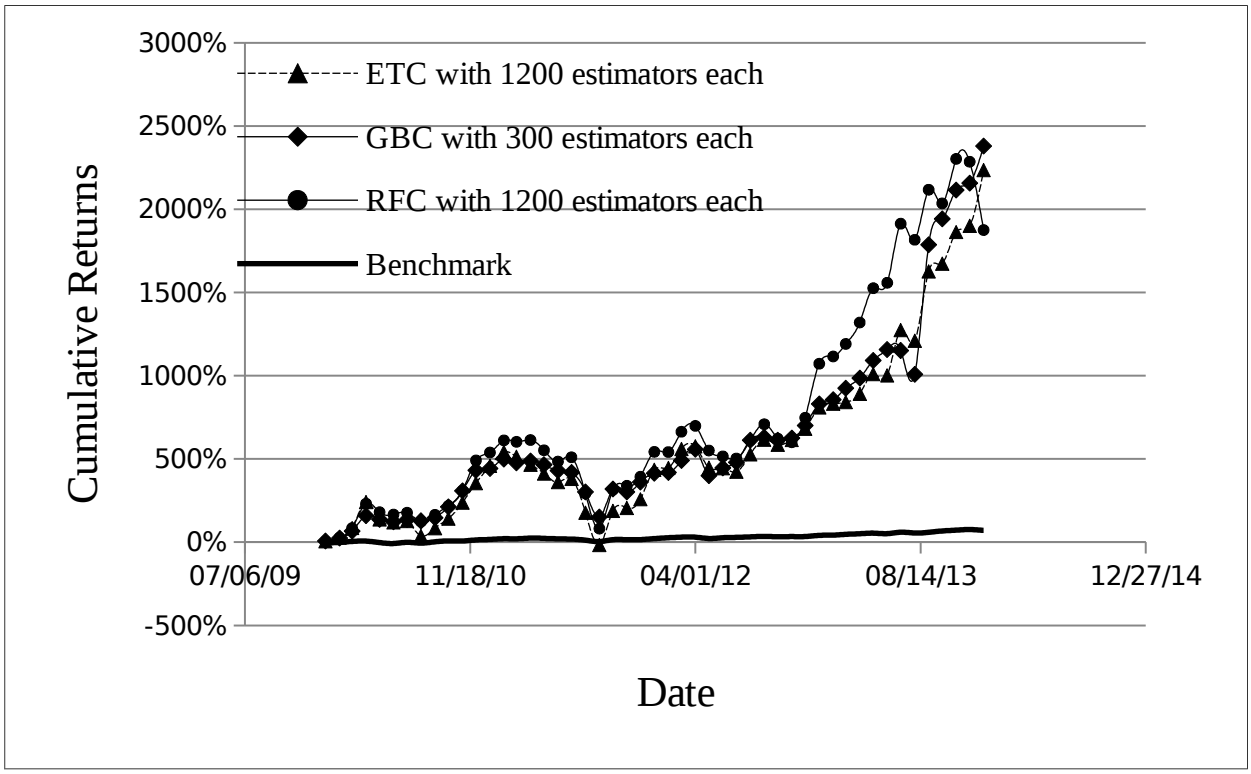

Figure 4: The graph shows the cumulative returns of each of the three algorithms when working with 100 automatically selected stocks (selected at the start of each month) and using the agreement of two classifiers to predict the trading stocks.

In Figure 3, we can see that all three algorithms seem to be prone to a heavy decline between 2011 and 2012, and we will see that this decline is less prominent in the two-classifier method. The Gradient Boosting Classifier outperforms the other two classifiers considerably over the time period following the mass decline. All three classifiers outperform the benchmark significantly.

In Figure 4, it should be noted that the maximum possible number of estimators for each classifier was selected, which is why the more complex Gradient Boosting algorithm has less estimators then the simpler Extremely Randomized Trees and Random Forest algorithm. Nonetheless, they all achieve similar performance, although the Gradient Boosting algorithm seems to outperform the others and maintains less volatility, especially towards the end of the time period.

5.4: SIMULATIONS WITH PARAMETER ADJUSTMENTS

Following the results that were observed in the previous section, we will now move on to investigate how fine-tuning the classifier parameters affects cumulative returns of the program. We chose to change two parameters in the

Gradient Boosting Classifier, namely the number of estimators in the classifier, and the maximum depth of each estimator in the classifier. It can be seen from Figure 5 that an increase in number of estimators yields better cumulative returns and more resistance to negative changes in the market.

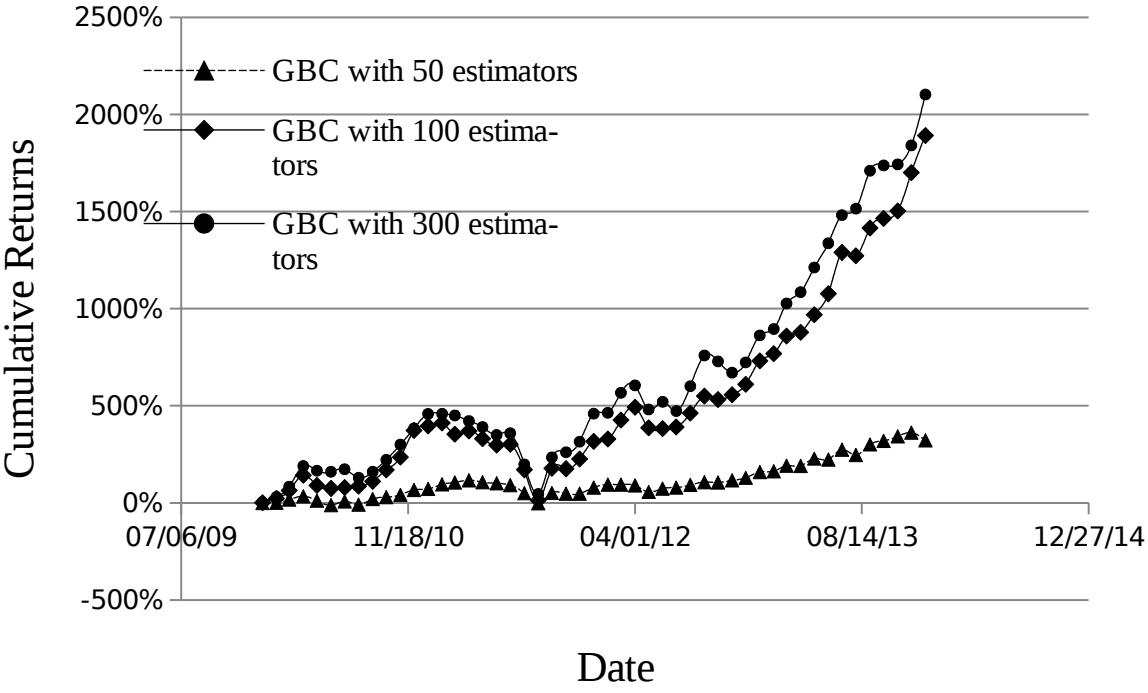

Figure 5: the graph shows the change in cumulative returns as we change the number of estimators (trees) in the gradient boosting classifier when only one classifier is used and stocks are automatically selected.

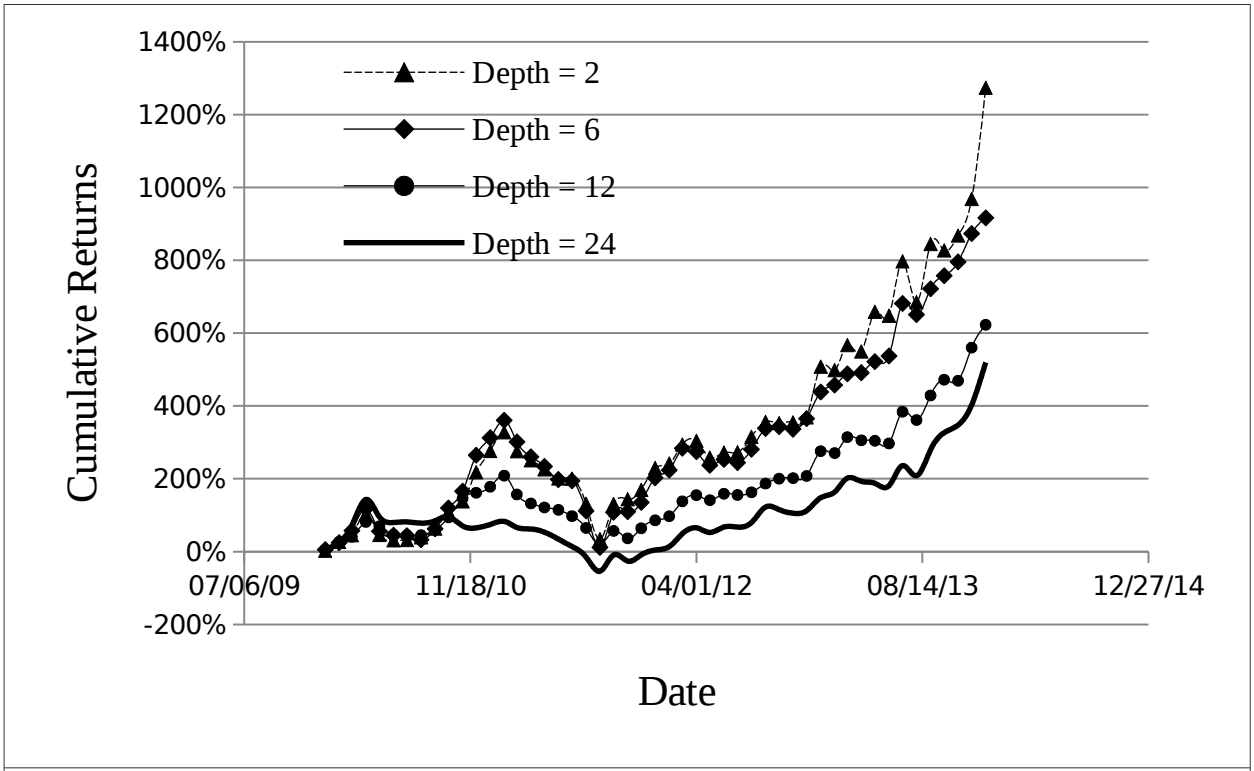

Figure 6: The graph shows the change in cumulative returns as we change the maximum depth of the estimators (trees) in the Gradient Boosting Classifier when only one classifier is used and stocks are automatically selected.

From Figure 6, the reader can observe from the graph that a decrease in the maximum depth allowed yields better cumulative returns. This is most likely due to the classifier not being prone to over-fitting when the depth is limited. While the changes were only performed on the Gradient Boosting classifier, the construction of the other two classifiers is similar and the results are almost the same (we will not show those results due to lack of space). This is evident from when we fine-tuned the classifiers earlier in the research section of Quantopian and arrived at similar results. It can be shown that, in general, increasing the number of estimators and decreasing the maximum depth of the estimators will increase the predictive performance of ensemble methods.

5.5: PERFORMANCE INDICATORS RESULTING FROM SIMULATIONS

Following the fine-tuning investigation, we start to compare performance indicators other than cumulative returns, using the average of their 12-month value over multiple simulations and taking them to a confidence level of 90%. We will start by comparing the averaged values of alpha and beta over a 12-month period for each of the three classifiers (taking the Standard and Poor 500 index, NYSE: SPY, as a benchmark) and for each of the classification scenarios (one classifier and two classifiers respectively). By observing the data in Table 6, it can be seen that in general the two-classifier method produces a higher value of alpha and beta than the one classifier method.

Table 6: The table compares the average values of the alpha and beta coefficients over 12-month periods for each of the three classification methods when used in simulation over the time-period 2010 to 2015.

	12-month Alpha		12-month Beta	
	One Classifier Method	Two Classifiers Method	One Classifier Method	Two Classifiers Method
Random Forest Classifier	0.40	1.29	1.89	2.79
Extremely Randomized Trees Classifier	0.40	1.05	1.25	2.77
Gradient Boosting Classifier	0.62	1.37	1.74	4.70

When looking for a good trading strategy, it is preferable to find one that has low correlation to the market (in this case, the market is considered to be the Standard and Poor 500 index, NYSE: SPY). A low long-term beta value and a high long-term alpha value for a trading strategy are desirable characteristics of a trading strategy because they indicate low market correlation and high “risk-free” return respectively.

From observing the data in Table 6 we can deduce that, while most strategies have undesirably high values of beta, they also possess large values of alpha and the two classifiers method produces higher alpha than its one classifier

counterpart on average. This leads us to presume there is value in these strategies from a financial perspective, although they would need further work to decrease market correlation. We will also compare the ratio indicators which give the reader some indicator of return versus risk for each of the classifiers in each of the classification scenarios, but in different ways. It should be noted that the ratios will seem unusually high compared to traditional methods, and this is expected with the high cumulative returns we saw earlier. The next table, Table 7, compares the Sharpe, Sortino, and Information ratios for all three ensemble methods for both the one classifier and two classifier methods.

Table 7: The table compares the average values of the Sharpe, Sortino and Information ratios over 12-month periods for each of the three classification methods when used in simulation over the time-period 2010 to 2015.

	Sharpe Ratio		Sortino Ratio		Information Ratio	
	One Classifier Method	Two Classifiers Method	One Classifier Method	Two Classifiers Method	One Classifier Method	Two Classifiers Method
Random Forest Classifier	2.26	3.42	4.06	5.28	0.11	0.10
Extremely Randomized Trees Classifier	2.68	3.24	4.07	3.25	0.10	0.10
Gradient Boosting Classifier	3.61	3.84	5.28	5.73	0.15	0.13

We can observe from the values in Table 7 that in general, the Gradient Boosting Classifier method produces higher performance ratios than its counterparts and this may indicate that the Gradient Boosting Classifier method results in higher returns with lower risk relative to other trading strategies. The reader can also see that the two-classifier method produces higher performance ratios than the one classifier method in most of the cases, except for the Information Ratio in which both methods have similar performance. The final two performance indicators, outlined in Table 8 below, are the volatility and maximum draw-down, both of which describe the risk associated with the trading algorithm.

Table 8: The table compares the average values of the volatility and maximum draw-down indicators over 12-month periods for each of the three classification methods when used in simulation over the time-period 2010 to 2015.

	Volatility	Maximum Draw-down

	One Classifier Method	Two Classifiers Method	One Classifier Method	Two Classifiers Method
Random Forest Classifier	0.24	0.35	11.55%	21.45%
Extremely Randomized Trees Classifier	0.23	0.49	11.69%	25.25%
Gradient Boosting Classifier	0.22	0.38	24.00%	24.02%

It can be observed from Table 8 that in general, the two-classifier method has higher volatility than the one classifier method. This may be because of the greater condition placed on the one classifier method (i.e.: to be sure of the result with a certain probability defined beforehand) than that of the two classifiers method. Reading the three previous tables, namely Tables 6, 7, and 8, we can see that almost all trading strategies have high volatility and large draw-downs, despite them having large amounts of cumulative return and good return-versus-risk ratios. It is worth noting that the two classifier methods, on average, have higher return and risk ratios than their one classifier counterparts.

6. CONCLUSIONS

It has always been a difficult task to predict the stock market, and even after many attempts by researchers to restore confidence in stock market prediction, there are many other researchers and economists that believe the market is random and cannot be predicted. It is our aim that the simulation results presented in this project provide the reader with a new positive look towards machine learning and stock market prediction.

The simulation results and investigation show both advantages and disadvantages of using ensemble methods in algorithmic trading; all three classification methods were able to *outperform the benchmark* in cumulative returns *over a long trading period* and had *significant alpha coefficients*. Additionally, all three classification methods outperformed the methods from previous literature in predictive accuracy. However, when our methods were applied in algorithmic trading, they all showed volatile and risky trading behavior, and they all had an undesirably large correlation to the market. Through basic parameter adjustment (such as maximum depth and number of estimators) and changing of trading strategies (such as one and two classifier methods), we were able to greatly modify the performance of all classifiers and their simulated trading portfolios.

It is in our opinion that the deployment of the algorithmic trading strategies discussed in this paper is a good move by any prospective investor. The strategies outlined in this project provide a solid foundation for further improvement, and they can be deployed once they have been adjusted to have less volatility and risk. The use of these ensemble methods specifically, and machine learning in general, in real algorithmic trading in the future is realistic, and we expect further work and research on them.

The field of machine learning algorithms is a very vast one and we have only explored the field of ensemble methods in this project. There are many new machine learning methods being developed that have been shown to excel at time-series prediction. For example, deep neural networks have recently shown good performance in time-series prediction, and there have been some attempts at using them on stock market data. [22] It is also possible to create ensemble methods from deep neural networks and apply similar ideas presented in this project to those ensemble methods.

We have also only explored one specific trading strategy that encompassed our prediction algorithms. The field of algorithmic trading has many different trading strategies, and these can differ in the inputs used to feed the machine learning algorithm, or in the way the strategy handles the output of the machine learning algorithm. For the former, it would be wise to investigate the use of both fundamental data and technical indicators together as inputs to the predicting algorithm, and assess the results. For the latter, it would also be wise to investigate a larger number of output prediction classes than the three target classes we used in this project.

REFERENCES

1. Mladjenovic, Paul. *Stock investing for dummies*. John Wiley & Sons (2005).
2. Levišauskait, Kristina. "Investment Analysis and Portfolio Management."*Leonardo da Vinci programme project* (2010).
3. Bauman, Mark P. "A review of fundamental analysis research in accounting."*Journal of Accounting Literature* 15 (1996): 1.
4. Edwards, Robert D., John Magee, and WH Charles Bassetti. *Technical analysis of stock trends*. CRC Press (2007).
5. Glantz, Morton, and Robert Kissell. *Multi-Asset Risk Modeling: Techniques for a Global Economy in an Electronic and Algorithmic Trading Era*. Academic Press (2013).
6. Hendershott, Terrence, Charles M. Jones, and Albert J. Menkveld. "Does algorithmic trading improve liquidity?" *The Journal of Finance* 66.1 (2011): 1-33.
7. Chaboud, Alain P., et al. "Rise of the machines: Algorithmic trading in the foreign exchange market." *The Journal of Finance* 69.5 (2014): 2045-2084.
8. Friedman, Jerome H. "Greedy function approximation: a gradient boosting machine." Annals of statistics (2001): 1189-1232.2.Tin Kam (1995); Random Decision Forest.
9. Li, Ping, Qiang Wu, and Christopher J. Burges. "Mcrank: Learning to rank using multiple classification and gradient boosting." Advances in neural information processing systems. (2007).

10. Geurts, Pierre, Damien Ernst, and Louis Wehenkel. "Extremely randomized trees." Machine learning 63.1 (2006): 3-42.
11. Boehmer, Ekkehart, Kingsley YL Fong, and Juan Julie Wu. "International evidence on algorithmic trading." AFA 2013 San Diego Meetings Paper. (2014).
12. Kimoto, Tatsuya, et al. "Stock market prediction system with modular neural networks." Neural Networks, 1990., 1990 IJCNN International Joint Conference on. IEEE (1990).
13. Naeini, Mahdi Pakdaman, Hamidreza Taremian, and Homa Baradaran Hashemi. "Stock market value prediction using neural networks." Computer Information Systems and Industrial Management Applications (CISIM), 2010 International Conference on. IEEE (2010).
14. Huang, Wei, Yoshiteru Nakamori, and Shou-Yang Wang. "Forecasting stock market movement direction with support vector machine." Computers & Operations Research 32.10 (2005): 2513-2522.
15. Shah, Vatsal H. "Machine learning techniques for stock prediction." Foundations of Machine Learning, Spring (2007).
16. Andersen, André Christoffer. A Novel Algorithmic Trading Framework Applying Evolution and Machine Learning for Portfolio Optimization. Diss. Master's Thesis, Faculty of Social Science and Technology Management, Department of Industrial Economics and Technology Management (2012).
17. Shen, Shunrong, Haomiao Jiang, and Tongda Zhang. "Stock market forecasting using 65machine learning algorithms." (2012).
18. Lin, Yuling, Haixiang Guo, and Jinglu Hu. "An SVM-based approach for stock market trend prediction." IJCNN, The 2013 International Joint Conference, IEEE, (2013).
19. Gupta, Abhishek, and Samidha D. Sharma. "Clustering-Classification Based Prediction of Stock Market Future Prediction." IJCSIT) International Journal of Computer Science and Information Technologies 5.3 (2014): 2806-2809.
20. Mark Dunne. "Stock Market Prediction". University College Cork (2015).
21. Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830 (2011).
22. Jia, Hengjian. "Investigation Into The Effectiveness Of Long Short Term Memory Networks For Stock Price Prediction." arXiv preprint arXiv:1603.07893 (2016).
23. Elman, Jeffrey L. "Distributed representations, simple recurrent networks, and grammatical structure." Machine learning 7.2-3 (1991): 195-225.
24. Hendershott, Terrence, Charles M. Jones, and Albert J. Menkveld. "Does algorithmic trading improve liquidity?" The Journal of Finance 66.1 (2011): 1-33.
25. Friedman, Jerome H. "Greedy function approximation: a gradient boosting machine." Annals of statistics (2001): 1189-1232.
26. Breiman, Leo. "Random forests." Machine learning 45.1 (2001): 5-32.
27. Geurts, Pierre, Damien Ernst, and Louis Wehenkel. "Extremely randomized trees." Machine learning 63.1 (2006): 3-42.
28. Brock, William, Josef Lakonishok, and Blake LeBaron. "Simple technical trading rules and the stochastic properties of stock returns." The Journal of finance 47.5 (1992): 1731-1764.
29. Atsalakis, George S., and Kimon P. Valavanis. "Surveying stock market forecasting techniques–Part II: Soft computing methods." Expert Systems with Applications 36.3 (2009): 5932-5941.
30. Fama, Eugene F. "Random walks in stock market prices." Financial analysts journal 51.1 (1995): 75-80.
31. Shah, Vatsal H. "Machine learning techniques for stock prediction." Foundations of Machine Learning| Spring (2007).
32. Nuti, Giuseppe, et al. "Algorithmic trading." Computer 44.11 (2011): 61-69.
33. Tudball, Dan. "Quant Insights." Wilmott 2016.82 (2016): 27-30.