

April 2019

# Trading & Investing System Development: A System of Systems

Alexander Jay Wurts  
*Worcester Polytechnic Institute*

John William Battles  
*Worcester Polytechnic Institute*

Nicholas Scott Houston Albert  
*Worcester Polytechnic Institute*

Follow this and additional works at: <https://digitalcommons.wpi.edu/iqp-all>

---

## Repository Citation

Wurts, A. J., Battles, J. W., & Albert, N. S. (2019). *Trading & Investing System Development: A System of Systems*. Retrieved from <https://digitalcommons.wpi.edu/iqp-all/5391>

This Unrestricted is brought to you for free and open access by the Interactive Qualifying Projects at Digital WPI. It has been accepted for inclusion in Interactive Qualifying Projects (All Years) by an authorized administrator of Digital WPI. For more information, please contact [digitalwpi@wpi.edu](mailto:digitalwpi@wpi.edu).

# Trading & Investing System

## Development: A System of Systems

AN INTERACTIVE QUALIFYING PROJECT REPORT SUBMITTED TO THE FACULTY OF  
**WORCESTER POLYTECHNIC INSTITUTE** IN PARTIAL FULFILLMENT OF THE  
REQUIREMENTS OF THE DEGREE OF BACHELOR OF SCIENCE



**AUTHORED BY:**

NICHOLAS S.H. ALBERT  
JOHN W. BATTLES REL-1806  
ALEXANDER J. WURTS - REL-11

**DATE:**

30 APRIL 2019

**REPORT SUBMITTED TO:**

REINHOLD LUDWIG, PH.D.  
GBETONMASSE SOMASSE

## Authorship Page

<b>Section</b>	<b>Primary Author(s)*</b>
Chapter 1	1.1 Nicholas Albert 1.2 Nicholas Albert 1.3 Nicholas Albert 1.4 Nicholas Albert
Chapter 2	2.1 John Battles 2.2 John Battles
Chapter 3	3.1 Nicholas Albert 3.2 Nicholas Albert 3.3 Nicholas Albert 3.4 John Battles 3.5 John Battles 3.6 John Battles 3.7 Alexander Wurts
Chapter 4	Alexander Wurts

Chapter 5	Alexander Wurts
Chapter 6	Nicholas Albert
Chapter 7	John Battles
Chapter 8	Alexander Wurts
Chapter 9	Alexander Wurts
Chapter 10	Alexander Wurts

\*All non-primary authors of a section were editors

# Table of Contents

<b>Authorship Page</b>	<b>1</b>
<b>Table of Contents</b>	<b>4</b>
<b>Table of Figures</b>	<b>8</b>
<b>Table of Tables</b>	<b>9</b>
<b>Chapter 1: Introduction and Overview of the Research</b>	<b>10</b>
1.1 Statement of the Problem	10
1.2 Statement of Problem Importance	10
1.3 Objectives	11
1.4 Brief Summary of Literature and Statement of Creativity	11
1.5 Summary	12
<b>Chapter 2: Trading and Investing</b>	<b>13</b>
2.1 General Overview	13
2.1.1 Difference Between Investing and Trading	13
2.2 Basic Macroeconomics	14
<b>Chapter 3: Trading Systems</b>	<b>17</b>
3.1 Types of Investing	17
3.1.1 Value Investing	17
3.1.2 Growth Investing	17
3.2 Types of Trading	18
3.2.1 Swing Trading	18
3.2.2 Short Term Trading	18
3.2.3 Fundamental Trading vs Technical Trading	18
3.3 Data Sources	19
3.3.1 Data Sources for Investing	19
3.3.2 Data Sources for Technical Trading	19
3.3.3 Data Sources for Fundamental Trading	20
3.4 Time Frames	20
3.4.1 Time Frames	20
3.4.3 Short Term Trading Strategies	21
3.4.2 Long Term Investing Strategies	22
3.5 Trading Brokerage Accounts & Platforms	22
3.5.1 Types of Brokerage Accounts	23
3.5.2 Types of Trading Platforms	24

3.6 Trading System Objectives	24
3.7 Trading System Parts	25
3.7.1 Selecting Stocks	26
3.7.2 Entry Rules: When to Buy an Asset	26
3.7.2.1 Order Types	27
3.7.3 Exit Rules	28
3.7.4 Position Sizing Rules	29
<b>Chapter 4: Optimizing and Analyzing Trading Systems</b>	<b>31</b>
4.1 Optimization	31
4.2 Walk Forward Analysis	31
4.2.1 How to perform a Walk Forward Analysis	31
4.2.2 Understanding the results of a Walk Forward Analysis	32
4.3 Using walk forward results to optimize the algorithm	35
4.4 Comparing Algorithms	36
4.5 How to Optimize and Analyze a System of Systems	36
<b>Chapter 5: Trading System #1 Machine Learning Trader</b>	<b>38</b>
5.1 Overview	38
5.2 Stock Selection Criteria	38
5.2.1 Selected Stocks based on Criteria in 5.2	39
5.3 Stock Price Data Preprocessing	39
5.4 Machine Learning	41
5.4.1 Random Forest	42
5.4.2 Bagging	42
5.5 When to run algorithm	43
5.6 Money Allocation	43
5.6 Selling	44
5.7 Backtesting	44
5.8 Results	44
<b>Chapter 6: Trading System #2</b>	<b>46</b>
6.1 Method of Trading	46
6.2 Strategy	46
6.2.1 Screener	46
6.2.2 The Seven Criteria	46
6.2.3 Stocks that Pass Screener Criteria	47
6.2.4 Allocation	47
6.3 Risk management	47
6.4 Backtest	48

<b>Chapter 7: Trading System #3</b>	<b>50</b>
7.1 Method of Trading	50
7.2 Strategy	50
7.2.1 Use a screen to determine high growth stocks	50
7.2.2 Uses 5 criteria	50
7.2.3 Ideally Leaves a Maximum of 10 Stocks	51
7.3 Allocation	51
7.4 Risk management	51
7.5 Backtest	51
7.6 Performance	53
<b>Chapter 8: System of Systems</b>	<b>54</b>
8.1 Overview	54
8.2 Stock Selection: John's and Nicholas's System	54
8.3 Asset Allocation: John's System	54
8.4 Buy Timing: Alex's System	54
8.5 Sell Timing: General System	55
8.6 System Retirement Criteria	55
8.7 Long Term Implementation	55
8.8 Intermediate Term Implementation	55
8.9 Short Term Implementation	56
8.9 Optimization	56
8.9.1 Machine Learning Optimization	56
8.9.2 Position Closing Parameters	57
8.9.3 Asset Allocation	58
<b>Chapter 9: Results for System of Systems</b>	<b>59</b>
9.1 Subsystem individual results	59
9.2 Overall Results	59
9.2.2 Long Term Results	60
9.2.3 Medium Term Results	61
9.2.4 Short Term Results	62
<b>Chapter 10: Conclusions</b>	<b>63</b>
10.1 Summary	63
10.1 Is it safe to trade with real money?	63
10.2 Problems and Issues	64
10.3 Future Work	64
<b>References</b>	<b>66</b>





## Table of Figures

Figure 1: The Different Phases of the Macroeconomic Cycle	15
Figure 2: Heikin Ashi Transformation Example	39
Figure 3: Cumulative Summation Equation.	39
Figure 4: Fractional Differentiation Example	40
Figure 5: $(1-B)^d$ expanded version.	41
Figure 6: Expansion of $w$	42
Figure 7: Expansion $X$	42
Figure 8: Definition of $X$ tilda.	42
Figure 9: Formula for calculating $w$ iteratively.	43
Figure 10: Weights minimum cut off	43
Figure 11 Trading System One Results	46
Figure 12: FinViz Screener with Variables Applied	49
Figure 13: Criteria and Parameters for Backtest	50
Figure 14: Results from the Backtest	50
Figure 15: Equity Graph of the Backtest	51
Figure 16: Backtest Trades	51
Figure 17: System Three Backtest	54
Figure 18: System Three Equity History	54
Figure 19: Long Term Results	62
Figure 20: Intermediate Results	63
Figure 21: Short Term Results	64

# Table of Tables

Table 1: Brokerage and Trading Platforms	23
--	----

# Chapter 1: Introduction and Overview of the Research

## 1.1 Statement of the Problem

The purpose of this Interactive Qualifying Project is to understand the financial market and to learn different techniques in order to build a system of trading systems. It will cover economics and stock market basics, the process of selecting stocks that have a high performance, and the system of systems that was created. It is difficult to have a consistent return on investment for amateur traders, which is contributed by not understanding which stocks to choose and what parameters should be used when selecting stocks. Patterns are difficult to find in the market, therefore most systems are only effective for certain periods of time and for specific stocks or assets. Learning the skills needed for smart decision making takes a significant amount of time, especially when an individual takes part in trading daily.

## 1.2 Statement of Problem Importance

In order to be able to retire at a reasonable age and enjoy, one must have the funds to do so. If done correctly, investing helps build an individual's wealth throughout their lifetime to make this a reality. It is critical to learn how the stock market works as well as the underlying economics to be able to invest wisely and effectively. By not investing, one will lose a significant amount of money in the long term due to missed compound interest. It is in the best interest of every individual to learn how the markets work, to learn the skills needed to make smart decisions, and capitalize on it in order to grow their wealth.

## 1.3 Objectives

The objectives of this project are to provide the reader with background material to enable them to implement our trading systems and eventually implement their own. A large component of this project was creating and developing multiple trading/investing systems that were later combined in order to make a system of systems. Our team aggregated the best features of each individual trading system into one to create this system of systems that produces a better return on investment. This involved putting to use machine learning through the use of a Bagging Classifier and a Random Forest Classifier. The outcome was a system that performed much better than DOW or the S&P 500. This system was optimized to produce the maximum profit.

## 1.4 Brief Summary of Literature and Statement of Creativity

This paper has expanded and improved upon ideas from articles on various topics. We started with ideas from numerous articles and from there we created three individual systems that were combined into one large system with artificial intelligence capabilities.

[“Searching for Future Stock Market Winners”](#) [1] is about what criteria should be used for finding stocks with high future potential. The data for this is based on prior “winners”. This helped us generate ideas on different criteria we should use to help predict what stocks would perform well.

[“Micro-Cap Stocks Outperform, Often in Consecutive Years”](#) [2] explains how over time small companies generally outperform larger or blue-chip companies. This strategy requires a lot of patience since it generally only works in the long term. Since we knew we wanted our trades to be on the side of short term, we stayed away from small companies.

[“Stock market index prediction using artificial neural network”](#) [3] is a study about “the ability of artificial neural network (ANN) in forecasting the daily NASDAQ stock exchange rate [...]” Like them, we used past data to train our system in order for it to learn patterns and to produce a larger return.

## 1.5 Summary

This report first introduces the background material required to understand trading and the basics of the stock market. This gives the readers an understanding so that when reading about the three different systems and the system of systems, they will understand why certain decisions were made. It then introduces investing strategies along with the trading systems our team created. Finally, it combines the trading systems into a conjoined system that uses the strengths of each individual system to make a better system of systems. The goal is to produce a system of systems that performs better than the DOW or the S&P 500.

## Chapter 2: Trading and Investing

### 2.1 General Overview

The two types of people who trade assets are professional and retail traders. Professional traders are just that, professionals. They are paid by someone else to manage that person's money and to invest it so that it grows. A retail trader is someone who trades their own money for their own gain. A couple of important differences is that professional traders trade significantly more money than a retail trader and also have a fiduciary duty to act in the best interest of their clients. What this means is that a professional trader will generally stick to one or two strategies that they have validated as being successful. Retail traders are free to trade however they like since any money lost is their own.

#### 2.1.1 Difference Between Investing and Trading

There are two main methods for making a profit from assets, and most often associated with the stock market, investing and trading. The greatest difference between the two is the base strategy. Investors look for the asset to grow in value over some period of time while traders look to exploit the volatility of the asset, wanting the price to change either increasing or decreasing in value. A second difference is the length of time that each type holds the asset. An investor looks to buy for the long-term while a trader may buy and sell an asset anywhere from days to seconds. Each method has its advantages and disadvantages.

Investing is often considered a more simplified approach when compared to trading because of the market's history of growing on average 10% yearly and the less time it requires overall. Investors typically focus on assets fundamental factors and long-term growth potential.

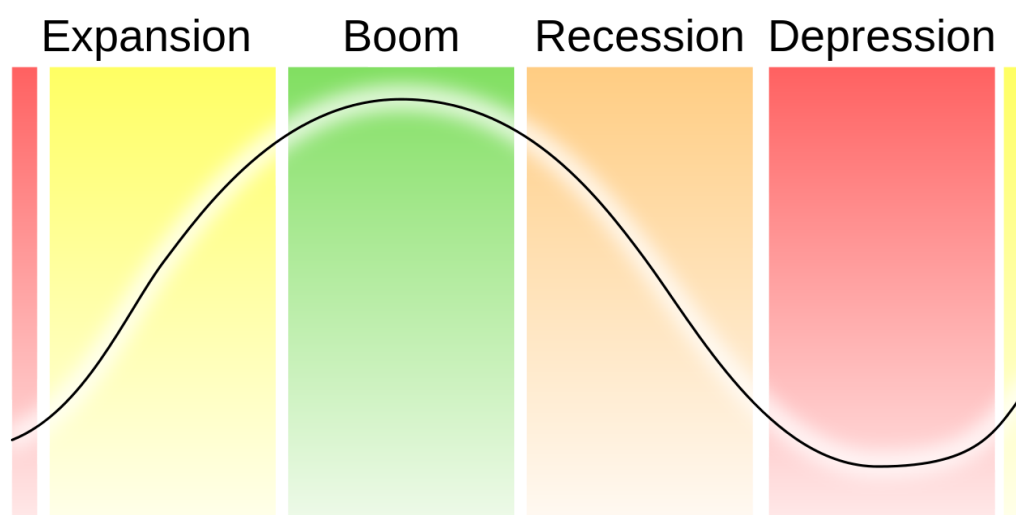
They are more willing than traders to weather market-wide fluctuations. Once the decision to purchase an asset is made an investor will check the asset price periodically and decide whether to continue holding the asset, to sell, or buy more. Investing comes at the cost of being dependent on the conditions of the market. This means that at certain times it can be very difficult to buy or sell an asset due to low volumes being traded at the time. An investor can also be subject to market-wide conditions. A prime example is during the 2008-09 recession when the markets crashed and some investors lost upwards of 50%.

Trading requires more research and involvement than investing due to the rapidity that is involved in trading. A trading mentality focuses more on an assets technical factors and short term outlook versus the value of the asset. Trading can result in significantly higher returns than investing. However, with higher returns comes higher risk which is why being a successful trader requires more time than investing. This time is spent both researching and monitoring the assets. Due to the shorter time frame involved with trading, a trader must keep a constant watch on the price of an asset. Another consideration for traders is the tax ramifications, trading profits can be taxed at the higher income tax rate than the capital gains tax rate that investing involves. As the use and spread of computers expand using computers to run an algorithm has evolved and become more prevalent. Computers give the advantage of having unparalleled reaction times to changes in the market and can give a trader the slight advantage they need.

## 2.2 Basic Macroeconomics

The economy as a whole goes through a periodic cycle, as seen in Figure 1. This is known as the business cycle and consists of expansion, contraction, and recovery. Generally, phases of expansion and boom, periods where unemployment regresses and GDP increases, are followed with a recession phase. Recessions are contraction periods where there is an

overall downturn in economic activity. They start during the peak of the economic expansion and follow until the start of the recovery phase. Recovery or depression phase is the period where the economy starts to bounce back and regains economic productivity and unemployment start rising again.



**Figure 1:** The different phases of the macroeconomic cycle.

There are four assets that are traded; equities, bonds, currencies, and commodities. Equities are more commonly known as stocks or shares. Stocks are essentially pieces of a company that are owned by individuals or groups. There are a couple of types of stocks that each have certain attributes, common and preferred. Common stocks are the most populous and represent ownership of a portion of the company or a claim to the dividends paid by the company. Preferred stocks do not guarantee the same voting or ownership stake in the company as common stocks, however, preferred stocks are guaranteed a fixed dividend in perpetuity and are paid off before common stocks in the case of bankruptcy. Bonds are a fixed income investment that represents a loan made by an investor to a borrower. Bonds are most commonly issued by governments at all levels; local, state, and federal. Bonds can also be sold by companies as an alternative to selling stocks in the company. Currencies are traded in pairs



as relative values to each other, as one increase in value the other decreases with respect to the first. The foreign exchange or forex market is the largest market in the world. The last asset, commodities, are tangible real-world products. Commodities include; oil, livestock, wood, metals, and grains among others. The commodity market is generally used to buy futures of these products, which in turn sets the prices of the products when they reach the market.

Depending upon which phase of the cycle the economy is in impacts how financial institutions trade and which asset they base their plan around. During expansions historical data has shown that the stock market experiences the highest annual growth among the asset classes, however, during recessions stocks have the greatest losses. Bonds are the opposite of stocks with lower returns during expansions and higher returns during recessions. The foreign exchange market depends on which pairs of currencies are being traded and how the respective countries' economies are performing. For example, if the US is experiencing a recession and the UK is experiencing an expansion then the value of the dollar with respect to the English pound will decrease. However, if the US and UK are experiencing recessions but the UK has a worse recession then the dollar will increase in value with respect to the English pound. Commodities are similar to stocks in that during an expansion the returns are higher and during a recession the returns are small or losses. This is due to the fact that the commodities market is just that a market for commodities or physical goods. During an expansion people and companies have excess money and are looking to expand which requires goods to do creating higher demands and during a recession the opposite is true, people and companies have less money and are therefore less likely to spend on goods they do not need decreasing demand.

## Chapter 3: Trading Systems

### 3.1 Types of Investing

#### 3.1.1 Value Investing

Value investing is an investment strategy where stocks are chosen if they are being traded for less than their book value. Individuals who practice this tactic are looking for the stocks that they believe are undervalued by the market. Though value investing works very well after market sell-offs, it is complicated to do after a bull market is in progress. [4]

Benjamin Graham is known as the “father of value investing”. Warren Buffett was a student of Graham, and both of them were very successful value investors. Buffett promoted Graham’s approach and attributed a lot of his success to his teacher. [5]

#### 3.1.2 Growth Investing

Growth investing is an approach to investing that focuses on capital appreciation. Growth investors usually invest in companies whose earnings are expected to grow at a rate that is above average compared to its respective industry or the overall market. This tactic is appealing to many investors because purchasing stock in new and growing companies have the possibility to have extremely high returns if the companies are successful. The downside to this is that these companies tend to be very high risk. [6]

## 3.2 Types of Trading

### 3.2.1 Swing Trading

Swing trading requires holding an asset, either long or short, for more than one trading session, but no longer than several weeks or a couple of months. The goal of this technique is to encapsulate a piece of a potential price movement. More or less, swing trading is recognizing where an asset's price is most likely going to move and capitalizing on it. [7]

### 3.2.2 Short Term Trading

Short term trading is potentially very beneficial but it comes with high risk. A short term trade can last for as little as a seconds to several days. In order to be successful with this strategy, one must have knowledge of the asset that they are trading and be able to identify the patterns from its history. [8]

### 3.2.3 Fundamental Trading vs Technical Trading

Fundamental and technical trading are two of the big major schools of thought when participating in investing in the markets. Fundamental trading is a strategy of assessing securities by trying to measure the intrinsic value of a stock where technical trading only focuses on the stock's price and volume. Technical traders use multiple indicators and look for trends to analyze stocks in order to decide which stocks are worth investing in. [9]

## 3.3 Data Sources

### 3.3.1 Data Sources for Investing

In order to trade algorithmically, the program needs raw data. This can be obtained from different websites. Stock pricing data can be acquired from many websites. For daily open, high, low, and close data Yahoo Finance is a good source. They have five or more years of data for nearly every stock. Minute data usually costs money to acquire, however, Oanda offers a limited amount of minute data on their website.

Company valuation data is also important for algorithmic programs. Company information data can be found in a company's quarterly earnings reports. They are a standardized document released by every public company that gives important details regarding how well the company is performing. While the original source is valuable, it is helpful to have simplified data. Yahoo finance is a good source of simplified data as well as Finviz and Investopedia.

In general, valuation data is used for long term trading because it better represents the movement of the stock over a long period of time. Stock price data is used for technical trading because the short term stock price is not heavily affected by the company value.

### 3.3.2 Data Sources for Technical Trading

Technical traders look for trends in the market and use multiple different indicators to decide what stocks are worth trading. Examples of indicators used by technical traders include the Relative Strength Index (RSI), Simple Moving Average (SMA), Bollinger Bands. Since stocks almost never move in a straight line, trends are not always easy to spot. An uptrend,

though the price might drop from time to time, has a positive slope where a downtrend is the opposite.

### 3.3.3 Data Sources for Fundamental Trading

Fundamentalists use both macroeconomic (economy and industry conditions) and microeconomic factors (management of the companies) in order to decide what is worth investing in. The intent of this method is to provide the investor with a quantitative value that they can compare to a stock's current prices to determine whether the stock is over or undervalued.

## 3.4 Time Frames

### 3.4.1 Time Frames

Determining the desired time frame for holding assets is vital to the desired goal of the trader. The three time frames are long, intermediate, and short. The time frame selected can impact which asset classes are chosen to trade.

Long term trading focuses on investments that are going to be held for more than a year. These are investments that are shown to have steady growth over the course of their life and most often include; exchange-traded-funds (ETFs), index funds (i.e. SPY), and bonds. Long term assets have several advantages and disadvantages over medium and short term assets. Long term asset profits are taxed at the capital gains tax level while the other time frames are taxed at the higher income tax level. Long term investments generally also pay higher dividends which can act as a supplementary or replacement income. The downside to long term trading is that the capital used is tied up for the long term. If another, more lucrative, opportunity comes

along it can be difficult to liquidate. Long term trading is also a bet that in the long run, an investment will have a greater return than another asset in a shorter time period.

Intermediate term trading focuses on assets that will be held for a period of a week up to a year. This can involve using both trading or investing methods to generate a profit.

Intermediate term trading assets can include all four types of asset classes. Intermediate term trading can have a greater return than long term trading and less risk than short term trading.

Fees can be lower than long term trading but due to a higher number of trades, the total cost of fees can exceed that of long term trading. Profits are also taxed at the higher income tax rate than the capital gains rate. Due to the shorter time frame than the long term, medium term trades can be easily liquidated in the event of a market crash or a better opportunity.

Short term trading focuses on assets held for at most several days. Traders using this time frame look for any movement in an asset's value. With short term trading, the profit is made from minute changes in the asset's value. Due to the inherent aspect of the assets being held for a short time, a trader using this time frame would be highly liquid and able to get in and out of positions quickly as the market dictated. However, short term trading is very time intensive due to the constant buying and selling and the oversight required with that. Similar to medium term trading any profits are taxed at the income tax level.

### 3.4.3 Short Term Trading Strategies

here are a few types of strategies for short term trading; day trading, swing trading and scalping. Day trading is when a trader buys and sells assets but have closed their position by the end of trading each day. Day traders use technical analysis to determine when to open and close positions and avoid any drastic changes that can occur due to reports or stories being released after hours. Swing trading is when a trader bases their buy and sell rules on the trends

a stock may follow. Swing traders look for the signs that a trend may be ending and want to capitalize on the volatility of the asset price that occurs between trends. Scalping involves exploiting the price gaps in spreads. A scalper will generally buy at the bid price and sell at the ask price to get the difference in the spread. Scalpers positions are most often small but numerous and look for stable markets that are not prone to drastic swings.[10]

#### 3.4.4 Intermediate Length Trading

The two main strategies for intermediate trading are; position and swing trading. In position trading an individual will look at long term charts and their own method to determine the trend of a market. Position traders will often look to ride the “wave” that is caused by a trend and will most oftenly ride the trend until it breaks. Some consider position trading to be a passive type when compared to other strategies. The main difference between short term swing trading and intermediate term swing trading is the time period of the trends observed. An intermediate term trader will look for trends that occur over weeks or months.[10]

#### 3.4.2 Long Term Investing Strategies

There are two long term investing strategies; active and passive. An active investor is when the individual applies more time to their investment plan. This can include monitoring the news, researching potential investments more, and just keeping a watchful eye on their current investments. Active investors are generally people who are looking to beat the average return of the stock market. A passive investor is an individual who will find investments, buy them, and then hold. They are looking for long term gains with little to no work done on their part. Passive investors will most often hold their investments for a considerable length in time regardless of how the market is performing. This results in passive investors rarely beating the market.

### 3.5 Trading Brokerage Accounts & Platforms

Brokerage accounts are financial accounts where investors deposit money that later can be used to buy investments like stocks and mutual funds. The investor, similar to owning any regular bank account, can sell investments and withdraw the money within his/her account at any desired time. Incomes gained through brokerage accounts are also taxable. There are a multitude of brokerage firms that offer different types of accounts. Some are full service whereby investors are charged higher commission fees and provided with extensive advice. Others provide less service hence charging their clients significantly less than full service brokerage accounts.

Trading Platforms are software constructs that enable for buying, selling and managing market positions. They provide traders with market insight and decision making tools supplying charting tools, real-time price quotes, news feeds and even premium research. Some platforms target specific asset classes like currencies, securities, options, or futures.

**Table 1:** Brokerage and Trading Platforms

Brokerage Platforms	Trading Platforms
<ul style="list-style-type: none"> <li>● Charles Schwab</li> <li>● E*Trade</li> <li>● Vanguard</li> <li>● Merrill Lynch</li> <li>● Morgan Stanley</li> <li>● Wells Fargo Advisors</li> </ul>	<ul style="list-style-type: none"> <li>● Interactive Brokers</li> <li>● TradeStation</li> <li>● TDAmeritrade</li> <li>● Robinhood</li> <li>● MetaTrader</li> </ul>



### 3.5.1 Types of Brokerage Accounts

There are two types of brokerage and trading accounts and depending upon what a potential investor or trader wants should be taken into consideration when selecting an account. The first is a cash account. Cash accounts require the initial capital to be supplied up front by the person opening the account. Each trade is then done with this available capital with the individual able to spend as much as they have in their account. The second type is a margin account. Margin accounts allow an individual to borrow against the capital they have in their account. Any debts due to borrowing are charged a daily interest based upon the prime rate plus whatever the platform charges. Individuals are also subject to margin calls if the value of their assets falls below the amount covered by their capital. A margin call requires the individual to add more capital in order to maintain their positions. Not all assets are available in a margin account due to the necessity for there to be a large volume of the asset. Margin accounts are most common in the Forex market.

### 3.5.2 Types of Trading Platforms

There are many different types of trading platforms now available covering every imaginable market. The most common are Oanda, MT4, and TradeStation. Oanda is primarily used for the forex market and offers many helpful tools for trading. There are over 70 currency pairs and an account can be opened for as little as \$1. Oanda is also regulated by government agencies allowing traders to know that any trades they are making are legal and secure. TradeStation is another highly popular platform that allows traders access to domestic markets and has a plethora of ancillary features. TradeStation allows for traders to create their own

strategies in the programming language EasyLanguage as well as allowing for in depth analysis of the various tradable assets.

### 3.6 Trading System Objectives

The primary objectives of a trading system is to produce a high return on investment, remove emotion from the trading process, and reduce time spent trading. Accomplishing these three objective produces a system of systems that is profitable and effective.

The return on investment objective is most often the primary reason why a system is developed. A system is successful for this objective when it is able to produce more money than it is originally allocated. For example if a system was allocated \$25,000 and it was able to successfully increase that amount to \$26,000 over the course of a year it would accomplish this objective.

The next objective, removing emotion from trading, is one of the reasons trading systems can be more effective than humans. Systems are effective at trading based on specific set rules and not deviating from them. However humans are easily swayed by emotion. For example if a trade is losing value a human may feel holding onto the stock is better because it will increase again. A trading system however will sell at the set stop loss value regardless. While the human could be correct, the stop loss is set to reduce loss and not following it greatly increases risk.

The third objective is saving the trader time. Since the system is doing the trading for the trader, the trader is able to spend more time selecting stock and potentially trading himself. Or the trader can use the time to improve the system. The additional time gained can be used for any number of tasks.

Overall trading systems are designed to make money. However different systems and different markets have slightly altered objectives to maximize profit. When creating a machine it is important to have the market and time frame objectives in mind.

## 3.7 Trading System Parts

A trading system is a set of sub-tasks that when combined automatically trade a given asset. They are usually designed to run autonomously trading without direction. However, some trading systems can be used to provide an indicator that a human can then use to make the final decision. A standard trading system consists of the following parts: stock selection, buying, selling, and position sizing/allocation.

### 3.7.1 Selecting Stocks

Stock selection is a key component in producing a profitable system. If the stock is trending down significantly over the past few months it is unlikely the system can trade it effectively. Varying fundamental and technical analyses can be used for this process. A stock type commonly used with precision trading systems is called a Smoothie. There is security with a high average true range, high volume, and high moving price. They provide high liquidity due to volume and high volatility to maximize profit potential.

Stock filtering software like Finviz [11] can be used to create a custom filter for stock selection and then monitored over time to only trade the assets captured in the filter.

### 3.7.2 Entry Rules: When to Buy an Asset

Position creation timing is key to a profitable algorithm. In other words, when the algorithm decides to buy can make or break an algorithm. There are many ways to create entry

rules. An effective algorithm could consist of a complex set of criteria required to buy, or based on technical indicators such as multiple moving averages.

A complex system can have many criteria to buy. Some of these could be after a stock rises two percent above a minimum, or if the relative strength index (RSI) is above 70 considered overbought, or both. The key is combining these smaller indicators in an effective way that enables buying at the minimum value.

A more simple moving average based entry strategy consists of two moving averages usually a short one like ten days and a longer one such as twenty-five days. Every time the ten day crosses from below to above the twenty-five is a buy signal, and every time it crosses in the other direction is a sell signal.

Both these methods involve pattern analysis and attempting to find a profitable strategy in seemingly random stock prices. A more advanced method is leaving the pattern recognition to advanced statistical, mathematical or machine learning systems. A machine learning approach can train on historical data and then produce buy signals when the previous stock prices matches an older pattern. There are many types of machine learning algorithms that can work. However, some are better suited for stock data than others. They are effective because they can methodically find patterns in high dimensional data much better than humans. Humans are good with 2D graphs, and certain 3D charts. but with anything greater, such as the N-dimensional market data, humans have a difficult time.

#### 3.7.2.1 Order Types

After an entry rule is activated the type of order needs to be decided. The various order types for buying and selling are: market orders, stop orders, and limit orders.

Market orders are performed at the present market price. However, this price can change between when the order is initiated and when the order is completed. This is because of the buy-sell spread.

Stop orders are orders placed that do not execute immediately. When a stop buy order is initiated it waits for the stock to rise above a certain price before buying. However, these choices are rarely used. A stop loss order sells the stock when it goes below a certain level. This is valuable when an asset is being held, and the trader wants to minimize loss if it starts to go down. A stop order is always executed at a higher price for a buy and a lower price for a sell.

A limit order is the opposite of a stop order. A limit buy order is executed if the stock price goes below a certain level. A limit sell order sells when the stock goes above a certain level. Unlike a market order, a limit order is always executed at the specified price or lower for a buy, and higher for a sell.

The benefit of the stop and limit orders is that they can be more precise when compared to a market order. As well as providing a risk reducing measure when the stock is sold automatically to reduce loss or take profit.

A more advanced order is the trailing stop loss. This is used to maximize profit. It works by selling the stock after it has decreased a set amount from the maximum. For example, if a stock was bought a \$100 per share and it rises to \$110 then down to \$90. A trailing stop loss of 3% would be sell the asset at \$106.7 because it has decreased 3% from the maximum. If a standard stop loss of 3% below the purchase price was used, it would have sold at \$97 for a profit of -\$3 as opposed to \$6.7.

When shorting stock the terms are slightly different. The term shorting refers to when a short position has been created on a stock. Covering refers to closing the position

### 3.7.3 Exit Rules

Exit rules can be just as, if not more, important as buy rules, exit rules are the key to both taking a profit and avoiding an unnecessary loss. The ideal exit rules would close a position when the stock has reached a maximum before a fall. However, the ideal is not always achievable. The next best is setting criteria to sell based on length of time held, profit taking threshold, stop loss threshold, and indicator based. By setting these criteria well an algorithm can perform close to the ideal.

When trading short term if an algorithm is effective at predicting stocks intraday, it should not hold trades for a period longer than that. Therefore a maximum position holding time should be set at a day to automatically sell stocks at that point. Or all positions can be exited at the stock market close. The same strategy can be used for different time periods by altering the time threshold.

A trailing stop loss is used as described earlier to achieve a profit close to the maximum. While the goal is to always make a profit, sometimes it is best to sell a position before too much money is lost.

A profit taking threshold can be set to guarantee a profit when a position has increased a certain amount. While this may reduce the profit from a stock that increases after being sold, it will guarantee some profit. The goal of this strategy isn't to have one trade that produces the majority of profit, it's to have lots of small but profitable trades.

### 3.7.4 Position Sizing Rules

The final part of a trading system is the position sizing strategy. The simplest, but the riskiest method is to utilize all available money for every trade. For example for every buy signal

buy as many stocks as possible, and for every sell, sell everything. This is the most risky, but also has the maximum profit potential.

Another strategy is to slowly buy and then slowly sell. Every time a buy rule is activated the algorithm will purchase a set amount of stock and slowly increase its position until it starts selling to decrease the position. This averages out the buying price and selling price and significantly reduces the risk compared to buying and sell all at once.

The optimal strategy is to allocate based on perceived risk. A position with a higher risk should have less money because it minimizes the loss potential. Less risky positions should be allocated more money because they have less potential to fall. This strategy reduces risk as much as possible to allow for consistent, safe growth.

# Chapter 4: Optimizing and Analyzing Trading Systems

## 4.1 Optimization

Optimization is a way to improve an algorithm by altering the key variables to maximize the objective function. The objective function defines a way of determining how effective an algorithm is. An example objective function is a return on investment function, where an algorithm with a higher return would be better than one with a smaller return. The optimization step is after the algorithm has been developed thoroughly and all aspects have been considered. Incorrect use of optimization can lead to false conclusions. The idea is to optimize the algorithm on a certain set of data, and then test on another set. The two sets should be disjoint with no overlapping data. System and algorithm are used interchangeably in this section.

## 4.2 Walk Forward Analysis

One of the most common algorithm testing methods. It is when you use past data as if it was coming in live and trade based on it. Then you can see how much money the algorithm would have made if it were trading real money during that time period. One can look at the return over the testing period and determine how successful algorithm is.

### 4.2.1 How to perform a Walk Forward Analysis

Many trading systems have walk forward analysis built into their software. This section goes over how to do a walk forward analysis yourself.



To perform a walk forward analysis a stock needs to be selected and its historical price data downloaded. The data should generally be more than one hundred data points or more than three months of data. A starting account budget of \$100,000 is generally good because commissions do not play a significant role and the algorithm can allocate assets more precisely.

The algorithm should be fed one data point at a time. The algorithm then determines what action to do given the data point and the data it has stored on the previous points. Some algorithms need a warm-up phase to collect enough stock prices to perform its analysis.

If the algorithm decided to buy some of the cash held should be converted to stock based on the quantity the algorithm bought. If the algorithm decides to sell the stock should be converted to cash based on the amount sold. After every new point is fed to the algorithm its total value of cash plus stock should be saved for analysis afterward.

Each data data point should be sent to the algorithm and the above steps should be repeated until the data set has been exhausted. At the end convert any outstanding stocks to cash and the final cash value represents the final walk forward equity value.

#### 4.2.2 Understanding the results of a Walk Forward Analysis

Walk forward analysis returns many valuable metrics about the algorithm. They can be used to optimize the algorithm. However, incorrect analysis of metrics can lead to inaccurate results. The final results for an algorithm should always be on data never tested before. Implicit cheating or optimizing your algorithm to work well on the test data will not improve the over algorithm. All key metrics need to be considered for an algorithm and the goal of the algorithm needs to be kept in mind. If the goal of the algorithm was low risk, a low ROI, but minimal drawdown would still be considered effective. Below are the key metrics and how to use them.

Return on Investment (ROI) is the best overall metric for walk forward analysis. It is the percent change from the original price. For example, if the initial amount allocated was \$100,000 and the final result after the walk forward was \$110,000 the ROI would be 10%. A higher number is better and represents a large amount of money earned. However, it is important to compare ROI against other algorithms or baselines run on the same period. The S&P 500 ETF, SPY, can be used as a good comparison. If the ROI for SPY was 10% over a year and the system ROI was 6% it means that the money could have made more money if it had been invested in SPY.

Number of trades is good to keep track off. Too many means commission will have a significant effect on the ROI. Lower is generally better and 1 can even be fine as long as the ROI is at the desired value.

The final portfolio value reflects the ROI. However, it represents the absolute price instead of the percent change.

Average trade ROI is good for understanding the individual trades. Average trade ROI will usually not be the same as the over ROI, because the average trade ROI is not weighted. For example, if the overall ROI was 10%, but the average trade ROI was -5% this signals that a few trades made a majority of the money. This is not a good sign because it shows the algorithm is inconsistent.

Win percentage represents what percentage of trades were profitable. If a trade created at a stock price of \$80 and then sold at \$82 it would be a profitable trade. A losing trade would be a buy in at \$82 and a sell at \$80. The count of profitable trades over the count of total trades would be the win percentage. If all the trades were profitable it would be 100%. A larger percentage is better.

Average holding period is the average time each position is held before being sold. This can be helpful when trying to optimize an algorithm for a certain time period. If the goal is to trade intraday the average holding period should be less than a day. If it is longer than a day then the exit strategy needs to be modified.

Maximum drawdown is the maximum distance between any high and its following greatest low. If the algorithm started with \$25,000 and it decreased down to \$24,000 before climbing consistently until \$26,000 with no other decreases. The maximum drawdown would be \$1,000.

Any of the above metrics aggregated for hourly, or daily results can also be valuable. This only works when the algorithm has access to minute or hour data. It is common for more volatility around 9:30 am when the markets open, so the algorithm may be more effective then. Therefore the algorithm could only be run at that time and not later to maximize ROI.

In addition to key metrics visualizing the results in the following ways can be very valuable.

Plotting all buys, and sells overlayed on the algorithm equity history and current stock price. The trades can be analyzed to see determine whether the trades are being sold at the correct time to view how many trades were profitable at one point, but were then sold for a loss.

A histogram of trade ROI and a histogram of trade holding length provide an improved way of understanding the two metrics. It can help visualize which threshold, stop loss, profit taking, or time expiration is being activated the most. If a majority of trade ROI's are next to the stop loss threshold you know the stop loss is being activated often. If the trades are spread out it can be assumed the maximum time criteria is being activated.

### 4.3 Using walk forward results to optimize the algorithm

Using the key metrics the algorithm can be optimized on the training data, and then tested on the testing data. Here a few examples are presented on how to adjust an algorithm based on certain results.

Trades are sold too long after the maximum. This is caused by incorrectly set sell criteria. A fix would be to add a trailing stop loss to sell stocks a certain percentage after a local maximum.

Positions are held too for the goal time period. For example, when performing intraday trading a stock is held for 1 week. This problem can be fixed by adding a trade timeout of 1 day so a trade never exists for more than 1 day. This will allow the algorithm to focus on a single time period and be more effective.

Trades are sold prematurely because of hitting a stop loss before an increase. This is caused by a stop loss being too restrictive. While it does help reduce the maximum loss for any stock it also can reduce the gain. It is unlikely that any algorithm will buy at the bottom so a good stop loss will allow for some downward movement before moving up. A potential fix would be increasing the stop loss slightly to allow for a larger drawdown.

The algorithm's overall return is positive but less than expected. This can be caused by not allocating enough money to each trade or the algorithm. To fix this increase the size of each trade and allocate more money to the algorithm to create more risk, but a potentially higher ROI.

More than 50% of trades are winners but the algorithm is still losing money. This is caused by an ineffective allocation algorithm. It needs to be fine tuned to allocate more money to less risky stocks, and less money to risky stocks. This will reduce the risk and increase the potential for gain.

## 4.4 Comparing Algorithms

Trading and investing algorithms are compared most often with annualized return on investment. In other words, it is the percent increase in money if extrapolated to, or averaged over one year. If an algorithm returned 5% over 4 months then it could be expected to earn 15% over one year. Or if an algorithm earned 24% over 2 years its annualized return on investment would be 12%.

Another method of comparison is by directly comparing each key metric across both algorithms. It is important to note that if the numbers are close to each other the result from either algorithm could be larger. It is important to determine whether the difference is of statistical significance.

## 4.5 How to Optimize and Analyze a System of Systems

The same methods used to optimize individual algorithms can be used to optimize a system of systems. The key addition is that instead of just changing parameters you can swap algorithms in and out, and allocate money corresponding to the best performing algorithm. Thus, if one algorithm is effective from 9am-12pm it can replace an algorithm that is ineffective from 9am-12pm.

The goal for the system of systems is essential to optimize the ROI. A system cannot be expected to be highly profitable at everything, and it does not need to be. A system strategy should be chosen and strictly followed to achieve the best ROI. An example strategy could be to only focus on trading tech stocks intraday.

The base system work off could be one that allocates money equally to each system and runs on all stocks continuously. This is a solid base point to optimize from.

Once the system is created it is optimized on the training data, then evaluated on the testing data. Training data is the data used to train and optimize the system. The algorithm should be run on the data multiple times until it finds the combination of settings that produce the best ROI. The testing data is a set of data that is only used for producing final results. If the algorithm is modified after running it on the testing set, a new testing set should be chosen that the algorithm hasn't used before. Once those settings have been selected it is run on the testing data to produce the final results.

## Chapter 5: Trading System #1 Machine Learning Trader

### 5.1 Overview

This algorithm is heavily based on the book *Advances in Financial Machine Learning* [12]. Its goal is to take the concepts from the book and extend them using ideas from papers, and ideas of my own. It was extended using the Heikin Ashi [13] transform and used different buying and selling methods to increase the likelihood of a profitable trade. All code was developed in Python using Numpy [14][15] and Pandas[16] for data processing and formatting, Scikit-learn[17] for Machine Learning, and Matplotlib [18] for graphing.

### 5.2 Stock Selection Criteria

The stock selection criteria were as follows. The stock needed to have a positive trend over the last two months, and a positive over the past half year. A beta of 0.5 and 1.5 is good to reduce volatility and allow for better swing trading. A RSI of less than 60 will make sure the stock is on an uptrend but not overbought. A volume of greater than 1 million per day will allow plenty of liquidity for trading.

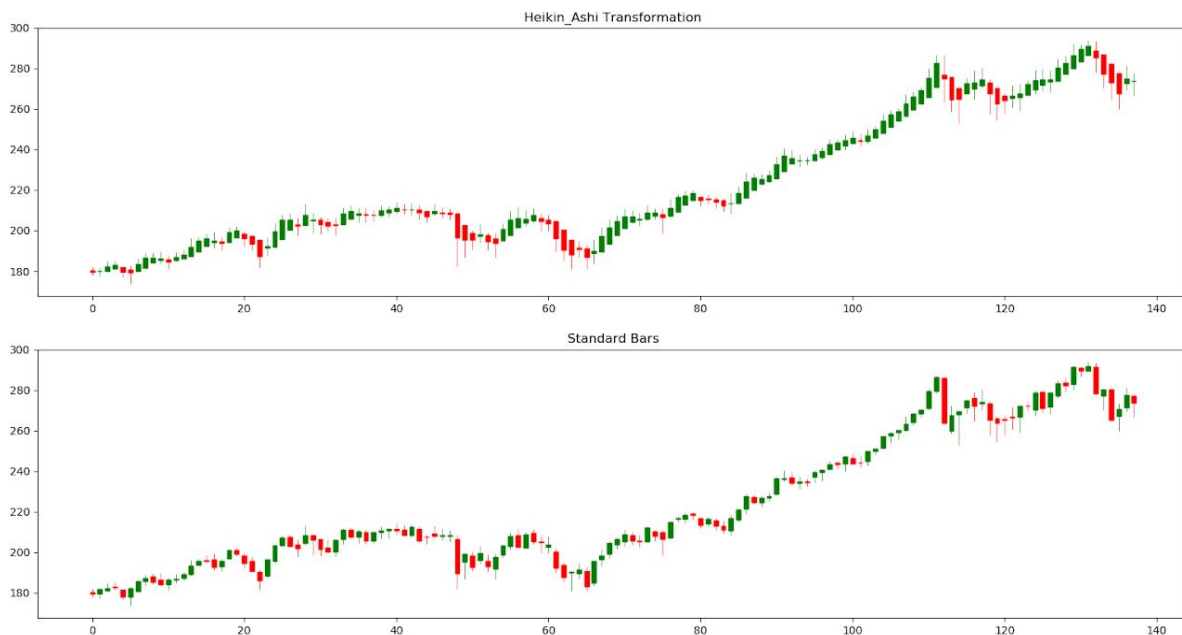
#### 5.2.1 Selected Stocks based on Criteria in 5.2

The stock selected from the criteria in 5.2 were JD, CZR, GPK and BP.

## 5.3 Stock Price Data Preprocessing

### 5.3.1 Transforming Yahoo Data into Bars

The data processing begins by downloading the previous five years of stock data from Yahoo Finance. Then the open, high, low, close and volume are extracted. Each day is represented by one bar, or a single set of Open, High, Low, Close, and Volume values. The daily bars are aggregated into bars that have the same amount of cumulative volume and may cover multiple days. The amount varies based on the stock. After the stock has been aggregated it is sent through a Heikin Ashi transform which makes the data easier to process for the machine and more likely to find a pattern. An example Heikin Ashi transform is shown in Figure 2.



**Figure 2:** Heikin Ashi Transformation Bars (top chart) vs Standard Bars (bottom chart). There is a noticeable difference in smoothness. When the chart is rising around point 100 on the x axis the bars are much more consistent, and when the price is dropping near the right side it



averages them into one large downtrend. The instrument used was the SPY ETF from 2013 to 2018.

### 5.3.2 Cumulative Summation

Training data selection occurs through a cumulative summation algorithm. This algorithm selects time periods after a price movement in the 75% percentile. The algorithm for cumulative summation is shown in Figure 3.

$$\begin{aligned} S_t^+ &= \max \{0, S_{t-1}^+ + y_t - E_{t-1} [y_t]\}, S_0^+ = 0 \\ S_t^- &= \min \{0, S_{t-1}^- + y_t - E_{t-1} [y_t]\}, S_0^- = 0 \\ S_t &= \max \{S_t^+, -S_t^-\} \end{aligned}$$

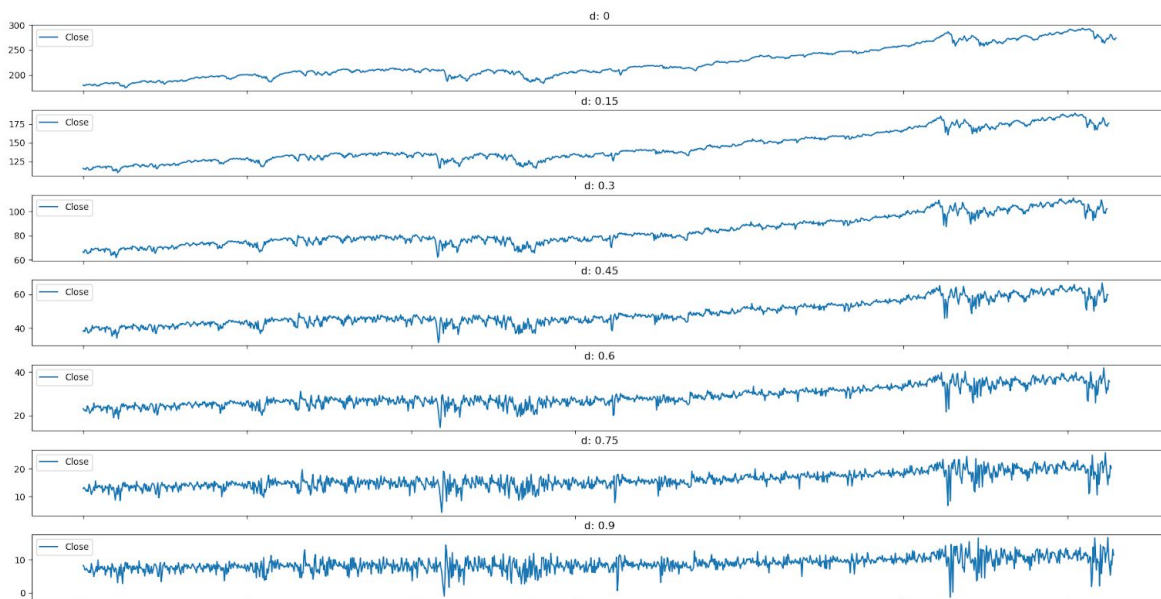
**Figure 3:** Cumulative Summation Equation. [12] S+ and S- are values that are updated at every bar. When they reach a threshold a flag is updated indicating a training sample will be removed.

This percentile is calculated by taking the high minus the low and finding the mean and standard deviation for all days in the training set. It then uses the data to calculate the 75th percentile of movement. It uses this as the threshold for saving training data points. The next step is determining the training points used for training the machine learning agent. The algorithm looks at the data following the last value in the training point. Then a barriers are placed five days in the future, 1% above, and 1% below the last price. The first time the stock moves through any barrier it is recorded. The stock is then labeled as 1, 0, or -1 corresponding to 1% above, the time barrier, and 1% below respectively.

### 5.3.3 Fractional Differentiation

#### 5.3.3.1 Overview

Once the training values are selected the data is fractionally differentiated [11]. To perform fractional differentiation the data is sorted from newest to oldest. Older data is generally less valuable when predicting the stock price. Additionally we care about how much the price is changing, the derivative, as well as what the value of the stock is. A weighting function based on a parameter  $d$  decides how much each past stock price will be weighted. For example the most recent stock price will be weighted 1.0 and as it moves back it will be weighed less until a stock 10 days ago is weighted 0. An example of varying  $d$  values is shown in Figure 4 below. This transformation narrows the range of values the algorithm needs to train for and effectively combines the rate of change with the stock value.



**Figure 4:** Fractional differentiation using different  $d$  values.  $D$  is the level fraction of differentiation. When  $d$  is 0 there is no differentiation. A  $d$  of 1 is equivalent to a first order derivative. A  $d$  between 0 and 1 such as 0.5 represents fractional differentiation. Notice how the

range of the y axis shrinks from top to bottom. As the graph moves from top to bottom the derivative plays a more significant role. The X axis units are arbitrary and represents time moving from left to right.

### 5.3.3.2 Method

Consider the backshift operator,  $B$ , applied to a matrix of real-valued features  $\{X_t\}$ , where  $B^k X_t = X_{t-k}$  for any integer  $k \geq 0$ . For example  $(1-B)^2 = 1 - 2B + B^2$ , where  $B^2 X_t = X_{t-2}$ , so that  $(1-B)^2 X_t = X_t - 2X_{t-1} + X_{t-2}$ . Now replacing  $k$  with  $d$ , we can form the following equation shown in Figure 5. Figure 5 is a dot product between  $B$  and the summation the only parameter  $d$  is used for calculating every term. When  $B$  is factored out the resulting equation is  $W$  shown in Figure 6. Figure 7 shows the orientation of  $X$  to produce the correct result when dot product with  $w$ . Figure 8 created  $X$  tilda, the result  $W$  dot product to  $X$ .  $X$  tilda will be the stock price values fed into the Random Forest algorithm.

$W$  can be costly to compute, however it can be computed iteratively using the formula shown in Figure 9. When calculated it can be multiplied by  $X$  to produce  $X$  tilda.  $W$  only needs to be calculated once for a given  $d$ , and then used for many different stock price samples.

$W$  is an infinite series that approaches zero. Eventually the dot product values reach a value that is too small to be valuable. This threshold is  $\tau$ . Once this threshold is set the first value in  $w$  that is less than this value is noted, and the remaining  $w$  values are cut off. This dramatically reduces calculation time, and only utilizes the data values that have an effect on the result. Figure 10 shows the equation for the cut off where  $l^*$  is the first index in  $w$  the absolute value of  $w$  is less than  $\tau$  ( $|w_{l^*}| < \tau$ ).

$$\begin{aligned}
(1 - B)^d &= \sum_{k=0}^{\infty} \binom{d}{k} (-B)^k = \sum_{k=0}^{\infty} \frac{\prod_{i=0}^{k-1} (d - i)}{k!} (-B)^k \\
&= \sum_{k=0}^{\infty} (-B)^k \prod_{i=0}^{k-1} \frac{d - i}{k - i} \\
&= 1 - dB + \frac{d(d - 1)}{2!} B^2 - \frac{d(d - 1)(d - 2)}{3!} B^3 + \dots
\end{aligned}$$

**Figure 5:**  $(1-B)^d$  expanded version. [12]

Then  $w$  can be defined as seen in Figure 5 where as  $i$  increases the value of  $w$  decreases.

$$\omega = \left\{ 1, -d, \frac{d(d - 1)}{2!}, -\frac{d(d - 1)(d - 2)}{3!}, \dots, (-1)^k \prod_{i=0}^{k-1} \frac{d - i}{k!}, \dots \right\}$$

**Figure 6:** Expansion of  $w$ . [12]

$$X = \{X_t, X_{t-1}, X_{t-2}, X_{t-3}, \dots, X_{t-k}, \dots\}$$

**Figure 7:** Expansion  $X$  [12]

$$\tilde{X}_t = \sum_{k=0}^{\infty} \omega_k X_{t-k}$$

**Figure 8:** Definition of  $X$  tilda. [12]

$$\omega_k = -\omega_{k-1} \frac{d - k + 1}{k}$$

**Figure 9:** Formula for calculating w iteratively. D is the parameter used for determining how rapidly the equation approaches 0. [12]

$$\tilde{\omega}_k = \begin{cases} \omega_k & \text{if } k \leq l^* \\ 0 & \text{if } k > l^* \end{cases}$$

**Figure 10:** Weights minimum cut off.  $l^*$  is the first index in w the absolute value of w is less than tau ( $|w_k| < \tau$ ). K is index into w. W tilda is the result. [12]

#### 5.3.4 OHLC combination

Each day consisting of the open, high, low and close is then flattened into an one dimensional array with four values for each day. Each training point is then saved to a file with its corresponding label.

### 5.4 Machine Learning

Machine Learning uses data and optimization algorithms to attempt and find the optimal result given a set of data and an objective. They work by learning how the training data corresponds to the answer usually using labeled data and then tries to generalize it to test images it has never seen before. It is a python implementation of the most commonly used algorithms including Linear Regression, SVM, Logistic Regression, Random Forest, and many more.

For this system the machine learning algorithm was trained on the past 5 years of daily data for a given stock not including the section of training data 5 months long. This provided a

good sized training set along with a testing set that changed every program run. A link to the code is provided in the index.

#### 5.4.1 Random Forest

Random Forest is a classification algorithm. When given an input vector it can label the vector a certain pre-specified class. An example of a simple random forest is one that can classify a number as positive or negative. It would accept a vector of one value long and return a 1 if it is positive, and 0 if it is negative. It would have one node that has a threshold at 0. If the number is greater than or equal to zero it is positive. If it is less than 0 it is negative. This idea can be scaled up to more complex ideas such as predicting when to buy or sell a stock. The input is the stock price history, and the output is a number -1, or 1, where -1 is buy and 1 is do nothing. For this project the random forest used has 2000 nodes. This enables it to more effectively learn the patterns in the stock price movement. Random forest was originally developed by J. D. Hunter [19].

This project used the sklearn random forest implementation of the Random Forest algorithm.

#### 5.4.2 Bagging

A bagging classifier takes a training set and divides it up into multiple sections. Each section trains a different classifier. For example if there are one hundred training points, and each classifier is trained on ten data points. There would be ten classifiers. Once they are trained every new data point is classified based on the average of the ten classifiers. This average creates a more effective algorithm because each classifier specializes in a section of training data. If the algorithm trained primarily on data trending down, it would be good at recognizing if the pattern would continue.

This project used the sklearn bagging classifier implementation for the trading algorithm. It used four random forest classifiers as the four trained classifiers. The reasoning behind four classifiers, is that each classifier can be specialized in a certain time period. The usual economic cycle can range from a few months to a few years. Since this algorithm is trained on 5 years of data when divided by four that leaves a little over a year for each classifier. This should effectively group each economic cycle into its own classifier. The bagging classifier used for this project was from Scikit-learn.

## 5.5 When to run algorithm

The algorithm is executed every time the cumulative summation described earlier moves up 1.5% or down 1.5% in a single movement.

## 5.6 Money Allocation

The primary algorithm works by trading a single stock. However the algorithm can be run multiple times and pointed at different stocks to reduce risk. The algorithm works by allocating its money all at once. For every buy signal it buys as many stocks as possible. For every sell signal it sells as many stocks as possible.

## 5.6 Selling

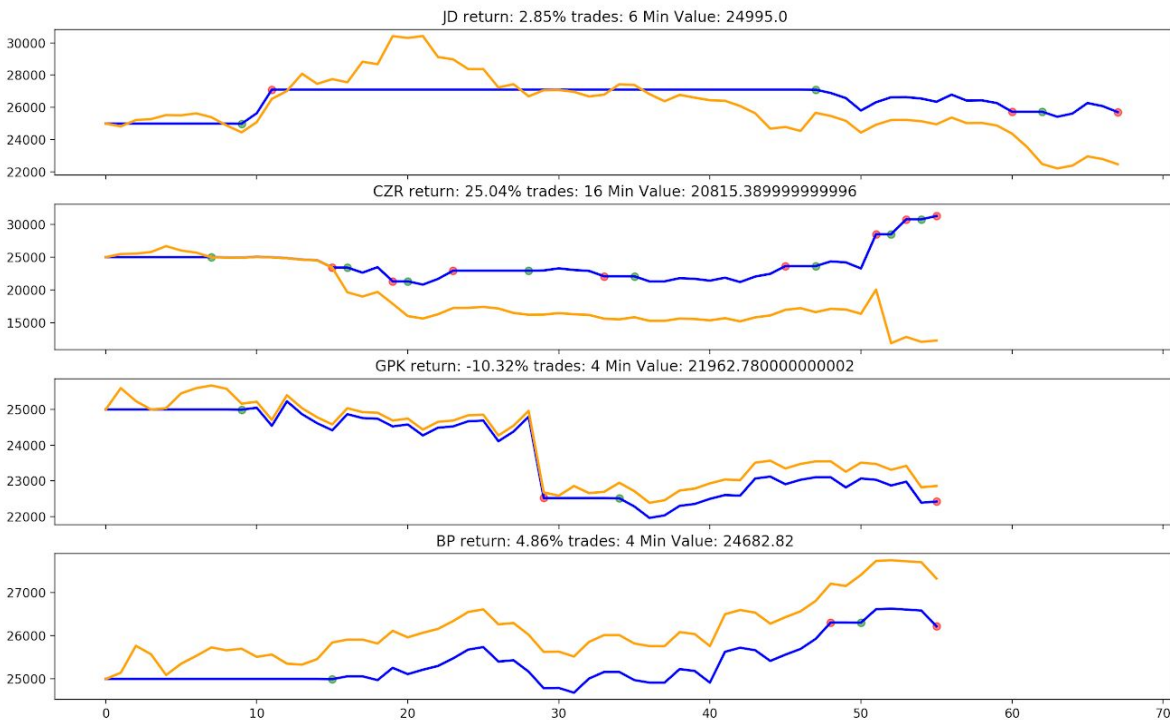
Stock sold at either 3% above or below purchase price or within 5 days of the purchase. 3% chosen because it is roughly the 75th percentile of price movement (high - low) for any stock in one day is around that amount. Therefore, if a stock moves this much in one day from open to close a significant event has most likely happened and it should sell.

## 5.7 Backtesting

The algorithm was back tested on 60 consecutive trading days. It performs a walk forward analysis focusing on ROI. It includes a \$5 commission on all buys and sells to reduce overtrading. The backtesting code was written from scratch to increase customizability. This customizability enabled the use of Sklearn for the random forest implementation, and the ability to produce improved results visualizations over other software.

## 5.8 Results

Figure 11 shows the walk forward results from 4 stocks each allocated with \$25,000 at the start. The average ROI of the algorithm is 23.56% annualized. The test covers a length of 60 trading days.



**Figure 11:** System #1 walk forward analysis



The stocks were JD, CZR, BP, and GPK. Test data is chosen randomly from five year data set and the random tree algorithm is trained on other part of the data. The orange line is the stock price. The blue line is the system's equity history. Red dots are sell points, green dots and buy points. The graph was produced using Matplotlib.

## Chapter 6: Trading System #2

### 6.1 Method of Trading

This system used a manual approach where time was put into studying stock market patterns as well as using backtesting and different criteria in order to narrow down the selection of stocks to choose from. All buy and selling was done manually as well. This allowed the ability to get out of a trade and sell the stock when something unexpected happened so that there was more growth than loss. Other than the use of the screener mentioned below, all of the analysis is done by the user.

### 6.2 Strategy

#### 6.2.1 Screener

The goal was to create a screener to assist the user in finding stocks that will maximize their investment. This screener finds stocks that are undervalued but are expected to have a high return on investment.

#### 6.2.2 The Seven Criteria

This screener utilizes these seven criteria: price over \$1, a price-to-earnings ratio over 20, a forward price-to-earnings ratio over 15, earnings per share growth this year over 20%, return on investment over 25%, and price-to-sales ratio over 3. These criteria helped narrow down the selection of stocks the user has to analyze and also gives them the ability to choose what they want to invest in from that given list.

### 6.2.3 Stocks that Pass Screener Criteria

After the screener has been run, it will leave the user with around ten stocks to analyze and look through. To create a manageable portfolio, the user should narrow down the selection of stocks down to about four or five based on their knowledge of the market and patterns that they might see in those given stocks.

### 6.2.4 Allocation

The allocation of money, in the end, is up to the user. It is safer to distribute the money equally over all of the stocks until you can safely decide how to split the money up if so desired.

## 6.3 Risk management

When making any kind of investment, the investor is taking on the risk that comes along with it. This system dictates that if the price of the stock drops 15% below the priced at which it was purchased, the stock will be sold. The risk decreases when the investor distributes their cash over multiple stocks/assets, so this is another way that the risk will be managed.

Filters: 7		Descriptive(1)		Fundamental(5)		Technical(1)		All(7)	
Exchange	Any	Index	Any	Sector	Any	Industry	Any	Country	Any
Market Cap.	Any	P/E	Over 20	Forward P/E	Over 15	PEG	Any	P/S	Over 3
P/B	Any	Price/Cash	Any	Price/Free Cash Flow	Any	EPS growth this year	Over 20%	EPS growth next year	Any
EPS growth past 5 years	Any	EPS growth next 5 years	Any	Sales growth past 5 years	Any	EPS growth qtr over qtr	Any	Sales growth qtr over qtr	Any
Dividend Yield	Any	Return on Assets	Any	Return on Equity	Any	Return on Investment	Over +25%	Current Ratio	Any
Quick Ratio	Any	LT Debt/Equity	Any	Debt/Equity	Any	Gross Margin	Any	Operating Margin	Any
Net Profit Margin	Any	Payout Ratio	Any	Insider Ownership	Any	Insider Transactions	Any	Institutional Ownership	Any
Institutional Transactions	Any	Float Short	Any	Analyst Recom.	Any	Option/Short	Any	Earnings Date	Any
Performance	Any	Performance 2	Any	Volatility	Any	RSI (14)	Any	Gap	Any
20-Day Simple Moving Average	Price 10% above \$	50-Day Simple Moving Average	Any	200-Day Simple Moving Average	Any	Change	Any	Change from Open	Any
20-Day High/Low	Any	50-Day High/Low	Any	52-Week High/Low	Any	Pattern	Any	Candlestick	Any
Beta	Any	Average True Range	Any	Average Volume	Any	Relative Volume	Any	Current Volume	Any
Price	Over \$1	Target Price	Any	IPO Date	Any				Reset (7)

**Figure 12:** FinViz Screener with Variables Applied

## 6.4 Backtest

The backtest for this screen came from the free service provided by marketinout.com. Shown below is the backtest and it's results. The dates that were used for this backtest were 1 January 2019 to 16 April 2019. Figure 13 shows the criteria used as well as the parameters set in place for the backtest. Figure 14 is the results of the test. The overall profit was \$3,355, with a growth of 34%. Over this span of time, there were a total of seven trades made. Figure 15 shows the equity graph, which shows the performance of this backtest versus the S&P 500. Lastly, Figure 16 shows the seven trades that were made during this period of time.


<b>Strategy Name</b>	IQP			 Edit Strategy
<b>Position Type</b>	Long			
<b>Strategy Definition</b>				
<b>Criteria for Opening a Position:</b>	Price Is Greater Than 1 Price Is Above SMA(20) P/E Is Greater Than 20 EPS, 5-Year Growth, % Is Greater Than 20 ROI, 5-Year Avg Is Greater Than 80  <b>daily</b> periods for all technical indicators			
<b>Backtest Parameters</b>				
<b>Initial Capital:</b>	\$10,000	<b>Stop Loss:</b>	20%	
<b>Capital at Risk:</b>	20% per trade	<b>Trailing Stop Loss:</b>	15%	
<b>Portfolio Max Size:</b>	20 positions	<b>Period:</b>	1/1/2019 - 12/31/2019	
<b>Testing Model:</b>	Close prices			
<b>Comm. per Trade:</b>	\$0.00			

Figure 13: Criteria and Parameters for Backtest

<b>Results</b>				
<b>Total Profit:</b>	\$3,355	<b>Total Trades:</b>	7	
<b>Capital Growth:</b>	34%	<b>Profit Trades, % of Total:</b>	86%	
<b>Profit Factor: [?]</b>	13.66	<b>Max Drawdown</b>	2%(\$265)	
<b>Expected Payoff: [?]</b>	\$479.28	<b>Abs Drawdown</b>	0%(\$0)	
<b>Payoff Ratio: [?]</b>	2.28	<b>Avg Market Impact [?]</b>	Minor 😊	

Figure 14: Results from the Backtest

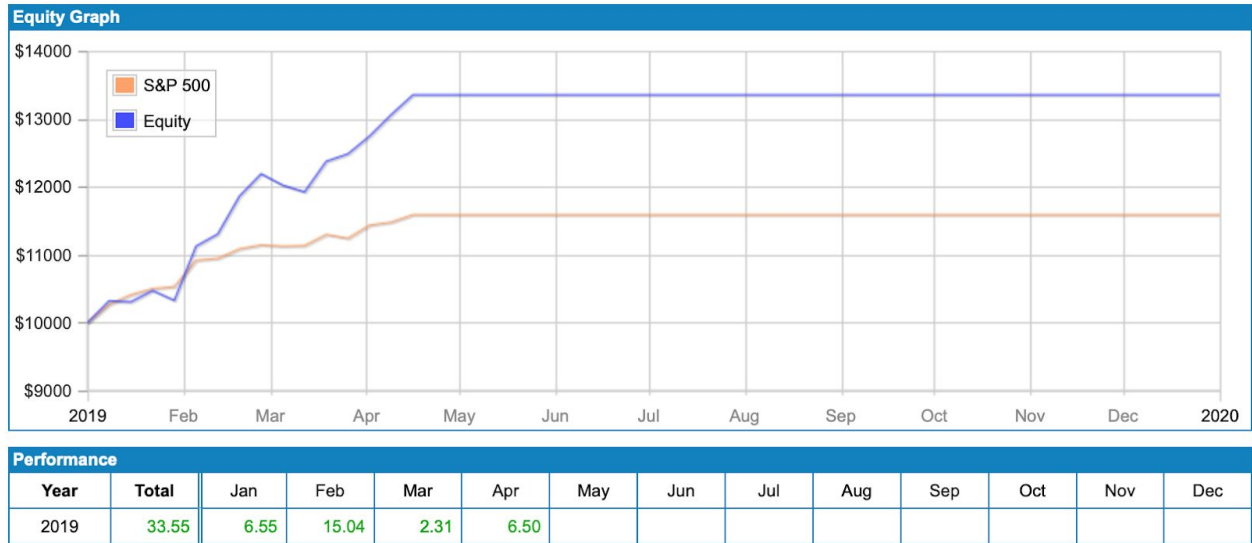


Figure 15: Equity Graph of the Backtest

Performed Trades:		All											
Symbol	Entry Date	Entry Price, \$	Exit Date	Exit Price, \$	Shares	Profit, \$	Exit On	Market Impact [?]					
MED	03/08/2019	131.47	04/15/2019	137.62	15	92.25	Period End	●					
ATHM	01/25/2019	76.02	03/08/2019	81.60	25	139.50	Stop Loss	●					
ATHM	01/09/2019	81.65	01/14/2019	70.61	24	-264.97	Stop Loss	●					
ANET	01/08/2019	214.28	04/15/2019	327.34	9	1,017.54	Period End	●					
VEEV	01/04/2019	90.99	04/15/2019	132.63	21	874.44	Period End	●					
TPL	01/02/2019	566.32	04/15/2019	894.12	3	983.40	Period End	●					
SPGI	01/02/2019	169.85	04/15/2019	216.47	11	512.82	Period End	●					

Figure 16: Backtest Trades

\*This backtest has different parameters due to a lack of options but is more or less the same as the strategy stated above.

## Chapter 7: Trading System #3

### 7.1 Method of Trading

This system is run manually with the operator issuing all buy and sell orders as well as placing all trailing stops. Once the screen is run any further analysis is also performed by the operator. This includes looking at various reports about the company as well as looking into any published financials of the company. The only automated part is the trailing stop which once set at the specified percentage will execute remotely.

### 7.2 Strategy

#### 7.2.1 Use a screen to determine high growth stocks

This system looks for stocks that are undervalued by the general market and thus have a high potential for gains. Also known as value investing.

#### 7.2.2 Uses 5 criteria

The screener uses 5 criteria to filter the undesired stocks out. These are; market cap less than \$1 billion, price greater than \$1, an average 10-day volume greater than 10,000, a Price/Sales ratio for the trailing twelve months in the lowest 10% of the market, and an estimated Earnings Per Share growth for the next year in the highest 20% of the industry. Looking for small and micro-cap companies filters out companies that have less room for growth while the minimum price of \$1 removes any highly volatile penny stocks. The 10-day volume average allows for easy liquidity. The P/S screen finds companies that have had low sales and

when combined with the estimated EPS growth looks for companies looking to grow their sales and thus increase the value of their company.

### 7.2.3 Ideally Leaves a Maximum of 10 Stocks

Once run the screen has returned at most 10 stocks to be examined further. This analysis is done manually by looking through various financial and news reports about the company and sector.

## 7.3 Allocation

The strategy for this system is to evenly allocate the stocks based upon how many stocks made it through the screening process. If five stocks are selected then 20% of the total capital available will be used for each stock. By not being dependent upon one stock it limits the

## 7.4 Risk management

Risk management is one of the most important and difficult aspects of any systems. When do you cut your losses and take profits? In this system, the risk is mitigated by a trailing stop loss of 5% and using the screener itself. The screener rarely selects stocks from the same sector due to the screen requirement being dependent upon the EPS growth in the highest 20% of its industry.

## 7.5 Backtest

The backtest for this screen came from the free service provided by marketinout.com. Due to this being the free version it only tested for the current year to date. The criteria are also

slightly varied when compared to the final screen. As shown in Figure 17 the return was 17% over the observed time frame and when compared to the S&P 500, as shown in Figure 18, beats it.


<b>Strategy Name</b>	growth		
<b>Position Type</b>	Long		
<b>Strategy Definition</b>			
<b>Criteria for Opening a Position:</b>	Capitalization: Micro-cap Price Is Greater Than 1 Average Volume, min Is Greater Than .01 Price/Sales to Industry, % Is Less Than 10 EPS, 5-Year Growth, % Is Greater Than 100  <b>daily</b> periods for all technical indicators		
<b>Backtest Parameters</b>			
<b>Initial Capital:</b>	\$10,000	<b>Trailing Stop Loss:</b>	5%
<b>Capital at Risk:</b>	50% per trade	<b>Period:</b>	1/1/2019 - 12/31/2019
<b>Portfolio Max Size:</b>	5 positions		
<b>Testing Model:</b>	Close prices		
<b>Comm. per Trade:</b>	\$0.00		
The report displays results for the current year only. Please sign up to view the full report.			
<b>Results</b>			
<b>Total Profit:</b>	\$1,651	<b>Total Trades:</b>	4
<b>Capital Growth:</b>	17%	<b>Profit Trades, % of Total:</b>	75%
<b>Profit Factor: [?]</b>	9.94	<b>Max Drawdown</b>	2%(\$185)
<b>Expected Payoff: [?]</b>	\$412.69	<b>Abs Drawdown</b>	0%(\$0)
<b>Payoff Ratio: [?]</b>	3.31	<b>Avg Market Impact [?]</b>	Minor 

Figure 17: System 3 Backtest

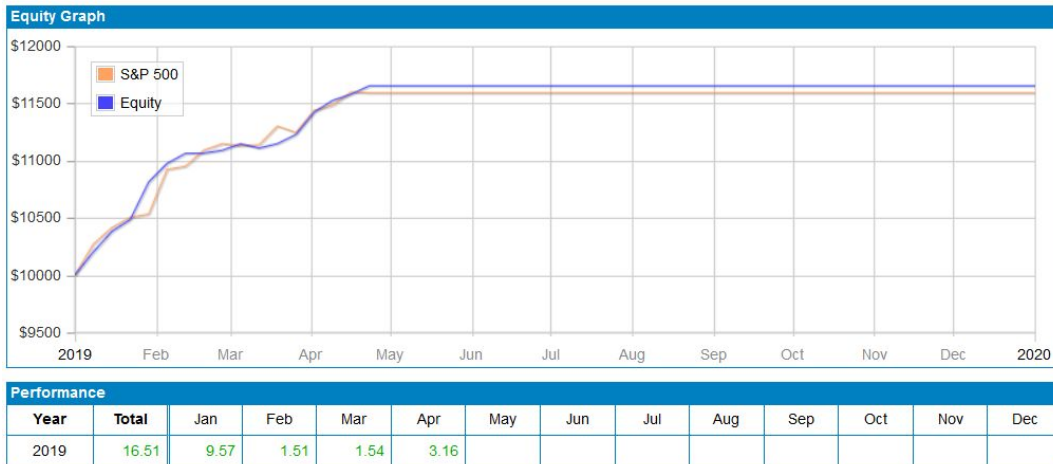


Figure 18: System 3 Equity History



## 7.6 Performance

The backtest had a return of 17% over the time period observed. When annualized this becomes a return rate of 40.83%.

## Chapter 8: System of Systems

### 8.1 Overview

The system of systems combines parts of the individual systems to create a better combined system. The resulting system trades multiple assets using the same strategy. It does not involve turning individual systems on and off, but rather a improved combination of systems. The main parts of a trading system are the stock selection, asset allocation, buy timing, and sell timing. Each of these parts were taken from systems #1-3. Stock selection uses the same method as system #2 and system #3. Asset allocation is based on system #3. Buy timing is based on system #1's Machine Learning approach. Sell timing is based on a general approach from all three systems.

### 8.2 Stock Selection: System #2 and #3

The stock selection criteria is based upon a minimum price and 10-day volume, max capitalization, P/S compared to the market, and the projected EPS growth compared to the industry. The read is referred to sections 7.2.2 and 6.2.2 for more details.

### 8.3 Asset Allocation: System #3

Allocating the selected stocks based upon those stocks that exhibit the most desirable traits and values. Refer to section 7.3 for a more detailed description.

## 8.4 Buy Timing: System #1

The algorithm utilizes the buying system used in system 1's Machine Learning algorithm. The stock price data is fed into the algorithm and if the algorithm returns a -1 it buys and if it returns a 1 it does nothing. Refer to Chapter 5 for a more detailed description of how the algorithm works.

## 8.5 Sell Timing: General System

The system of system utilizes three sell strategies. The first one is based off a trailing stop loss with a threshold varying by the time period used. For short term use the stop loss is set at 1%, for intermediate 3% and for long term 5%. The stoploss covers changes between -5% and 5%. The second strategy is a profit taking measure which automatically takes profit when the strategy increased by 3% in the short term, 5% in the intermediate term, and 10% in the long term. The final strategy is length of position. For short term the position is sold after 3 days, for intermediate term 8 days, and for longterm the position is kept until one of the above sell criteria is met.

## 8.6 System Retirement Criteria

The system is disabled if it loses a specified amount of money in a specified amount of time. The retirement thresholds are -10% in a month, -7% in a week, and -15% since creation. It will run until one of these criteria are met, or it is manually disabled.

## 8.7 Long Term Implementation

Buying is limited to once per week and selling is unrestricted.

## 8.8 Intermediate Term Implementation

Buying is limited to every other day and selling is unrestricted

## 8.9 Short Term Implementation

Buying and selling is unrestricted.

## 8.9 Optimization

The System of Systems has a lot of parameters that could be varied to optimize the system. We choose to focus on the machine learning aspects of position creation, the position closing parameters, and asset allocation. All the optimization was always done on a separate set of data than the training and testing set, to reduce overfitting and false results.

### 8.9.1 Machine Learning Optimization

Machine learning models come with many parameters called Hyperparameters. They are parameters that the machine cannot effectively learn and relies on the programmer to find the optimal value. Two two most significant hyperparameters are the number of classifiers in the bagging classifier, and the number of classifiers within the random forest algorithm.

We started out with a very large amount of one thousand classifiers in the bagging algorithm, and then five hundred classifiers within each of the bagging classifiers creating a total of fifty thousand small classifiers. The size of this algorithm was too large for the size of the data we were working with. Five years of daily minus the testing set comes out to about 1100 days of data. Then the preprocessing algorithm reduces the amount of training data significantly. Since there are more thousands more classifiers than data points it could

hypothetically memorize every training point. This would be inefficient though because it wouldn't generalize to new data well. We reduced down to 4 bagging classifiers and 2000. This enabled the system to generalize more effectively and produced a better ROI

The next hyperparameter was the threshold for when the cumulative summation algorithm would activate. Originally it was set at a constant threshold for every instrument (stock). This caused the algorithm to flag too many or too few training examples producing bad results. In order to set a better threshold we raised it to the 75th percentile of daily motion defined by the high minus low for every day. This greatly standardized the threshold and brought the number of training examples into a better range.

### 8.9.2 Position Closing Parameters

Choosing the best closing parameters required hundreds of back testings systematically looking for the best result. The equity history visualization used for showing the results proved a good tool in finding the best values. It allowed us to see which threshold was being hit and how to adjust them to maximize the ROI. An important threshold was the time stop loss value. It was initially set to 3% below the purchase price. The profit taking price was set 5% above the purchase price. We noticed that positions that increased 3-5% followed by 6-8% drop were not being sold. This was caused by our stop loss and profit taking values being set incorrectly. To improve this we added a trailing stop loss in place of the standard stop loss. This guaranteed profit as long as the position increased more than the trailing stop loss. For example if the trailing stop loss was 3% and the position increased by 3.5% there would be a return of 0.5%.

Choosing the trade timeout period was optimized using the same process as above. We would look to see when a stock was being sold and whether the threshold could be changed to

increase the overall ROI. Many times the algorithm was selling too early and not realizing the maximum gains. When we increase the timeout length, the ROI increased by 2-3%.

### 8.9.3 Asset Allocation

Initially we had a progressive allocation strategy. Each instrument was given an equally amount of money. Each time they received a buy alert they purchased twenty percent of their allocated money. This process would repeat until they were fully allocated, or a sell criteria was hit and all assets were sold. This was a successful strategy but it severely limited ROI. The strategy was not allocating enough money to create a substantial return. To fix this issue we increased the risk and made every buy signal purchase as many stocks as possible. Since the algorithm was already profitable this change increased the return by about five times.

## Chapter 9: Results for System of Systems

### 9.1 Subsystem individual results

Each subsystem was profitable and the final system of systems was much more successful than any individual system. The three systems through different approaches all earned money over the period they were tested. They produced a return that was more significant than the SPY ETF. More detailed results are included in Chapter 6, 7, and 8 for each individual system.

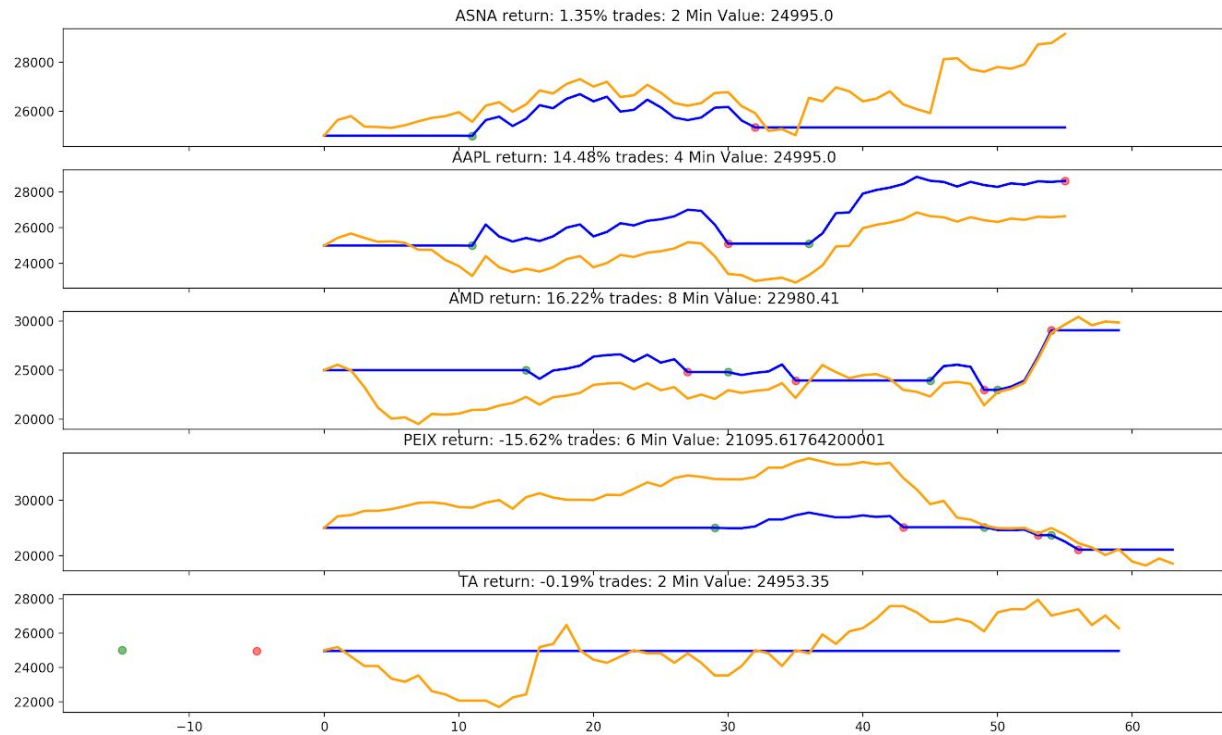
### 9.2 Overall Results

The average ROI of the system of systems across all time periods was 26.67% annualized which is significantly better than the S&P 500 at 9.7% [20]. The number of trades is reasonable and the resulting commission does not eat into the over ROI very much. The number of positive stocks backtests was more than negative backtests which shows the algorithm is net positive, however the amount positive needs to be considered.

The equity graph provides a visual understanding of the backtesting process. The orange line is the stock value if \$25,000 worth of stock was bought and held the length of the test period. The blue line is the equity curve of the system of systems. A green dot is a buy signal, and a red dot is a sell signal. Every green dot must be followed by a red dot. The dots can be connected to represent a single position. The distance apart of these dots represent the length of time the specific position was held. Additionally, whether the connected dot is lower, higher, or at the same level shows which selling criteria was met. This graphic effectively displays a majority of the optimization metrics at once.

For each time period tested the equity curve visualization provides evidence that the holding period and buying/selling criteria are set effectively to maximize profit. For the long term, there is no expiration timer which is shown through the long position holds. The profit taking is shown to be effective because it sells at the peak multiple times. It is early sometimes but it still represents a significant gain. Some of the backtests show little to no trading. This is because the system of systems buy signal was never activated, and the system did not want to trade the selected stock.

### 9.2.2 Long Term Results



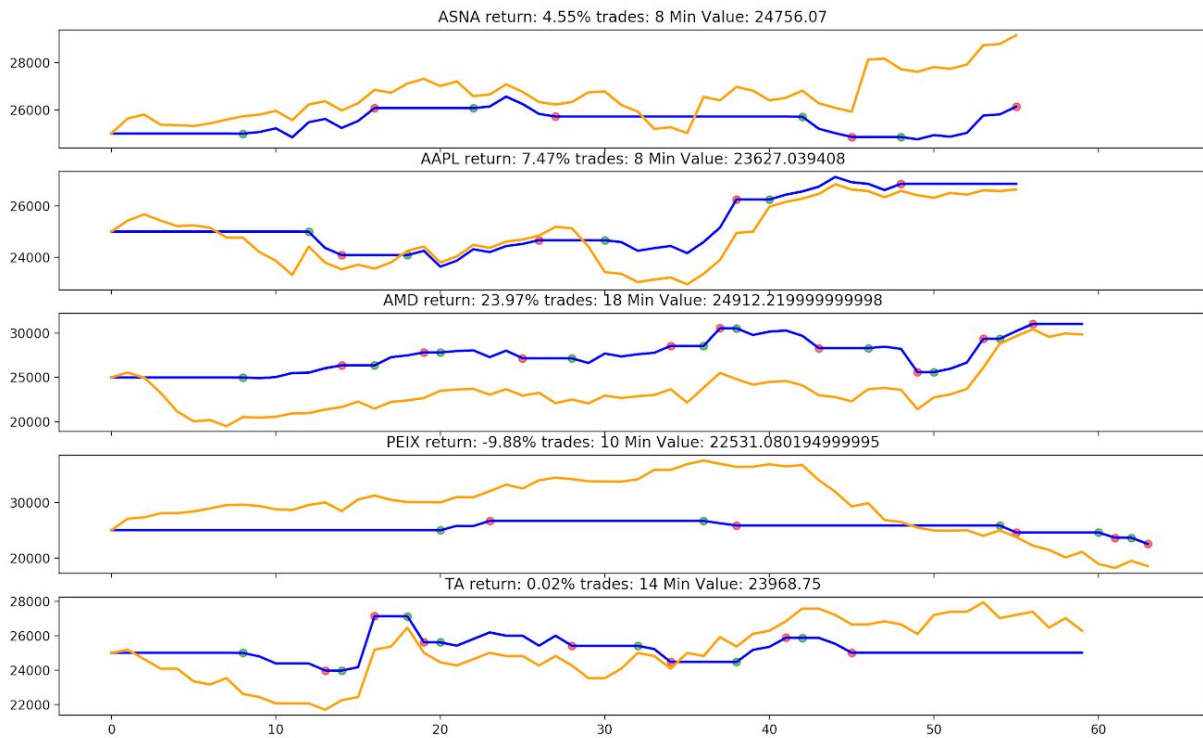
**Figure 19:** Long Term Time Frame Results.

The results are visualized in Figure 19. The ROI over the 60 day period was ROI: 4.05%. The **annualized ROI was 17.05%**. This result is notably worse than intermediate or short term results. However it does show effective trading with its AAPL return, and the AMD return. Both



these backtests show the algorithm capitalizing on primary upward movements and reducing drawdown. It also traded minimally and still returned a very good ROI when compared to SPY.

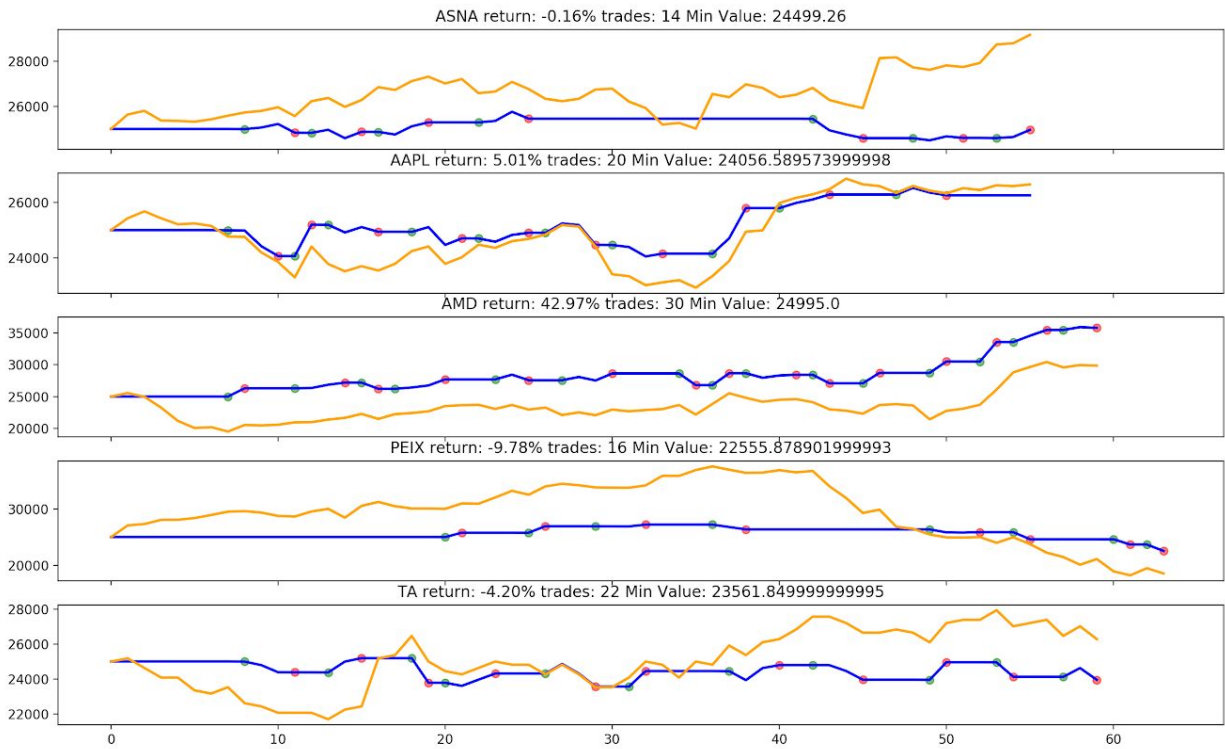
### 9.2.3 Medium Term Results



**Figure 20: Intermediate Time Length Results:**

The results are visualized in Figure 20. The ROI over the 60 day period was ROI: 6.53%. The **annualized ROI was 27.43%**. This result was better than the long term results mainly due to the fact that it traded less when PEIX. As compared to above where it did not sell until PEIX was much lower. Additionally, it was able to capitalize on more gain from ASNA than the long term results.

## 9.2.4 Short Term Results



**Figure 21: Short Term Results**

The results are visualized in Figure 21. The ROI over the 60 day period was ROI: 8.46%. The **annualized ROI was 35.53%**. The short term back test was the most successful of the three backtests. The most notable result is from AMD. It was able to produce a return of 42.97% even when the AMD stock itself was closer to a 20% ROI.

## Chapter 10: Conclusions

### 10.1 Summary

The System of Systems was much more effective than any of the underlying systems and represents a successful aggregation of subsystems. Throughout the process, we have learned a significant amount investing and trading. At the start, we had little to no experience trading or investing and had knowledge of how to develop an autonomous trading system. By the end, we were able to create an advanced and very profitable algorithm that produces a ROI significantly greater than the SPY S&P 500 ETF (9.7% average yearly). The goal of developing a system a novice trader could use to increase his trading performance was successful. This algorithm could be deployed alongside a day trader and produce a sizeable ROI with no additional work.

### 10.1 Is it safe to trade with real money?

The overall ROI is noteworthy however more testing is required to analyze the risk. Larger returns usually come with more inherent risk. Over the multiple tests, it has been consistently positive, which is a good sign. It could be used for real trading because of its ability to produce a high return. Even if given a small amount of money a ROI of 26.67% will return more money if applied to \$1000 compared to \$2500 dollars with a ROI of 9.7%. However, it should not be the sole trading program. It should control only a small amount of money and run on multiple stocks to minimize risk. The code required to access live data and perform trades needs to be completed in order for real money to be tested. Before it trades with real money paper trading should be tested to make sure the system is trading as expected.

Another use could be as a technical indicator that is used by a human while trading. This provides a second layer of risk prevention and adds a second viewpoint on whether the trade will be profitable or not.

## 10.2 Problems and Issues

The most worrying thing about the results is the risk factor. When run on one stock it has the potential to lose a significant amount of money, however, when the results are averaged over 5 stocks the majority are winners and outweigh any of the losers, even when they are significant losers.

The largest problem with the project was developing an intelligent random forest algorithm. There are many hyperparameters that need to be tuned in order to create a profitable algorithm. Many iterations and thousands of backtests were performed in order to find the parameters that produced a significant ROI.

Another challenge was connecting the Random Forest algorithm to a backtesting program. There are no free backtesting algorithms online that have a connection to the Scikit learn Python library. Therefore we had to create one from scratch in order to test our algorithm. This was a simple process to start however when we needed to add more advanced features like trailing stop losses and better visualizations it became more difficult.

## 10.3 Future Work

The next step to improve the system would be implementing Neural Network buying signals. Neural networks better learn the trends and should be able to create better buying and selling signals that produce more profit. Additionally, a machine learning or neural network

algorithm could be trained to decide to sell as well. Selling based on the current criteria is effective but there are opportunities to be more efficient and maximize profit.

## References

- [1] W. Thorp, "Searching for Future Stock Market Winners," May 2018. [Online]. Available: <https://www.aaii.com/journal/article/searching-for-future-stock-market-winner>. [Accessed 18 February 2019].
- [2] AAI Staff, "Micro-Cap Stocks Outperform, Often in Consecutive Years," May 2018. [Online]. Available: <https://www.aaii.com/journal/article/micro-cap-stocks-outperform-often-in-consecutive-years>. [Accessed 23 February 2019].
- [3] A. Moghaddam, M. Moghaddam and M. Esfandyari, "Stock market index prediction using artificial neural network," 1 July 2016. [Online]. Available: <http://www.scielo.org.pe/pdf/jefas/v21n41/a07v21n41.pdf>. [Accessed 2 March 2019].
- [4] A. Hayes, "Value Investing Definition," 25 March 2019. [Online]. Available: <https://www.investopedia.com/terms/v/valueinvesting.asp>. [Accessed 27 March 2019].
- [5] E. Norris, "Benjamin Graham: The intelligent investor," 22 May 2018. [Online]. Available: [https://www.investopedia.com/articles/07/ben\\_graham.asp](https://www.investopedia.com/articles/07/ben_graham.asp). [Accessed 31 March 2019].
- [6] J. Chen, "Growth Investing," 16 May 2018. [Online]. Available: <https://www.investopedia.com/terms/g/growthinvesting.asp>. [Accessed 25 March 2019].

- [7] C. Mitchell, "Swing Trading," 8 March 2019. [Online]. Available: <https://www.investopedia.com/terms/s/swingtrading.asp>. [Accessed 29 March 2019].
- [8] C. Seabury, "Mastering Short-Term Trading," 6 February 2019. [Online]. Available: <https://www.investopedia.com/articles/trading/09/short-term-trading.asp>. [Accessed 29 March 2019].
- [9] C. Majaski, "Fundamental vs. Technical Analysis: What's the Difference?," 4 April 2019. [Online]. Available: <https://www.investopedia.com/ask/answers/difference-between-fundamental-and-technical-analysis/>. [Accessed 5 April 2019].
- [10] K. Zucchi, "4 Common active trading strategies," 4 Aug 2018. [Online] Available: <https://www.investopedia.com/articles/active-trading/11/four-types-of-active-traders.asp> [Accessed 20 April 2019]
- [11] Finviz, 2019. [Online]. Available: <https://finviz.com/>. [Accessed 22 April 2019].
- [12] M. Lopez de Prado, *Advances in Financial Machine Learning*, Wiley, 2018.
- [13] N. Sharma and C. Chauhan S., "Heikin-Ashi Transformation and Its Effect on Neural Network Learning for Stock Market Data," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 4, no. 12, pp. 21608-21611, 2016.
- [14] Travis E, Oliphant. *A guide to NumPy*, USA: Trelgol Publishing, (2006).
- [15] Stéfan van der Walt, S. Chris Colbert and Gaël Varoquaux. The NumPy Array: A Structure for Efficient Numerical Computation, *Computing in Science & Engineering*, 13, 22-30 (2011), [DOI:10.1109/MCSE.2011.37](https://doi.org/10.1109/MCSE.2011.37)
- [16] Wes McKinney. *Data Structures for Statistical Computing in Python*, Proceedings of the 9th Python in Science Conference, 51-56 (2010)
- [17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp.

2825-2830, 2011.

- [18] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90-95, 2007.
- [19] T. Kam Ho, "Random Decision Forests," AT&T Bell Laboratories, Murray Hill, 1995.
- [20] S. Nickolas, "SPY: SPDR S&P 500 Trust ETF," Investopedia, 14 October 2018. [Online]. Available: <https://www.investopedia.com/articles/investing/122215/spy-spdr-sp-500-trust-etf.asp>. [Accessed 22 April 2019].

## Appendix A: Program Code

The link to a website where all the code for this project is hosted. On the website is running instructions. <https://github.com/AJWurts/InvestmentTrading>

All running instructions assume Python 3.0+ is installed. Use either python3 or python to access the correct version of Python. The following libraries are required to run this program.

- Scikit-learn
- Numpy
- Scipy
- Pandas
- Matplotlib

### Code Structure Overview

- **Folder: Initial Results**
  - Contains results from an early iteration of the program
- **Folder: backtesting**
  - Contains code that runs the backtest
    - Folder: algorithms
      - Code for the individual algorithms tested
        - Algorithms.py
          - Contains the code for multiple algorithms including the Machine Learning Algorithm used for the System #1 and System of Systems.
    - **backtester.py**
      - Code that runs a walk forward analysis on a set of algorithms and stocks. Running instructions:
        1. Open a terminal and navigate to InvestmentTrading

- Required to run `training_data_creator.py` and `machinelearning/randomforest.py` on specified ticker in order to run this file.
  - 2. Type: `python3 backtesting/backtester.py [stock tickers here]`
    - Example: `python3 backtesting/backtester.py AMD`
    -
- **trade.py**
  - Contains the class structure for some of the objects used in the back tester.
- **Folder: data**
  - Contains the original yahoo data files and any meta files used throughout the program process.
- **Folder: final\_results**
  - Contains equity history visualizations for all tests performed. Final results images
- **Folder: machinelearning**
  - Contains all files required for creating training data in the format required for Random Forest.
    - Folder: `saved_classifiers`
      - Contains all weights for every stock saved in individual files
    - **bars.py**
      - Contains the code used to produce bars.
    - **fracdiff.py**
      - Calculates the weights for a given `d` and applies the transformation to a pandas dataframe.
    - **randomforest.py**
      - Trains a Random Forest algorithm on a given ticker
        - Running instructions
          1. Open a terminal and navigate to `InvestmentTrading`
          2. Type: `python3 machinelearning/randomforest.py [ticker]`
            - Run `training_data_creator.py [ticker]` first
            - In place of `[ticker]` use a stock ticker such as `AMD`
    - **sampling.py**
      - The code for cumulative summation
    - **triplebars.py**
      - Contains multi-threading code for applying triple barrier thresholds and labeling for all training samples within `training_data_creator.py`
- **Folder: mlsize**
  - Contains text files with the size of a given stock's random forest input size to be used in the backtester.
- **Folder: thresholds**
  - The thresholds used with cumulative summation for each ticker.
- **Folder: tradehistory**
  - Contains the trade history for each stock tested in the backtester.
- **training\_data\_creator.py**
  - Takes a yahoo data file and converts the data into a csv that can be used with `randomforest.py` for training a trading system.
- **README.md**
  - Instructions for how to run `main.py` and required libraries.



- **main.py**  
Main program files.