Time Frame Trading Algorithms

A. L. Gillies
M. R. Gadouleau
MEng Computer Science

***Abstract*—**
Algorithmic trading is characterised by an entirely hands off approach to stock market trading. All data manipulation, mathematical inference, machine learning and trade execution is done autonomously. With this approach, how much of an improvement can be gained over a standard interest rate provided by a high street bank, in the time frame given?

Using the average interst rate calculated from British banks in conjunction with implementation of both statistical and machine learning techniques show that algorithmic trading can improve the annual return on investment over a given time frame.

This paper will consider two possibilities for implementation of the system, a purely statistical method, relying on known practices and techniques, and a hybrid system incorporating both statistical reasoning and machine learning. The known statistical practices are mostly used by human traders to allow for data insight and are well vetted. (Murphy 1999) The machine learning techniques are widely used in other contexts, with limited academic papers being available for this area.

The Support Vector Machine that made up the Machine Learning component of the testing outstripped all statistical methods by a significant margin, which in turn returned significantly higher results than is available through any high street bank.

This project was a success but contains an unbound task, there is always going to be a better method. The methods used within this paper are the result of limited time and resources, any extension to this project would reduce these limitations. Possible extensions include a wider range of machine learning techniques and using data of a finer grain, over a wider time frame.

Keywords - Algorithmic, Machine Learning, Statistics, R, Trading, Stocks

I. Introduction

The stock market has been an early adopter of technology since its inception, as companies want to get an edge over their fellows and thus earn the most money. The first computer usage in the stock market was in the early 1970s with the New York Stock Exchange introducing the Designated Order Turnaround system (DOT). This allowed for bypassing of brokers and routed an order for specific securities to a specialist on the trading floor (Hasbrouck et al. 2013). The definition of 'broker' is an entity that buys and sells goods or assets for others, in this case 'securities' which are defined as a financial instrument that holds monitary value. Since this point the use of machines to allow for increase throughput and speed has been pandemic. It was inevitable that computers would be used to aid in the decision making process of what to buy or sell and when. This was shown to be very effective and got significant traction in the financial market in 2001 with the showcase of IBMs MGD and Hewlett-Packard's ZIP (Tesauro & Das 2001). These two algorithmic strategies were shown to consistently outperform their human counterparts. They were both based on papers from 1996 showing that academic conception of algorithmic applications in financial markets has been present for several decades (Gjerstad & Dickhaut 1998), (Cliff & Bruten 1998). Whilst in the current day over one billion shares are traded every day, this would not be possible without computerised assistance.

The aim of any algorithmic trading system is to 'beat the market': to buy low and sell high to gain the most capital over any given time frame, be that a day, a year, or a decade. This paper will look into the challenge faced by these algorithms and if, using available tools, it is possible to outstrip any high street banks offering rates. This will be achieved through simulation of a stock market, using real stock data, and the simulation of buying and selling these stocks using an algorithm. The current base bank interest rate is set at 0.5% by the Bank of England (Bank of England 2018), with the average individual savings account or ISA over a fixed term of 1 year having a value of 1.85% (Murray 2018)

This project will be based around the deliverables in table I.

II. Related Work

As has been mentioned previously, this field is made up of companies or researchers that are unwilling to share their recent breakthroughs. This results in a copious amount of research being available for techniques that are seen as antiquated, with very little being published on state-of-the-art. Any modern technique being employed is done so by an institution which uses it to profit from the stock market, this 'edge' will be guarded until it is no longer useful.

The available papers that are based in this field are focused on statistical implementation. With multiple books being published on the topic. Some based on single strategies but most being compendiums of multiple strategies, for example (Murphy 1999). There are a few papers that include more complex models such as the Black-Scholes Model (Saad et al. 2015) which are seen as pivotal to the world of economics and finance, but have little benefit to computational finance. The same can be said for papers concerning the application and implications of Brownian motion, or the effect of the Efficient Market Hypothesis (Meng et al. 2016). Although applicable to the modern world of finance as a whole, they are almost unusable within the context of this paper. This is due to the highly fluid and abstract nature of these theorems that are used

Unique ID	Deliverable	Description
DL1	Simulate the financial market	Have data for at least 10 companies for at least a year, with data for each minute where data is available.
DL2	Allow buying and selling of stocks	Have a functional buying and selling mechanism, with the data collected for each transaction processed.
DL3	Implement statistical methods	Implement as many statistical methods as are beneficial to allow for the insight into the data for each stock.
DL4	Implement a purely statistic strategy	Using just the statistical methods implemented in DL3, create a strategy that will buy and sell stocks to maximise profit made over the time frame given.
DL5	Create a hybrid strategy	Implement a machine learning trading strategy that uses the stock data as well as any statistical methods that are helpful to maximise profit made over the time frame given.
DL6	Implement tracking systems	Implement graphical and table outputs for the results of the computer logic and trading performance.
DL7	Create a testing criteria	Create a method with which to test the strategy so as to avoid over fitting.

Table I
DELIVERABLES

to try to explain the world of economics more as a whole instead of giving finer details on more specific topics within this area.

This results in a juxtaposition of copious amounts of research into the statistical methods that are available and their reasoning, with a significant paradigm shift into the research that is based around machine learning in this context.

The most useful piece of research for the statistical methods is (Murphy 1999). This book is a compendium of trading strategies that were used in the time period of 1950 to 1980, with ample detail on market analysis and any human readable charts that can be used within a trading scenario, whilst also giving details of the philosophy behind the trading. It also compares the use of computerised aids within a financial market to a purely human system. Electronic and Algorithmic Trading Technology (Kim 2007) contains multiple strategies and specific statistical methods that are useful within this context. It also shows a significant amount of detail concerning the surrounding systems, for example matching software that is used within a stock exchange that is not relevant. Written more from the view of a stock exchange, significant time and detail is given to the automation of‘order flow’ more commonly known as the matching of buying and selling parties within a stock exchange. Trading Commodities and Financial Futures (Kleinman 2005) is written very much for the purpose of teaching human traders. With sections on the psychology of trading and the history of trading. The techniques used can also be applied within computerised finance and are discussed in great depth. ‘Dark markets asset pricing and information transmission in over-the-counter markets’ (Duffie 2012) contains very little in the way of strategies but is useful to give a wider picture of the context and legal aspects of trading. The data aspect touched upon within this book is very useful in inspiring the data formatting and data manipulation that is to be used throughout this paper.

The limiting factor for papers within this domain is that they are not within the field of machine learning in addition to being within the domain of computer finance. The inverse can be said for papers within the field of machine learning. Those that fall into both categories are limited in other ways. Some being too general to be of significant use, others being more in line with an academic look at the history of the domain. ‘Evaluating machine learning classification for financial trading: An empirical approach’ (Gerlein et al. 2016) is one of the papers that was found to have test results but was limited in that no strategies were provided and very little detail is given about the very basic models that are employed within the paper. The point is made that complex models have downsides and that Occam’s Razor may play a role. ‘Predicting Stock Price Direction using Support Vector Machines’ (Madge 2015) is a very relevant paper that provides insight into the predictive powers of an SVM when used in conjunction with momentum and volatility measures. (Madge 2015) is limited in that testing was done with very general data and shows very little promise with respect to short term predictions. ‘Visual knowledge discovery and machine learning for investment strategy’ (Wilinski & Kovalerchuk 2017) is very useful in providing detailed information regarding machine learning data manipulation before input but does not specify any details of machine learning, just that it will be used in future. ‘Buy Low, Sell High: A High Frequency Trading Perspective’ (Cartea et al. 2014) is based within the world of High Frequency trading and whilst this would be useful to an extension of this paper, we do not take into account market reactions to actions taken within this paper so it is beyond scope.

III. SOLUTION

The solution is outlined in several sections that split the task into the stages that were followed in order to test the viability of all claims within this paper.

A. Simulation

This section will outline in detail all components of the simulation and their interconnectivity. The purpose of this section is to give details on all work that was done outside of the implementation of the algorithms tested. Simply implementing the mathematics would not allow for testing of these algorithms.

1) Methodology: The logic behind the simulation was to allow trading at a fine grain level, which needed to be permitted by the data used to perform the algorithms. The data that was available grows from 'tick data' or data for every trade executed, to 'minute data', giving details of all trades within that minute, and growing to 'day data' and beyond, which are all the same as minute but with a widening ranges. A drawback was that 'tick data' was found to be too costly for the project so 'minute data' was used. Minute data provides an open, high, low, and close price for a stock in the given minute, which gave plenty of data points per day of trading. A simplification was created at this point, trading was only done on the price that opened the minute. This was done to reduce the complexity of calculating any inter-minute values. The running of the simulation was based on the progression through each date of available trading and for each day iterate through every minute of the trading day, starting at 09:30 until 16:00. Each trading day has 390 points at which trading is possible, if the data permits it. This is available for 382 days in the 18 months of data that was used.

The programming language that was chosen for this task was R (Team 2013). This was chosen over other languages due to its extensive data manipulation capabilities as well as the contained mathematics functions such as standard deviation and differentiation. The data manipulation allows for large sets of incomplete data to be sub-sampled and manipulated into a usable form. The mathematical functions will form a basis for the statistical methods that are to be implemented.

2) Data: 45 different stocks were used in this paper, taking from each the maximum number of data points available between 01/03/2016 and 01/09/2017. This gave an 18 month window wherein the first 6 months did not allow for trading to occur, only for data collection and training of models (which will be discussed in depth further in the paper) and the final 12 months were allocated for testing each algorithmic approach. There were some limitations with the data that was used. Due to the volatility of the stock market and the fact that some days trading was halted for a myriad of reasons, the target of 390 data points per day, for 18 months, was not achieved. Trading is not available on weekends, on bank or national holidays, or if trading is halted for some other reason. This meant that the solution had to take into account holes in the data. These gaps are also not regular. Any given day can be cut short or a whole day could be taken without a pattern. Gaps also exist in minutes that no trades were executed, which removes any values from the data and removes the possibility of trading within these minutes.

The data was taken from the NASDAQ, the second largest stock market in the world. The number of stocks used was arbitrarily big enough to provide enough data to test any given algorithm, some of which were found to prefer an abundance of data whilst others were much more short sighted and volatile. The most frequently traded stocks were chosen to allow for the most complete data.

The data was first downloaded in the form shown in Table II, it was then converted using an R script into the form shown in Table III. Thus reducing computational overhead from running the simulation as all data manipulation had been done prior to the data being used. In the final form no other alterations had to be made to the data other than set data types.

Some examples of the formatted data in use can be seen in Figures 1,2,3,4,5,6,7, and 8.

3) Set-up: The simulation was set up in such a way so as to minimise the impact caused by gaps in the data. This meant using the data to provide the iteration behaviour. The data input in the form of Table III was cast for each row using a typecast function to allow for correct manipulation of numerical values. The dates were converted to numerical values for correct comparison, as were the times. The stock

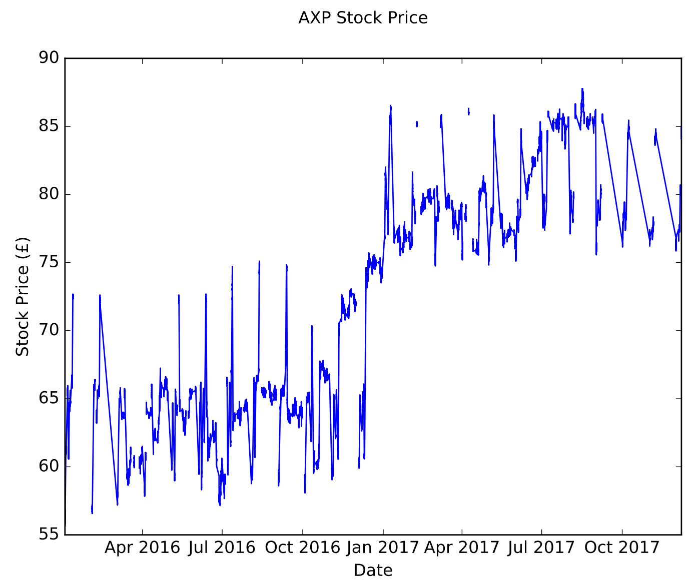

Figure 1. A graph showing the AXP stock price over the time frame used.

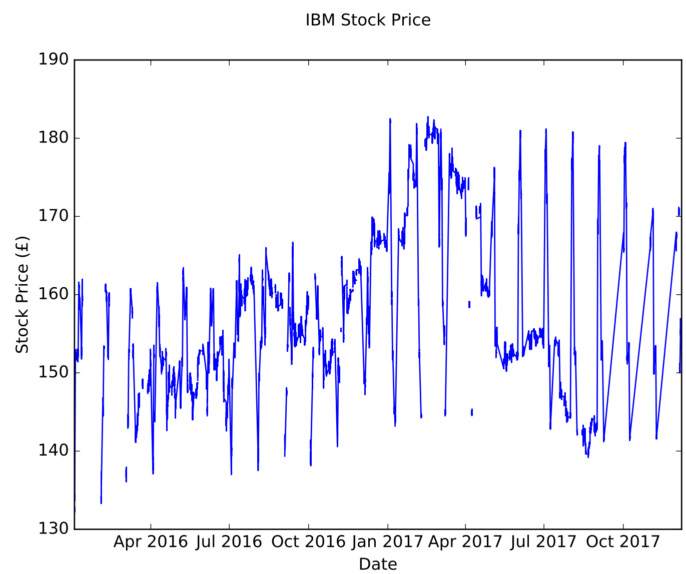

Figure 2. A graph showing the IBM stock price over the time frame used.

Ticker	Date	Time	Open	High	Low	Close
AA	01/03/16	09:30	9.1100000	9.1300000	9.1100000	9.1100000
AA	01/03/16	09:31	9.1000000	9.1200000	9.0800000	9.0900000
...	...	...	...	...	...	...

Table II
DATA OF A SINGLE STOCK, IN THIS CASE AA.

Date	Time	AA	AAPL	ADBE	AIG	...
01/03/16	09:30	9.11	97.66	86.15	50.59	...
01/03/16	09:31	9.10	97.58	86.22	50.67	...
...	...	...	...	...	...	...

Table III
FORMATTED DATA OF ALL STOCKS USED.

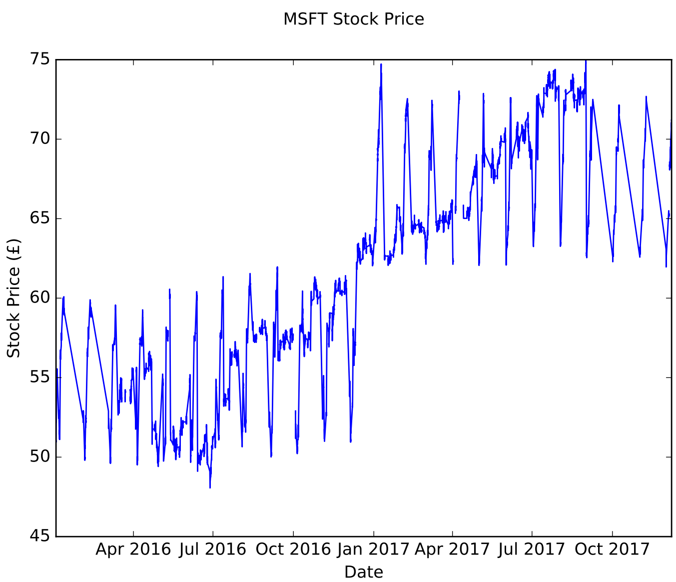

Figure 3. A graph showing the MSFT stock price over the time frame used.

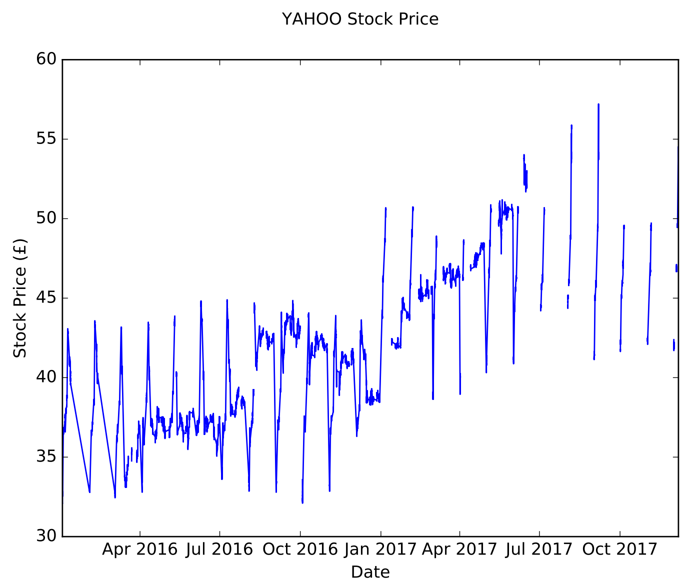

Figure 4. A graph showing the YAHOO stock price over the time frame used.

values were also numerical.

The simulation then uses all unique dates within the Dates column to allow for iteration through the table at the highest level, through each individual day. Once the data had been sub-sampled for that day, then a list of unique minutes was created. Once the data had been further sub-sampled to the current minute, each of the stock columns were iterated through and tested for NA, or Not Available, values. If an NA value is present for that stock, then it is passed over. If the value is available however then two functions are used, 'shouldBuy' and 'shouldSell'. The first used is 'shouldBuy'; this will take input of current date, current time, and current stock, to allow for a decision to be made on whether to buy an amount of that stock or not. These functions are based on manipulations of three other tables, 'Active', 'Sold', and 'Ledger'. These are shown in tables IV, V, and VI, respectively.

Each also comes with its own population function, 'Buy', 'Sell', 'Update'. The first two are called if any decision is made. If a stock's value is such that the 'shouldBuy' function returns true then the 'Buy' function is called, taking input of date, time, stock, and amount. The amount is dictated by the 'amountShouldBuy' function that is an extension of the 'shouldBuy' function that returns the amount of the current stock that should be bought. This is then added as another row to the 'Active' table, as this is now an active stock that can be sold. The amount of any given stock that should be purchased is calculated through another function 'getAmount'. This will test all calculations done in 'shouldBuy' and calculate a confidence value that will then, using the amount of liquid capital available, dictate the amount of the stock that will be bought. The 'shouldSell' function is then queried every time an updated value is available for any given row in the 'Active' table. Using the new value a decision will be made if the stock should be sold, if this is found to be the case then the 'Sell' function is called. The function will remove the row of the 'Active' table that corresponds to this block of stock to be

Unique ID	Date Bought	Time Bought	Stock	Number of Shares	Cost per Share
...	...	...	...	...	...

Table IV
ACTIVE

Unique ID	Date Bought	Time Bought	Stock	No. of Shares	Cost per Share	Amount Bought	Date Sold	Time Sold	Price per Share	Amount Sold
...	...	...	...	...	...	...	...	...	...	...

Table V
SOLD

Date	Total Value	Stock Value	Capital Value
...	...	...	...

Table VI
LEDGER

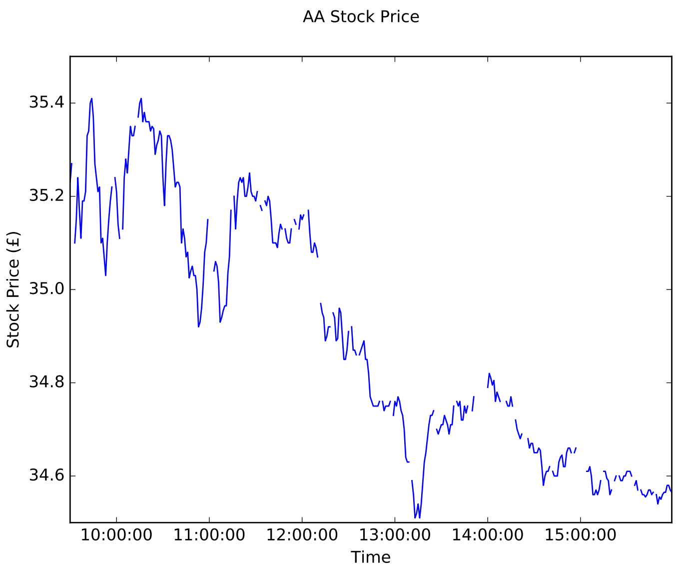

Figure 5. A graph showing the AA stock price over a single day.

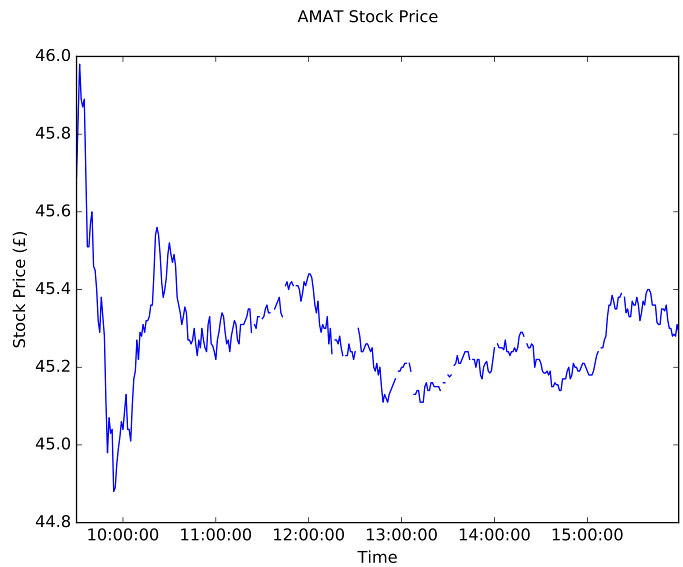

Figure 6. A graph showing the AMAT stock price over a single day.

sold and add it to the 'Sold' table along with all the details of the transaction.

The final table is 'Ledger' which uses the function 'Update'. this is called on a regular basis to allow for tracking of the algorithm throughout the simulation. All overall details of capital are stored in this table. The 'Date' column is a concatenation of both 'Time' and 'Date' in any other table to allow for minute by minute updates of the simulation or if this is not required then daily updates are possible.

4) Functions: Shown in Table VII are some of the more regularly utilised functions in the simulation and are the key underpinnings of the statistical and machine learning techniques that are used throughout the simulation.

B. Statistical Methods

Once a working simulation had been established and all functions had been shown to be working, statistical methods had to be implemented. These were separated into two groups, the first are technical overlays, methods that provided insight into the data by providing numbers that are on the same scale as the price itself. Whilst the other group, technical indicators give insight into the data through numbers that are in no way related to the stock price, and thus allow much more insightful comparisons to be made inter-stock as opposed to intra-stock.

Function	Arguments	Details
Buy()	Date, Time, Stock, Amount	Adds a row to the 'Active' table.
Sell()	Row Data, Date, Time, Stock, Stock Price	Adds a row to the 'Sold' table and removes the corresponding row of the 'Active' table.
Update()	Date, Total Capital	Adds a row to the 'Ledger' table for re-tracking of the simulation
shouldBuy()	Date, Time, Stock	Calculates using any method set if the current stock should be bought at the current price.
shouldSell()	Unique ID, Date, Time	Calculates if the given block of stocks should be sold at the current price.
amountShouldBuy()	Date, Time, Stock	This is a function that is used to find a confidence value in the current stock price and thus returns the amount of the given stock that should be bought also taking into account the amount of liquid capital that is available at the current time.
getMax()	List	Get the maximum value of the given list.
getMin()	List	Get the minimum value of the given list.
getAverage()	List	Get the mean value of the given list.
getSD()	List	Get the standard deviation of the given list.
getXDate()	Date, X	Get the date X days ago
getXClose()	Date, Stock, X	Get the close X days ago.
getXOpen()	Date, Stock, X	Get the open X days ago.
getXDataPoints()	Date, Time, Stock, X	Get the last X data points for the given stock
getXDayDataPoints()	Date, Stock, X	Get all data for the day X days ago.
getXSincePrice()	Date, Time, Stock, Price	Get the number of data points between the current data point and the last data point that had the given price.
getTotalGainLoss()	Date, Time, Stock, X	Over the last X data points get the total gain and loss.

Table VII
BASIC FUNCTIONS

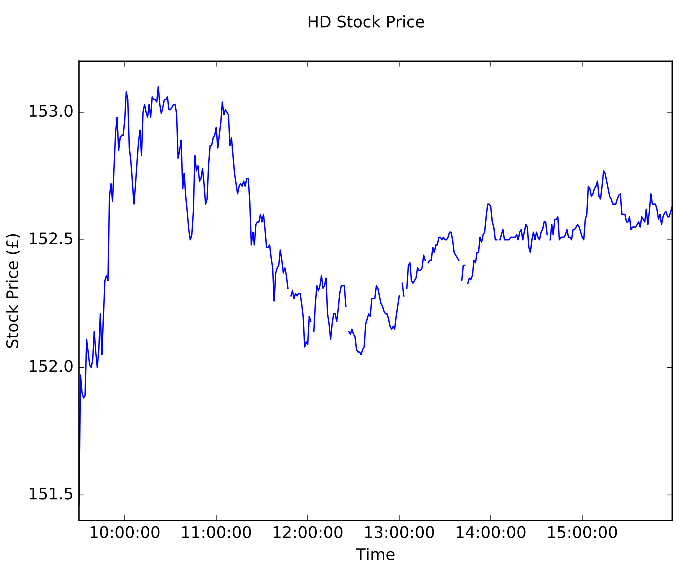

Figure 7. A graph showing the HD stock price over a single day.

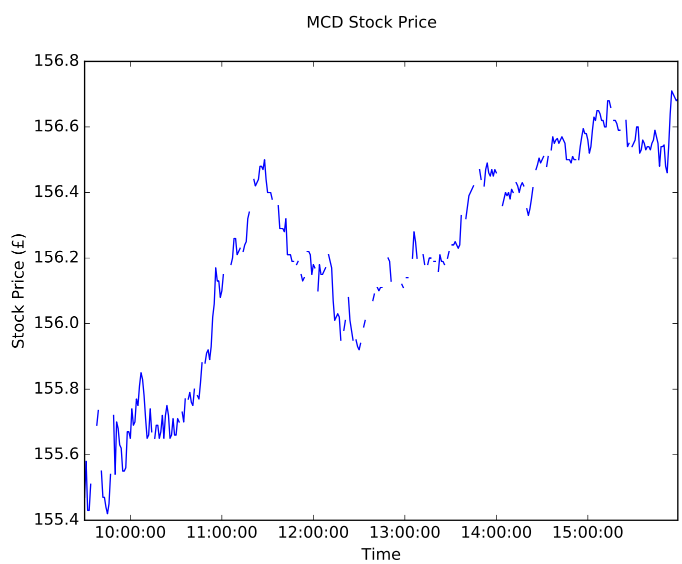

Figure 8. A graph showing the MCD stock price over a single day.

That is comparing two stocks is easier if technical indicators are used as opposed to using technical overlays which are very dependant on the stock price and do not translate as well to comparisons between two unique stocks.

An example of a technical overlay is the Bollinger Band, the calculation of which will be shown in the following section. This technical overlay, shown in figure 9, clearly has an output range that is dictated by the stock price. An example of a technical indicator is the MACD, this will be explained further in the paper. This can be seen in figure 10 and shows that the stock price has no bearing on the final value of the indicator.

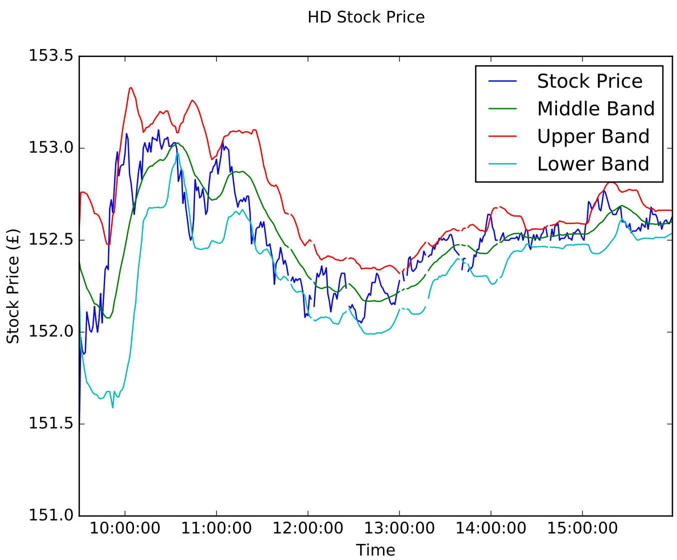

Figure 9. A graph showing the change in the value of Bollinger Bands as the value of the HD stock changes.

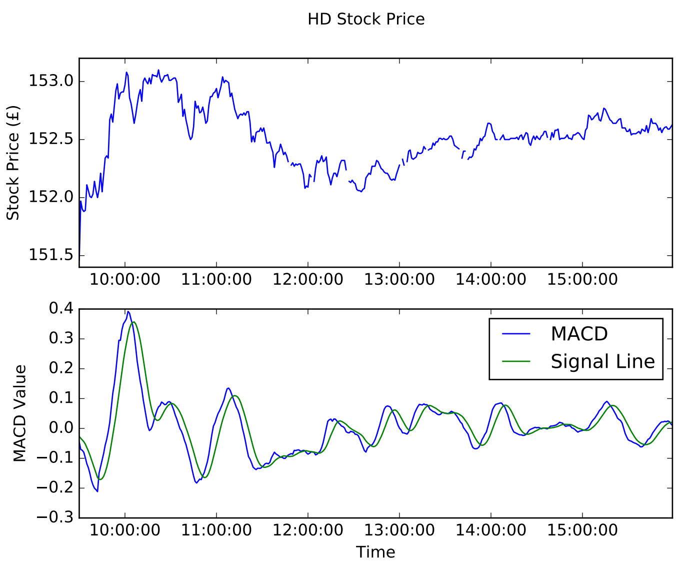

Figure 10. A graph showing the change in the value of MACD as the value of the HD stock changes.

1) Technical Overlays: Defined as a statistical method that is dependant on the price.

Bollinger Bands - (Bollinger 1992) - These are volatility bands placed above and below the current stock price and are based on the standard deviation. These are designed to give an indication of how the volatility of a given stock changes over time. For any given stock over a time frame X, the three bands are calculated as such:

Upper band $=$ Simple Moving Average(X) $+$ (Standard Deviation$(X) * 2$)
Middle band $=$ Simple Moving Average(X)
Lower band $=$ Simple Moving Average(X) $-$ (Standard Deviation$(X) * 2$)
The standard time period is 20 days.

Chandelier Exit - (Elder 2002) - Developed by Charles Le Beau, this is designed to help stay within a trend and not to exit early. In the case of an uptrend the Chandelier Exit will typically be below the stock price and the inverse is true in the case of a downtrend. To calculate it over a time period X:

Long = High(X) - (Average True Range(X) * 3)
Short = Low(X) + (Average True Range(X) * 3)
The standard time period is 22 days.

Ichimoku Cloud - (Murphy 1999) - A multifaceted indicator developed by Goichi Hosoda, a Japanese journalist. This is an average based trend identifying indicator based on the standard candlestick charts. This indicator is used as a basis in a number of other theories including Target Price Theory. There are five plots within Ichimoku Cloud.

Using time period X.
Tenkan-sen or Conversion line = (High(X) + Low(X)) / 2 - Default X is 9
Kijun-sen or Base line = (High(X) + Low(X)) / 2 - Default X is 26
Senkou Span A or Leading span A = (Conversion Line + Base Line) / 2
Senkou Span B or Leading span B = (High(X) + Low(X)) / 2 - Default X is 52
Chikou Span or Lagging span = CloseXPeriodsAgo(X) - Default X is 26

Kaufman's Adaptive Moving Average (KAMA) - (Kaufman 1998) - This indicator is designed to remove market noise during volatile periods. It takes three parameters, X, Y, and Z. X is the number of periods that is used by the first step of the calculation, known as the efficiency ratio. This will be shown later. The second is the number of periods for the first and fastest exponential moving average or EMA. Third is the number of periods for the second and slowest EMA. The defaults for these values are (10, 2, 30).

Efficiency Ratio = Change/Volatility
Change = Absolute Value(Close(Now) - CloseXPeriodsAgo(X)) - Default X is 10
Volatility = Sum(Absolute Value(Close - CloseXPeridsAgo(X))) - Default X is 1, this sum is done 10 times, for the last 10 changes in price.

The next stage of KAMA is the calculation of a smoothing constant.

Smoothing Constant = ((Efficiency Ratio * (fastest SC - slowest SC)) + slowest SC)^2

Final stage is the use of the previous KAMA value to calculate the next value.

New KAMA = Previous KAMA + (Smoothing Constant * (Current Price - Previous KAMA))

Keltner Channels - (Keltner 1960) - Very similar to Bollinger Bands but instead of using standard deviation average true range is used. Created by Chester Keltner, this indicator is made up of three lines in a similar way to Bollinger Bands.

Upper Channel Line = Exponential Moving Average(X) + (2 * Average True Range(Y))
Middle Line = Exponential Moving Average(X)
Lower Channel Line = Exponential Moving Average(X) - (2 * Average True Range(Y))
Default X = 20, Y = 10

Moving Averages - (Murphy 1999) - These can come in multiple forms with multiple names. The catch all term for this type of smoothing average is a moving average. The most simple is known as a Simple Moving Average (SMA). This takes the average of the last X data points. There is no weighting or extra steps. A more complex version is the Exponential Moving Average (EMA), this uses weighting to give the more recent values more significance in the calculation. The initial value of EMA is the same as the SMA for the same period as EMA requires an initial value.

EMA_{Today} = (Current Stock Price * K) + ($EMA_{Yesterday}$ * (1 - K))
K = 2 / (N + 1)
N = Number of periods over which the EMA is applied

Moving Average Envelopes - (Murphy 1999) - Based on a Moving Average, this is a percentage based envelope that provide parallel bands above and below the Moving Average. Gives an indication of trends in the data as well as an indicator for stocks that are overbought and oversold when the trend is flat.

Upper Envelope = MovingAverage(X) + (MovingAverage(X) * Y)
Lower Envelope = MovingAverage(X) - (MovingAverage(X) * Y)
Typical values X = 20 and Y = 0.025

Parabolic SAR - (Wilder 1978) - SAR stands for ‘stop and reverse’, this was called a Parabolic Time/Price System. This indicator follows the stock price as the trend is formed, and will then ‘stop and reverse’ when the trend ends, to follow the new trend. This is one of the more complex indicators.

In the case of a rising SAR:
EP or extreme point is a variable that is equal to the highest value of the current uptrend.
AF or acceleration factor is a variable that starts at 0.02 and is increment by 0.02 each time the EP is changed, meaning that it is incremented each time a new high is reached. The maximum value of AF is 0.2.

$SAR = SAR_{Yesterday} + AF_{Yesterday}(EP_{Yesterday} - SAR_{Yesterday})$

In the case of a falling SAR:
This uses the same variable names but inverse behaviour, the EP is equal to the lowest point in the current downtrend. AF is the same but is incremented when EP reaches a new low.

$SAR = SAR_{Yesterday} - AF_{Yesterday}(EP_{Yesterday} - SAR_{Yesterday})$

These are used to indicate a trend and once a price falls to the other side of the value calculated in the current trend, that trend is over and SAR will flip to the opposite trend.

Pivot Points - (Murphy 1999) - An overlay used to indicate directional movement and then shows these in potential support and resistance levels. These are predictive indicators and they exist in multiple forms, the most well known are the standard, Denmark, and Fibonacci versions. These are calculated using the previous days high, low, and close values and are then not recalculated throughout the trading day. A calculation has multiple components, the pivot point, multiple supports, and multiple resistances.

Standard Pivot Points.
Pivot Point = (High + Low + Close)/3
Support One = (PP * 2) - High
Support Two = PP - (High - Low)
Resistance One = (PP * 2) - Low
Resistance Two = PP + (High - Low)

Denmark Pivot Points. These are the most complex calculations as they have conditional statements in them and do not use the same calculation methods as the other two.
Pivot Point = X / 4
Support One = (X / 2) - High
Resistance One = (X / 2) - Low

Where X is calculated as:
If Close $<$ Open: X = High + (2 * Low) + Close
If Close $>$ Open: X = (2 * High) + Low + Close
If Close = Open: X = High + Low + (2 * Close)

Fibonacci Pivot Points. These are similar to the standard pivot points but use different spacing techniques, namely the Fibonacci sequence.
Pivot Point = (High + Low + Close)/3
Support One = PP - (0.382 * (High - Low))

Support Two = PP - (0.618 * (High - Low))
Support Three = PP - (1 * (High - Low))
Resistance One = PP + (0.382 * (High - Low))
Resistance Two = PP + (0.618 * (High - Low))
Resistance Three = PP + (1 * (High - Low))

Price Channels - (Murphy 1999) - Using three calculations to show an upper, lower, and middle bound, used to indicate the start of an upwards or downward trend.

Upper = High(X)
Center = (High(X) + Low(X)) / 2
Lower = Low(X)
The default X value is 20.

2) Technical Indicators: Defined as a statistical method that is not dependent on the price.

Aroon - (Chande & Kroll 1994) - Aroon is indicative of the strength of the current trend. It was designed to be similar but uniquely different to standard momentum oscillators, which focus on price relative to time, Aroon focuses on time relative to price. It has two components, Up and Down, both are expressed as percentages. Up will maximise on an upward trend and Down will maximise on a downward trend.

Aroon Up = ((X - DaysSinceHigh(X))/X) * 100
Aroon Down = ((X - DaysSinceLow(X))/X) * 100
Default value of X = 25.

Aroon Oscillator - A join of the two values of Aroon into a single value.
Aroon Oscillator = Aroon Up - Aroon Down

Average True Range (ATR) - (Wilder 1978) - Developed as a measure for volatility, ATR has been used in a wide variety of applications outside of the financial world. The initial idea was based around a concept called True Range, calculated as such:

The greatest of:
High(X) - Low(X)
ABS(High(X) - PreviousClose)
ABS(Low(X) - PreviousClose)

This was then used in conjunction with the previous true range to calculate the new ATR.

New ATR = ((Prev ATR * (X-1)) + TR) / X - Default X is 14

BandWidth - (Murphy 1999) - One of the two indicators derived from Bollinger Bands, the other being %B. This is a single value that takes all Bollinger Bands as components.
Bandwidth = ((Upper Band - Lower Band) / Middle Band) * 100

%B Indicator - (Murphy 1999) - Another Bollinger Band derivative, %B indicator gives an indication as to the relationship of the current price and the Upper and Lower Bollinger Bands.
%B = (Current Price - Lower Band) / (Upper Band - Lower Band)

Commodity Channel Index (CCI) - (Lambert 1980) - Used to show a comparison between the current price and the average price over a given timespan. Uses multiple other calculations as component parts, including Simple Moving Average, Typical Price, and Mean Deviation.

Typical Price = (High + Low + Close) / 3

Mean Deviation = SUM(ABS(Period Value - Period Average)) / X
Default X = 20. Find the sum of the deviation from the average value of the last 20 periods within each period.

CGI = (Typical Price - X period SMA of Typical Price) / (0.05 * Mean Deviation)// Default X = 20.

Coppock Curve - Developed by Edwin Coppock. Using a Weighted Moving Average as well as a period based Rate Of Change, this simple indicator as been used by many as a sell indicator as the value crosses the positive-negative boundary. It is calculated as the WMA of the ROC plus the ROC over a different period.

Coppock Curve = WeightedMovingAverage(X, RateOfChange(Y)) + RateOfChange(Z)
Default X = 10, Y = 14, Z = 11.

DecisionPoint Price Momentum Oscillator (PMO) - An oscillator that is calculated as a smoothed version of the rate of change using the exponential moving average as part of the smoothing process.

Smoothing Multiplier = (2 / Time period)
Custom Smoothing Function = Close - Smoothing Function(previous day) * Smoothing Multiplier + Smoothing Function(previous day)
PMO Line = 20-period Custom Smoothing of (10 * 35-period Custom Smoothing of (((Today's Price/Yesterday's Price) * 100) - 100))
PMO Signal Line = 10-period EMA of the PMO Line

Detrended Price Oscillator (DPO) - (Murphy 1999) - Used to identify cycle details. High, low, and cycle length can be calculated.

DPO = Price X/2 + 1 periods ago less the X-period simple moving average.

Mass Index - (Murphy 1999) - Volatility indicator used to show a trend reversal before it occurs. Originally developed by Donald Dorsey.

Single EMA = 9-period exponential moving average (EMA) of the high-low differential
Double EMA = 9-period EMA of the 9-period EMA of the high-low differential
EMA Ratio = Single EMA divided by Double EMA
Mass Index = 25-period sum of the EMA Ratio

MACD (Moving Average Convergence/Divergence Oscillator) - (Appel 2005) - Is said to be one of the most effective momentum indicators as well as being very simplistic to perform.

MACD Line: (12-day EMA - 26-day EMA)
Signal Line: 9-day EMA of MACD Line

MACD Histogram - (Murphy 1999) - Developed by Thomas Aspray as a development on the MACD, to pre-emptively detect crossovers between the two lines in MACD.

MACD Histogram: MACD - Signal Line

Percentage Price Oscillator (PPO) - (Murphy 1999) - A momentum oscillator that is related to MACD. Calculated as a percentage showing the relationship between two moving averages.

Percentage Price Oscillator (PPO): (12-day EMA - 26-day EMA)/26-day EMA * 100
Signal Line: 9-day EMA of PPO
PPO Histogram: PPO - Signal Line

Pring's Know Sure Thing (KST) - (Pring 2002) - Using the smoothed rate of change over four different length periods, this momentum oscillator gives a more well based indication of movement than a typical momentum oscillator.

RCMA1 = 10-Period SMA of 10-Period Rate-of-Change
RCMA2 = 10-Period SMA of 15-Period Rate-of-Change
RCMA3 = 10-Period SMA of 20-Period Rate-of-Change
RCMA4 = 15-Period SMA of 30-Period Rate-of-Change
KST = (RCMA1 * 1) + (RCMA2 * 2) + (RCMA3 * 3) + (RCMA4 * 4)
Signal Line = 9-period SMA of KST

Pring's Special K - (Pring 2002) - This momentum oscillator is a concatenation of three different velocities, short, medium, and long term, to provide more stable prediction of movement. This is a sum of the three different velocities which are each made up of a weighted sum of different components.

Special K = 10 Period Simple Moving Average of ROC(10) * 1
+ 10 Period Simple Moving Average of ROC(15) * 2
+ 10 Period Simple Moving Average of ROC(20) * 3
+ 15 Period Simple Moving Average of ROC(30) * 4
+ 50 Period Simple Moving Average of ROC(40) * 1
+ 65 Period Simple Moving Average of ROC(65) * 2
+ 75 Period Simple Moving Average of ROC(75) * 3
+100 Period Simple Moving Average of ROC(100)* 4
+130 Period Simple Moving Average of ROC(195)* 1
+130 Period Simple Moving Average of ROC(265)* 2
+130 Period Simple Moving Average of ROC(390)* 3
+195 Period Simple Moving Average of ROC(530)* 4

Rate of Change (ROC) and Momentum - (Murphy 1999) - A pure momentum oscillator used in many other indicators.

ROC = [(Close - Close n periods ago) / (Close n periods ago)] * 100

Relative Strength Index (RSI) - (Wilder 1978) - A range of zero to one hundred, RSI, a momentum oscillator, gives an indication of the speed and change of price movements.

RSI = 100 - (100 / 1 + RS)
RS = Average Gain / Average Loss

StockCharts Technical Rank (SCTR) - A collection of indicators that can be condensed into a single value for comprehensive comparison between multiple stocks, allowing for stock ranking to occur. Also very useful as individual components for data insight. The first step of calculation is the six components of the SCTR. These are calculated and given a set weighting and then once calculated, summed together as a multiple of their weighting.

Long Term Indicators
Percentage of current price above or below the EMA(200 days). Weighting of 30%.
RateOfChange(125 days). Weighting of 30%.

Medium Term Indicators
Percentage of current price above or below the EMA(50 days). Weighting of 15%.
RateOfChange(20 days). Weighting of 15%.

Short Term Indicators
Slope of Percentage Price Oscillator Histogram (PPO) over the last 3 days. Weighting of 5%.
Relative Strength Index (RSI) for the last 14 days. Weighting of 5%.

Slope - A very simple idea. The main concept is to calculate the line of best fit over a given time frame to show to trend over that time frame. This is a very simple tool used to give general trends.

Stochastic Oscillator - (Murphy 1999) - Developed by George C. Lane. A momentum indicator that uses the close data along with the range between the high and low values over a given period to show the current momentum. Lane states; this “follows the speed or the momentum of price. As a rule, the momentum changes direction before price.”

%K = (Close - Lowest Low over X)/(Highest High over X - Lowest Low over X) * 100.
%D = Simple Moving Average of %K over Y Periods.
Default value of X is 14. Default value of Y is 3.

StochRSI - (Chande & Kroll 1994) - Using Relative Strength Index or RSI, StochRSI is a measure of RSI relative to the max range of RSI over a set period. This indicator has a range of 0 to 1 with 0 indicating the lowest point over the period with 1 indicating the highest point over the period.

StochRSI = (RSI - Lowest Low RSI over X) / (Highest High RSI over X - Lowest Low RSI over X)
Default X value is 14.

TRIX - (Hutson 1983) - A triple smoothed Exponential Moving Average is used to calculate the percentage change over the last period.

Single Smoothed EMA = EMA of Close over X periods.
Double Smoothed EMA = EMA of Single Smoothed EMA over X periods.
Triple Smoothed EMA = EMA of Double Smoothed EMA over X periods.
TRIX = Single period percentage change in Triple Smoothed EMA.

True Strength Index - (Blau 1995) - Using two double smoothed price changes this is a momentum oscillator with the benefit of being relatively resistant to noise. Made up of two double smoothed price changes and the TSI calculation, this is a relatively simple indicator.

Double Smoother Price Change.
PC = Current Price minus Prior Price
Single Smoothed PC = EMA of PC over X periods.
Double Smoothed PC = EMA over Y periods of Single Smoothed PC.
Default X value is 25. Default Y value is 13.

Double Smoothed Absolute Price Change.
Absolute PC = ABS(Current Price minus Prior Price)
Single Smoothed PC = EMA of APC over X periods.
Double Smoothed PC = EMA over Y periods of Single Smoothed APC.
Default X value is 25. Default Y value is 13.

True Strength Value.
TSI = 100 * (Double Smoothed Price Change / Double Smoothed Absolute Price Change)

Ulcer Index - (Martin & Mccann 1992) - This volatility indicator was originally designed to measure downside risk in mutual funds, although it has now been re-purposed.

Percent-Drawdown = ((Close - Max Close over X periods)/Max Close over X periods) * 100
Squared Average = ((Sum of Percent-Drawdown over X periods)^2)/X
Ulcer Index = $\sqrt{SquaredAverage}$
Default X value is 14.

Ultimate Oscillator - Developed by Larry Williams. This is a triple time frame based momentum oscillator. The use of multiple time frames is limit the effect that noise can have on a typical momentum oscillator. There are several steps to the Ultimate Oscillator, all of which rely on Buying Pressure, BP, and True Range, TR.

BP = Close - Min(Low, Prior Close)
TR = Max(High, Prior Close) - Min(Low, Prior Close)

Average X = (BP Sum over X periods) / (TR Sum over X periods)
Average Y = (BP Sum over Y periods) / (TR Sum over Y periods)
Average Z = (BP Sum over Z periods) / (TR Sum over Z periods)

Ultimate Oscillator = 100 * ((4 * Average X)+(2 * Average Y)+Average Z)/(4+2+1)
Default values are X = 7, Y = 14, Z = 28.

Vortex Indicator - Developed by Etienne Botes and Douglas Siepman, based on the work of Welles Wilder and Viktor Schauberger. Using the relationship between two oscillators, one capturing positive trend movement and the other capturing negative, the vortex indicator is adept as showing the bias of the data.

Positive and negative Vortex Measurement.
+VM = ABS(Current High - Prior Low)
-VM = ABS(Prior Low - Current High)

Positive and negative Vortex Measurement over X.
+VMX = Sum of +VM over X periods
-VMX = Sum of -VM over X periods

True range over X.
TRX = Sum of True Range over X periods

Positive and negative Vortex Indicator over X.
+VIX = +VMX/TRX
-VIX = -VMX/TRX

Default X value is 14.

The crossovers of these two values is then used to identify the start and end of a trend and the direction of said trend.

Williams %R - (Murphy 1999) - Developed by Larry Williams and based on the Stochastic Oscillator developed by George C. Lane. This is the inverse of the Fast SO as the FSO reflects the relationship between the Close and the Lowest Low over a given period, whilst this reflects the relationship between the Close and the Highest High. This momentum indicator has the same benefits and drawbacks as the Stochastic Oscillator.

%R = (Highest High over X - Close)/(Highest High over X - Lowest Low over X) * -100
Default value of X is 14.

3) Testing: Once each of these methods had been implemented they were tested. With each being subtly different, the exact testing methods used for each one differed slightly. An example; with Moving Averages two were calculated with the difference being the length over which they were calculated. Then the relationship between these two was observed and if the shorter-term average was significantly higher than the longer-term average then an upward trend was being observed, when the short-term average was significantly lower than the long-term average then the opposite was true. Once each has been tested individually they can also be used in conjunction with other techniques, which has also been shown in the results section.

C. Machine Learning

Machine Learning is a very broad topic; much of contemporary computer science is based on it or consists of research done around it. The main technique that is being applied here is that of a Support Vector Machine or SVM. This maximises the distance between two clusters, and is based in statistical learning theory. Using a kernel mapping, mapping a vector into a higher dimensional space, allows for linear separation to be performed, even on nonlinear datasets if the correct kernel mapping is chosen. Linear separation is the key to this machine learning technique, it maximises the distance between the known elements of each of the two classes using what is basically a constraint satisfaction problem. This 'margin' between the two classes is our confidence in the separation, if the margin is small then there is very little distance within the higher dimensional space between the two classes meaning a smaller confidence of success, if the margin is large then there is a higher confidence in the successful determination of a given points class. (Wilson 2008)

1) Set-Up: The set-up of data and input revolved around two functions. These are 'getPeaks' and 'getTroughs'. They take input of date, time, stock, and X. X is the number of data points that will be used to make a decision about the buying or

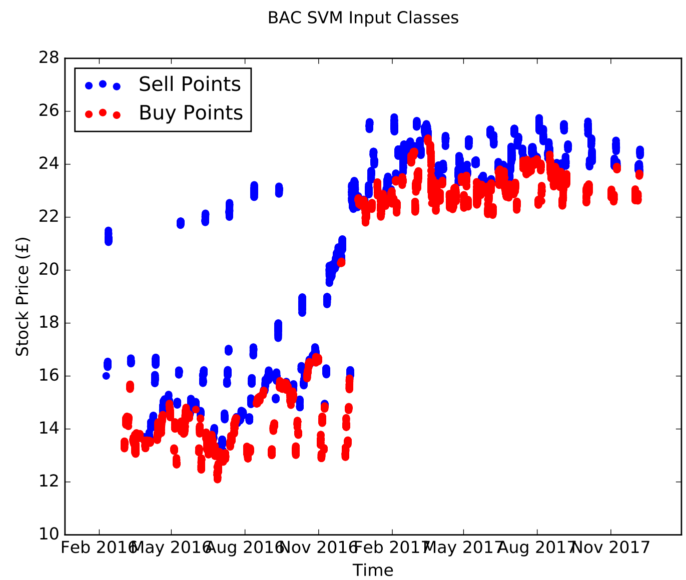

Figure 11. A graph showing the resulting data of the 'spikes' technique used to create the input data for both SVMs.

selling potential of the given data point. After X/2 data points have passed, a decision will be made using X/2 data points either side. 'getPeaks' will take this input, sub-sample the data and decide if this point was optimal to sell at, meaning it is a peak. 'getTroughs' will do the opposite, it will decide if this is a point at which it would have been optimal to buy. This data is then stored in tables 'SVMBuyData' and 'SVMSellData'. These are in the form of tables VIII and IX.

These tables are then used as input to the SVM learning function that is found as part of the e1071 CRAN library, which contains all set-up and query functions for the SVM (Meyer et al. 2017).

The functions 'getPeaks' and 'getTroughs' function in a similar way to 'shouldBuy' and 'shouldSell', they are very tune-able and are able to be changed to very different techniques without having an effect on the main functionality of the simulation. An example of a technique that could be used within these functions is differentiation. This technique is designed around the idea that data can be modelled as an equation and when the gradient of the line is 0, this is a stationary point, these will either be peaks or troughs. The second derivative is then used in conjunction with the data either side of the current point to decide if this stationary point is a peak or trough. This is an example of one of the many techniques used. Other techniques used are;
Rolling average - Take the highest points over variable length rolling average.
Spikes - Take any point as a peak if both neighbours are lower than that value and the inverse for troughs.
Median - Take the top and bottom X% of values.

An example of the results of these techniques is shown in figure X, showing the results of the 'spikes' technique.

Date	Time	Stock Value	Should Buy
01/03/16	09:30	9.11	False
...	...	...	...

Table VIII
SVMBuyData

Date	Time	Stock Value	Should Sell
01/03/16	09:30	9.11	False
...	...	...	...

Table IX
SVMSellData

2) Query: The query function within the e1071 library is very simple, given an SVM that has learnt its separator a function call will return the group that the given point belongs to.

This system is managed within the 'shouldBuy' and 'shouldSell' functions that were discussed within the statistical methods section. These have been modified so as to retrain and query the 'shouldBuySVM' and the 'shouldSellSVM' each time that they are called. Firstly one of these functions will be called within the simulation, for simplicity, 'shouldBuy' will be used as an example. 'shouldBuy' is called, within this function is the call to create and train the SVM associated with this function, namely 'shouldBuySVM', trained on the dataframe that is shown in table VIII. The data within this dataframe has been continuously updated throughout the simulation so as to avoid bulk computation when the SVM is used. This is done by continual calls to the 'getPeaks' function on regular intervals. Once the SVM has been trained, it is queried with the latest price for the given stock. This will return a class and then a decision will be made based on the class that has been returned.

IV. Results and Evaluation

The statistical methods throughout this paper have been tested in slightly unique ways, as explained within the testing section. This will be outlined for each of the methods. The machine learning methods will also be outlined but individually.

A. Technical Overlays

Bollinger Bands - Buy criteria will be based on percentage values in conjunction with the spread of the upper and lower band, giving an indication of volatility. The sell criteria is purely based on the relationship between the current price and middle band. Single variable of X was tested at X = 20, 15, and 35, with 20 being the default.

Chandelier Exit - The buy criteria is based on the relationship between the current moving average and the stock price at any given moment. The sell criteria are purely based on the conjunction between the long and short values of this overlay. Single variable of X was tested at X = 22, 17, and 27, with 22 being the default.

Ichimoku Cloud - The buy and sell criteria are based on the relationship between the A and B leading span with the other values providing momentum information. This has five variables and was not tested at any other values besides default.

KAMA - Allowing for clear identification of turning points, the buy and sell criteria are based on the difference between the KAMA value and the stock price. This has three variables and was not tested at any other values besides default.

Keltner Channels - Uses the same criteria as Bollinger Bands just with slightly different behaviour due to the differences between them. Takes two variables with default X = 20 and Y = 10, was tested at (15, 10), (25, 10), (20, 5), and (20, 15).

Moving Averages - Both Exponential and Simple moving averages were tested with basic comparison filters to give buy and sell criteria. Takes a single input and has no default, was tested at 5, 10, 15, 20, 25, and 30.

Moving Averages Envelopes - The buy criteria are based on the relationship between stock price and the lower envelope with the sell criteria being based on the relationship between the stock price and upper envelope. Takes two variables with default X = 20 and Y = 0.025. Tested at (15, 0.025) and (25, 0.025).

Parabolic SAR - Tracking of rising and falling SAR values gives an inference as to the correct time to buy and sell and these have been used as the criteria. This has no variables to be tested.

Pivot Points - The criteria for all three types of pivot point are similar with the only caveat being the number of support and resistance values that are taken into consideration as the buy and sell criteria change along with the value of the pivot point. These were only tested at default configurations.

Price Channels - Buying criteria is based on a moving average value in conjunction with a downward trend. Selling criteria is based purely on an inverting upward trend. These channels take a single variable with default value of 20, was then tested at 15 and 25 as well.

B. Technical Indicators

Aroon - Buy and sell criteria both based on the relationship between a moving average and the current stock price in conjunction with the trend information provided by the relationship between the up and down values. Single input value tested at 25 (default), 20, and 30.

Aroon Oscillator - Very similar to the Aroon test criteria but instead of using the relationship between up and down only the oscillator value is used. Single input tested at 25 (default), 20, and 30.

Average True Range - Using a moving average with bounded buy and sell criteria, ATR is used to indicate volatility of the price at a given point and this will impact the bounds used. This has no variables that needed to be tested.

BandWidth - Buy and sell are both based on percentage criteria of the bandwidth value as it is an inference of Bollinger Bands. Using Bollinger Bands as a base, this has a single input and was tested at 20 (default), 15, and 25.

%B Indicator - The same as BandWidth as they are both Bollinger Band derivatives. Both derivatives were tested at the same values as well.

Commodity Channel Index - Percentage criteria for buy and sell using the final CGI value. Taking a single input variable that was tested at 20 (default), 15, and 25.

Coppock Curve - The buy criteria is based on a moving average and the stock price; the sell criteria was based on the sign of the indicator. This takes three variables and was only tested at the default values.

DecsionPoint Price Momentum Oscillator - Using the value of the indicator in conjunction with a moving averages relationship with the stock price; buy is based on the moving averages relationship and not cut short by using the momentum indicator, the same is done in the inverse for sell. This was only tested at the default values.

Detrended Price Oscillator - This was tested and found to have limited use as a single indicator. Takes a single input and was tested at 15, 20, 25, and 30.

Mass Index - The buy and sell criteria are based on the calculation of the current trend and then use the indicator the show a trend reversal before the trend changes. This was tested at default values.

Moving Average Convergence/Divergence Oscillator - Used within the buy and sell criteria in the same way as all other momentum oscillators; in conjunction with a moving average and bounds. This was tested at default values.

MACD Histogram - Used identically to MACD, with the limited availability of inter-value relationship information. This was tested at default values.

Percentage Price Oscillator - Percentage bound for buy and sell criteria based on previous values of the PPO. This was tested at default values.

Pring's Know Sure Thing - Another momentum oscillator tested in conjunction with a moving average. This was tested at default values.

Pring's Special K - Using a percentage bound based around past values to give an indication of movement prediction. This was tested at default.

Rate of Change and Momentum - Used in conjunction with a moving average, this most basic momentum oscillator is the foundation for most of the more advance versions and so is tested in the same way. This takes a single input with no default and was tested at 5, 10, 20, 30, 50, and 100.

Relative Strength Index - Using previous values of RSI, bounds are calculated at the upper and lower range of the value 0 to 100 to give an indicator of the optimal times to buy and sell. The bounds have an initial value and are changed based on the data. This takes a single input with 10, 20, and 30 being tested.

StockCharts Technical Rank - Using the final value of SCTR, as well as the relationship between individual components, a confidence is calculated with respect to the changing of the price, as well as the direction of the change. This was only tested at default values.

Slope - Using the current slope value over the last set number of data points an inference can be made as to the future direction of the stock price. This projection is then used to indicate buying and selling patterns. Taking a single input, this was tested at 10, 20, and 30.

Stochastic Oscillator - Using the momentum value in conjunction with a moving average, confidence in the projection of price is known and thus buy and sell behaviour is simple. This takes two input values, with defaults (14, 3), this was tested along with (12, 3), (16, 3) and (14,2), (14, 4).

StochRSI - Using the previous values of the indicator upper and lower bounds are calculated to allow for the inference of buying and selling behaviour. This takes a single input with a default value of 14, this was tested, as was 10, 12, 16, and 18.

TRIX - Using percentage bounds, the current TRIX value is compared to a set of previous TRIX values and decisions on buying and selling are made based on this relationship. This takes a single input and was tested at 10, 20, and 30.

True Strength Index - Using percentage bounds based on previous TSI values, buying and selling behaviour is inferred. With two input values, the default (25, 13) was tested. As was (20, 13) and (30, 13).

Ulcer Index - The volatility shown using this indicator is used with regard to bounds initially set over a moving average and altered to reflect the volatility of the market. A single input, with default value 14, was tested along with 10 and 18.

Ultimate Oscillator - The smoothed momentum that this indicator shows is used in addition to a moving average to allow for inference within a trend to show turning points within the data and reduce the possibility of leaving a trend early. This takes three input variables that were tested at (7, 14, 28), (6, 12, 24), and (8, 16, 32), with the first being the default values.

Vortex Indicator - The relationship between the two oscillators calculated is used as trend and momentum information and all behaviours are inferred from these values and their relationship. With single input default value of 14, 10 and 18 were tested as well.

Williams %R - Used in an identical way to the Stochastic Oscillator, as this is an inference of that. Taking a single input value, along with the default 14, 10, and 18 were also tested.

C. Support Vector Machines

The first part of the testing of the support vector machine, or SVM, was to perform a search over variable space to find the best variables to use for testing. The variables that need to be tested consist of the data manipulation to start with, then the kernel that was to be used, as well as degree, C, and gamma, which are inputs to the function. These are all used within each of the SVMs, each of which are since class classifiers.

Data manipulation was split into differentiation, rolling average, spikes, and median. All discussed in section C.1. The kernel choice, also discussed in section C.1, is between a linear kernel, polynomial kernel, sigmoid function, and a radial basis function. These are all available through the e1071 library (Meyer et al. 2017). The degree, C value, and gamma are all searched via the library's built in grid search function.

D. Results

This section shows the best results that could be achieved within three sections of the paper. The first section is a purely statistical method used alone. The second is a conjunction of two or more statistical methods. The third is any formulation of support vector machine. These are shown in tables X, XI, and XII, respectively.

The aim of the project was to achieve a higher percentage in a given time frame that would be available from a high street bank. The values of which were set at 0.5% at the lowest (Bank of England 2018) and 1.85% at the highest (Murray 2018). As is clearly shown in Table X the use of any of the top ten statistical measures is enough to achieve this goal using even the highest fixed percentage individual savings accounts over the same time frame. The individual method results are then surpassed by the conjunction results show in Table XI, which are then again surpassed by the results of the SVM seen in Table XII.

The most successful results actions are shown in figure 12. This shows the points at which the most successful SVM bought and sold this specific stock. This gives an indication as to the success of this algorithm. The reason for no actions being made in the first six months is due to this being designated as a learning period and is not within the allotted trading time frame.

V. Conclusions

This section outlines the trends seen within the results that this paper has found and will also discuss the possible routes that could be taken if this project were to be extended.

A. Trends

As the results in Table X so clearly show it is very possible to outstrip the target set by a typical high street bank, using only statistical methodology. The table also shows a clear bias towards functions that base their methodology around the identification of the end of trends and the resultant information provided. This is a general trend however as there are several that are not based around this idea. Instead they are complete packages that calculate momentum and

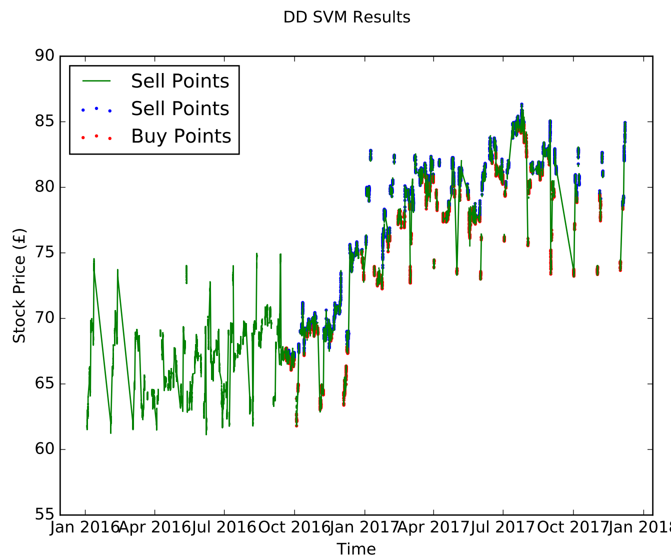

Figure 12. A graph showing the buying and selling points of both SVMs trained on the stock DD, as well as the stock price of DD.

trend information and use it within either long-reaching or complex systems. Another trend is seen in Table XI, the conjunction of a method based around the identification of trends and another method based around more complex moving averages give very promising results. The final results in Table XII gives a clear indication as to the best direction for future research. The benefits of machine learning within this context cannot be overstated. The results of the best of the SVM tests clearly show that statistical methods have a place within this field, it is just relegated to data manipulation to help improve the input for machine learning techniques.

B. Extension

An extension to this project that would be a boon to the testing would be a finer granularity of data. The data used in this paper was limited by the financial aspect of data acquisition; the paper and its findings were not hindered in this but it was a known limiting factor. Another data extension to be considered is the use of live data. This data would be taken periodically from a stock exchange to provide a more realistic approach to live trading. We maintain the statement that statistical methods are useful and will be needed no matter how advanced the project's machine learning techniques become. The number of data manipulation and extraction techniques that would be tested would increase, this is an open-ended task as no data would be perfect. The final extension to the would be testing a wider range of machine learning techniques. The use of deep learning within contemporary computer science is significant and this would be an obvious next step for this area.

References

Appel, G. (2005), *Technical analysis: power tools for active investors*, Financial Times/Prentice Hall.

Method	Parameters	Result
Parabolic SAR	Default	8.94%
KAMA	Default	8.29%
Ichimoku Cloud	Default	8.02%
MACD	Default	7.12%
Mass Index	Default	6.78%
Bollinger Bands	25	5.83%
StockCharts Technical Rank	Default	5.41%
DecisionPoint Price Momentum Oscillator	Default	5.16%
Stochastic Oscillator	14, 4	4.92%
Pivot Points	Default	2.19%

Table X
TOP 10 INDIVIDUAL RESULTS

Methods	Parameters	Result
MACD + KAMA	Default, Default	11.39%
Bollinger Bands + DecisionPoint Price Momentum Oscillator	25, Default	11.23%
MACD + Parabolic SAR	Default, Default	10.86%
MACD + Mass Index	Default, Default	10.84%
Stochastic Oscillator + KAMA	14, 4, Default	10.83%
MACD + Pivot Points	Default, Default	10.44%
Stochastic Oscillator + Parabolic SAR	14, 4, Default	8.18%
Bollinger Bands + KAMA	25, Default	6.96%
StockCharts Technical Rank + Parabolic SAR	Default, Default	6.19%
Bollinger Bands + Pivot Points	25, Default	6.02%

Table XI
TOP 10 CONJUNCTION RESULTS

Method	Parameters	Result
SVM	Differentiation + Sigmoid Function	18.41%
SVM	Median + Sigmoid Function	17.98%
SVM	Rolling Average + Sigmoid Function	16.17%

Table XII
TOP THREE SUPPORT VECTOR MACHINE RESULTS

Bank of England (2018), ‘Bank of England Statistical Interactive Database — Interest & Exchange Rates — Official Bank Rate History’. **URL:** *http://www.bankofengland.co.uk/boeapps/iadb/Repo.asp http://www.bankofengland.co.uk/boeapps/iadb/repo.asp*

Berutich, J. M., López, F., Luna, F. & Quintana, D. (2016), ‘Robust technical trading strategies using GP for algorithmic portfolio selection’, *Expert Systems with Applications* **46**, 307–315.

Blau, W. (1995), *Momentum, direction, and divergence*, Wiley.

Blokker, J. & Project, C. S. T. (2008), ‘The Application of SVM to Algorithmic Trading’, pp. 1–4.

Bollinger, J. (1992), ‘Using Bollinger Bands’, *Stocks & Commodities* **10**(2), 47–51.

Cartea, Á., Jaimungal, S. & Ricci, J. (2014), ‘Buy Low, Sell High: A High Frequency Trading Perspective’, *SIAM Journal on Financial Mathematics* **5**(1), 415–444.

Chande, T. S. & Kroll, S. (1994), *The new technical trader : boost your profit by plugging into the latest indicators*, Wiley.

Cliff, D. & Bruten, J. (1998), ‘Less Than Human: Simple Adaptive Trading Agents for CDA Markets’, *IFAC Proceedings Volumes* **31**(16), 117–122.

Duffie, D. (2012), *Dark markets asset pricing and information transmission in over-the-counter markets*, Princeton University Press.

Elder, A. (2002), *Come into my trading room: A complete guide to trading*.

Gerlein, E. A., McGinnity, M., Belatreche, A. & Coleman, S. (2016), ‘Evaluating machine learning classification for financial trading: An empirical approach’, *Expert Systems with Applications* **54**, 193–207.

Gjerstad, S. & Dickhaut, J. (1998), ‘Price Formation in Double Auctions’, *Games and Economic Behavior* **22**(1), 1–29.

Hasbrouck, J., Sofianos, G. & Sosebee, D. (2013), ‘New York Stock Exchange Systems and Trading Procedures’, *Journal of Chemical Information and Modeling* **53**(9), 1689–1699.

Hutson, J. K. (1983), ‘TRIX : Triple Exponential Smoothing Oscillator’, **3**(c), 1–6.

Iqbal, J., Ahmad, I. & Schmidt, G. (2012), ‘Can online trading algorithms beat the market? An experimental evaluation’, *3rd Student Conference on Operational Research* **22**(Scor), 43–52.

Kaufman, P. C. (1998), *Trading Systems and Methods*, 5th edn, John Wiley & Sons.

Keltner, C. W. (1960), ‘How To Make Money in Stocks’, *New York* **33**(10), 16–18.

Kim, K. (2007), *Electronic and Algoithmic Trading Technology: The Complete Guide*, Academic Press, an imprint of Elsevier.

Kleinman, G. (2005), *Trading commodities and financial futures: a step-by-step guide to mastering ...*

Krollner, B., Vanstone, B. & Finnie, G. (2010), ‘Financial time series

forecasting with machine learning techniques: A survey', *European Symposium on Artificial Neural Networks: Computational and Machine Learning* pp. 25–30.
Lambert, D. (1980), 'Charting The Market', *Commodities* .
Lima, L. (2017), 'Modeling of the financial market using the two-dimensional anisotropic Ising model', *Physica A: Statistical Mechanics and its Applications* **482**, 544–551.
Madge, S. (2015), 'Predicting Stock Price Direction using Support Vector Machines', *Independent Work Report* .
Martin, P. G. & Mccann, B. B. (1992), *The Investor's Guide to Fidelity Funds Winning Strategies for Mutual Fund Investing Publisher's Cataloging in Publication Data*, Venture Catalyst.
Meng, X., Zhang, J.-W. & Guo, H. (2016), 'Quantum Brownian motion model for the stock market', *Physica A: Statistical Mechanics and its Applications* **452**, 281–288.
Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F., Chang, C.-C. & Lin, C.-C. (2017), 'Package e1071'.
URL: *https://cran.r-project.org/web/packages/e1071/e1071.pdf*
Murphy, J. J. (1999), *Technical analysis of the financial markets : a comprehensive guide to trading methods and applications*, New York Institute of Finance.
Murray, A. (2018), 'Fixed-rate accounts reach two-year highs: here are the best'.
URL: *https://www.telegraph.co.uk/investing/bonds/our-pick-of-the-best-fixed-rate-savings-bonds/*
Pring, M. J. (2002), *Technical Analysis Explained: The Successful Investor's Guide to Spotting Investiment Trends and Turning Points*, McGraw-Hill.
Saad, S., Mohd Jaffar, M. & Sains Komputer, F. (2015), 'Review on the structural approach of the Black-Scholes model', **050084**(10).
Team, R. C. (2013), 'The R Project for Statistical Computing', *Http://Www.R-Project.Org/* pp. 1–12.
URL: *https://www.r-project.org/*
Tesauro, G. & Das, R. (2001), High-performance bidding agents for the continuous double auction, *in* 'Proceedings of the 3rd ACM conference on Electronic Commerce - EC '01', pp. 206–209.
Wilder, J. W. (1978), *New concepts in technical trading systems*, Trend Research.
Wilinski, A. & Kovalerchuk, B. (2017), 'Visual knowledge discovery and machine learning for investment strategy', *Cognitive Systems Research* **44**, 100–114.
Wilson, M. (2008), Support Vector Machines, *in* 'Encyclopedia of Ecology', Elsevier, pp. 3431–3437.