Gerard J. O’Rourke

Dr. Coleman

CSC-434-A: Thesis

May 10, 2021

Experiments in Algorithmic Trading

**Abstract**

Algorithmic trading is an important part of the modern-day stock market. It is estimated that roughly 50% of the trading volume in America come from algorithms rather than human decision makers. In order to gain a better understanding of this, we created an algorithmic trading program in Python. We were successfully in creating a functional program for algorithmic trading. Though the strategy itself needs refining, we were able to meet our software engineering goals and created a program that is highly flexible and maintainable.

**Background**

Algorithmic trading can be defined many ways. Investopedia defines algorithmic trading as, “use of process- and rules-based algorithms to employ strategies for executing trades” (Chen, 2020). There are many different ways of doing algorithmic trading. We decided to base our algorithm on a trend trading strategy that uses technical analysis. The bot would attempt to identify and capitalize on upward momentum of an asset’s price while selling when the price of an asset went down.

Technical analysis refers to a set of methods, primarily involving statistics and pattern recognition, of trends within an asset’s price. Almost any asset can be traded using technical analysis. Technical analysis is contrasted with fundamentals analysis, wherein a trader analyses the underlying fundamentals of a business such as economic factors, product success, earnings numbers, etc.

Consider the example of two traders: Alice and Bob. Both are interested in possibly investing money in Google. Alice uses a fundamentals-based strategy whereas Bob uses a technical strategy. To make her decision, Alice would look at the fundamentals of Google’s business: It’s last quarterly earnings, potential for future growth, new products, the general state of the economy, etc. To make his decision, Bob is more concerned with the statistics: The volume of Google’s price, the how fast it is risen or fallen, where the price is in relation to various moving averages, etc.

Both are valid ways of trading, though their aims are somewhat different. A fundamental-based strategy like the one Alice is using is well suited to more long-term investing, though there are ways of doing fundamentals analysis for short term trading. The technical strategy that we implemented is meant for more short-term trading where positions may only be held for minutes at a time.

**Technical Indicators**

To make decisions, our program used three main indicators: Simple Moving Average (SMA), Average directional Index (ADX), and Moving Average Convergence Divergence (MACD). We also experimented with Exponential Moving Averages (EMA), though this constituted a relatively minor portion of our experimentation.

The simple moving average (SMA) is calculated taking the closing price of mean of a set number of past candles and plotting that mean to the y-axis. The SMA can be thought of as a support or resistance level. If the price is above a particular moving average, it is considered to be in an up-trend. If it is below that moving average, it is considered to be in a down-trend. The SMA tends to give support and resistance If the price crosses the moving average, that is considered an indication of a trend reversal. Note that the more candles that are used to calculate a moving-average, the more significant it is. The price crossing the two hundred-period moving average is much more significant than the price crossing the fifteen-period moving average.

The average directional index (ADX) is an indicator meant to measure the trend of a stock. It measures the strength of a stock’s trend: Either up or down. The calculations for ADX are somewhat complex, but always work out such that ADX is measured between zero and one hundred. The higher the ADX, the stronger the trend, either up or down.

The SMA and ADX tell us which direction the price is trending in a quantifiable manner, but they do not tell the program when to buy a stock. For this we use the moving average convergence divergence or MACD.

MACD is calculated by subtracting the twenty-six-period exponential moving average (EMA) from the twelve-period EMA.[[1]](#footnote-1) A line called the signal line is calculated which is the nine-period moving average of the MACD. MACD revolves around the relationship between the MACD line and the signal line. When the signal line crosses above the MACD, this is considered a “bullish” sign (i.e., the stock will go up). When the signal line crosses below the MACD, this is considered a “bearish” sign (i.e., the stock will go down).

There are various ways of interpreting these indicators. Our program implements a bullish trend following strategy: Basically, the program tries to identify an uptrend and capitalize on it. The initial configuration was that if the signal line crosses above the MACD, if the ADX shows a clear trend by being greater than twenty-five, and the price was above the fifty-period SMA, the bot would buy. We later experimented with various configuration of these and other indicators.

It must be noted that technical indicators are not infallible. It is a well-known part of technical analysis that there is an element of risk with every trade that is made. At best, we were only able to achieve a 62.5% win rate, meaning only 62.5% of trades made money.

**Software Engineering Philosophy**

Our software engineering philosophy guided the design and implementation of our program.

We decided to emphasize maintainability and design over strategy. There was limited time for this project, thus we had to decide which features to implement and which not to. We decided to emphasize having a well written piece of code over having a profitable strategy. The reason for this is that the way it is implemented now, the strategy can be changed very easily with very minimal work. Had we not spent as much time planning the program’s architecture, this would not have been possible, and it would have been less time efficient to alter the strategy.

We also decided to not “reinvent the wheel” in our design. We did not write our own code where we could use an external library. One of the advantages of Python is that it has a wealth of external libraries that we were able to draw upon. This practice saved time and meant our code contained less errors. We used the “pandas” library for data analysis. We used the “ta” library for technical analysis. The “alpaca\_trade\_library” was used to implement communication with Alpaca, the brokerage our program uses. We also used several internal Python libraries in the course of making our program.

We also tried to make our code as modular as possible. This meant that it could have its function changed easily. For example, the API and strategy objects are handled in separate objects. Furthermore, the strategy is not incredibly asset specific. A user could easily rewrite the code to attach the bot to a brokerage for cryptocurrencies rather than equities.

We view our program more as a platform rather than a single bot for a single strategy. In this sense, it can be redone to trade any asset where technical analysis is applicable.

**Basic Functionality and Architecture**

When the program starts executing, a main function is called. The main function displays a menu in the Python terminal which the user can use to start the bot in real time mode or back testing mode.

Real time mode is for the bot to trade stocks on Alpaca using API credentials. Currently, the API credentials are hardcoded in the program. Though this would normally be considered a major security risk, the credentials connect to a paper trading[[2]](#footnote-2) account that does not have access to any real money. We decided that though this is technically a bad security practice, there was no real risk if the API keys were compromised because they only allow for algorithms to interface with the market (they do not allow for account information to be changed), this account does not control real money, and these keys can be disabled from within the account if they are compromised.

Right now, the bot is written to trade SPY (a popular ETF that follows the S&P 500) based on one-minute candles. The bot will use an API object to communicate with the Alpaca servers and retrieve market data. Once it has the market data, the data is passed to the strategy object which evaluates whether or not the trade is a good trade. If the algorithm deems it is a good trade, the bot places the order to the Alpaca servers. The bot adds stop loss and take profit orders[[3]](#footnote-3) to the order, so it does not actually have any logic for closing an order manually.

It is worth noting that the program does not make trades between 9:45 a.m. EST and 4:00 p.m. EST. The reason is that our program cannot make trades when the stock market is closed, and we have to wait until fifteen minutes after market open (9:30 a.m. EST) because our current Alpaca subscription does not allow the API to access data newer than fifteen minutes. The program has to trade using fifteen-minute old data.

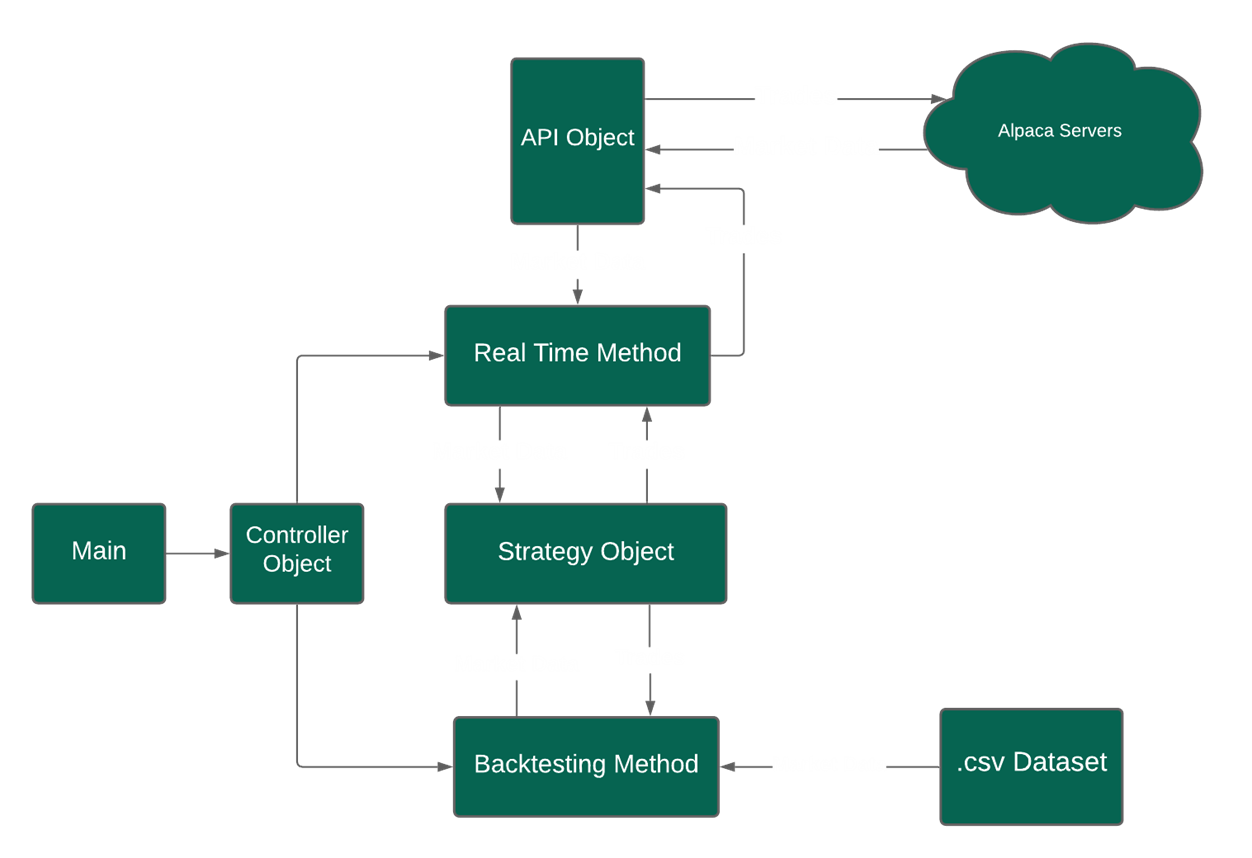


Figure : Program Architecture

If the user decides to call the back testing method, the program will read candle data for a particular asset from a .csv file and the back testing method will use it to simulate the passage of time on stock market by iterating through the candles. For each candle, a pandas data frame containing the candle and the past thirty candles will be passed to the strategy object. The strategy object will treat this data the same way it would real time data from the stock market and will make predictions based on the data.

If the strategy object decides to make a trade, it will signal the backtesting object. The back testing object will then simulate an open trade and record the results. When the test is complete, the bot will display statistics from the test to the terminal.

**Advantages and Disadvantages of Python**

This program was developed in Python. Python had a variety of advantages and disadvantages for programming this program. To start off, Python is easy to write, and change compared to other languages. This meant that those a lot of time went into the theory, architecting, and testing of the bot, the actual development time was comparatively quick, though there were some time-consuming bugs.

Furthermore, there are many libraries for Python. Part of the reason the development time was shorter was because there are many libraries for Python. We used the ta library for technical analysis, pandas for providing data frames, as well as the alpaca\_trade\_api for connecting to the Alpaca servers.

The major downside of Python is that being an interpreted language is that it is slow. This impacted backtesting with large datasets. The bot was designed to work with one-minute candles, but these datasets tend to be very large, and testing took prohibitive times as the strategy algorithm became more complex. Though we do not know for certain, we believe there might be inefficiencies in the TA library, particularly related to the ADX library. The ADX library nearly doubled runtime over the MACD alone. The problem seemed to be worse as larger datasets were used.

It is worth noting that this did not affect the real time portion of the program. Even on the worst configurations, the bot still takes less then a minute to process a candle. The bot only trades based off of one-minute candles, so the bot only needs to process one candle a minute in order to trade on the live market.

This time does become problematic when testing large dataset. One dataset we found consisted of the one-minute candles of bitcoin between January 1st, 2012 and December 31st, 2020. This was contained in a .csv file with several million rows. Processing this dataset was relatively impossible with the resources we had available.

**Backtesting Results**

Despite some troubles, we did have some valuable findings in our backtesting results. The main takeaway was that the bot was generally profitable. Most configurations did not lose money for any dataset. The bot was tested using the following datasets:

* A small excerpt of the aforementioned Bitcoin data with one-minute candles for a period of roughly 43.7 hours.[[4]](#footnote-4)
* The daily candles for SPY from May 2, 2020 to May 2, 2021; from May 2, 2011 to May 2, 2021; and from May 2, 2001 to May 2, 2021.
* SPY one-minute candles from 5/2/20 – 5/2/21
* Data from the Australian Dollar trading against the U.S. Dollar using one-minute candles from May 2, 2020 to May 2, 2021.
* Data from the Euro trading against the U.S. Dollar using one-minute candles from May 2, 2020 to May 2, 2021.
* Data from Tesla’s (ticker: TSLA) stock using one-minute candles from May 2, 2020 to May 2, 2021

It must be noted that these were not ideal datasets for the work being done. It must be noted that datasets with minute candles for the stock market can be hard to come by and often require money. Following the presentation, we were able to find a source that provided this data for free. Due to the fact that the runtime of constraints, we opted for more breadth of testing rather than depth, testing more datasets with fewer strategies rather than fewer datasets with every strategy. We were able to record the backtesting results. Please note that this is only a selection of the actual results, distilled for the most important times.

|  |  |  |  |
| --- | --- | --- | --- |
| **Selected Backtesting Results** | | | |
| Dataset | MACD Profit | MACD + ADX Profit | Underlying Asset Change |
| Bitcoin One Minute Candles | 4.63% | 1.30% | 2.82% |
| SPY Daily Candles 5/2/20 – 5/2/21 | 5.30% | 1.21% | 47.57% |
| SPY Daily Candles 5/2/11 – 5/2/21 | 19.27% | 18.65% | 206.34% |
| SPY Daily Candles 5/2/01 – 5/2/21 | 19.97% | 18.55% | 229.05% |
| AUD/USD Minute Candles 5/2/20 – 5/2/21 | -0.66% | 0.84% | 20.41% |
| TSLA Minute Candles 5/2/20 – 5/2/21 | 9.21% | 34% | 1.29% |
| EU/USD Minute Candles 5/2/20-5/2/21 | -0.74% | 1.77% | 9.52% |
| SPY Minute Candles 5/2/20 – 5/2/21 | -1.31% | 4.39% | 47.62% |

In general, it would appear that the bot only works in strong uptrends. This is to be expected because the strategy was designed with uptrends in mind. The bot in its current form does not make short positions (though there is some code to make implementing them easier).

Depending on the dataset, using ADX may or may not be helpful to the strategy. Except for the foreign exchange datasets, using ADX always increased the win rates of our trades. It decreased profitability on the datasets using daily candles but increased profitability on datasets using minute candles.

We also tested various simple moving averages using the fifty-period, twenty-six-period, and twelve-period, as well as the twenty-period exponential moving average. The first did not seem to have an affect while all the others decreased the profitability. It must be noted here that these were not as extensively tested as the other two. At one point, we changed our implementation of MACD to sell when the signal line crossed below the MACD line. This had no effect on our trades and only increased the run time of the program. This is likely because MACD is a lagging indicator, meaning it takes several candles to respond to market conditions. By the time the signal line crossed the MACD line, the stop loss had already been triggered. Most of our tests were done with the old implementation of MACD.

Our testing revealed that it the program in its current form seems to fill an oddly specific niche: The bot had the most success on datasets where the price increases, peaks, and then decreases. In both the Bitcoin and Tesla datasets, the prices increased (quite dramatically in the case of Tesla) and then decreased. In all other datasets, where there was a consistent increase, the bot was profitable, but underperformed the underlying asset. Except for Tesla and Bitcoin, a wealthy investor with one million dollars to invest would be better off that money by simply buying the underlying asset rather than investing it in a hedge fund managed by our program. In all honesty, it bears noting that Bitcoin has since dramatically increased in price since the end of dataset we used, though Tesla has fallen even lower.

**Possible Improvements to the Algorithm**

Though we were able to do much with this project, this project was developed with limited time. We only had so much time to research the theory, design and develop the bot, debug it, find data, and test the strategy. As such, though the basic part of the project is done, there are possible improvements that could be made that we did not have time to implement.

The algorithm’s intent was to capture the upside of the stock while avoiding the downsides. This did not work to the degree we expected it to. One possible improvement we could implement is adding more sophisticated logic to closing orders. Right now, our bot closes a position after its open position’s value moves 5% one way or another. We picked such a high number because the algorithm was originally designed with the cryptocurrency market in mind, which is notoriously volatile. This may be a big part of the reason that the bot suffers so much with data sets from the foreign exchange market, which is much less volatile.

Had there been more time to work on the project, improving the way the bot closes an order would be a worthwhile area of exploration. There are several methods for improving this.

Secondly, we only seriously tested three technical indicators, whereas there a number of other technical indicators that we could have added but did not have time to. Relative Strength Index and Bollinger Bands were possible candidates that might have improved our strategy’s performance.

Lastly, there were multiple variables for a number of different aspects of the strategy, such as the amount of the portfolio to invest, various variables related to the technical indicators, etc. Though it would have taken a while to implement, we could have used some form of machine learning, such as multiple linear regression or something similar, to find an optimal configuration.

Furthermore, we could have used machine learning or general automation to automate certain aspects of the testing process such as which combination of indicators work well with which kind of market. This would allow us to improve our algorithm both in general and tailor it to trade on the unique conditions present in different asset classes. As we have learned from our experimentation, what may work in the cryptocurrency market does not necessarily work in the foreign exchange market. Machine learning could allow us to build models to tailor our trading strategy to each asset class.

**Conclusion**

Overall, the project was a success. Though our strategy requires major refining and there certainly additional features we could have implemented, we did manage to create a working algorithmic trading platform, which was the main goal of the project. We met all of our goals in the area software engineering which was to create a functional robot that required minimal unnecessary writing of code, was module in architecture, and could be easily reworked. It was an excellent learning exercise, and this program definitely has potential to be continued.

Works Cited

Chen, James. “Algorithmic Trading Definition.” *Investopedia*, Investopedia, 13 Jan. 2021, [www.investopedia.com/terms/a/algorithmictrading.asp](http://www.investopedia.com/terms/a/algorithmictrading.asp).

Hayes, Adam. “Technical Analysis Definition.” *Investopedia*, Investopedia, 7 May 2021, [www.investopedia.com/terms/t/technicalanalysis.asp](http://www.investopedia.com/terms/t/technicalanalysis.asp).

Segal, Troy. “Fundamental Analysis.” *Investopedia*, Investopedia, 7 May 2021, [www.investopedia.com/terms/f/fundamentalanalysis.asp](http://www.investopedia.com/terms/f/fundamentalanalysis.asp).

Maverick, J.B. “How Is the Average Directional Index (ADX) Calculated, and What Is the Formula?” *Investopedia*, Investopedia, 9 Apr. 2021, [www.investopedia.com/ask/answers/112814/how-average-directional-index-adx-calculated-and-what-formula.asp](http://www.investopedia.com/ask/answers/112814/how-average-directional-index-adx-calculated-and-what-formula.asp).

Mitchell, Cory. “Moving Average Ribbon Definition and Uses.” *Investopedia*, Investopedia, 23 Mar. 2021, [www.investopedia.com/terms/m/movingaverageribbon.asp](http://www.investopedia.com/terms/m/movingaverageribbon.asp).

“Historical Data Feed :: Dukascopy Bank SA: Swiss Forex Bank: ECN Broker: Managed Accounts: Swiss FX Trading Platform.” *Dukascopy Bank SA*, 8 May 2021, [www.dukascopy.com/swiss/english/marketwatch/historical/?lang=en&ibref=8507](http://www.dukascopy.com/swiss/english/marketwatch/historical/?lang=en&ibref=8507).

“AUD USD Chart - AUD/USD Rate.” *TradingView*, 9 May 2021, [www.tradingview.com/symbols/AUDUSD/](http://www.tradingview.com/symbols/AUDUSD/).

“Tesla, Inc. (TSLA) Stock Price, News, Quote & History.” *Yahoo! Finance*, Yahoo!, 8 May 2021, finance.yahoo.com/quote/TSLA/.

Zielak. “Bitcoin Historical Data.” *Kaggle*, 11 Apr. 2021, [www.kaggle.com/mczielinski/bitcoin-historical-data](http://www.kaggle.com/mczielinski/bitcoin-historical-data).

“SPDR S&P 500 (SPY) Stock Price, News, Quote & History.” *Yahoo! Finance*, Yahoo!, 9 May 2021, finance.yahoo.com/quote/SPY?p=SPY.

“High Frequency Trading (HFT) Definition.” *Nasdaq*, [www.nasdaq.com/glossary/h/high-frequency-trading#:~:text=It%20is%20estimated%20that%2050,computer%2Dbacked%20high%20frequency%20trading](http://www.nasdaq.com/glossary/h/high-frequency-trading#:~:text=It%20is%20estimated%20that%2050,computer%2Dbacked%20high%20frequency%20trading).

1. An exponential moving average is like a simple moving average, but more recent prices are given more weight. [↑](#footnote-ref-1)
2. Paper trading is the practice of trading using imaginary money. This allows traders to test their strategies without risking any capital. [↑](#footnote-ref-2)
3. Stop Loss and Take Profit orders refer to a type of order on the stock market where the position is closed after it gains or losses a certain percentage of its value. [↑](#footnote-ref-3)
4. Due to the broad timeframe of the original and the somewhat peculiar format of the .csv file, we were unable to determine the exact time period covered by our excerpt. [↑](#footnote-ref-4)