Momentum Trading Algorithm

'Buy Low, Sell Lower'

K. Alexander, A. Jackson, J. Lyell

A Project Completed by the Trading Team in conjunction with

The University of Western Australia
September 2021 Quantitative Finance UWA

Contents

1 Introduction

1.1 Momentum Trading

"In stock market trading, momentum is the rate of acceleration of a security's price – That is, the speed at which the price is changing" [1].

Momentum is the price's tendency to continue rising or falling over a period, usually in response to both the price of the stock and volume of the stock being traded. The goal is to buy stocks that are rising and sell them when they look to have peaked. This can be automated with algorithms.

There are many ways to implement momentum trading. Various methods involving different parameters and different indicators can be used to trade on. Our goal is to identify a type of algorithm that is consistent across a range of popular cryptocurrencies.

2 Algorithm Selection

2.1 Overview

We decided to focus our efforts on a smaller selection of algorithms but use a wider range of parameters, to try and generate the best algorithm. We tested the classic moving average, both with and without a volume filter, an exponential moving average and a moving average crossover. These three quantitative algorithms were the most interesting to us and appeared to have the most promise from a very simple initial test.

2.2 Simple Moving Average

2.2.1 Without Volume Filter

The Simple Moving Average (SMA) indicator uses a moving average of recent price trends for a defined time period, as shown in Equation 1.

$$\text{SMA[n]} = \frac{A_1 + A_2 + ... + A_{n-1} + A_n}{n} \tag{1}$$

Where:
n is the number of samples in the selected time interval
A_n is the asset price at time n.

This indicator can be used to predict future price movements by considering the moving average and comparing this to the current trading price. For example, if in the current time period, the close price is higher than the moving average of previous periods, the coin is perceived to have strong positive momentum. In this example the algorithm would enter a position in the coin, in the hopes that the price will continue to rise. The algorithm would use the inverse to decide on a time to sell, attempting to sell when the momentum reverses just after the price has theoretically peaked.

Shortcomings of this indicator are its inability to consider trade volumes and to distinguish recent prices from older prices within its calculation. These flaws are addressed in the other moving average indicators analysed within this report.

2.2.2 With Volume Filter

To incorporate trade volume into our trading algorithm, we combined a volume filter with the SMA indicator described in the last section. The volume filter is almost identical to the price SMA, except averaging trade volume rather than trade price (see Equation 2).

$$\text{Volume SMA[n]} = \frac{V_1 + V_2 + ... + V_{n-1} + V_n}{n} \tag{2}$$

Where V_n is the volume of trades at the end of interval n

This volume filter is also used to predict bullish and bearish trends by comparing the most recent trading volume to the Volume SMA. For example, our algorithm would attempt to enter a position if the most recent volume was greater than the Volume SMA as this is indicative of a market movement. This was conditional on the Price SMA as described in our first algorithm, and both conditions had to be satisfied to enter a position.

2.3 Exponential Moving Average

The Exponential Moving Average (EMA) indicator is similar to the SMA, in that it uses an average of the price over previous time periods to decide on whether to buy or sell. However with an exponential moving average, more recent data is given an exponentially higher weighting than older periods, as shown in Equation 3.

$$\mathrm{EMA}[n] = A_n \times \frac{2}{n+1} + \mathrm{EMA}[n-1] \times \left(1 - \frac{2}{n+1}\right) \tag{3}$$

Where A_n is the price at the end of interval n

We decided to omit the volume filter from the Exponential moving average as it appeared to worsen our preliminary results.

2.4 Moving Average Crossover

Moving Average Crossover analysis is a technique which uses two SMA indicators of significantly different time inverval values (values of n), often labeled Long SMA (larger n) and Short SMA (smaller n). This technique seeks to compare these two SMAs rather than the most recent coin price. This comparison attempts to illustrate trends rather than instantaneous movements. A 'Golden Cross' is often used when the Short SMA rises above the Long SMA indicating a positive trend. Conversely, a 'Death Cross' describes the instance when the Short SMA dips below the Long SMA.

3 Algorithm Optimization

3.1 Overview

For the project this semester, trading teams were given 30-minute interval data for seven popular Cryptocurrencies, namely, ADA, BTC, DASH, ETH, LTC, NEO, XRP. This data spanned the year 2020, and included the time, trade volume and the pening, closing, highest and lowest prices for that interval. The goal was to optimize a basic momentum trading python script to maximize profit, based on the provided data. Whilst we trained this algorithm with data from 2020, we still attempted to optimize the algorithm for future coin performance, as many consider 2020 to be an overly successful year for Cryptocurrencies. It is also worth noting that a 0.01% fee for each trade was included. Our optimization approach involved determining the main parameters for each of the previously mentioned strategies and optimizing for these based on their profits for all seven coins.

3.2 SMA

3.2.1 Without Volume Filter

The single most important parameter for the SMA strategy (without the volume filter) is the n value, which determines the number of past price points to include in the price average (Price Window). Since the time data is in 30-minute time steps, we expected the number of points to be relatively low and ran simulations for the entire year for n ranging from 0 to 50. These simulations would return a percentage profit over the year for each coin, to allow for coin comparison regardless of their individual prices. These differing coin profits for each value of n were averaged, with the highest profit pertaining to the optimized value/s of n.

3.2.2 With Volume Filter

The SMA strategy with the volume filter has two key parameters to consider, the number of past prices to include in the price average (Price Window n_P) and the number of past volume points to include in the volume average (Volume Window n_V). As with the previously described optimization process, we also ran year long simulations for each coin for both values of n ranging from 0 to 50. Note, since these parameters are not in dependant, we had to run these simulations for every possible combination of n_P and n_V, i.e. [n_P=0 & n_V=0, n_P=0 & n_V=1, . . . , n_P=50 & n_V=49 , n_P=50 & n_V=50]. The profits from these simulations were once again averaged across all seven coins to select the optimal n_P, n_V pair.

3.3 Exponential Moving Average

The EMA strategy's only key parameter to optimize is the number of past price points to include in the Price Window, as with the SMA strategy. This single parameter was optimized through the same process as the SMA strategy.

3.4 Moving Average Crossover

The Moving Average Crossover strategy requires the optimization of size of the Short Price Window (n_Short) and the size of the Long Price Window (n_Long) simultaneously. As with the SMA including volume filter strategy, all possible combinations of n_Short and n_Long were considered with each ranging between 0 and 50. This allowed for the strategy to be optimized for a pair of n_Short and n_Long values.

4 Algorithm Performance

4.1 Simple Moving Average

4.1.1 Without Volume Filter

Figure 1 shows the percentage profit versus the Price Window size for all seven coins. It is immediately obvious that LTC dramatically outperformed all the other coins for values of $n < 35$. This was a trend for all of the strategies and hence suggests that its incredible performance was almost entirely in dependant of any momentum strategy employed. It is clear that there is a shared peak around low values of n and some much smaller individual spikes for $10 < n < 20$. The final results, of the percentage profit averaged over all the coins is shown in Table 1. The optimum parameter found for this strategy was a Price Window of 2.

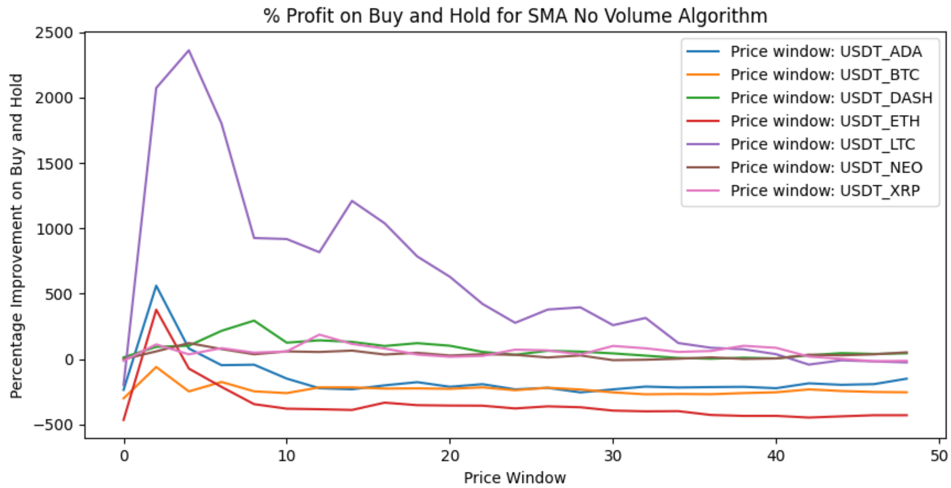

Figure 1: Profit % of each coin, with varied Price Window (n values)

Table 1: Net Return of SMA without filter for varied Price Window

Price Window (n)	Net Return	Price Window (n)	Net Return
0	-1201.01	26	-281.33
2	3215.24	28	-341.87
4	2381.89	30	-486.99
6	1743.31	32	-463.03
8	670.03	34	-695.47
10	369.89	36	-747.83
12	377.73	38	-720.14
14	683.42	40	-779.35
16	493.09	42	-826.58
18	234.81	44	-813.51
20	-20.1	46	-833.21
22	-228.26	48	-778.12
24	-434.04		

4.1.2 With Volume Filter

Figure 2 shows the percentage profit (averaged over all coins) for all different values for the Price and Volume Window. It is obvious that the pair, $n_P = 2, n_V = 2$ was clearly the top performer for this strategy. Interestingly, it appears that there is a strong gradient with respect to changes in the Price window, but little to no gradient with respect to changes in the volume window. This suggests that trade volume is also largely dependant on trade prices and does not deviate enough to dramatically effect the algorithm. Since there are 625 data points for this algorithm, the most interesting results are shown in Table 2.

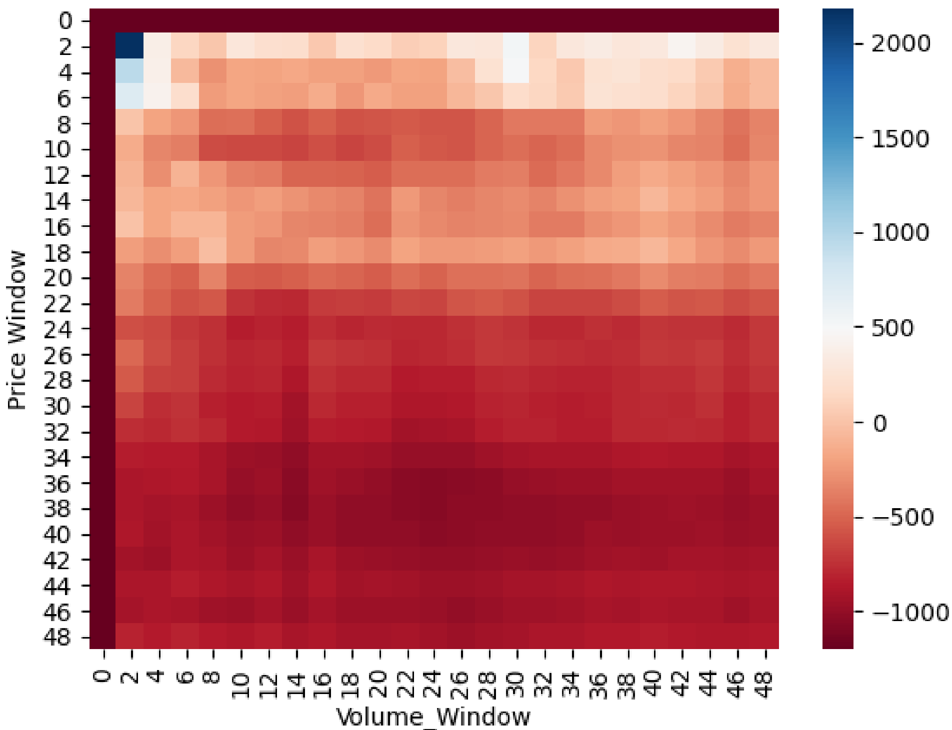

Figure 2: Profit % of each coin, with varied Price Window and Volume Windows (n values)

Table 2: Net Return of SMA with volume filter for varied Price and Volume windows

Price (n_P)	Volume (n_V)	Net Return	Price (n_P)	Volume (n_V)	Net Return
2	2	2184	10	2	-142.14
2	4	382.18	10	4	-338.65
2	6	131.51	10	6	-381.04
2	8	28.11	10	8	-618.62
2	10	283.08	10	10	-624.59
2	12	202.06	10	12	-627.64
2	14	197.55	10	14	-656.46
2	16	35.16	10	16	-594.93
4	2	950.19	12	2	-98.76
4	4	397.38	12	4	-299.2
4	6	-59.76	12	6	-93.46
4	8	-287.34	12	8	-260.98
4	10	-172.58	12	10	-362.15
4	12	-188.82	12	12	-386.03
4	14	-161.74	12	14	-496.16
4	16	-198.92	12	16	-499.42
6	2	721.53	14	2	-67.11
6	4	413.7	14	4	-175.14
6	6	196.37	14	6	-165.99
6	8	-227.1	14	8	-203.67
6	10	-176.99	14	10	-253.37
6	12	-198.05	14	12	-215.48
6	14	-210.5	14	14	-270.77
6	16	-137.93	14	16	-316.2
8	2	8.68	16	2	-6.21
8	4	-191.2	16	4	-174.01
8	6	-254.14	16	6	-83.5
8	8	-458.87	16	8	-83.89
8	10	-435.29	16	10	-234.37
8	12	-525.05	16	12	-256.93
8	14	-588.31	16	14	-336.16
8	16	-523.67	16	16	-346.22

4.2 Exponential Moving Average

Figure 3 shows the percentage profit versus the Price Window size for all seven coins. Once again, LTC greatly outperformed the other coins and a Price Window value, $n = 2$ was the optimized parameter value. The net profit results can be found in Table 3

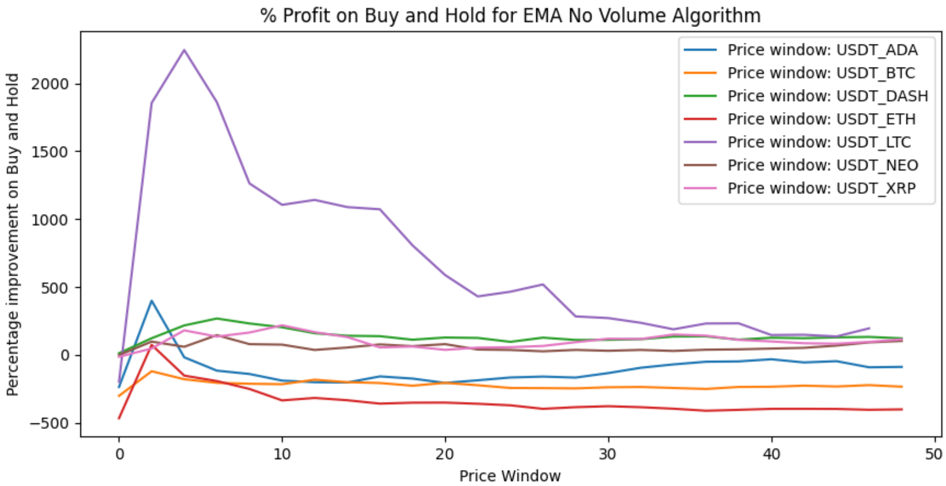

Figure 3: Profit % of each coin, with varied Price Window (n values)

Table 3: Net Return of EMA without filter for varied Price Window

Price Window (n)	Net Return	Price Window (n)	Net Return
0	-1201.01	26	-60.05
2	2478.27	28	-271.13
4	2356.33	30	-214.88
6	1897.93	32	-206.44
8	1137.35	34	-202.95
10	865.31	36	-160.69
12	808	38	-188.67
14	680.5	40	-241.92
16	621.5	42	-266.37
18	294.78	44	-266.37
20	72.54	46	-198.62
22	-118	48	-381.48
24	-123.1		

4.3 Moving Average Crossover

Figure 4 shows the percentage profit (averaged over all coins) for all different combinations of Short and Long Price Windows. This algorithm produced the most interesting results from all the algorithms as well as the largest domain for profitability shown by the large blue area. This suggests that this algorithm would be more reliable for a diverse range of coins and economic climates. The optimized pair selected was $n_{Short} = 24, n_{Long} = 40$.

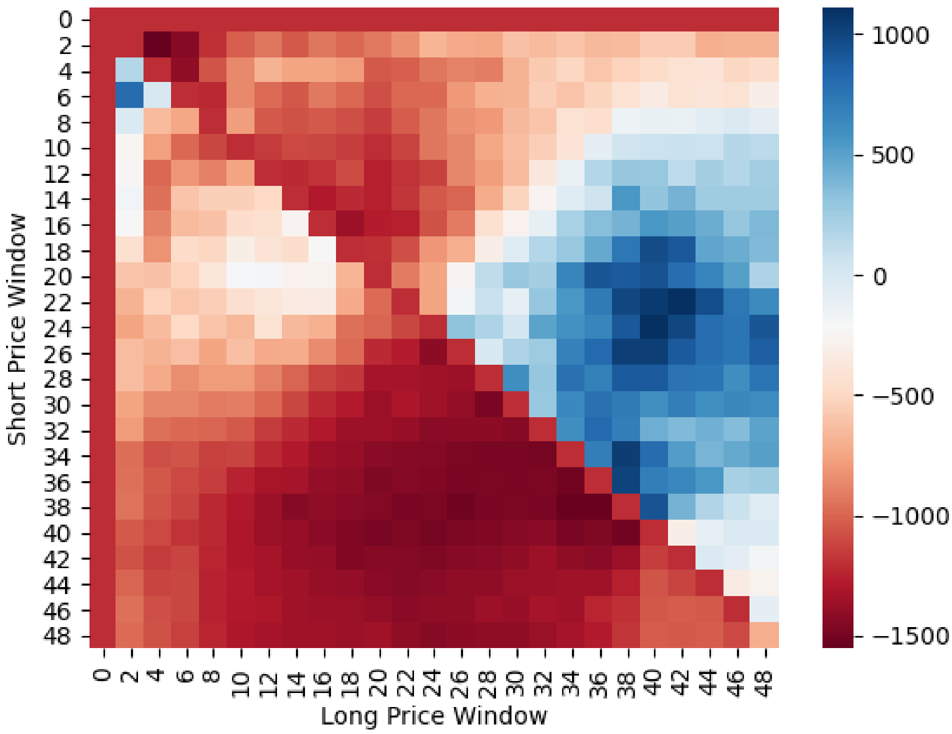

Figure 4: Profit % of each coin, with varied Long Window and Short Windows

4.4 Algorithm Results

After optimizing our parameters as discussed above, we compared the returns of each algorithm with said parameters to find the best algorithm overall. Figure 5 shows the resulting break down of each algorithm against all 7 coins and the average return. All algorithms average return was positive relative to the standard buy and hold strategy, with SMA no volume filter returning the best with an average improvement of +459.32%. The Crossover algorithm was the most consistent algorithm, however, its average improvement was the worst at +159.17%.

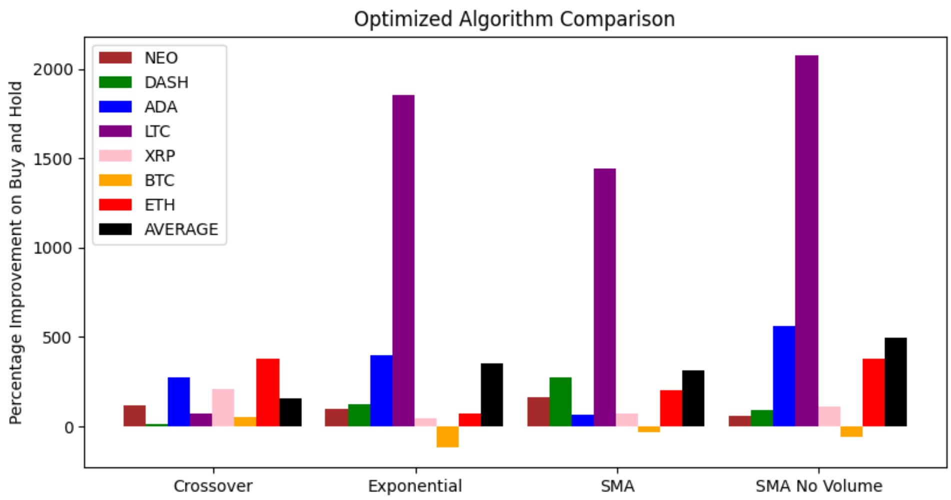

Figure 5: Algorithm with Optimal Parameter Comparison

4.5 Coin Selection

As mentioned previously, our coin selection was relatively straight forward with LTC outperforming all other coins by a significant margin. Figure 6 highlights this with a comparison between all coins for our chosen algorithm (SMA no volume filter).

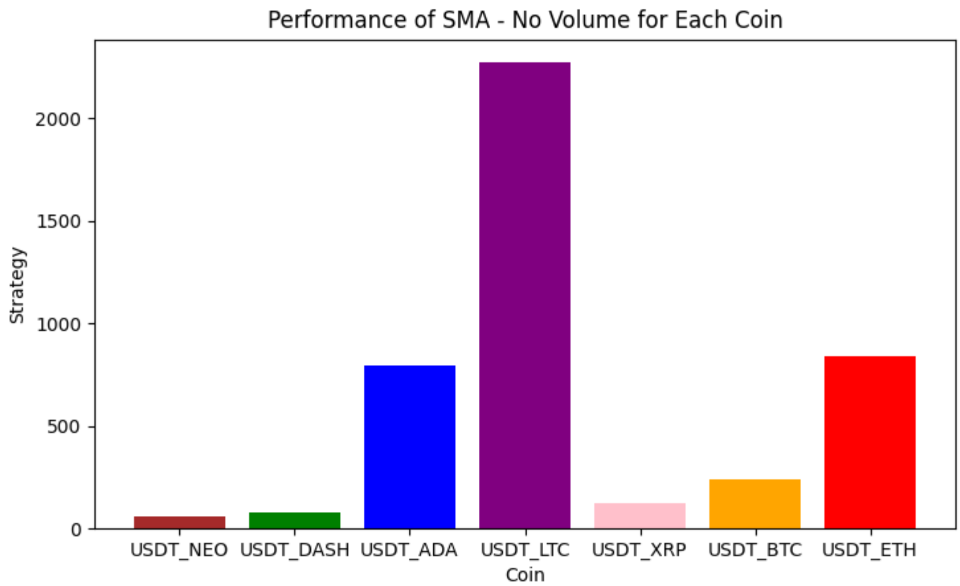

Figure 6: Volume No Filter Returns with Optimal Parameters

5 Summary and Findings

5.1 Discussion

Our project this semester took a very quantitative approach to momentum trading, not only in algorithm selection and testing but also in result analysis. The selected algorithm and coin (SMA no volume on LTC) produced extremely promising results over the time period and coins tested (See Figure 7). This algorithm, however, was not extensively tested over a multitude of coins or against different market conditions and could potentially produce volatile results. In addition, the algorithm makes a large amount of trades in a small time frame which is optimal in this test scenario with flat rates, but could be impractical if implemented with standard fees. It will be interesting to see the results of a back-test with unseen data in the follow-up trading report.

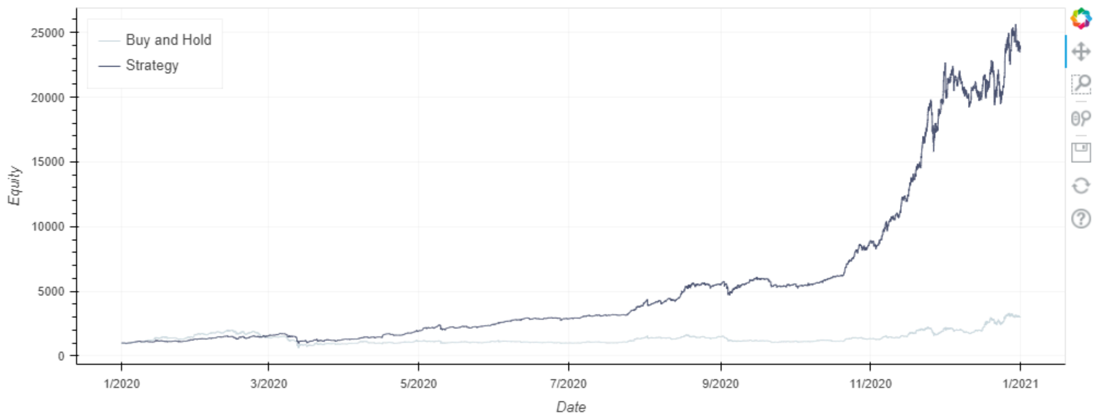

Figure 7: Volume No Filter Returns with Optimal Parameters

References

[1] R. Dhir, *Momentum*, May 2021. [Online]. Available: `https://www.investopedia.com/terms/m/momentum.asp`.