Refined Hammer Line Trading Strategy

Final report for COMP 4971C

Nov 2021

WANG, Ruida

676914295@qq.com

Supervised by

David Rossiter

Keywords: Hammer Line, derivative estimation, dynamic return rate, and long-shorting trading strategy

Abstract

This research will discuss the formation and detail of a trading strategy based on the Hammer Line Entry Signal and named the Refined Hammer Line. Firstly, using Hammer Line as the entry signal and stop-loss basis, then applying the derivative estimation to judge the trend of the market, if the market is in a growing trend, performs buying strategy if it is in dropping trend, performs shorting strategy. Then the return rate is dynamic to the market trend judgment. After every optimization step, we test it in the Bitcoin to USD market (noted as BTCUSD). Finally, we have created an algorithm that can perform considerable benefits in most cases no matter how the market performs under historical data performance under applying the same set of strategies. The return rate of our trading strategy achieves 744 times of return rate compared to the 295 times return rate of the buy-and-hold strategy with the maximum draw-down rate of 40% (a considerably low draw-down rate in the market of cryptocurrency). Then, to avoid the problem of overfitting for the trading strategy in the BTCUSD market, the same strategy is also tested in the Ethereum to USD market (denoted as ETHUSD). It also gives a relatively good result.

Table of Contents

Figure table

Chart table

1. Introduction

1.1 Original idea basis

There is an ever-lasting conflict about the relation between past and future in the financial market. In practice, it is impossible to make a financial decision without historical data. Nevertheless, an old saying that the past financial market does not mean the future keeps on proving itself multiple times. History has always proved that sometimes the majority believe that the market will grow and return with a great crisis. Despite the trend for financial research to feed artificial intelligence a massive amount of data and test for the outcome, this research uses a different approach that only focuses on local data and uses the vast amount of historical data for testing. Our algorithm can fix the problem that the algorithm based on vast amounts of data does not apply well to aggressive and new markets like the cryptocurrency market. However, the most profitable chances are all lies in such market.

The original idea for coming up a trading strategy like this that when doing the pre-research searching about stock trading strategies, an interesting result is there are a lot of trading strategy that have been hotly discussed both on YouTube and everywhere on the discussion community. For over 20 years, on the Internet many articles keep on show up to introduce those kind of trading strategies, some of them even been included in lecture notes of some statistical analysis courses [1]. However, many of the trading strategy have not been rigorously proved by financial research (this is concluded from the fact that it is hard to find any related research report on Google Scholar) it means that there may exists a lot of valid and profitable trading strategies that is not carefully investigated through research. The trading strategy that this study will discuss is such kind of strategy, the entry signal for the strategy is called "**Hammer Line**".

1.2 Software Basis

In this project, I use programming-based test for the trading strategy and diagram analysis to test how profitable the strategy is. Here are the environment build softwares and statistical analysis tools that are used in the project

- IntelliJ IDEA for the IDE

- Java Adopt Open JDE (HotSpot) version 16.0.1
- Java SDK 16
- Reader and Writer based on Utils written by Professor *P. N. Hilfinger* in UC, Berkeley
- Test environment library given by CS 61 BL instruction team
- Self-written data analytic tools

1.3 Source of data

All cryptocurrency data that this project use is get from *Crypto Data* [2] using Gemini Exchange data of Bitcoin and Ethereum:

- Bitcoin to USD (denoted as BTCUSD): from 13:00 p.m. 8th October 2015 to 0:00 27 September 2021
- Ethereum to USD (denoted as ETHUSD): from 13:00 9th May 2016 to 0:00 29th September 2021.

Both data has the data set of an hour and records every hour’s Open, High, Low, Close, Volume of the trade.

And the stock data this project use in the final part of the report is get from Bloomberg terminal.

2. Some Fundamental Elements of this study

2.1 Price of exchange

Since in the data set used for this project, every hour of historical record of Bitcoin to USD and Ethereum to USD contains more than one parameter (in particular open, close, high and low in that interval). But most of the data analysis tool in this project use requires single input for a certain period also for the buy-in and sell-out. Thus, in this project, the average of open and close for a specific hour is considered as the price for calculation.

$$price = \frac{open + close}{2}$$

2.2 Least Square Method

In the trading strategy, one of the essential procedures is to judge the current trend of the market to decide whether to do short or buy trading. In this research, such a trend will be identified using the least square method to estimate one set of previous data to estimate the linear expression of price. The formula gives as follows:

Given a set of data that has data density of one hour per number:

$$\{x_1, x_2, \cdots \cdots x_n\}$$

Firstly, calculate the average of x and i, denoted by: $\bar{x}$, $\bar{y}$

The trend of such set of data is calculated by:

$$k = \frac{\left(\sum_{i=1}^{n} x_i \cdot i\right) - n\bar{x}\bar{y}}{\left(\sum_{i=1}^{n} x_i^2\right) - n\bar{x}^2}$$

2.3 Moving average

Due to the rapid change of price in bitcoin and all other cryptocurrencies, the current price sometimes cannot represent the current market trend. Thus, to smooth the growth and draw of the market, this research uses a simple moving average of a particular hour instead of price to determine the market trend. Calculated by the formula:

Given a set of data of price with one hour per data point:

$$\{x_1, x_2, \cdots\cdots x_{n*24}\}$$

Moving average is calculated by:

$$n\ day\ MA = \frac{\sum_{i=1}^{n*24} x_i}{n}$$

Note that since the analysis will heavily rely on the moving average of the market in the latter part of the report, the notation for the moving average is described as follows. If we use 3 hour 20-Day MA, it contains two parts, assume it is 16:00 24th Nov, then the 3 hour 20-Day MA represents the set of 20-Day MA of 16:00, 15:00, and 14:00. For example, the 20-Day MA of 16:00 is calculated as follows: beginning at 16:00 24th Nov get past 20 days price data in the hour (i.e., totally 480 data points of price), then take the average of them. Thus, we get the 20-Day MA of 16:00 24th Nov.

Thus, the notation can be described as follows:

<time interval of sets> + *<moving average types>*

In this research, sometimes the time interval may be not in hours but in days (to simplify the notation) the 3 days 20-Day MA is equivalent to 72 hours 20-Day MA (i.e., using a data set that contains 72 elements which represent each hour's 20-Day MA in the past 72 hours).

2.4 Trading cost

Although the transaction between cryptocurrencies and the trade between accounts is almost free in the bitcoin market, when liquidating the bitcoin to USD or other currencies, the platform will draw the transaction cost of 0.1% per trade [3]. Thus, for every trade that we simulate during this research, we will deduct the amount of money we get by 0.1% for both buy and sell trading.

2.5 Shorting success rate

In the algorithm implemented by this research, at some point, we will introduce a short strategy into our algorithm. And as said by a professional in finance research, due to experience, the average rate of successful shorting is 40% to 60%. For simplicity, whenever we are going into short trading, a successful shorting judgment is made, with a success rate of 50%. This rate represents an abstraction of all possible situation that may happen for example, no chance to do the shorting, the shorting interest rate is not suitable and so on and so forth.

2.6 Buy and hold and Compare basis

Buy and hold is a passive investment strategy in which an investor buys the security and hold then for a considerable long period regardless of fluctuations in the market, this strategy can represent the basic return rate of an investment. Since different market have different nature character (for example, return rate, risk and so on) and the return rate of buy and hold strategy can be the basis of judging the performance of all other trading strategies. That means how good a trading strategy performs is determined by for how long and how much it bit the buy and hold strategy.

To give readers a better understanding about how the strategy in this paper performs, all the result conclusion part of algorithm will be represented by a graph. An example graph is as follows:

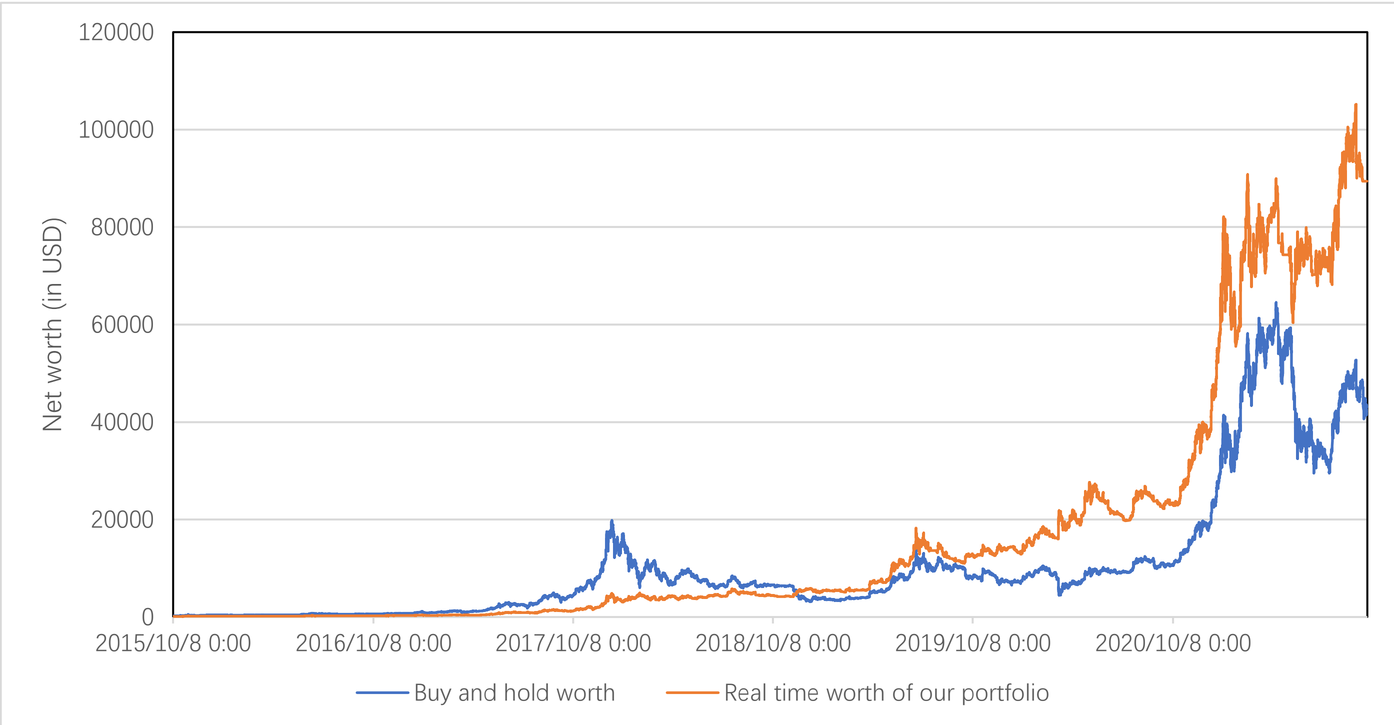

Fig I Sample result graph

This graph demonstrates the usual pattern of result graph. If not specified, the blue in the graph represents the net worth of buy and hold worth of the given equity and the orange line represents the outcomes of the algorithm that the graph is represented. The vertical axis represent the net worth has the unit of USD for all result graph in this report.

3. Trading algorithm

3.1 General view of trading algorithm

Nowadays, most algorithms use a massive amount of historical data to estimate the market trend and predict the future short-term price of the stock market. For example, some algorithms use Bayes' Theorem to determine the estimated direction of the market [4]; other use Benford's law to evaluate the distribution to determine the current trend of the market [5]. They are all wonderful algorithms with solid proof and splendid outcomes in most traditional markets like Stock or Foreign exchange.

However, it's also a common sense that the past does not mean the future; that is especially true for the cryptocurrency market. As history proved, this market may be hugely affected by government policy and investors' confidence in cryptocurrency. Thus, a trading algorithm based on a long time of historical data analysis to predict future price has a high chance of not being able to handle such cases.

As the future is hard to determine, and the past is not that reliable, the analysis of the current market trend becomes the only choice we can make when composing an algorithm that fits cryptocurrency. That thought leads to the idea of analysis of moment of market.

3.2 Entry signal

The entry signal for the algorithm is also the spark point for the whole project. During previous research for this project, an exciting fact shows up; there exists a specific pattern for each growth and fall of price. Whenever the market is in the early stage of growth, the hammer pattern shows up on the candle chart of the stocks. Like the figure below shows:

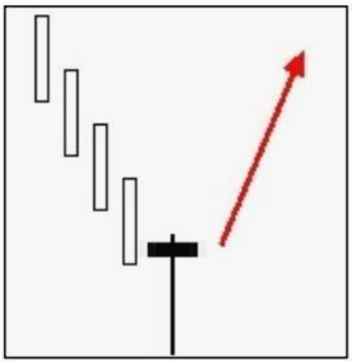

Fig II: Hammer Line example

This graph shows a typical pattern for the Hammer Line entry signal when the market is in a massive move. By pervious analysis, the market has high chance to have a significantlly up as this signal exisits

In this research, to quantify the existence of the hammer line, we judge whether the body of the hammer exists at the upper 70% part of the candle and whether the volume of the trade is smaller than 30% of the whole high–low line. Then this period is judged as a buy-in candle. And it will be the time to enter the market at next hour.

Also, to prevent the massive move of cryptocurrency from sending out a fake signal of growth, in this algorithm, the “certain period” mentioned above should be a balance point between representing the momentum of the market and not missing the chance of making a profit. After trying multiple choices between one hour to serval days, it turns out that choosing 24 hours as the period of judging is a balanced choice (choice can lead to more profit).

Moreover, since the shorting algorithm will be used later in the program, the entry signal for shorting is just the other way round, the reverse of the hammer line. Example as the figure below shows

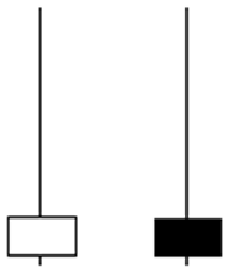

Fig III Reverse Hammer Line

The pattern can be explained by the trading psychology as follows: During the trend of trading, whenever a clear hammer shows up, it means that there is a rural battle between buyers and sellers during the movement, at the beginning of the specific period, the seller takes control of the market, which pull down the market price for a large scale, however, at the end of the given period, the buyer finally wins the battle and push the price of stock up for a large scale. That means, during the hammer period, there was a considerable downtrend followed by a significant uptrend, and the existence of such a signal show that it’s time for us to catch the trend.

3.3 Exit Signal: Stop-loss & expected rate of return per trade

After implementing the entry signal, it is necessary to consider the existing signal. A proper pout signal should be just on the highpoint of the market, and a reasonable stop-loss should prevent the massive loss from a significant fall of the market.

At this point of the program, the expected return rate is the best data from history, 16% per trade.

Since it is unlikely that the user of one trading algorithm can fully trust an algorithm when the market is in a huge wave, and the investors' assets vaporize hugely. Thus, for a good trading algorithm, the stop-loss must be included. In this part of our algorithm, the stop-loss signal is temporarily the lower bottom line for the Hammer Line and tested for the result of the entry candle.

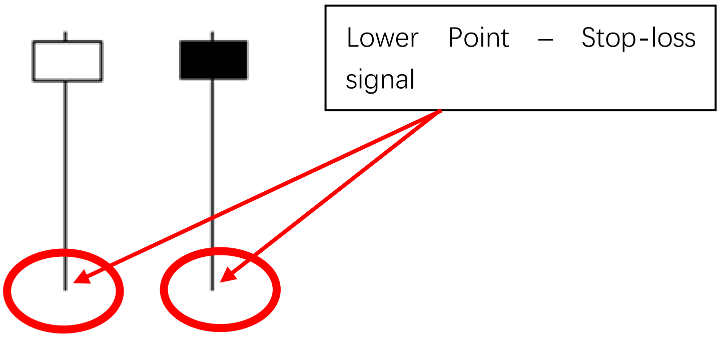

Fig IV Stop-loss Example

Beginning from this point, the first version of the algorithm is finished, and the result are shown as follows (all the price data rounded to nearest dollar):

Year	Begin date	End date	Begin Money this year	End Money this year	CAGR	Draw down rate	Total number of winning trades	Total number of losing trades
2015	2015/10/8	2016/1/1	120	1780	49.96%	39.09%	8	4
2016	2016/1/1	2017/1/1	180	436	142.22%	4.27%	23	14
2017	2017/1/1	2018/1/1	436	7878	1707.36%	29.03%	74	38
2018	2018/1/1	2019/1/1	7878	3910	-50.36%	63.75%	95	83
2019	2019/1/1	2020/1/1	3910	7523	92.39%	46.73%	119	107
2020	2020/1/1	2021/1/1	7523	34274	355.58%	2.12%	140	118
2021	2021/1/1	2021/9/27	34275	46879	36.77%	52.06%	156	130
Total								
2015-2021	2015/10/8	2021/9/27	120	46879	38965.81%	2.12%	156	130

Chart I Result chart – Basic algorithm in BTCUSD

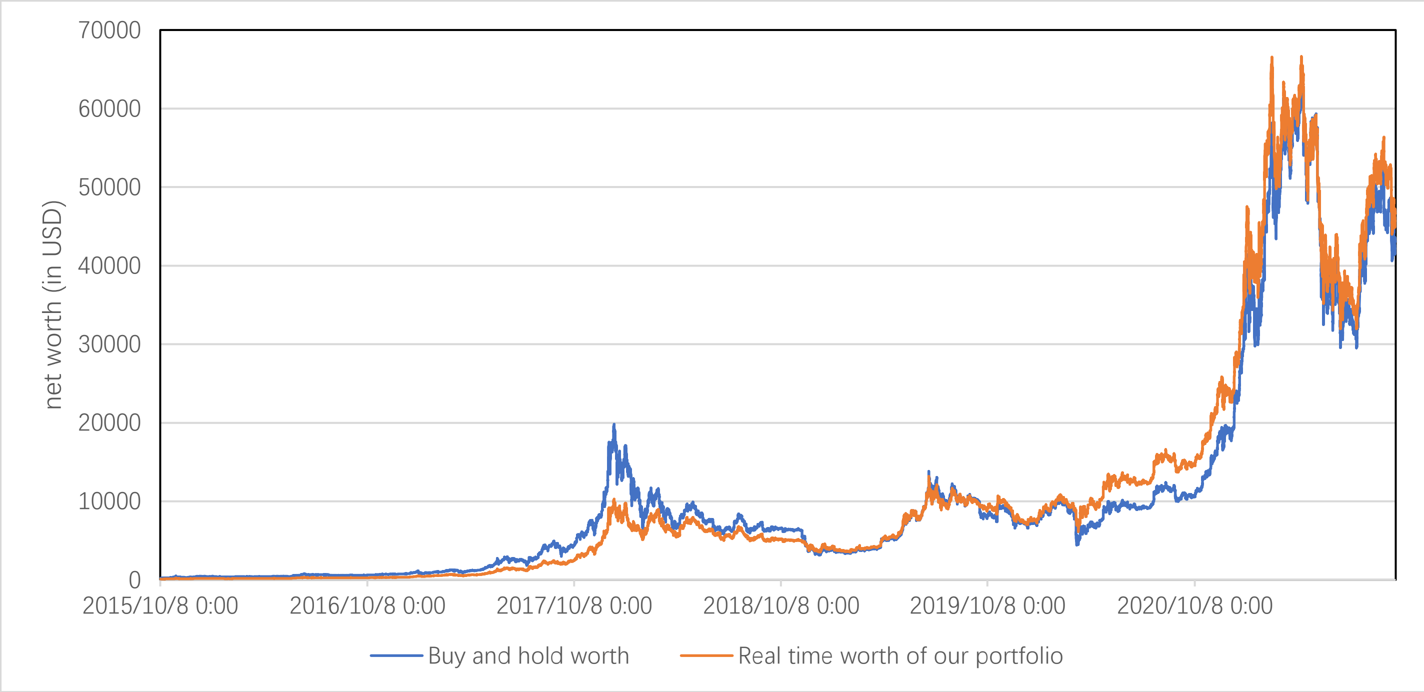

Fig V Performance of basic algorithm in BTCUSD

3.4 Trend judgement

After testing with the previously given entry and out signal, the result shows that our strategy does not perform as well in performance as it should be. In most of the time, the compare to the buy and hold strategy, our algorithm does not win the fight of return rate in the long run.

It is not hard to see that most winning trades happen at the uptrend of the whole market, and most of the losing trades are in the market's downtrend. That makes sense since it is easy for every algorithm to be profitable when the whole market is growing. Thus, a typical thought for preventing loss is to avoid any trading during the downtrend. This naturally led to the idea of judging the trend of the market.

The core of this research is to get the most profitable algorithm by the local value of the market. With the inspiration of covering a curve with multiple straight lines (a usual idea to represent a curve by computer), this research uses the least square method to get the k-value of the local market prices.

Though there are serval ways to estimate the k-value of the current market and multiple intervals of time to choose from to get the k-value, the standard of a "good" estimation is clear: When the market is in an uptrend, the value of our measure should be positive, and if the market is in a downtrend, it should be negative. Four sets of estimation of k-value are used in the research. And their plot is demonstrated below. (Since the whole trend of bitcoin various too much, only present the most represented graph of 2019 in here):

Fig VI k of 3 hours price

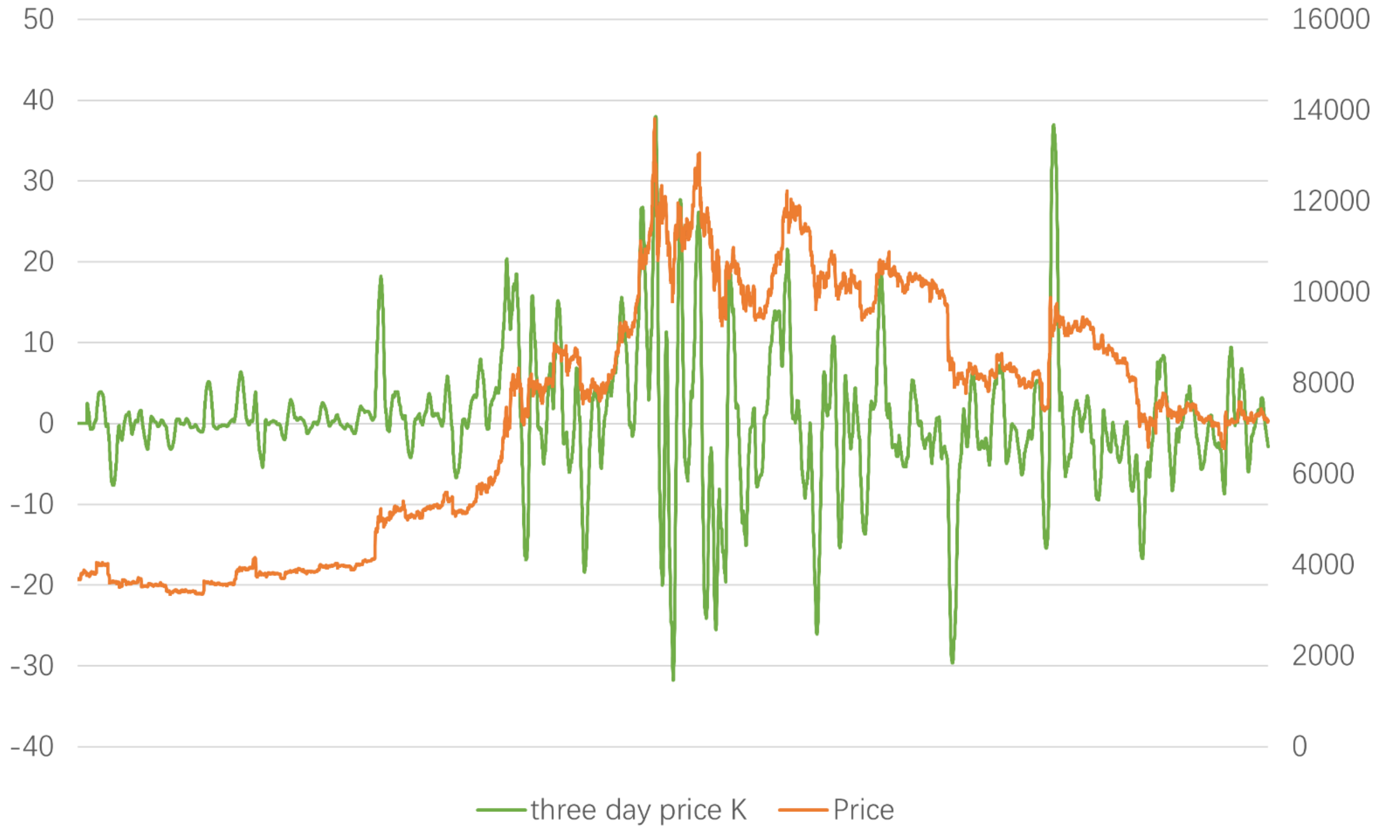

Fig VII k of 3 days price

Fig VIII k of 3 hours 20-Day MA

Fig IX k of 3 days 20-Day MA

The way that these four graphs are formed can be described as follows. Each data point means the estimated value of K at that hour, and it is attained by applying the least square method to the 3 (or 72) price or 20-Day MA before it and getting the k. And notice that in **Fig VI** and **Fig VII**, they use k of 3 hours (or days) price, which means apply the least square method to the past 3 (or 72) hourly price data to get the k.

On the analysis of the diagrams, it is easy to find that for every estimation of k-value using price, the wave of the values varies so huge that any form of stable trading is impossible for such patterns. After analysing two diagrams using MA, it is trivial that the estimation using three days has a significant lag of the market, resulting in a terrible result to a market that waves as huge as cryptocurrency. And all the above estimation has resulted in a relatively low return rate compared to the analysis of 3 hours of 20-Day MA.

Thus, to conclude, the 3-hour estimation interval using 20-Day MA as a data source has the highest chance of leading to the most profitable algorithm. Therefore, the first adjustment that is needed to our algorithm is to avoid any trading when the market analysis signals a downtrend for the current market. By using this estimation of the trend of the market. With an iterating best rate of return per trade (14%) based on the historical data, we can get the resulting form as follows (all the price data rounded to the nearest dollar):

Year	Begin date	End date	Begin Money this year	End Money this year	CAGR	Draw down rate	Total number of winning trades	Total number of losing trades
2015	2015/10/8	2016/1/1	120	180	49.96%	39.09%	8	4
2016	2016/1/1	2017/1/1	180	421	134.44%	4.27%	21	13
2017	2017/1/1	2018/1/1	422	4135	880.15%	29.03%	63	35
2018	2018/1/1	2019/1/1	4135	3272	-20.88%	37.19%	69	53
2019	2019/1/1	2020/1/1	3272	5177	58.23%	54.40%	88	70
2020	2020/1/1	2021/1/1	5177	25965	401.57%	2.12%	108	80
2021	2021/1/1	2021/9/27	25965	33027	27.20%	54.53%	126	96
Total								
2015-2021	2015/10/8	2021/9/27	120	33027	27422.22%	2.12%	126	96

Chart II Result chart – performance with trend judgement in BTCUSD

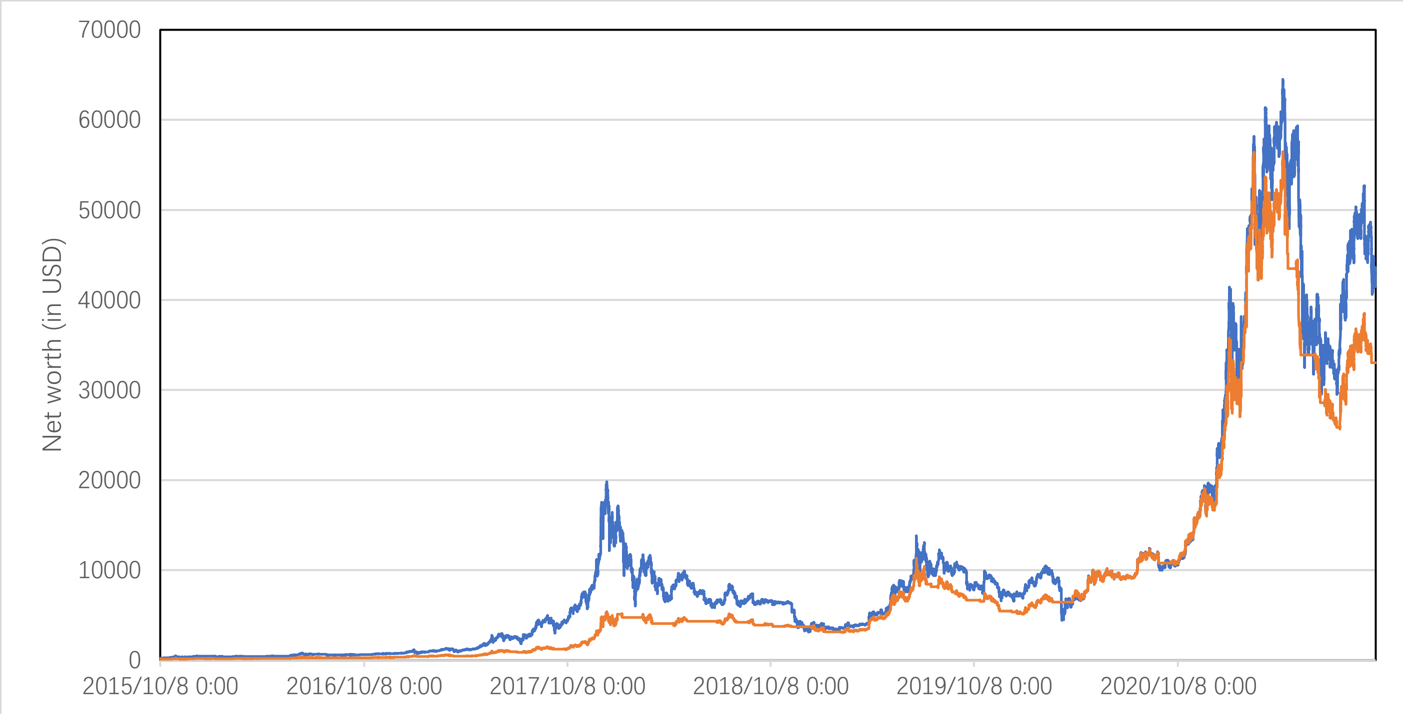

Fig X Performance of algorithm with trend judgement in BTCUSD

From the above analysis result, one conclusion we can draw is that although in this algorithm, the market trend estimation can get rid of the effect of the market draw-down (like the days in 2018, where the whole market experienced a significant downtrend, but the worth of our portfolio is still stable). That means the optimization reaches its target "prevent the huge loss caused by the huge draw-down trend of the market."

3.5 Dynamic return rate

From the resulting diagram above, although optimizing trend judgment can prevent loss caused by the vast market draw-down. Another fact is that with the optimization, the final return rate is not better than the market not mentioned compared to the last algorithm.

After analysing the diagram for why the total return rate is worth the base mark, an apparent result is that with fixed return rate judgment (in the last two results 10% and 14% separately), the net worth of our portfolio cannot follow the growing for the market. That circumstance was most evident in 2017. During that year, the market experienced colossal growth. Yet, the portfolio's worth did not grow that much, and it will be affected by some draw-down of the market, which causes the net worth of our portfolio to lag the market after 2017 significantly.

As for the way to solve this problem. If analysis from the other side of the problem. We can find that the problem for the algorithm is that it can signal entry when the market is in a growth trend. Still, the fixed return rate per trade makes the algorithm sometimes lag behind the best existing point, sometimes exist trade earlier than the growing trend over. It will be unreasonable to have a "general" rule for return rate per trade for the ever-changing market. But one thing for sure is that a good "existing point" is around the point where the market's growth ends to maximize the earning. With that thought, the algorithm can be optimized by getting the real-time tendency of the market, as long as it goes negative (representing the growing trend is over) sell out the bitcoin. By doing this, there should be a better chance for us to catch up with the whole market growth trend.

After applying the return rate optimization, the resulting chart for the algorithm looks as follows (all the price data rounded to the nearest dollar):

Year	Begin date	End date	Begin Money this year	End Money this year	CAGR	Draw down rate	Total number of winning trades	Total number of losing trades
2015	2015/10/8	2016/1/1	120	181.559	51.30%	37.81%	4	1
2016	2016/1/1	2017/1/1	181.559	395.1554	117.65%	4.27%	35	8
2017	2017/1/1	2018/1/1	395.1554	4895.502	1138.88%	38.65%	50	10
2018	2018/1/1	2019/1/1	4895.502	3995.326	-18.39%	30.93%	72	15
2019	2019/1/1	2020/1/1	3995.326	9639.253	141.26%	36.72%	100	20
2020	2020/1/1	2021/1/1	9639.253	41812.99	333.78%	2.12%	129	23
2021	2021/1/1	2021/9/27	41812.99	66633.84	59.36%	14.36%	155	27
Total								
2015-2021	2015/10/8	2021/9/27	120	66633.84	55428.20%	2.12%	155	27

Chart III Result chart - performance with trend judgement and dynamic return in BTCUSD

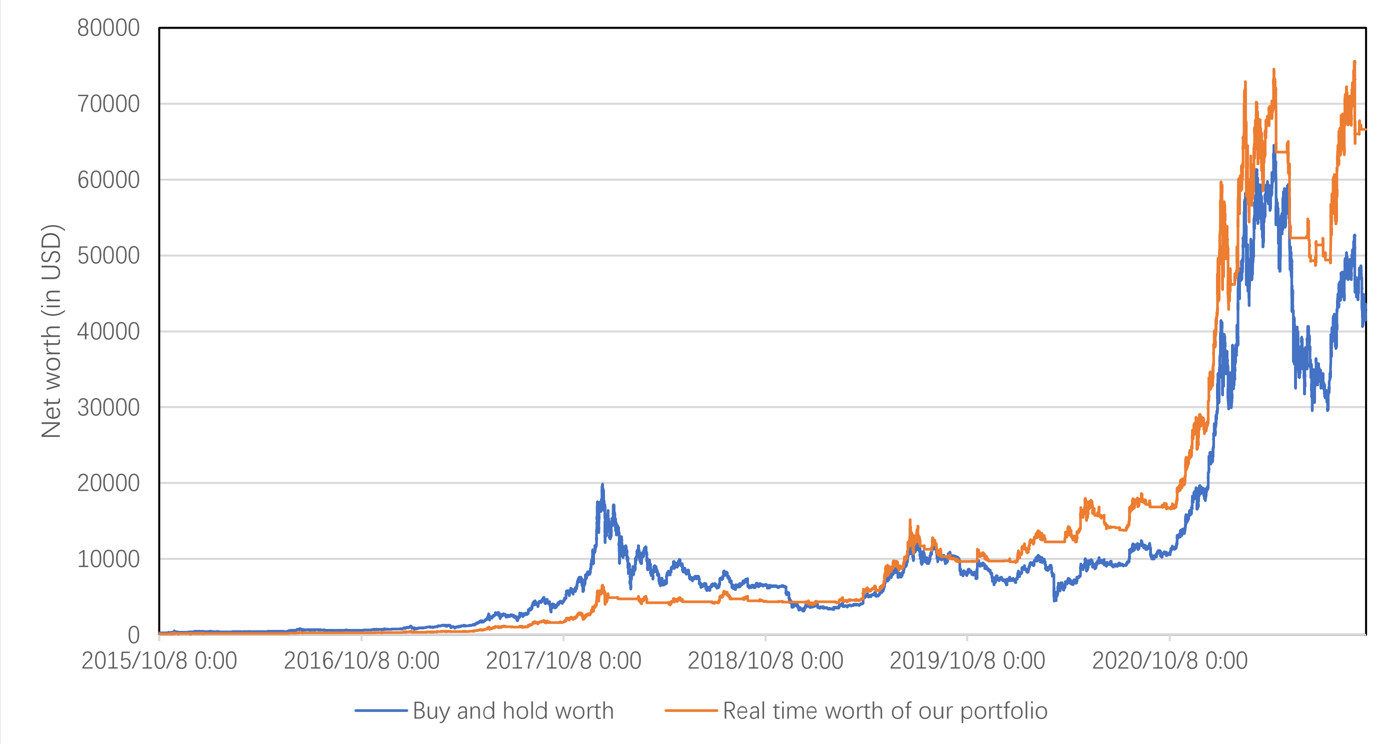

Fig XI Performance of algorithm with trend judgement and dynamic return in BTCUSD

3.6 Shorting strategy

This optimization to our algorithm comes from the idea that when in a short trend, it is easier for a trader to lose money no matter what buy trade he does. However, another trading uses the down of a stock to make profits, which is called short.

Fundamentally, what short do is a trader (denoted as trader A) lends stock from another trader (denoted as trader B) and sells immediately for cash. Then A should return the same number of stocks to B before a specific period. The shorting strategy is a bet against that the price will go down for the stock, and A can buy in stocks at a lower price than the price he got from B.

However, things did not go as smoothly as expected. There is a big chance that it is impossible to do short trading at some point that does short may get a higher return. Thus, when introducing the short algorithm, a judge of successful short rate is also introduced. From an experienced bond manager, the success rate for doing successful short trading is around 50% that will be used to judge the successful return rate.

We can lose money in a market downtrend if we do nothing and wait for the wave to go, although it is not a bad idea. However, by doing so, we may miss a lot of chances to make profits. Thus, why do we not use the downtrend that we judge to do short trade to get more profits?

The entry signal for the shorting is just the reverse of the buy trading, which is the reverse hammer. It can be defined as follows. For one day candle, whenever the body of hammer exists at the point that is lower than 30% part of the candle and whether the volume of the trade is smaller than the 30% of the whole (high – low) line. Then this period is judged as a short in a candle. And it will be the time to enter the market at the next hour (if the successful shorting test is passed).

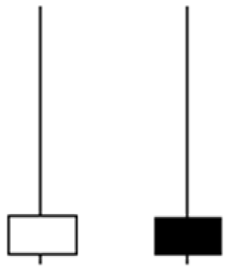

Fig XII Short entry signal - Reverse hammer

Since the risk may occur in making short trade, whenever we enter the short trade, 20% of the money in our account is saved to prevent risk.

Also, the existing signal for short trading is the reverse to those in the buy trade, which is when the trend judgment (three-hour 20 Day MA k) is bigger than 0 (that can conclude the downtrend is over in this trade) buy in the bitcoins and return to the trader that we lend from. The stop loss signal is also reversed to the buy trade, set it to be the upper line of the hammer that triggers the trade.

By doing a shorting successful rate estimation, the data in the research will add a level of randomness. To prevent that from affecting the result, from now on, the return rate we mentioned about this algorithm is the 300-times test results' average to avoid outliers affecting the judgment. And since it is unreasonable to show the every-time randomness, the resulting chart will be formed by the data around the area of the average return rate result.

300-times number is chosen in the balance of the big-number principle for the random variable statics and the computation power for the device. Alternative like 100-times and 1,000-times trials has also been tested. The former cannot represent the outcome of the strategy, and the latter consumes so much computation power, and the result does not increase that much accuracy.

The result shows as follows (all the price data rounded to the nearest dollar):

Year	Begin date	End date	Begin Money this year	End Money this year	CAGR	Draw down rate	Total number of winning trades	Total number of losing trades
2015	2015/10/8	2016/1/1	120	170	41.62%	39.91%	4	4
2016	2016/1/1	2017/1/1	170	381	124.36%	4.27%	43	19
2017	2017/1/1	2018/1/1	381	3619	849.20%	38.65%	60	28
2018	2018/1/1	2019/1/1	3619	5012	38.48%	14.94%	94	42
2019	2019/1/1	2020/1/1	5012	13165	162.68%	39.76%	135	56
2020	2020/1/1	2021/1/1	13165	57556	337.18%	2.12%	172	69
2021	2021/1/1	2021/9/27	57556	89440	55.40%	14.98%	211	78
Total								
2015-2021	2015/10/8	2021/9/27	120	89440	74433.06%	2.12%	211	78

Chart IV Result chart – performance with all optimizations in BTCUSD

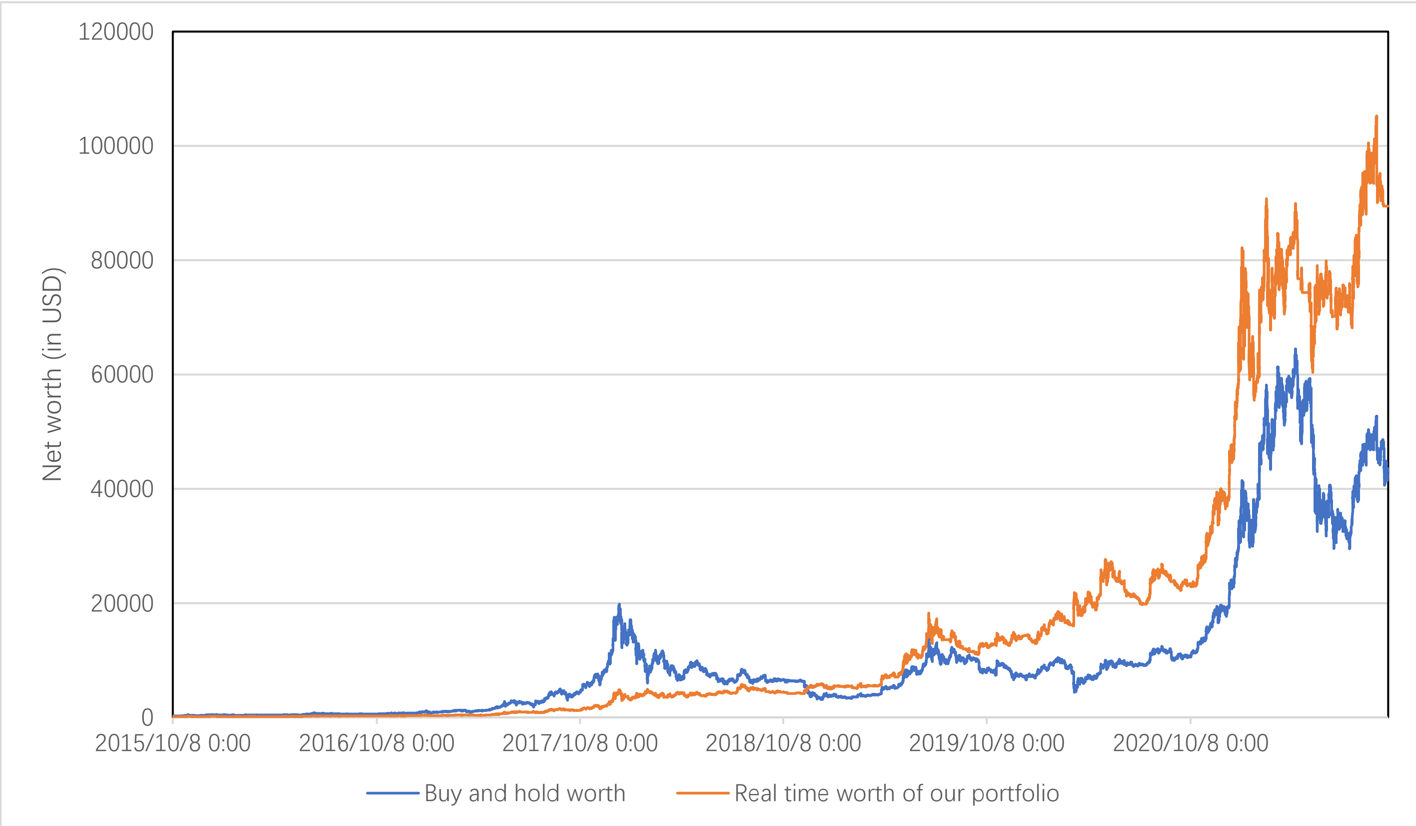

Fig XIII Performance of algorithm with all optimizations in BTCUSD

3.7 Step-by-Step conclusion

This section will draw a whole view of the trading algorithm. The step-by-step procedure of our algorithm can be implemented as follows:

1. Trend judgment: using past 3 hours 20-Day MA to get the current trend of the market
2. Entry:
   a) If in uptrend and buy-in hammer shows up for last 24-hour hammer, execute buy-in.
   b) If in a downtrend and short hammer shows up for the previous 24-hour hammer, perform short.
3. Out:
   a) In buy trading:
      i. If the trend judgement shows the up trend is over ($k < 0$), sell out the bitcoin
      ii. If the price is lower than the bottom line of the hammer leads to buy-in, execute a stop-loss
   b) In short trading:
      i. If the trend judgement shows the downtrend is over ($k > 0$), buy the bitcoin to return.
      ii. If the price is higher than the bottom line of hammer leads to short, execute a stop-loss

3.8 Conclusion for algorithm

The trading algorithm described above has the following benefits compared to the buy-and-hold strategy:

1. It can follow most of the market's uptrend an do buying and downtrend and do shorting and generate a stable income no matter the market trend.
2. It has a meager draw-down rate compared to the market (for the cryptocurrency market, less than 40% of draw-down is a relatively stable rate for the call)
3. It can adequately handle the trade when the market is in huge moves (This can be concluded from the performance of the algorithm in 2020 and 2021)
4. This algorithm does not depend on the historical data to feed and train the program, which can be directly used to do the trading to all other markets, especially to the new market that does not have that long history.

4. Application to other markets

4.1 ETH

Year	Begin date	End date	Begin Money this year	End Money this year	CAGR	Draw down rate	Total number of winning trades	Total number of losing trades
2016	2016/5/9	2017/1/1	5	2.87	-42.57%	71.88%	28	12
2017	2017/1/1	2018/1/1	2.87	252	8678.63%	33.56%	63	17
2018	2018/1/1	2019/1/1	252	1191	372.57%	24.93%	99	23
2019	2019/1/1	2020/1/1	1191	1938	62.67%	33.96%	141	38
2020	2020/1/1	2021/1/1	1938	17244	789.88%	2.71%	185	48
2021	2021/1/1	2021/9/29	17244	38270	121.93%	51.93%	217	59
Total								
2015-2021	2016/5/9	2021/9/29	5	38270	765295.45%	2.71%	217	59

Chart V Result chart - performance in ETHUSD

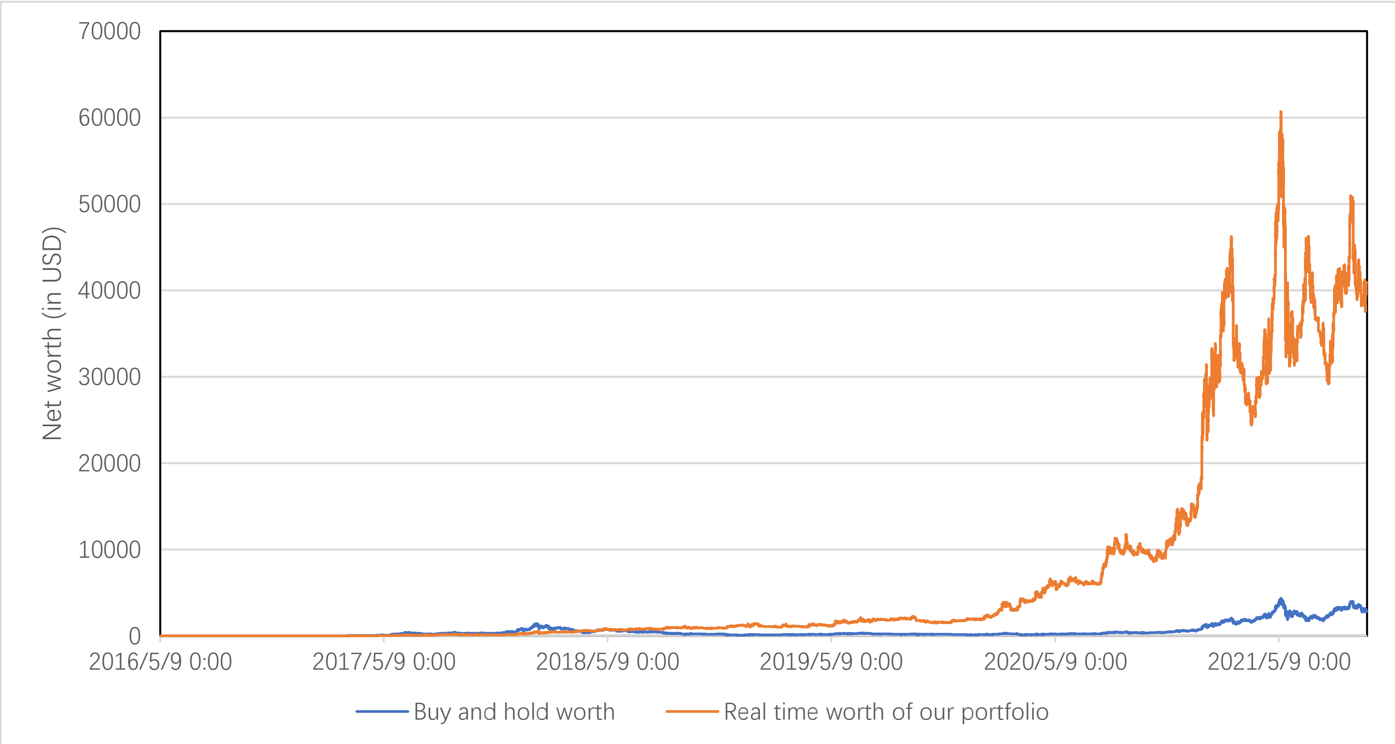

Fig XIV Performance of algorithm in ETHUSD

From the above result, we have seen that in the final period of the graph, the outcome of the trading strategy is ahead considerably more profitable than the buy and hold strategy. Another figure is shown below with different vertical axes to illustrate the market trend better and give the readers a better understanding of how this algorithm performs.

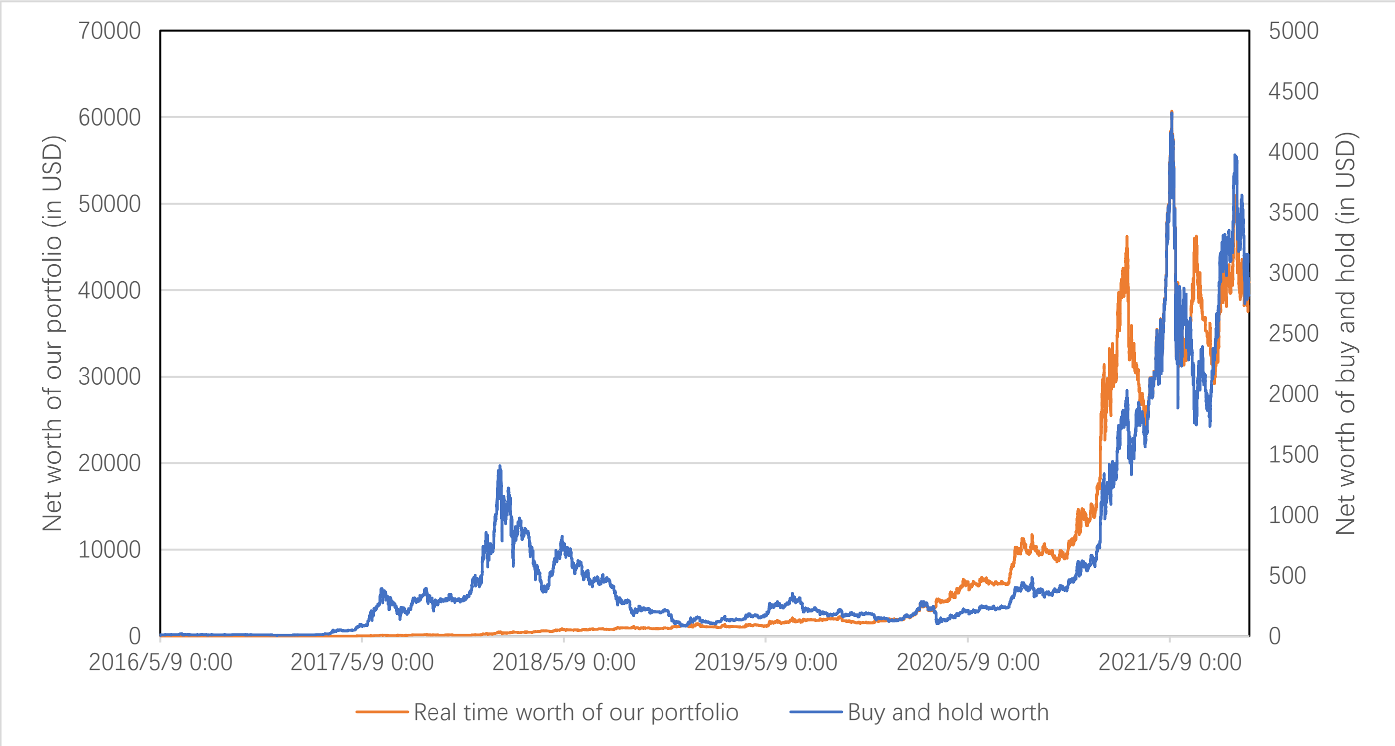

Fig XV Performance of algorithm in ETHUSD in different scale

5. Conclusion

In this report, we use Hammer Line as an entry signal and stop-loss bond optimized by the derivative estimation to the market trend and buy and shorting trading pairs. In the end, we discovered that such a strategy could perform a considerably higher return rate than the market and lower risk relative to the market (the return rate of the market of Bitcoin is around 297 times and the algorithm we used achieves approximately 744 times and the draw-down for the algorithm is controlled under 40% in every year). The key idea for this strategy is to use a short period of historical data to grasp the current trend of the market and perform buy or short trading in the market trend that suits the strategy.

There is still much to be discovered in the trading strategy. For example, what if we apply other signals to replace Hammer Line will give us a more accurate entry signal? Or is the arithmetic moving average the best way for this algorithm? Will exponential moving average perform a better outcome? Is the 3-hour time period the most suitable for this algorithm? Though much has been done in this project, much more is still needed to be discovered.

References:

[1] Think Markets (n.d.) Hammer Line *Price Action Trading Guide*

https://www.thinkmarkets.com/tfxmain/media/img/pdf/Price-Action-Trading-Guide.pdf

[2] Hisotrical data in Crypto data: https://www.cryptodatadownload.com/data/gemini/

[3] Transaction fee for Bitcoin: https://bitflyer.com/en-us/commission

[4] Chandra, R., & He, Y. (2021). Bayesian neural networks for stock price forecasting before and during COVID-19 pandemic. *PloS One*, *16*(7), E0253217.

[5] Weng, C., Shomali, H., & Golden Gate University. (2012). An Application of Benford's Law in Trading Volume and Accounting Numbers: A Comparison between the U.S. and Taiwan Stock Markets.