UNIVERSITY OF STRATHCLYDE

Department of Mathematics and statistics

MM908: Research project

Leveraging the technical competence of a stock for the purpose of trading

by
Rishabh Gupta
Student, MSc Quantitative finance, 21-22

Statement of work in project

The work contained in this project is that of the author and where material from other sources has been incorporated full acknowledgement is made.

Rishabh Gupta

15th August, 2022

Supervised by

Radovan Vojtko (Industrial Supervisor, CEO of Quantpedia.com)

Filip Kalus (Quantitative/IT developer, Quantpedia.com)

Dr. Mohammud Foondun (Academic Supervisor, Course Director, MSc Quantitative finance, University of Strathclyde)

Contents

1. Abstract

As per Ocean Tomo Consulting, in 1975, more than 80% of corporate value reflected in the S&P 500 was tangible assets, while intangible assets comprised less than 20% of market capitalization. By 2020, the game was completely changed, and nearly 90% of the market value of the S&P 500 resided in intangible assets. [14] This research paper is intended to apply the above theory in the algorithm trading world. There exists a trading strategy based on patents (intangible assets) in the database of Quantpedia.com. The below paper is written on checking the robustness of that strategy by backtesting it through changing various market indicators. The backtesting results lead to the conclusion that the stocks with high PTM ratio should be bought and stocks with low PTM ratio should be sold in a group of tertiles when we weigh them through the market value weighted method and rebalance the portfolio yearly.

2. Introduction

2.1 A brief primer on Patents and their relation to financial markets

- A patent is defined as an intellectual property that grants its owner the right to legally prevent others from producing, utilizing or commercializing a design, formula, or other invention for a limited period of time in exchange for disclosing the invention in an enabling manner.[1] The exclusive rights that the patent provides helps a company from interference by other players in the market who wish to do the same. Patents are only issued for a specific amount of time, often 20 years from the filing date. [2]
- "As per the United States Patent and Trademark Office, anyone who creates a novel and valuable technique, machine, manufacturing, or composition of matter, or who discovers a novel and useful improvement thereto, may be given a patent." [2]
- The examples of patented inventions are endless, from standard products like the telephone, dishwasher, and lightbulb to Boeing's Water Harvesting system and Amazon's click system for order processing. [2] [10]

Patents and financial markets

When a firm's market value rises or falls, its stock price fluctuations reflect the factoring in of newly available information which affect investor's expectations of how the stock will perform. [3] Apart from the usual information like merger or acquisition, this new information can be in the form of patents granted. The idea of using patents as a factor for investment is not new. **Ocean Tomo**, **a patent consultancy, offers the OT300, a US equity index, which represents a portfolio of 300 companies that own the most valuable patents relative to their book value. It claims to outperform the S&P 500 by 1620 basis points since its inception till November 2015 by choosing firms according to their patenting behavior and quality**. [4]

Many research papers suggest that corporate innovations act as a growth booster for the stock returns, firm's market value and its capital structure. In many academic research papers, patent data has been used for various types of financial analysis, such as:

- In the paper: Steinbusch, P and Vodegel, D, May 2015, "Influence of the number of patents on stock performance", regression analysis has been used for analyzing the impact of number of patents on IBM's stock performance.

- The research paper by Peter Neuhäusler et al (2011) analyzed the firm's technology and its patenting on the firm's market value. It computed **Tobin's q ratio** (Total market value of the firm/Total asset value of the firm) and concluded that the number of patent citations and the family size (number of patent office's/countries at which a patent has been applied) of patents significantly positively influence the market value. [5]
- A negative relation has been proved between the number of patents (a large sample of US firms) and stock price crash risk in a study by Hamdi Ben Nasr et al (2021). [6] The study refers to **"Stock price crash risk"** as likelihood of experiencing extremely negative firm-specific returns.

This paper tries to shed light on patents differently. It is mainly concerned with noticing the stock price behavior following the days the patent is granted and then making a trade.

2.2 Motivation for improving the trading strategy

The development of Patent to market (PTM) ratio

Jiaping Qiu, Kevin Tseng and Chao Zhang came up with a measure "PTM ratio" to compute market value of a firm attributable to its patents, in their research paper "Patent to market premium". The authors are of the opinion that a hedging portfolio generates a monthly equity return of 71 basis points when we take a long position in a portfolio of tickers with high PTM ratio and short position in a portfolio of tickers with low PTM ratio. **Let's observe what the research paper tells us about the procedures and findings about PTM ratios.**

- PTM represents around 11% to 20% of market value of a firm.
- The researchers computed **correlation of PTM ratio with firm's characteristics (Explanatory variables)** like RDME (Research and development expenses to market value of equity), Market capitalization, innovative efficiency (measured based on patent citations as in Hirshleifer, Hsu, and Li, 2013), asset growth, etc.
- It was found that firms with higher PTM ratio have higher RDME and higher innovative efficiency.
- To prove that PTM ratio captures only distinct information from the equity return explanatory variables (mentioned above), the researchers used Fama-MacBeth cross sectional regression.
- For investigating asset pricing implications of a patent, the researchers used the Generalized method of moment test and it was proved that **PTM ratio is a priced risk factor**, meaning that it is a factor that affects price risk of a stock (risk that the value of a security will decrease). [23]

- They researched further to find sources of premium associated with the PTM ratio. They applied ***real options theory*** *(It states that a firm will not exercise its real option when the net present value of delaying is greater than exercising, Dixit and Pindyck, 1994)* to patents and found that firms with high PTM ratios are more flexible in delaying exercising of their option and are more robust to economic downturns.
- Both the value weighted and equal weighted portfolios show that portfolios with high PTM ratio generate excess returns.

The application of above findings by Quantpedia

- The work "Patent to Market equity factor" by Quantpedia.com studied the findings of the above research paper by Jiaping Qiu et al (2018) [7] and designed a trading strategy based on the PTM ratio.
- The strategy has applied a 2-day window for recording the stock market reaction around the patent granted date as per Kogan et al (2017) [21] and has used an inventory method of calculating depreciated cumulative market value of patent as per Andrea L. Eisfeldt et al (2013). [22]
- Their findings from backtesting have shown a Sharpe ratio of 0.16 and an indicative performance of 5.91%.
- ***The primary motivation behind improving is to rigorously backtest the strategy further by trying various other combinations of the investment universe.***

Further improvement in the strategy

Backtesting is the process of simulating a trading algorithm on historical data. By running a backtest, you can measure how the algorithm would have performed in the past in different market conditions, especially during catastrophic events like the credit and covid crisis. While past performance is not a guarantee of future results, an algorithm with a proven track record can give investors more confidence when deploying on live trading than an algorithm that hasn't worked favorably in the past. [8] We will deploy the backtest:

- On the historical financial data for the period "1st January 2005 to 31st May 2022",
- Using the latest patent data covered till May 2022,
- With new inputs in Scenario analysis like change in asset allocation weighting method, change in stock market reaction period around the patent filing date and change in rebalancing period,
- On the data which will provide us an opportunity to cover the new crisis period of the Ukraine Russian war, that adds a worst case scenario for stocks and best case scenario for commodities like Oil and Gas.

We will then observe the equity curves of the above backtests and make suitable conclusions about the robustness of the strategy.

2.3 Trading strategy in brief

1. Before making any trade, the strategy algorithm goes through a back computation procedure which is as follows:
   - The stock market reaction for all the stocks (after two days the patent is granted) is computed by computing the change in market capitalization (over SPY return). We call this the **market value** of a patent.
   - The Cumulative market value of patent over the previous years is then computed for each stock.
   - The market capitalization for each stock is then recorded at the end of the year.
   - Patent to market(PTM) ratio is computed by dividing the Cumulative market value by Market capitalization.
   - We then sort all the 166 stocks into deciles based on the PTM ratio.
2. The stocks which fall in the highest decile are bought and stocks which fall in the lowest decile are short sold.
3. The weights for long and short are decided through a value weighting asset allocation approach.
4. At the end of next year, the same process is repeated and both the long and short portfolios are rebalanced.

2.4 Introduction to Quantpedia and Quantconnect

Quantpedia.com is a database of trading strategies made from various sources like financial journals, academic papers, universities, conferences, etc. Each year, the team goes through thousands of research papers and extracts the trading rules from some selected papers in plain language. The rules are then backtested on **Quantconnect's** coding environment, and then strategy is finalized and updated on the database.

- The website has three types of screeners which help clients to navigate strategies based on academic research, the performance of the strategy and other criteria such as region, types of instruments traded, period of rebalancing, etc.
- It also provides a **portfolio management tool** to build a custom portfolio of the trading strategies and review their combined effect through the equity curve and drawdown graph.
- Finally, it has an incredible **portfolio analysis section** where almost every type of financial report and graph can be automatically generated for the custom portfolio made earlier. For example, risk

management metrics like VaR percentiles and EWMA percentiles, Portfolio construction reports like efficient frontier and portfolio risk parity curves, etc. [9]

Quantconnect.com [8]

It is an algorithmic trading browser-based platform that lets anyone design, test and execute trading strategies. It provides the financial engineers with market data and cluster computers to build and backtest trading strategies across multiple markets, including equities, derivatives and cryptocurrencies.

A brief primer about the platform and its functions

- **Business model -** QuantConnect provides free backtesting and allows co-located servers to run live trading algorithms for a small fee.
- **Data providers –Algoseek** provides all the historical data for US equities (listed/delisted) and index options to Quantconnect for backtesting. Similarly, many other companies are providing other pricing and sentiment data to Quantconnect, such as data related to Corporate buybacks provided by **Smart insider** and crypto price data provided by **Kraken and Binance**.
- **Backtesting python environment -** At the start of the algorithm, we need to set the following mandatory variables:
  - Start date of backtest.
  - End date of backtest
  - Cash at the start date of backtest
- Along with the above variables, essential variables to be used as **inputs relevant to the trading strategy** are set. For example, in our case, the following variables are set: -
  - Number of days around the patent filing date
  - Number of months to be covered for taking cumulative period for computing market value of the patent
  - Portfolio rebalancing period in months
  - Method of sorting the portfolio
  - Asset allocation weighting approach
- Some of the initial variables that are to be mandatorily set up are:
  - **Scheduled events -** Scheduled events let us trigger code to run at specific times of the day. For example, we can set a scheduled event like fetching historical data before the market opens or we can schedule rebalancing each week 30 minutes before the market closes.
  - **Add universe -** Two types of universes can be added – Coarse universe and Fine universe.

1. Coarse Universe selection allows us to pick a set of stocks by its volume, price, or whether it has fundamental data. This universe helps to narrow down our universe to liquid assets, or assets which pass a technical indicator filter.
2. Fine fundamental selection is performed on the output of the coarse universe. We can think of this as a 2-stage filter; first, the coarse universe can select all the liquid assets, and then the fine fundamental universe can select those that meet our targets.

- **Universe settings resolution** – Resolution is the period of a data bar. It helps to execute trades and get the equity curve daily.
- **Add equity/Add forex/Add crypto/ Add option** – It lets us add the data manually for US-based equities/other instruments as the case may be. This data is available in all the resolutions such as tick, second, minute, hour, and daily.

After setting the required inputs and deploying the backtest, the algorithm plots the equity curve on a new results page. The page also displays other important information such as daily performance charts, trades, runtime statistics, and logs. Let us have a look at one of our backtest results page:

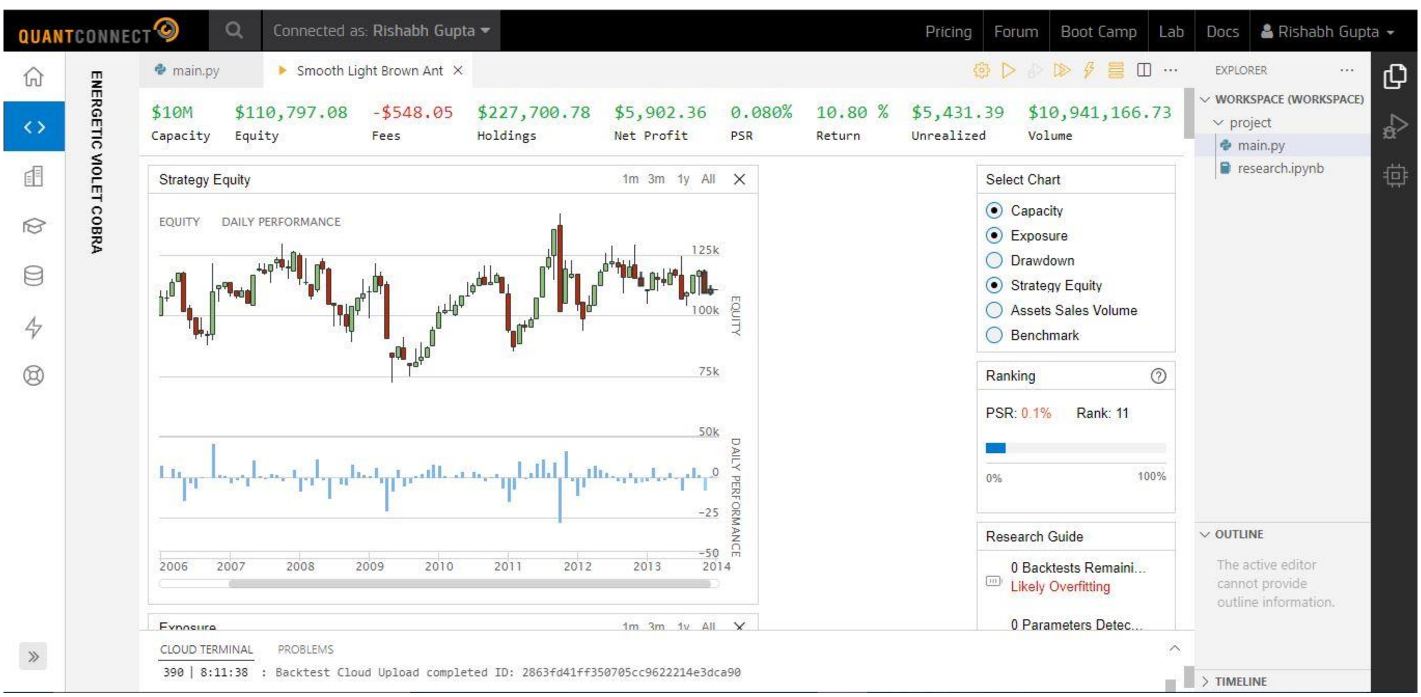


Figure 1: A glimpse of live results page when any algorithm is deployed. The content updates as the algorithm executes.

The banner at the top of the above picture shows the **runtime statistics,** which keep on updating as the backtest executes. The runtime statistics indicators are described as follows:

- **Capacity -** The maximum amount of money an algorithm can trade before its performance declines due to market impact.
- **Equity -** The total portfolio value if all of the holdings were sold at current market rates.
- **Fees** – Fees paid for all trades.
- **Holdings -** The absolute sum of the items in the portfolio.
- **Net profit** – Return (in Dollars) across the trading period. The net profit in the runtime statistics pertains to the last position that was held. [20]
- **Probabilistic sharpe ratio (PSR) -** The probability that the estimated Sharpe ratio of an algorithm is greater than a benchmark (1).
- **Return** – Rate of a return across the trading period. A return of 500% means that if we invest $100 at the start of the backtest period, our equity is converted into $600 at the end of the backtest period.
- **Unrealized -** The amount of profit a portfolio would make if it liquidated all open positions and paid the fees for transacting and crossing the spread.
- **Volume -** The total value of assets traded for all of an algorithm's transactions.

The page also displays a section for Overall statistics as shown below:

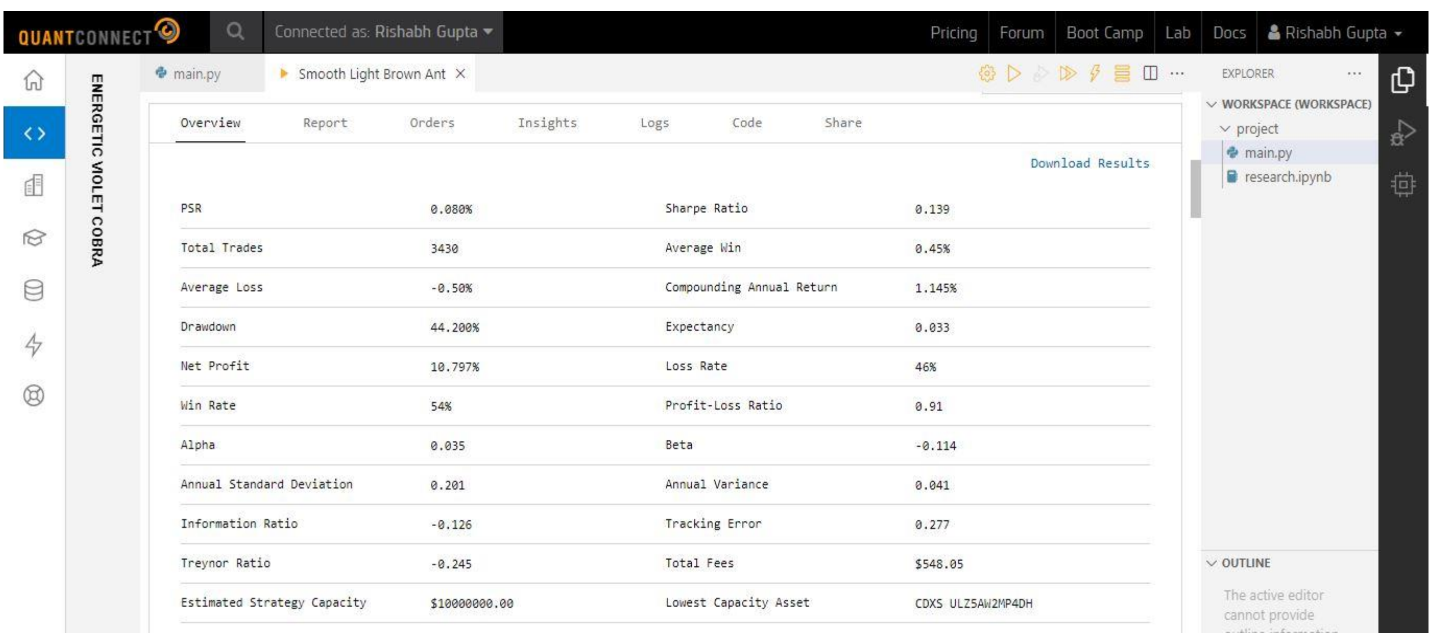


Figure 2: Overview tab in the results page showing performance metrics of a backtest.

The terms related to **overall statistics** are described as follows:

- **Total trades –** Total number of transactions executed (Both long and short).

- **Sharpe ratio** – A measure of risk adjusted return.
- **Average loss -** The average rate of return for unprofitable trades.
- **Average win -** The average rate of return for profitable trades
- **Drawdown -** The largest peak to trough decline in an algorithm's equity curve.
- **Compounding annual return -** The annual percentage return required to grow a portfolio from its starting value to its ending value.
- **Expectancy** – Expected return per trade.
- **Loss rate -** The proportion of unprofitable trades.
- **Win rate -** The proportion of profitable trades.
- **Profit loss ratio -** Average win rate to Average loss rate.
- **Alpha** – Excess return over benchmark.
- **Beta** – A statistical measure of stock volatility in relation to the market.
- **Annual variance** – Dispersion of annual returns from the mean annual return
- **Annual standard deviation** – Square root of annual variance.

- **Tracking error**[18] = $\sqrt{\text{Variance(Portfolio return} - \text{Benchmark return)}}$

- **Information ratio**[17] = $\frac{\text{Portfolio return} - \text{Benchmark return}}{\text{Tracking error}}$

- **Treynor ratio**[19] = $\frac{\text{Portfolio return} - \text{Risk free rate}}{\text{Portfolio beta}}$

3. Methodology

3.1 Data

The analysis was performed on historical patent data scraped from https://companyprofiles.justia.com/companies. **Justia** is an American website specializing in legal information retrieval. [15] There are 166 tickers mentioned on the website and they are US-based most traded liquid stocks. The data is in the form of the count of patents granted to those companies.

3.2 Hypothetical replication of Strategy

Below is the detailed process in MS Excel of what goes through in Quantconnect's beautiful algorithm. We have taken some random tickers and their count of the patent granted on random dates, which is shown below:

	Apple	FB	MSFT	AMAT	ADBE	ALGN	ARAY	Twitter	Coca cola	IBM	Intel	AMD	BASF	Shell	Exxon	Tesla	Pepsico
Year 2019																	
8-Jan-19	56	0	0	0	45	0	0	0	66	57	0	0	37	0	8	10	9
5-Feb-19	0	0	0	0	0	0	0	98	89	0	0	0	0	0	9	1	1
10-Mar-19	6	0	5	27	0	0	6	0	1	0	0	0	6	0	0	0	0
14-Sep-19	0	67	34	0	0	1	8	0	0	0	0	0	0	0	1	0	0
year 2020																	
19-Feb-20	0	0	0	0	0	0	0	1	1	0	0	0	0	0	22	0	0
16-Mar-20	1	0	1	11	0	0	1	0	44	0	0	0	23	0	0	0	0
7-Jul-20	0	12	1	0	0	1	46	0	0	0	0	0	0	0	1	9	5

Table 1: An illustration of actual data used in the backtest algorithm.

1. On the date when any company has been granted a minimum of one patent, the algorithm observes the stock market reaction two days after that date and computes the excess market move. The excess market move is measured by subtracting the change in SPY's price from the change in stock's price.

   SPY (SPDR S&P 500 ETF trust) is one of the most liquid ETFs on the US exchange that tracks the S&P 500 index, which comprises 500 large-cap US stocks. [13] **It is just like one stock representing 500 stocks.** The excess change % is then applied to market capitalization of the stock (as on stock market reaction cutoff date). We call this excess change ($) as the **market value** of the patent. We can see an example of computation of excess market move and then the market value in the below table:

	Market price of the ticker	Market capitalization of the ticker	SPY price
Patent filing date (8th Jan)	**$45**	**$1,000,000**	**$50**
11th Jan (2 days after filing)	**$50**	**$1,200,000**	**$55**
% Change in price	**10.54%**		**9.53%**
Excess change over SPY (10.54% - 9.53%)	**1.005%**		
Excess change over SPY ($) (1,200,000 * 1.005%)	**$12,060**		

Table 2: An illustration of computing the patent's market value using a 2-day window around the patent grant date.

This process is repeatedly done for every ticker for every patent filing date. In Quantconnect, the **closing adjusted price** is taken while computing the excess move. Let us move on to computing the market value of patents of all the tickers as shown in the below tables:

	Apple	FB	MSFT	AMAT	ADBE	ALGN	ARAY	Twitter	Coca cola	IBM	Intel	AMD	BASF	Shell	Exxon	Tesla	Pepsico
2019																	
8-Jan-19	56	0	0	0	45	0	0	0	66	57	0	0	37	0	8	10	9
5-Feb-19	0	0	0	0	0	0	0	98	89	0	0	0	0	0	9	1	1
10-Mar-19	6	0	5	27	0	0	6	0	1	0	0	0	6	0	0	0	0
14-Sep-19	0	67	34	0	0	1	8	0	0	0	0	0	0	0	1	0	0
Total Market value of patents in 2019 ($000's) (Row 1)	500	600	4,500	670	34,354	67,789	4,500	35,445	4,500	45,790	600	670	34,343	566	235	677	678
Total market value in 2019/(Growth rate + Gamma) ($000's) (Row 2)	1,429	1,714	12,857	1,914	98,156	193,683	12,857	101,273	12,857	130,829	1,714	1,914	98,124	1,616	670	1,934	1,937
Market capitalization at 2019 end ($000's)	3.17E+03	2.38E+03	6.43E+04	8.70E+03	1.75E+05	4.04E+05	4.43E+04	8.44E+05	6.77E+04	1.68E+05	2.56E+03	2.18E+03	2.58E+05	3.30E+03	1.34E+03	2.19E+03	2.19E+03
PTM ratio for 2019	0.45	0.72	0.2	0.22	0.56	0.48	0.29	0.12	0.19	0.78	0.67	0.88	0.38	0.49	0.5	0.881	0.885

Table 3: Computation of patent market values for the year 2019 (performed on data from Table 1). Here, Row 1 shows the sum of all the values calculated for every patent filing date as per method mentioned in Table 2. Say, $4,500,000 for MSFT is the total of all the excess change over SPY ($), i.e. for 5 patents as on 10th March and for 34 patents as on 14th September. Row 2 is representing the cumulative market value of every ticker for 2019. Since we are assuming that we are starting our backtest with 2019, we are capitalizing the excess change by growth rate and gamma.

	Apple	FB	MSFT	AMAT	ADBE	ALGN	ARAY	Twitter	Coca cola	IBM	Intel	AMD	BASF	Shell	Exxon	Tesla	Pepsico
2020																	
19-Feb-20	0	0	0	0	0	0	0	1	1	0	0	0	0	0	22	0	0
16-Mar-20	1	0	1	11	0	0	1	0	44	0	0	0	23	0	0	0	0
7-Jul-20	0	12	1	0	0	1	46	0	0	0	0	0	0	0	1	9	5
Total Market value of patents in 2020 ($000's) (Row 1)	550	670	4,950	73,700	37,790	74,568	4,950	3,899	4,950	50,369	66	737	37,778	8	6	4,569	744
Cumulative patent value in 2020 ($000's) (Row 2)	1,764	2,127	15,879	75,327	121,222	239,199	15,879	89,981	15,879	161,573	1,523	2,364	121,183	1,382	575	6,212	2,391
PTM ratio	0.31	0.31	0.31	0.98	0.31	0.31	0.31	0.04	0.31	0.31	0.04	0.31	0.31	0.01	0.01	0.74	0.31

Table 4: Computation of patent market values for the year 2020. Row 1 shows the sum of all the values calculated for every patent filing date as per method mentioned in Table 2. Row 2 is representing the Cumulative market value of every ticker for 2020 after adding the Row 2 values of Table 3 (after applying 15% depreciation). Say, for Apple, 1,764 = 550 + (1,429*0.85).

2. As shown above, we compute this **market value** at every year-end for every ticker **(Because we are taking the Cumulative period as one year)**. After computing the market value for that year, the Cumulative market value is then computed by the following formula:

$$Cumulative\ market\ value = Value\ of\ patent\ for\ current\ year + Value\ of\ patent\ for\ previous\ year * (0.85)$$

But when the **value of patent for previous year is not known**, then the method used is as follows:

$$Cumulative\ market\ value = \frac{Value\ of\ patent\ for\ current\ year}{(Growth\ rate + Gamma)}$$

We are taking a growth rate of 20%, Gamma of 15% and patent depreciation rate of 15% as per Jiaping Qiu et al (2018) [7].

Say for the first ticker "Apple", in 2019, the total market value is coming out to be $500,000 (Table 3). Let's assume we are starting our backtest with 2019 and we don't have market value of patents for the year 2018. So, the cumulative market value for 2019 will be 500,000/(0.20+0.15), i.e. $1,428,571.

3. Patent to market(PTM) ratio is then computed for that year for each ticker by the below formula:

$$PTM\ ratio = \frac{Cumulative\ market\ value}{Market\ capitalization}$$

4. We then sort the tickers into Deciles as per their PTM ratios, as shown below:

Decile	1	2	3	4	5	6	7	8	9	10
Value	0.176	0.212	0.326	0.456	0.49	0.548	0.7	0.82	0.88	0

Table 5: After sorting the PTM ratios of 2019 in ascending order, the decile values are computed. For example, 1^{st} decile of 0.176 is computed by first calculating the position of the required data point, i.e., 1*(n+1)/10 = 1.8. This implies the value of 1.8^{th} data point. So, the first data point is 0.12 and the second one is 0.19, so after interpolation, we get 0.12 + 0.8(0.19-0.12) which comes to 0.176. [24]

- In the above table, we see that the highest decile tickers are the ones which have a PTM ratio above 0.88, and the lowest decile tickers are the ones that have a PTM ratio below 0.176
- Tesla and PepsiCo have PTM ratios above 0.88 for 2019 and Twitter has PTM ratio below 0.176.

5. So we will long PepsiCo and Tesla and short sell Twitter.
6. The computation of Value weighting and quantity to be purchased/sold is shown below. The transaction fee of 0.05 basis point will also be deducted for every transaction.

Particulars	Tesla	Pepsico	Twitter
Position to be taken	Long	Long	Short sell
Market capitalization ($000's)	2,195	2,189	843,940
Value weights based on Market cap	50.07%	49.93%	100.00%
Cash in hand = $ 100,000			
Portfolio value ($)	50,068	49,932	100,000
Current market price at 2019 end	$2.40	$2.00	$6.00
Quantity to be long/short	20,862	24,966	16,667

Table 6: Computation of trading quantity after weighing tickers based on their Market capitalization

7. As we are rebalancing the portfolio every year, the algorithm will repeat the whole process (From computation of PTM ratio to Long/short sell tickers) again after liquidating all the positions at every year end.

3.3 Parameters tested for improvement

As stated above in section 2.4, the inputs will be our parameters which are tested by tuning them in a series of combinations. Apart from the original strategy, the below additional parameters are also tested simultaneously, keeping the original parameters constant.

Parameter	Original strategy	Other backtesting inputs				
Stock market reaction period	2 days	1 day	3 days	5 days	10 days	20 days
Asset allocation weighting method	Market value weighted	Equal weighted	Inverse volatility weighted			
Portfolio sorting method of ranking PTM ratio	Deciles	Vigintiles	Quartiles	Tertiles		
Portfolio Rebalancing frequency	Yearly	Monthly	Quarterly	Bi yearly		
Cumulative period of patents	1 year					

Table 7: Investment parameters to be tested. Cumulative period of patents will remain the same for every backtest, i.e. 12 months.

a) Let us understand the **asset allocation weighting approaches** in detail:

- **Market value weighting approach** – The weights to tickers are assigned in the proportion of their market capitalization to the total market capitalization of the portfolio.
- **Equally weighted approach** – Equal weights are assigned to all the tickers in the portfolio.
- **Inverse volatility weighted approach** – Here, assets are weighted in inverse proportion to the volatility of their returns. More volatile assets receive lower weights and vice versa. We are taking a lookback period of 60 days for the asset's volatility computation.

Volatility is measured by computing the standard deviation of the asset's return over the selected lookback period. An asset's inverse volatility is 1/(Standard deviation of the returns). Each asset is assigned weight equivalent to (Asset's inverse volatility)/ Total inverse volatility of all the assets. [11] For example:

Asset	Microsoft	Twitter
Volatility	5%	1%
Inverse volatility	1/5 = 0.2	1/1 = 1
Weightage	$\frac{0.2}{0.2+1} = 16.67\%$	$\frac{1}{0.2+1} = 83.33\%$

Table 8: An illustration of weighing tickers in a portfolio based on their past volatility.

b) **Ranking of tickers based on PTM ratio** – The tickers are sorted in the following groups as per their PTM ratios:
   - Deciles – 10 equal groups
   - Vigintiles – 20 equal groups
   - Quartiles – 4 equal groups
   - Tertiles – 3 equal groups

c) **Stock market reaction period** – We will observe the stock's movement over time of 1,2,3,5,10, and 20 days after the patent is granted to the company.

d) **Cumulative period** – It is the collective timeframe over which we calculate the market value of the patents. We are measuring the market value for a period of only 12 months.

e) **Rebalancing** – As the value of stocks changes over time, asset allocations can change accordingly. The rebalancing process returns the values of a portfolio's asset allocations to the levels defined by the investment plan. In our strategy, the rebalancing process will liquidate all the current positions after the set period, compute PTM ratios for the cumulative period and then take long and short positions. We will observe the results over the rebalancing periods of 1,6,12, and 24 months. [12]

4. Backtesting processing and results

We have backtested the strategy on two halves of the past data, i.e. 1st Jan 2005 to 31st Dec 2013 and 1st Jan 2013 to 31st May 2022, and also for the complete period from 2005 till 2022. We have kept the starting cash as $1,00,000. The **asset universe used for the backtest** is the daily data of SPY.

4.1 Results Part 1

Let’s observe the results obtained by running the algorithm on two halves of the data. For a quick analysis, we are looking at the main four performance parameters as shown:

	Original strategy		Vigintiles		Quartiles		Tertiles		Equal weight		Volatility weight	
	1st half	2nd half	1st half	2nd half	1st half	2nd half	1st half	2nd half	1st half	2nd half	1st half	2nd half
Sharpe ratio	0.138	0.428	0.155	-0.514	0.092	0.55	0.291	0.552	0.103	0.283	0.176	0.182
Net profit	-10.95%	70.03%	-80.42%	-119.31%	-5.95%	201.33%	55.92%	181.29%	-16.69%	-62.28%	-12.66%	-34.20%
Alpha	0.04	0.195	0.144	-0.48	0.015	0.062	0.039	0.054	0.02	0.122	0.094	0.074
Beta	0.097	0.098	0.109	-0.592	0.093	0.392	0.169	0.383	0.177	0.056	-0.047	0.05

	3 days reaction		5 days reaction		10 days reaction		20 days reaction		Monthly rebalance		Quarterly rebalance		Bi yearly rebalance	
	1st half	2nd half	1st half	2nd half	1st half	2nd half	1st half	2nd half	1st half	2nd half	1st half	2nd half	1st half	2nd half
Sharpe ratio	0.095	0.182	0.005	-0.098	0.316	-0.195	0.306	-0.244	0.318	0.032	0.177	-0.161	-0.144	0.222
Net profit	-42.99%	-74.66%	-83.62%	-93.54%	67.82%	-98.67%	71.77%	-93.64%	183.86%	-31.36%	-45.05%	-66.57%	-77.49%	29.14%
Alpha	0.046	0.111	0.009	-0.044	0.126	-0.12	0.111	-0.104	0.294	-0.005	0.177	-0.054	-0.055	0.059
Beta	-0.068	0.066	-0.095	-0.088	-0.088	0.002	-0.134	0.004	0.014	0.123	-0.12	0.101	0.053	0.088

Table 9: Brief results of 26 backtest operations. The purpose of dividing data is just to observe the performance of the strategy over the two periods separately and not to train and test the strategy by dividing into in-sample and out of sample data.

From the above table, we observe that:

- Net profit and Sharpe ratio are significantly different in both the periods when we are diversifying the portfolio into quartiles.
- Changing the asset allocation method and portfolio rebalancing period is not adding any value in the performance of the strategy.
- When it comes to observing the change in stock market reaction period, the backtest is showing positive profit only in the first half when computing the patent value at the end of 10 days and 20 days (after the patent is granted).
- Overall, there is no specific pattern visible in both the periods so as to make a conclusion.

4.2 Results Part 2

Let’s observe the results obtained by running the algorithm on the complete time period of data.

1. The below table shows the results when we are changing only the **Portfolio sorting method of ranking PTM ratio** and keeping other inputs as in original strategy as well as when we are changing only the **asset allocation weighting method** and keeping other inputs as in original strategy.

	Portfolio sorting method				Asset allocation method		
	Deciles	Vigintiles	Quartiles	Tertiles	Value weighted	Equal weighted	Volatility weighted
PSR	0.01%	0.00%	0.11%	0.54%	0.01%	0.00%	0.00%
Sharpe Ratio	0.288	-0.084	0.423	0.515	0.288	0.173	0.21
Total Trades	252	81	601	775	252	252	250
Average Win	10.38%	6.63%	-2.70%	2.30%	10.38%	8.11%	14.04%
Average Loss	-6.88%	9.46%	-3.45%	-1.76%	-6.88%	-7.49%	-8.05%
Compounding Annual Return	4.93%	0%	9.91%	10.96%	4.93%	-0.53%	1.41%
Drawdown	83.50%	100.10%	61.30%	48.30%	83.50%	89.50%	92.10%
Expectancy	0.357	-0.239	-0.867	0.377	0.357	0.115	0.418
Net Profit	131.44%	-100.15%	418.86%	511.99%	131.44%	-8.77%	27.57%
Loss Rate	46%	55%	39%	40%	46%	46%	48%
Win Rate	54%	45%	61%	60%	54%	54%	52%
Profit-Loss Ratio	1.51	0.7	-0.78	1.3	1.51	1.08	1.75
Alpha	0.097	-0.092	0.076	0.07	0.097	0.051	0.092
Beta	0.127	0.125	0.223	0.275	0.127	0.124	0.004
Annual Standard Deviation	0.372	1.005	0.22	0.177	0.372	0.124	0.438
Annual Variance	0.138	1.009	0.048	0.031	0.138	0.124	0.192
Information Ratio	0.075	-0.148	0.062	0.067	0.075	0.124	0.032
Tracking Error	0.396	1.015	0.25	0.207	0.396	0.124	0.466
Treynor Ratio	0.845	-0.671	0.418	0.333	0.845	0.124	21.494
Total Fees	$201.87	$100.01	$410.17	$320.93	$201.87	0.124	$208.49
Estimated Strategy Capacity	$150,000.00	$53,000.00	$12,000,000.00	$52,000,000.00	$150,000.00	0.124	$20,000.00

Table 10: Overall statistics of 7 backtest operations, where deciles and value weighted results are same as they belong to the original strategy. See Appendix A1.1 for equity curve.

2. The below table shows the results when we are changing only the **stock market reaction period** and keeping other inputs as in original strategy as well as when we are changing only the **frequency of rebalancing the portfolio** and keeping other inputs as in original strategy.

	Stock market reaction period					Rebalancing period			
	2 days	3 days	5 days	10 days	20 days	Yearly	Monthly	Quarterly	Bi yearly
PSR	0.01%	0.00%	0.00%	0.00%	0.00%	0.01%	0.00%	0.00%	0.00%
Sharpe Ratio	0.288	0.121	0.188	-0.048	0.082	0.288	0.25	0.107	0.018
Total Trades	252	256	260	264	241	252	2550	905	466
Average Win	10.38%	8.69%	14.49%	8.50%	16.84%	10.38%	1.65%	3.88%	5.02%
Average Loss	-6.88%	-5.38%	-9.90%	-17.03%	-12.94%	-6.88%	-1.34%	-3.50%	-4.50%
Compounding Annual Return	4.93%	-8.53%	-2.12%	-18.10%	-6.94%	4.93%	7.47%	-6.89%	-6.50%
Drawdown	83.50%	95.20%	95.10%	99.10%	93.60%	83.50%	63.30%	96.80%	91.90%
Expectancy	0.357	0.236	0.221	-0.225	0.13	0.357	0.186	0.024	0.027
Net Profit	131.44%	-78.87%	-31.15%	-96.92%	-71.43%	131.44%	250.66%	-71.18%	-68.98%
Loss Rate	46%	53%	50%	48%	51%	46%	47%	51%	51%
Win Rate	54%	47%	50%	52%	49%	54%	53%	49%	49%
Profit-Loss Ratio	1.51	1.61	1.46	0.5	1.3	1.51	1.23	1.11	1.12
Alpha	0.097	0.064	0.084	-0.024	0.037	0.097	0.167	0.075	0
Beta	0.127	-0.052	0.058	0.012	-0.046	0.127	0.072	0.007	0.073
Annual Standard Deviation	0.372	0.496	0.467	0.478	0.409	0.372	0.691	0.711	0.327
Annual Variance	0.138	0.246	0.218	0.229	0.167	0.138	0.478	0.506	0.107
Information Ratio	0.075	-0.033	0.022	-0.2	-0.1	0.075	0.135	-0.002	-0.2
Tracking Error	0.396	0.523	0.491	0.504	0.441	0.396	0.707	0.729	0.358
Treynor Ratio	0.845	-1.154	1.514	-1.876	-0.728	0.845	2.419	10.885	0.081
Total Fees	$201.87	$151.88	$86.20	$213.32	$288.58	$201.87	$2,085.29	$382.11	$114.42
Estimated Strategy Capacity	$150,000.00	$230,000.00	$120,000.00	$130,000.00	$540,000.00	$150,000.00	$92,000.00	$460,000.00	$150,000.00

Table 11: Overall statistics of 9 backtest operations, where 2 days and yearly rebalancing results are same as they belong to the original strategy. See Appendix A1.2 for equity curve.

Observations:

- Here we see far better performance than the part 1 results. Though the Inverse volatility weighting is giving a low beta and high alpha, but its net profit and Sharpe ratio is significantly less than the original market value weighting. Overall, changing the reaction period and asset allocation weighting method does not seem to add any value to the performance.
- We can also see a higher Net profit and Sharpe ratio when we diversify the portfolio into Tertiles and Quartiles as well as when we rebalance the portfolio monthly.

4.3 Results Part 3

After observing the part 2 results, we narrowed down our approach for backtesting by just changing the portfolio diversification method and then observing the behavior. We investigated on testing the strategy with **replacing Deciles with Quartiles as well as Tertiles in the original strategy** and then trying the previous combinations as below:

Parameter	Original strategy	Other backtesting inputs				
Stock market reaction period	2 days	1 day	3 days	5 days	10 days	20 days
Asset allocation weighting method	Market value weighted	Equal weighted	Inverse volatility weighted			
Portfolio sorting method of ranking PTM ratio	**Quartiles**					
Portfolio Rebalancing frequency	Yearly	Monthly	Quarterly	Bi yearly		
Cumulative period of patents	1 year					

Parameter	Original strategy	Other backtesting inputs				
Stock market reaction period	2 days	1 day	3 days	5 days	10 days	20 days
Asset allocation weighting method	Market value weighted	Equal weighted	Inverse volatility weighted			
Portfolio sorting method of ranking PTM ratio	**Tertiles**					
Portfolio Rebalancing frequency	Yearly	Monthly	Quarterly	Bi yearly		
Cumulative period of patents	1 year					

Table 12: Investment parameters to be tested for round 3 backtest. Cumulative period of patents will remain the same for every backtest, i.e. 12 months.

The below table shows the results when we are changing only the **asset allocation method** and keeping other inputs as in original strategy **(with Quartiles)** as well as when we are changing only the **frequency of rebalancing the portfolio** and keeping other inputs as in original strategy:

	Asset allocation method				Rebalancing period			
	Value weighted	Equal weighted	Volatility weighted		Yearly	Monthly	Quarterly	Bi yearly
PSR	0.11%	0.00%	0.00%		0.11%	0.00%	0.02%	0.00%
Sharpe Ratio	0.423	-0.04	-0.116		0.423	0.007	0.331	0.206
Total Trades	601	599	592		601	5947	2206	1137
Average Win	3.09%	2.80%	3.48%		3.09%	0.45%	1.12%	1.57%
Average Loss	-2.45%	-3.62%	-3.87%		-2.45%	-0.52%	-1.11%	-1.70%
Compounding Annual Return	9.91%	-4.27%	-8.47%		9.91%	-3.11%	7.36%	3.01%
Drawdown	61.30%	91.40%	94.30%		61.30%	83.70%	75.80%	77.90%
Expectancy	0.376	-0.021	-0.08		0.376	0.004	0.184	0.115
Net Profit	418.74%	-53.25%	-78.62%		418.74%	-42.33%	244.40%	67.75%
Loss Rate	39%	45%	52%		39%	46%	41%	42%
Win Rate	61%	55%	48%		61%	54%	59%	58%
Profit-Loss Ratio	1.26	0.77	0.9		1.26	0.87	1.01	0.93
Alpha	0.076	-0.015	-0.034		0.076	-0.018	0.057	0.035
Beta	0.223	0.088	0.062		0.223	0.254	0.299	0.222
Annual Standard Deviation	0.22	0.212	0.252		0.22	0.222	0.242	0.255
Annual Variance	0.048	0.045	0.063		0.048	0.049	0.059	0.065
Information Ratio	0.062	-0.335	-0.365		0.062	-0.305	0.01	-0.089
Tracking Error	0.25	0.257	0.293		0.25	0.249	0.262	0.281
Treynor Ratio	0.418	-0.096	-0.473		0.418	0.006	0.268	0.236
Total Fees	$410.12	$91.83	$60.76		$410.12	$439.85	$645.53	$328.20
Estimated Strategy Capacity	$12,000,000.00	$65,000.00	$79,000.00		$12,000,000.00	$3,300,000.00	$2,700,000.00	$17,000,000.00

Table 13: Overall statistics of 7 backtest operations, where value weighted and yearly rebalancing results are same as they belong to the original strategy. See Appendix A2.1 for equity curve.

The below table shows the results when we are changing only the **stock market reaction method** and keeping other inputs as in the original strategy **(with Quartiles)**. (See Appendix A2.2):

	Stock market reaction period					
	2 days	**1 day**	**3 days**	**5 days**	**10 days**	**20 days**
PSR	0.11%	0.00%	0.00%	0.00%	0.00%	0.00%
Sharpe Ratio	0.423	0.003	-0.233	-0.299	0.093	0.153
Total Trades	601	581	271	533	587	578
Average Win	3.09%	4.90%	10.33%	6.26%	4.56%	5.41%
Average Loss	-2.45%	105.19%	-11.21%	-4.14%	-4.62%	-3.77%
Compounding Annual Return	9.91%	-6.65%	-23.86%	-17.78%	-1.49%	-2.42%
Drawdown	61.30%	95.20%	99.60%	98.60%	90.00%	95.20%
Expectancy	0.376	-0.396	0.012	0.172	0.278	0.275
Net Profit	418.74%	-69.85%	-99.14%	-96.70%	-23.06%	-34.78%
Loss Rate	39%	42%	47%	53%	36%	48%
Win Rate	61%	58%	53%	47%	64%	52%
Profit-Loss Ratio	1.26	0.05	0.92	1.51	0.99	1.43
Alpha	0.076	-0.007	-0.075	-0.083	0.011	0.058
Beta	0.223	0.102	-0.158	-0.062	0.172	0.076
Annual Standard Deviation	0.22	0.31	0.372	0.293	0.261	0.42
Annual Variance	0.048	0.096	0.138	0.086	0.068	0.176
Information Ratio	0.062	-0.224	-0.397	-0.489	-0.183	-0.029
Tracking Error	0.25	0.341	0.414	0.338	0.291	0.444
Treynor Ratio	0.418	0.01	0.551	1.414	0.141	0.846
Total Fees	$410.12	$75.05	$27.31	$57.17	$117.20	$181.55
Estimated Strategy Capacity	$12,000,000.00	$5,100,000.00	$27,000,000.00	$210,000,000.00	$8,600,000.00	$7,900,000.00

Table 14: Overall statistics of 6 backtest operations. See Appendix A2.2 for equity curve.

The below table shows the results when we are changing only the **asset allocation method** and keeping other inputs as in original strategy **(with Tertiles)** as well as when we are changing only the **frequency of rebalancing the portfolio** and keeping other inputs as in original strategy:

	Asset allocation method			Rebalancing period			
	Value weighted	**Equal weighted**	**Volatility weighted**	**Yearly**	**Monthly**	**Quarterly**	**Bi yearly**
PSR	0.54%	0.00%	0.00%	0.54%	0.21%	0.61%	0.33%
Sharpe Ratio	0.515	0.102	0.21	0.515	0.46	0.537	0.484
Total Trades	775	795	783	775	8207	2853	1456
Average Win	2.33%	1.95%	2.00%	2.33%	0.30%	0.54%	0.94%
Average Loss	-1.77%	-2.21%	-1.89%	-1.77%	-0.26%	-0.49%	-0.74%
Compounding Annual Return	10.96%	0.53%	2.84%	10.96%	9.33%	11.45%	10.14%
Drawdown	48.30%	72.30%	51.80%	48.30%	42.70%	50.10%	45.00%
Expectancy	0.383	0.031	0.059	0.383	0.144	0.26	0.257
Net Profit	511.91%	9.66%	62.90%	511.91%	373.14%	560.99%	438.31%
Loss Rate	40%	45%	49%	40%	46%	40%	45%
Win Rate	60%	55%	51%	60%	54%	60%	55%
Profit-Loss Ratio	1.31	0.88	1.06	1.31	1.12	1.09	1.28
Alpha	0.07	0.007	0.015	0.07	0.059	0.069	0.065
Beta	0.274	0.128	0.183	0.274	0.266	0.321	0.274
Annual Standard Deviation	0.177	0.164	0.14	0.177	0.172	0.175	0.178
Annual Variance	0.031	0.027	0.02	0.031	0.03	0.031	0.032
Information Ratio	0.067	-0.285	-0.255	0.067	0.009	0.084	0.041
Tracking Error	0.207	0.214	0.189	0.207	0.204	0.2	0.207
Treynor Ratio	0.333	0.13	0.161	0.333	0.298	0.293	0.313
Total Fees	$320.91	$125.70	$148.13	$320.91	$1,614.79	$826.01	$430.50
Estimated Strategy Capacity	$52,000,000.00	$86,000.00	$120,000.00	$52,000,000.00	$9,300,000.00	$15,000,000.00	$41,000,000.00

Table 15: Overall statistics of 7 backtest operations, where value weighted and yearly rebalancing results are same as they belong to the original strategy. See Appendix A2.3 for equity curve.

The below table shows the results when we are changing only the **stock market reaction period** and keeping other inputs as in original strategy **(with Tertiles)**:

	Stock market reaction period					
	2 days	**1 day**	**3 days**	**5 days**	**10 days**	**20 days**
PSR	0.54%	0.00%	0.02%	0.00%	0.01%	0.11%
Sharpe Ratio	0.515	0.188	0.299	0.038	0.23	0.444
Total Trades	775	773	761	766	757	718
Average Win	2.33%	1.81%	1.74%	2.58%	1.88%	3.30%
Average Loss	-1.77%	-2.13%	-2.11%	-2.62%	-2.18%	-2.44%
Compounding Annual Return	10.96%	2.50%	5.26%	-1.89%	3.53%	10.24%
Drawdown	48.30%	82.10%	60.60%	90.80%	66.80%	69.30%
Expectancy	0.383	0.069	0.131	0.023	0.117	0.39
Net Profit	511.91%	53.73%	144.29%	-28.31%	83.13%	446.76%
Loss Rate	40%	42%	38%	48%	40%	41%
Win Rate	60%	58%	62%	52%	60%	59%
Profit-Loss Ratio	1.31	0.85	0.83	0.99	0.86	1.35
Alpha	0.07	0.025	0.039	-0.004	0.024	0.074
Beta	0.274	0.103	0.165	0.151	0.204	0.228
Annual Standard Deviation	0.177	0.173	0.173	0.204	0.173	0.207
Annual Variance	0.031	0.03	0.03	0.041	0.03	0.043
Information Ratio	0.067	-0.201	-0.119	-0.287	-0.179	0.061
Tracking Error	0.207	0.224	0.217	0.243	0.212	0.238
Treynor Ratio	0.333	0.316	0.314	0.051	0.194	0.404
Total Fees	$320.91	$133.92	$165.48	$184.01	$133.09	$324.76
Estimated Strategy Capacity	$52,000,000.00	$32,000,000.00	$120,000,000.00	$130,000,000.00	$32,000,000.00	$21,000,000.00

Table 16: Overall statistics of 6 backtest operations. See Appendix A2.4 for equity curve.

From the above tables, we observe that:

- Diversifying the portfolio into three groups is giving us higher Sharpe and Net profit ratio than diversifying into four groups.
- Though Tertiles with quarterly rebalancing is giving a higher Net profit ratio than Tertiles with yearly rebalancing, but it will cost more to the investor in terms of transaction fees. Also, the other performance measures like Alpha, Beta, Standard deviation, Treynor ratio, Profit-loss ratio and Sharpe ratio are very much equivalent to Tertiles with yearly rebalancing.

4.4 Conclusion

After observing all the results from the above scenarios we are of the **opinion** that if we

- ❖ compute the patent value after observing the stock's reaction after **2 days** of filing patent,
- ❖ **sort** the portfolio into **three groups** as per the PTM ratio,
- ❖ allocate **weights** to tickers as per their **market value**,
- ❖ and **rebalance** the portfolio **yearly,**

it will provide us a ***more robust and diversified*** strategy than before.

4.5 Historical crisis analysis of the final strategy

Let's compare the performance of strategy with the index (S&P 500) in the financial crisis that occurred in the past years (as visible on Quantpedia.com Portfolio analysis section)

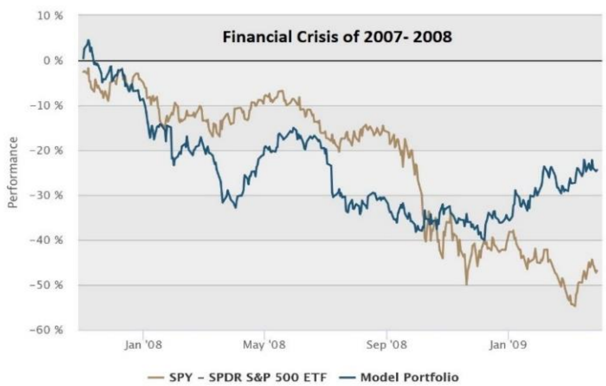


Figure 3: Comparison of the strategy with SPY at the time of subprime mortgage crisis, which was caused by collapse of US housing market and reduction of liquidity in global financial markets.

The portfolio returns are worse than the index returns till October 2008.

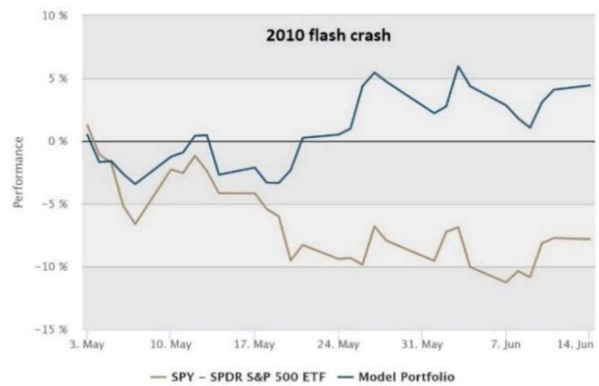


Figure 4: Comparison of the strategy with SPY during a **short term shock of 2010 flash crash**. This event occurred on May 6, 2010 when DJIA fell more than 1,000 points within 10 minutes. [25]

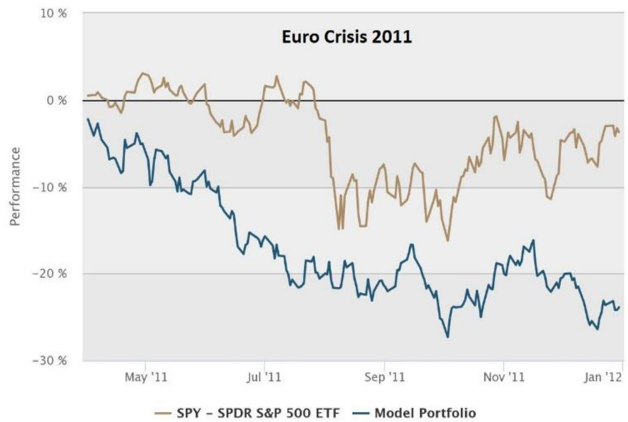


Figure 5: The debt crisis started with collapse of Iceland's banking system in 2008, followed by loss of confidence in European businesses and finally downgrading of Eurozone countries' debts by rating agencies. [26]

We see some opposite behavior of strategy around July 2011. After then, its trend is majorly aligned with the trend of the index.

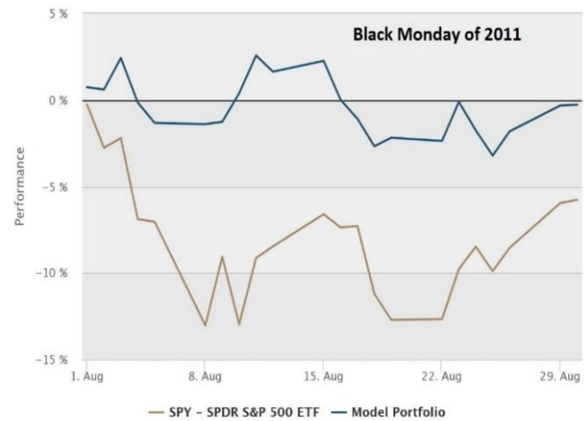


Figure 6: On 8th August, 2011, Dow jones lost 635 points due to downgrade of credit rating of America from AAA to AA+ by S&P. [27]

The returns of the strategy seem to have impacted majorly as they hover around 0% till 15th August and then they become negative and align with the trend of the index.

Figure 7: On 24th August, 2015 Shanghai composite index fell by 8.5% and Dow jones by 1000 points. It happened due to series of events that started with devaluation of Chinese yuan on 11th August, which led to the loss of $5 trillion to global markets. [28]. The strategy is performing worse than the index throughout the crisis period.

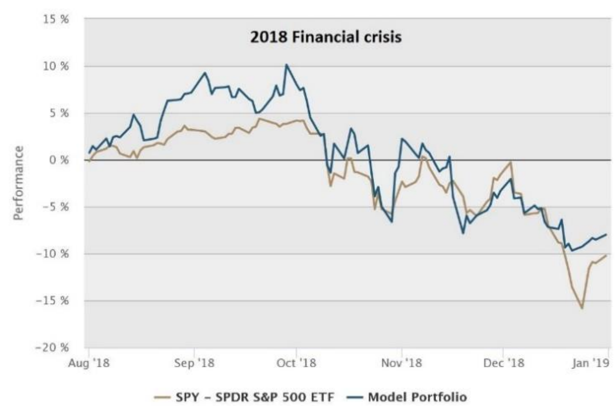


Figure 8: A combination of many factors contributed to crisis in 2018, like US-China trade tensions, a massive fall of 35% in oil prices, bitcoin crash and increase in Greek borrowing costs. Our portfolio is giving better returns till October 2018 and is seen following the path of SPY afterwards.

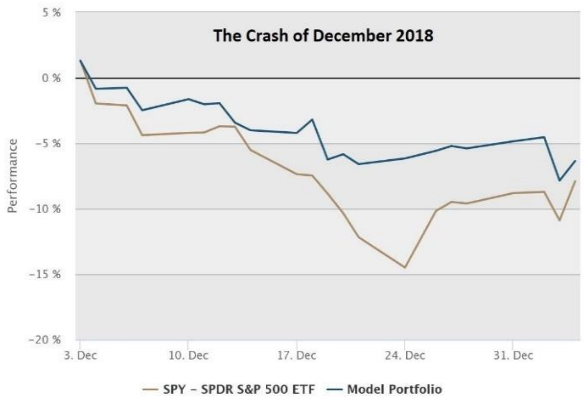


Figure 9: The S&P 500 dropped by more than 9% and Dow jones was down 8.7% in December 2018 as investors feared a central bank was poised to tighten monetary policy. [31] [32] US-China trade war also contributed to the crash. The tickers in our portfolio seem to be also impacted by the 2018 Christmas crash as evidenced by negative returns.

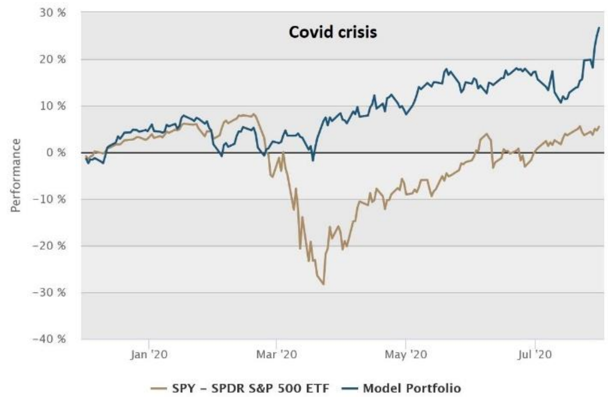


Figure 10: Our strategy is giving consistent positive returns across the whole initial period of covid crisis. Long position in ORCL, AMD and HPQ helped to make the curve flat as they did not plummeted/plunged heavily. Short sell position in DIS also helped the portfolio make positive returns as its price kept on going down consistently.

Correlation of cumulative returns of SPY with the cumulative returns of the strategy in crisis periods:

	Correlation with SPY
2018 financial crisis	93%
December 2018 crash	92%
Euro crisis 2011	69%
Credit crisis 2007-08	63%
Black Monday 2011	52%
Chinese turbulence 2015-16	46%
Covid crisis 2020	7%
Flash crash 2010	-45%

Table 17: Correlation of portfolio cumulative returns with SPY cumulative returns

Observations:

1. If we see the correlation of the strategy with the SPY, they mostly tend to move in the same direction in the majority of the crisis phases of the economy.
2. In events like Covid, Black Monday 2011, flash crash and 2018 crash, the strategy is performing better than the index. It may be due to the nature of the strategy. The strategy has to short sell 1/3rd of the stocks with low PTM ratios, which can result in a good performance when stock prices decline.

3. If we take a small example for comparing our portfolio with the index during the Chinese turbulence in 2015, our portfolio has some holding in KLAC (on 3rd January 2015). KLAC is a semiconductor manufacturing company and had plant in China as on June 2015. [29] KLAC stock dropped by 35% from 5th January, 2015 till 24th August, 2015, which have also contributed to the overall negative performance of the strategy, as visible in Figure 7.

	A	B	C	D	E	F	G	H	I
1	Time	Symb	Price	Quantit	Type	Statu	Value	Tag	
6	2006-01-04T05:00:00Z	KLAC	27.76309787	239	Market On Open	Filled	6635.38039		
34	2007-01-04T05:00:00Z	KLAC	27.3902257	518	Market On Open	Filled	14188.13691		
71	2008-01-03T05:00:00Z	KLAC	25.81998382	-473	Market On Open	Filled	-12212.85235		
147	2009-01-03T05:00:00Z	KLAC	12.67650881	-284	Market On Open	Filled	-3600.128503	Liquidated	
194	2011-01-04T05:00:00Z	KLAC	23.26962114	687	Market On Open	Filled	15986.22972		
279	2012-01-04T05:00:00Z	KLAC	29.30029897	-687	Market On Open	Filled	-20129.30539	Liquidated	
367	2015-01-03T05:00:00Z	KLAC	58.55272127	324	Market On Open	Filled	18971.08169		
415	2016-01-05T05:00:00Z	KLAC	59.3701076	-73	Market On Open	Filled	-4334.017855		
510	2017-01-04T05:00:00Z	KLAC	70.88148144	-251	Market On Open	Filled	-17791.25184	Liquidated	
523	2018-01-03T05:00:00Z	KLAC	97.05693646	131	Market On Open	Filled	12714.45868		
574	2019-01-03T05:00:00Z	KLAC	83.28377602	110	Market On Open	Filled	9161.215362		
620	2020-01-03T05:00:00Z	KLAC	173.0836312	2	Market On Open	Filled	346.1672625		
673	2021-01-05T05:00:00Z	KLAC	255.4683579	-58	Market On Open	Filled	-14817.16476		
729	2022-01-04T05:00:00Z	KLAC	431.7683771	-74	Market On Open	Filled	-31950.85991		
778									
779									

Figure 11: Long position in KLAC as on 3rd January, 2015 as visible in the backtest orders file (downloaded from Quantconnect.com backtest results page)

4. For crisis that happened from 2007 till 2011, the strategy's performance is less correlated with SPY whereas as we go further till 2018, it tends to correlate much highly with SPY. One of the reasons that can have a minor contribution can be – Till 2011, the portfolio contained many stocks that were not part of the top 500 list of S&P 500 and by 2018, some of them were converted into Fortune 500 stocks. [16] For Example, stocks like SANM, VMW, HMC and IVAC have never been in the SPY list. Stocks like TRMB was added to SPY in 2021 and ALGN was added in 2017.
5. At the time of **covid crisis**, AMD was performing well due to high demand for chip based devices. HP was also in the list due to increase in work and schooling from home. [33] [34]

	Time	Symbol	Price	Quantity	Type	Status	Value	Tag
615	2020-01-03T05:00:00Z	EA	104.634373	-4	Market On Open	Filled	-418.537491	
616	2020-01-03T05:00:00Z	NTAP	57.2382131	-51	Market On Open	Filled	-2919.14887	
617	**2020-01-03T05:00:00Z**	**AMD**	**47.94**	**174**	**Market On Open**	**Filled**	**8341.56**	
618	2020-01-03T05:00:00Z	TWTR	31.8	24	Market On Open	Filled	763.2	
619	**2020-01-03T05:00:00Z**	**HPQ**	**18.9813883**	**2384**	**Market On Open**	**Filled**	**45251.62968**	
620	2020-01-03T05:00:00Z	KLAC	173.083631	2	Market On Open	Filled	346.1672625	
653	2020-01-03T05:00:00Z	F	8.9826292	-1693	Market On Open	Filled	-15207.5912	
654	2020-01-03T05:00:00Z	T	24.2021924	1333	Market On Open	Filled	32261.52244	
655	2020-01-03T05:00:00Z	GE	91.5320338	108	Market On Open	Filled	9885.459646	
656	**2020-01-03T05:00:00Z**	**ORCL**	**50.6966318**	**598**	**Market On Open**	**Filled**	**30316.58581**	
657	2020-01-03T05:00:00Z	TSLA	88.002	113	Market On Open	Filled	9944.226	
658	**2020-01-03T05:00:00Z**	**DIS**	**146.51**	**-706**	**Market On Open**	**Filled**	**-103436.06**	
659	2020-01-03T05:00:00Z	GILD	58.0877119	-570	Market On Open	Filled	-33109.9958	
660	2020-01-03T05:00:00Z	EBAY	34.7193129	474	Market On Open	Filled	16456.95433	Liquidated
661	2020-01-03T05:00:00Z	LRCX	282.851798	-244	Market On Open	Filled	-69015.8386	Liquidated
662	2020-01-03T05:00:00Z	JNPR	22.4583796	-590	Market On Open	Filled	-13250.444	Liquidated
663	2020-01-03T05:00:00Z	AMAT	59.2077014	-1487	Market On Open	Filled	-88041.8519	Liquidated
664	2020-01-03T05:00:00Z	SMCI	29.5	31	Market On Open	Filled	914.5	Liquidated
665	2020-01-03T05:00:00Z	GM	36.28733	745	Market On Open	Filled	27034.06088	Liquidated

Figure 12: Long position in AMD, HPQ and ORCL along with short position in DIS as visible in the backtest orders file

5. Strengths and weaknesses of the analysis

The above study reflects the following strengths:

- New inputs and scenarios were introduced to an already designed trading strategy, making the paper innovative and insightful.
- Since the literature information is limited on application of patents, it makes the research stand out from others.
- Experimental approach was followed throughout the study, making the paper more interesting.

The above study reflects the following weaknesses:

- The study is limited by the size of patent data as well as sample size of backtesting period. A larger window of backtest period might have covered more short and long term shock periods. The paper lacks information on patent data as well. It is only covering the most traded 166 stocks in the US, whereas there are more than 2,500 companies listed on NYSE. [30]
- Since all the backtesting is being performed on the Quantconnect platform and not the usual python environment like Colab or Jupyter notebook, the reader might have to go through a lot of reading material to understand the working of Quantconnect.

6. Suggestions for further research

It is suggested that further work should be undertaken in the following areas:

- It would be interesting to stress test the strategy (final strategy chosen) on different hypothetical market scenarios so as to check the resilience of the portfolio.
- It will be important to explore the potential use of machine learning models like K-fold cross validation for backtesting the final strategy so as to observe the performance on both training and testing data and prevent overfitting.
- The same patent strategy can be tested with financial datasets of other geographies such as UK, Europe, Canada, and Japan.

7. References

[1] Wikipedia, https://en.wikipedia.org/wiki/Patent#cite_note-Herman-1

[2] Kenton W, July 2022, “Patent”, https://www.investopedia.com/terms/p/patent.asp

[3] Harrigan, K.R. “Patent value and the Tobin’s q ratio in media services”

[4] Thompson, M.J, March 2016, “Patenting and stock returns”, https://seekingalpha.com/article/3960532-patenting-and-stock-returns

[5] Neuhäusler P, Frietsch R, Schubert T and Blind K, 2011, “Patents and the financial performance of firms – An analysis based on stock market data

[6] Nasr H.B, Bouslimi L and Zhong R, 2021, “Do Patented innovations reduce stock price crash risk?”

[7] Qiu J, Tseng K and Zhang C, 2018, “Patent-to-market premium”

[8] Wikipedia, https://en.wikipedia.org/wiki/QuantConnect
Quantconnect, https://www.quantconnect.com/docs/v2/our-platform/welcome?ref=v1
Quantconnect, https://www.quantconnect.com/docs/v2/writing-algorithms
Quantconnect, https://www.quantconnect.com/docs/v2/our-platform/backtesting/results
Quantconnect, https://www.quantconnect.com/docs/v2/writing-algorithms/universes/key-concepts?ref=v1
Quantconnect, https://www.quantconnect.com/docs/v2/our-platform/backtesting
Quantconnect, https://www.quantconnect.com/tutorials/strategy-library/beta-factors-in-stocks
Greg Bland, August 2021, Quantconnect – A Complete Guide, https://algotrading101.com/learn/quantconnect-guide/

[9] Joe Marwood, December 2014, Quantpedia Review: 240+ Trading strategies, https://stocksoftresearch.com/quantpedia-review/

[10] Shikhar Sahni, How patents impact share prices of companies?, https://www.greyb.com/patents-impact-share-prices/#:~:text=Patent%20citations%20always%20have%20a,it%20brings%20to%20the%20firm.

[11] Composer, https://help.composer.trade/article/26-inverse-volatility-weighting#:~:text=An%20asset's%20inverse%20volatility%20is,inverse%20volatility%20of%20all%20assets

[12] James Chen, July 2022, Rebalancing, https://www.investopedia.com/terms/r/rebalancing.asp

[13] Steven Nickolas, July 2022, SPY ETF (The SPDR S&P 500 ETF Trust)
https://www.investopedia.com/articles/investing/122215/spy-spdr-sp-500-trust-etf.asp

[14] https://www.oceantomo.com/services/patent-indexes/ocean-tomo-300-patent-index/

[15] https://en.wikipedia.org/wiki/Justia

[16] https://en.wikipedia.org/wiki/List_of_S%26P_500_companies
https://gist.github.com/alexchinco/1ef9cb653334839f3c24

[17] Murphy C.B, October 2020, “Information ratio”, https://www.investopedia.com/terms/i/informationratio.asp

[18] CFI team, February 2022, “Tracking error”, https://corporatefinanceinstitute.com/resources/knowledge/trading-investing/tracking-error/

[19] CFI team, January 2022, “Treynor ratio”, https://corporatefinanceinstitute.com/resources/knowledge/finance/treynor-ratio/

[20] Khurana M, December 2017, “Understanding Runtime Statistic Net Profit”, https://www.quantconnect.com/forum/discussion/3141/understanding-runtime-statistic-net-profit/p1

[21] Kogan, Leonid, Papanikolaou D, Seru A, Stoffman N, 2017, “Technological innovation, Resource allocation and Growth”

[22] Eisfeldt A, Papanikolaou D, 2013, “Organization Capital and the Cross-section of expected returns”

[23] Kenton W, June 2021, “Price risk”, https://www.investopedia.com/terms/p/pricerisk.asp

[24] Cuemath, https://www.cuemath.com/data/decile/

[25] Kenton W, April 2022, “Flash Crash”, https://www.investopedia.com/terms/f/flash-crash.asp

[26] Kenton W, October 2021, “European Sovereign Debt crisis”, https://www.investopedia.com/terms/e/european-sovereign-debt-crisis.asp

[27] Wikipedia, February 2022, “Black Monday (2011), https://en.wikipedia.org/wiki/Black_Monday_(2011)

[28] The Economist, August 2015, “The cause and consequences of China’s market crash”, https://www.economist.com/news/2015/08/24/the-causes-and-consequences-of-chinas-market-crash

[29] KLA Tencor corporation Form 10-K, July 2015, https://www.sec.gov/Archives/edgar/data/319201/000031920115000053/klac10k2015.htm

[30] Statista research department, July 2022, “NYSE and Nasdaq monthly number of listed companies comparison 2018-2022, by domicile”, https://www.statista.com/statistics/1277216/nyse-nasdaq-comparison-number-listed-companies/

[31] Lahiff K, December 2021, “Why this month echoes December 2018 when stocks plummeted, according to top strategist”, https://www.cnbc.com/2021/12/08/why-this-month-echoes-december-2018-when-stocks-plummeted.html

[32] Isidore C, December 2018, “2018 was the worst for stocks in 10 years”, https://edition.cnn.com/2018/12/31/investing/dow-stock-market-today/index.html

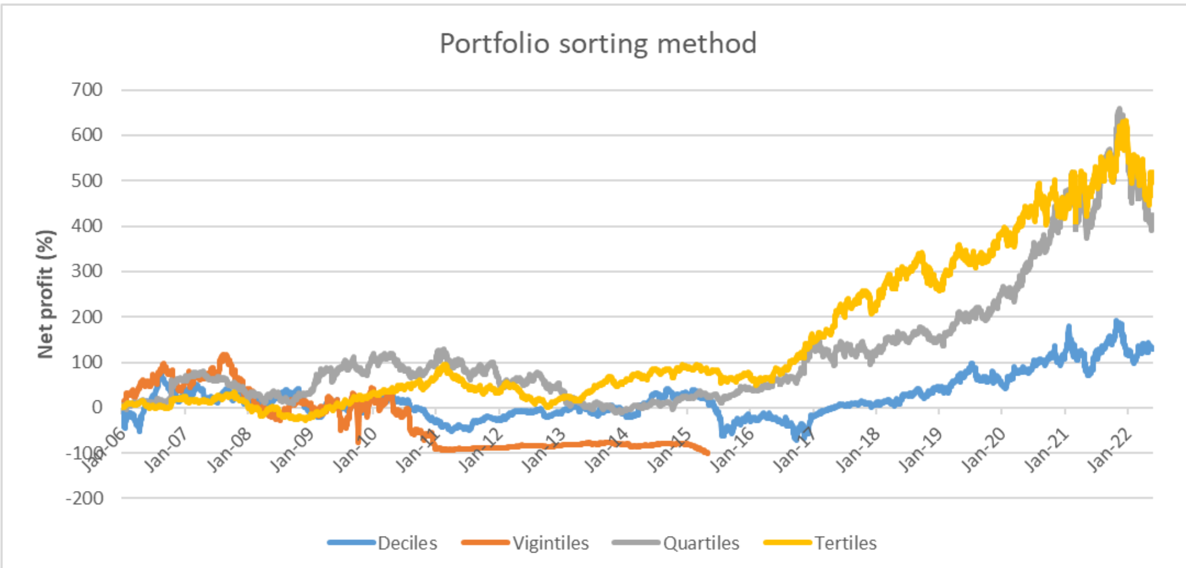
[33] Halloran S, August 2020, “Hewlett Packard gets $1B boost from COVID-19 working and schooling from home”, https://www.foxbusiness.com/markets/hewlett-packard-gets-1b-boost-from-covid-19-working-and-schooling-from-home

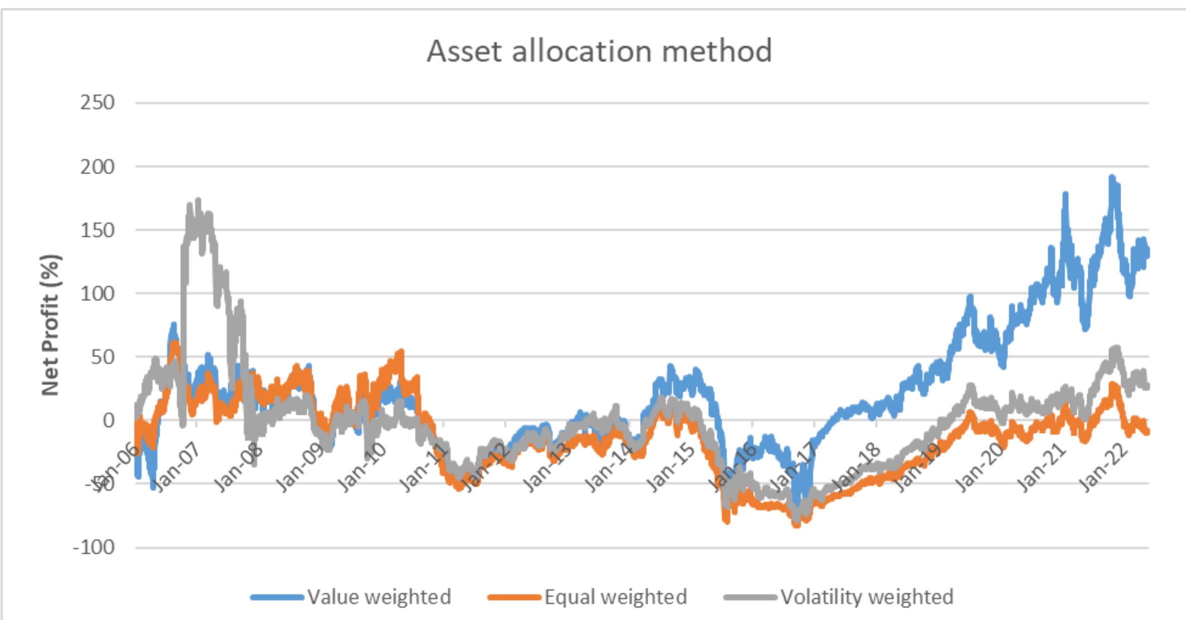
[34] Witkowski W, July 2020, “AMD is weathering coronavirus while biggest rival struggles”, https://www.marketwatch.com/story/amd-is-set-up-to-withstand-coronavirus-in-the-short-term-2020-04-16

8. Appendix

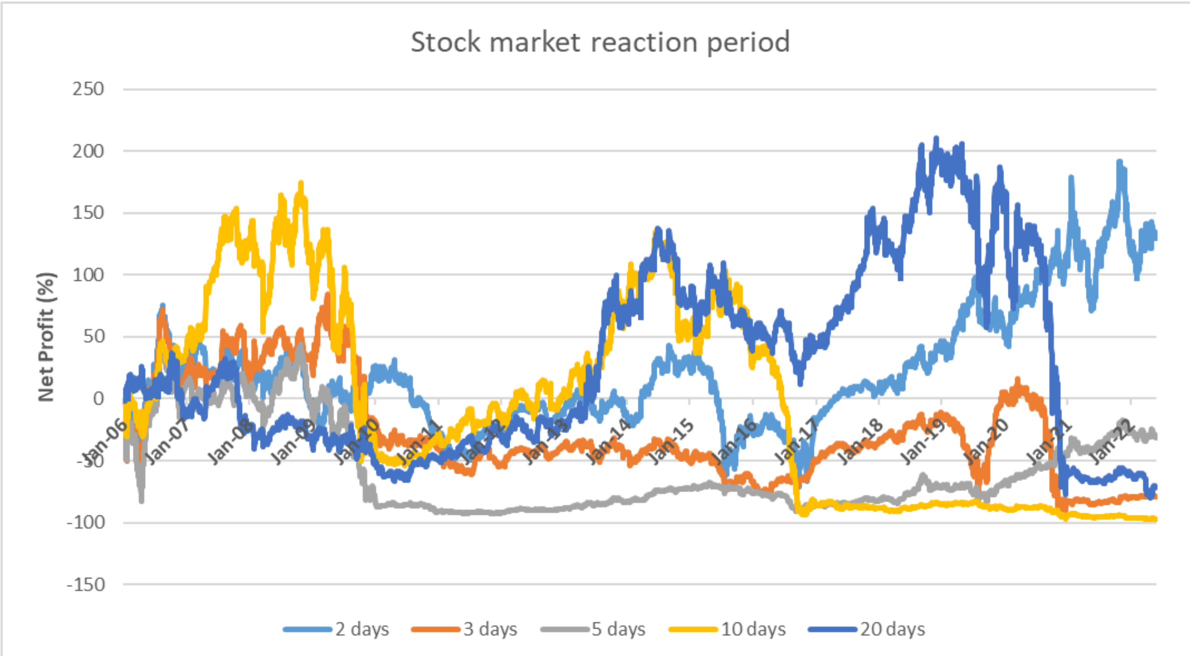
As stated in the methodology, several combinations of strategy parameters were backtested and the combination with overall statistics, especially higher Sharpe and Net profit ratio was selected and further analyzed. The algorithm is shown after the equity curves of all the backtests (with effect from Results - part 2).

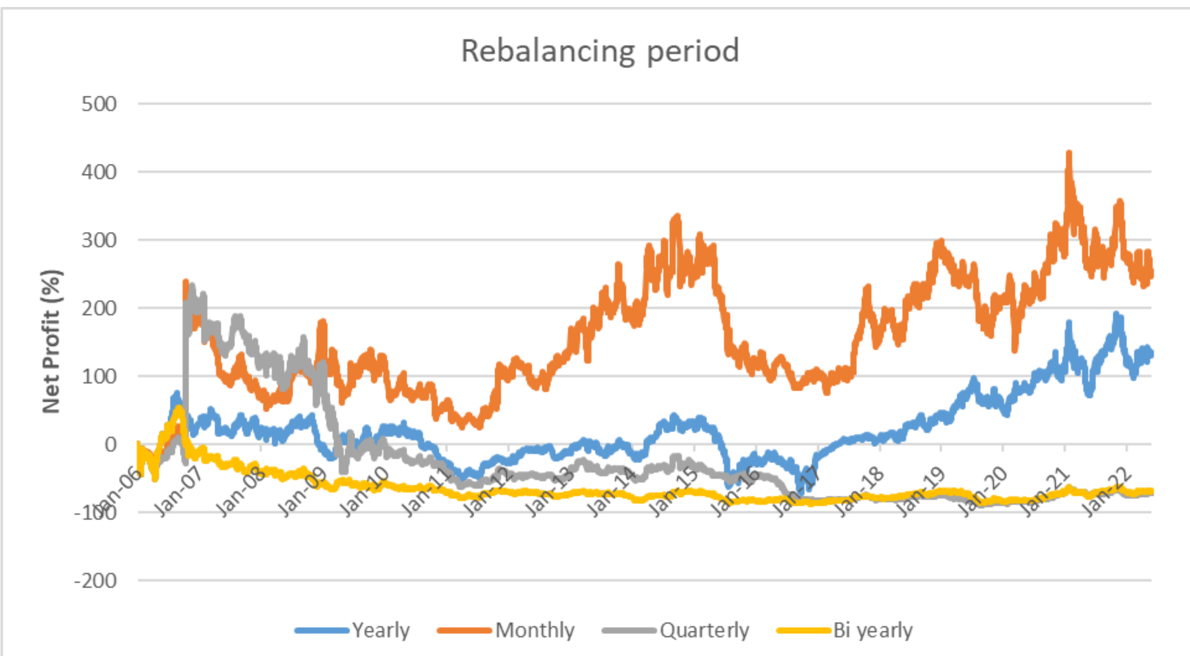
A1.1



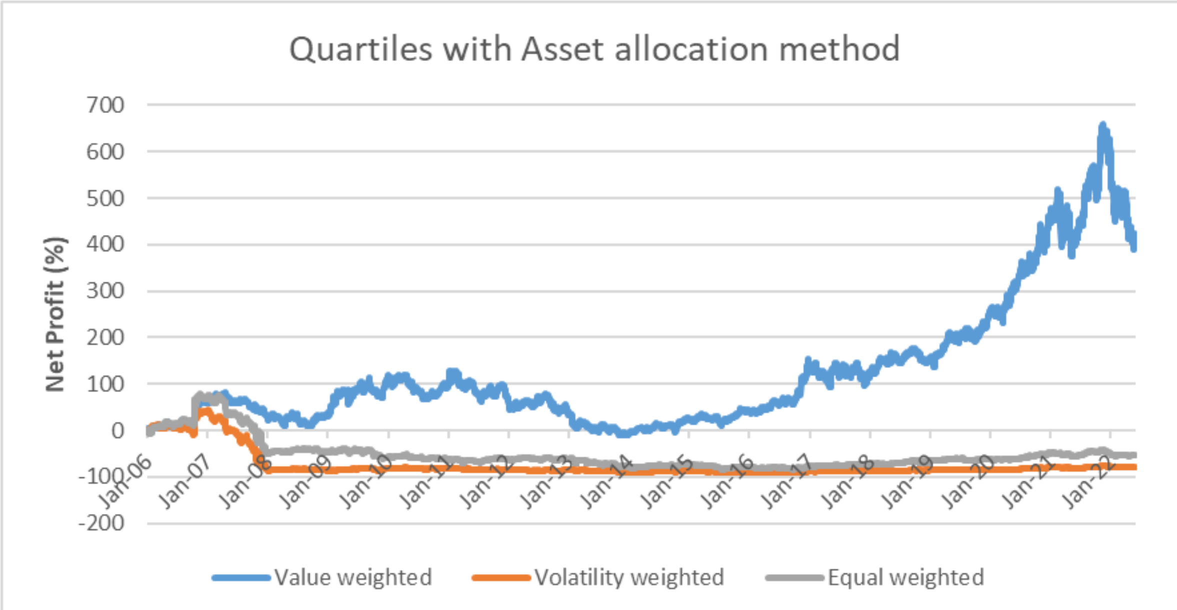


A1.2

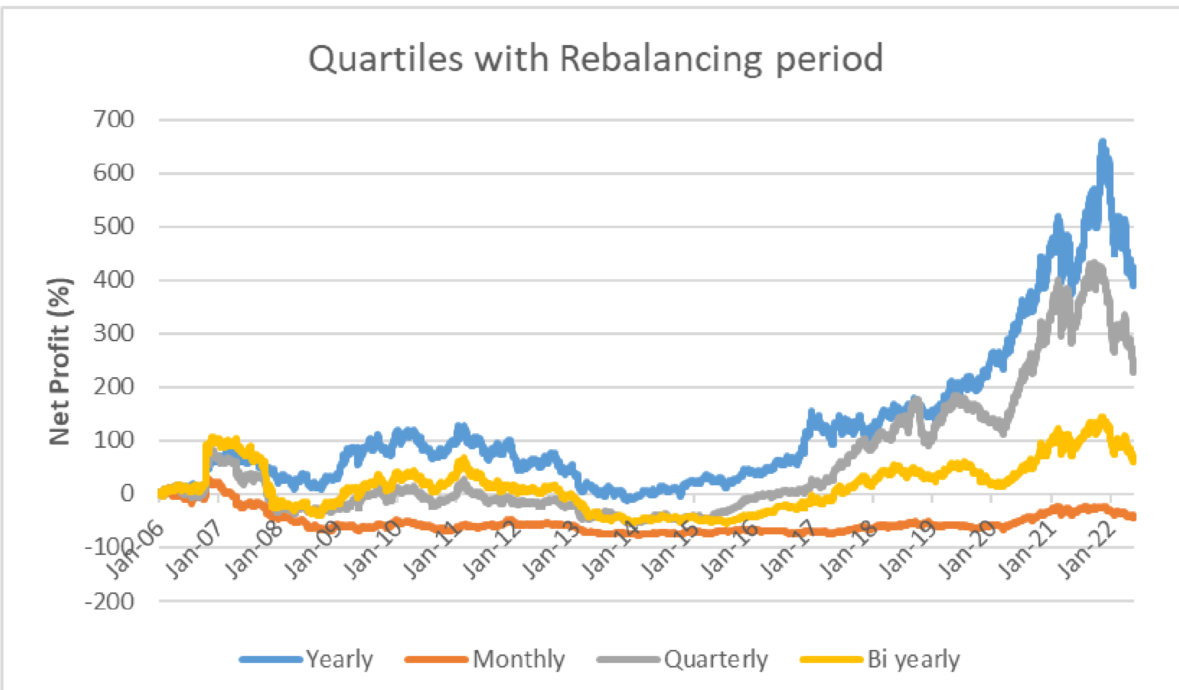

Stock market reaction period
Net Profit (%)
250
200
150
100
50
0
-50
-100
-150
Jan-06
Jan-07
Jan-08
Jan-09
Jan-20
Jan-11
Jan-12
Jan-13
Jan-14
Jan-15
Jan-16
Jan-17
Jan-18
Jan-19
Jan-20
Jan-21
Jan-22
2 days
3 days
5 days
10 days
20 days


Rebalancing period
Net Profit (%)
500
400
300
200
100
0
-100
-200
Jan-06
Jan-07
Jan-08
Jan-09
Jan-10
Jan-11
Jan-12
Jan-13
Jan-14
Jan-15
Jan-16
Jan-17
Jan-18
Jan-19
Jan-20
Jan-21
Jan-22
Yearly
Monthly
Quarterly
Bi yearly

A2.1

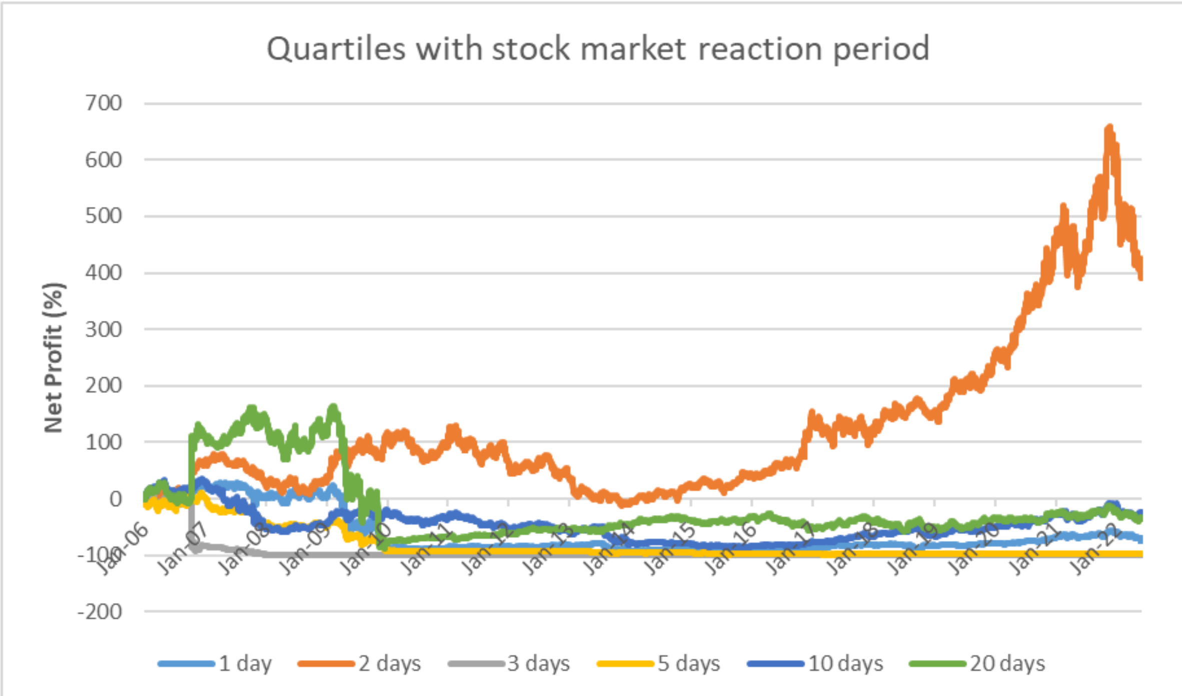


Quartiles with Asset allocation method
Net Profit (%)
700
600
500
400
300
200
100
0
-100
-200
Jan-06
Jan-07
Jan-08
Jan-09
Jan-10
Jan-11
Jan-12
Jan-13
Jan-14
Jan-15
Jan-16
Jan-17
Jan-18
Jan-19
Jan-20
Jan-21
Jan-22
Value weighted
Volatility weighted
Equal weighted

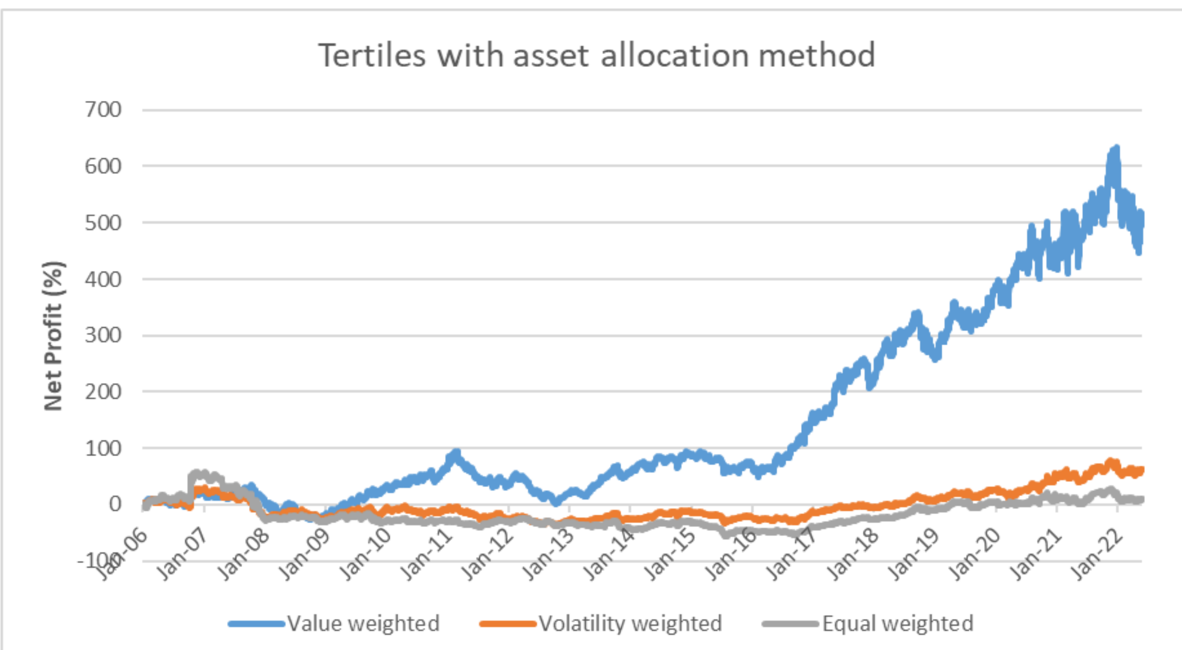


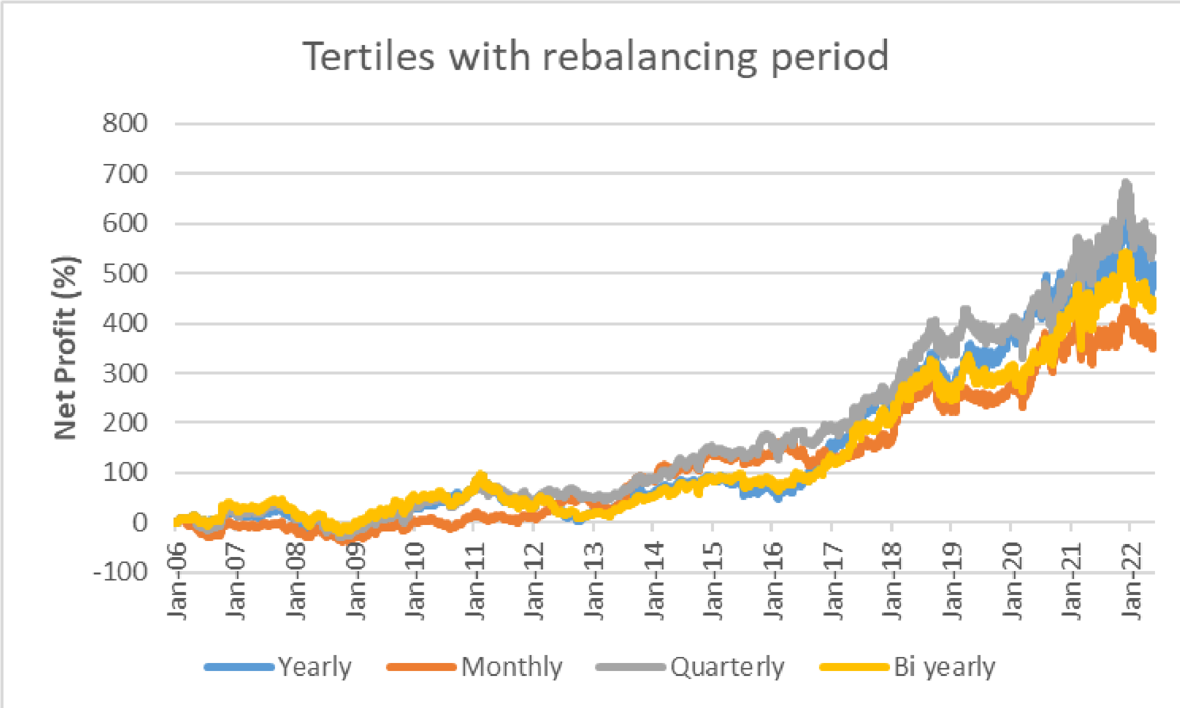
Quartiles with Rebalancing period
Net Profit (%)
700
600
500
400
300
200
100
0
-100
-200
Jan-06
Jan-07
Jan-08
Jan-09
Jan-10
Jan-11
Jan-12
Jan-13
Jan-14
Jan-15
Jan-16
Jan-17
Jan-18
Jan-19
Jan-20
Jan-21
Jan-22
Yearly
Monthly
Quarterly
Bi yearly

A2.2

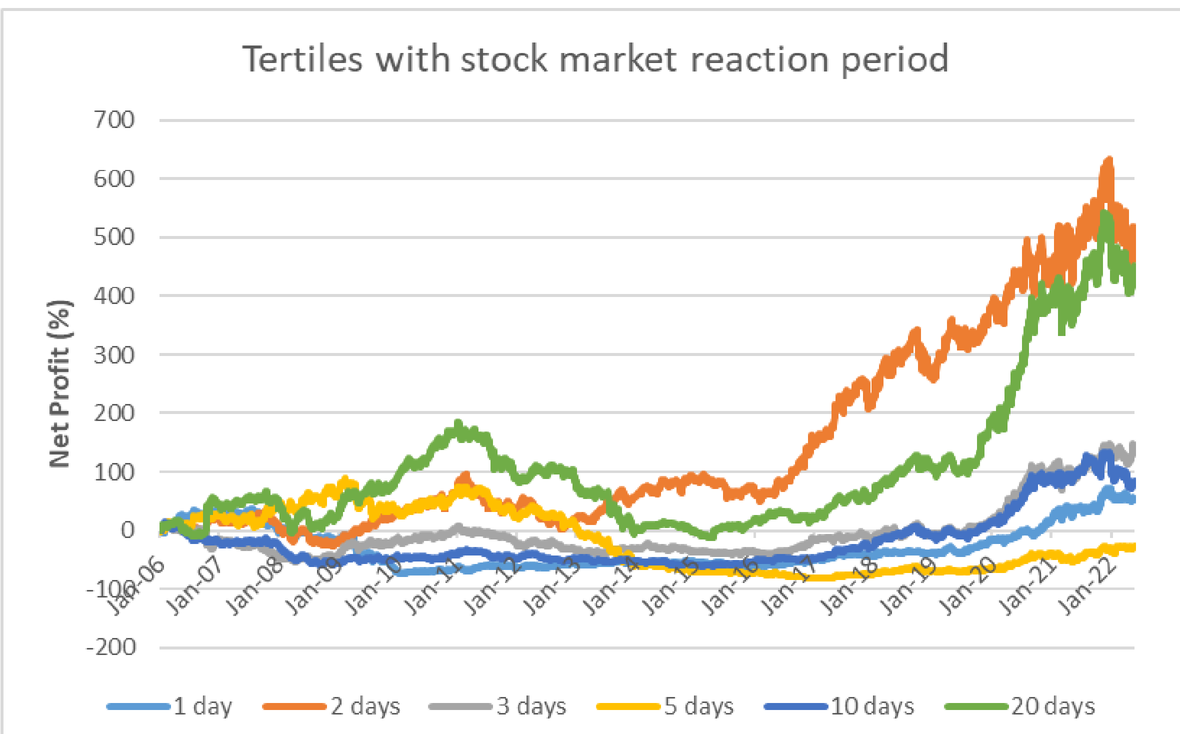

Quartiles with stock market reaction period
Net Profit (%)
700
600
500
400
300
200
100
0
-100
-200
Jan-06
Jan-07
Jan-08
Jan-09
Jan-10
Jan-11
Jan-12
Jan-13
Jan-14
Jan-15
Jan-16
Jan-17
Jan-18
Jan-19
Jan-20
Jan-21
Jan-22
1 day
2 days
3 days
5 days
10 days
20 days

A2.3


Tertiles with asset allocation method
Net Profit (%)
700
600
500
400
300
200
100
0
-100
Jan-06
Jan-07
Jan-08
Jan-09
Jan-10
Jan-11
Jan-12
Jan-13
Jan-14
Jan-15
Jan-16
Jan-17
Jan-18
Jan-19
Jan-20
Jan-21
Jan-22
Value weighted
Volatility weighted
Equal weighted


Tertiles with rebalancing period
800
700
600
500
400
300
200
100
0
-100
Net Profit (%)
Jan-06
Jan-07
Jan-08
Jan-09
Jan-10
Jan-11
Jan-12
Jan-13
Jan-14
Jan-15
Jan-16
Jan-17
Jan-18
Jan-19
Jan-20
Jan-21
Jan-22
Yearly
Monthly
Quarterly
Bi yearly

A2.4


Tertiles with stock market reaction period
700
600
500
400
300
200
100
0
-100
-200
Net Profit (%)
Jan-06
Jan-07
Jan-08
Jan-09
Jan-10
Jan-11
Jan-12
Jan-13
Jan-14
Jan-15
Jan-16
Jan-17
Jan-18
Jan-19
Jan-20
Jan-21
Jan-22
1 day
2 days
3 days
5 days
10 days
20 days

A3.1 Quantconnect algorithm

The below algorithm pertains to the final strategy chosen. For running the code, it has to be copied as it is and pasted in the Quantconnect's notebook (Can be done by creating a free account). After pasting it, press Ctrl + F5 to start the backtest and get the results.

```
#region imports
from AlgorithmImports import *
from enum import Enum
from dateutil.relativedelta import relativedelta
from pandas.tseries.offsets import BDay
from collections import deque
#endregion

class PortfolioWeighting(Enum):
    EQUALLY_WEIGHTED = 1
    VALUE_WEIGHTED = 2
    INVERSE_VOLATILITY_WEIGHTED = 3

class PatentToMarketEquityFactor(QCAlgorithm):

    def Initialize(self):
        self.SetStartDate(2005, 1, 1)
        self.SetEndDate(2022, 5, 31)
        self.SetCash(100000)

        # parameters
        self.reaction_period_after_patent:int = 2  # check for reaction of n days after patent grant
        self.d_period_after_patent:int = self.reaction_period_after_patent + 1 # n of needed daily prices for performance after
patent grant calculation
        self.d_volatility_period:int = 60      # daily volatility calculation period
        self.m_cumulative_period:int = 12       # calculate CPM value using n-month cumulative patent performance history
        self.m_rebalance_period:int = 12        # rebalance once a n months
        self.quantile:int = 3                  # portfolio percentile selection (3-tertile; 4-quartile; 10-decile and so on)
        self.portfolio_weighting:PortfolioWeighting = PortfolioWeighting.VALUE_WEIGHTED

        # assign larger daily period if volatility weighting is set
        if self.portfolio_weighting == PortfolioWeighting.INVERSE_VOLATILITY_WEIGHTED:
            self.max_period:int = max(self.d_volatility_period, self.d_period_after_patent)
        else:
            self.max_period:int = self.d_period_after_patent

        self.required_exchanges:list[str] = ['NYS', 'NAS', 'ASE']
```

```
        self.CMPs:dict[str, float] = {}                              # recent CPM value storage
        self.weights:dict[Symbol, float] = {}                        # recent portfolio selection traded weights
        self.patent_dates:dict[datetime.datetime, list[str]] = {}           # storing list of stocks keyed by their patent date
        self.market_moves:dict[str, list[tuple(float, datetime.datetime.date)]] = {} # storing all market moves in one year keyed by
stock's ticker

        # Source: https://companyprofiles.justia.com/companies
        csv_string_file:str = self.Download("https://www.dropbox.com/s/rckze5h1ldzndag/csvfile.csv?&dl=1")
        lines:list[str] = csv_string_file.split('\r\n')

        # select header, then exclude 'date'
        tickers:list[str] = lines[0].split(';')[1:]

        # store RollingWindow object keyed by stock ticker
        self.prices:dict[str, deque] = { ticker : deque(maxlen=self.max_period) for ticker in tickers }

        for line in lines[1:]:
            if line == '':
                continue

            line_split:list[str] = line.split(';')
            date:datetime.datetime = datetime.strptime(line_split[0], "%d.%m.%Y").date()

            # initialize empty list for stock's tickers, which have patent in current date
            self.patent_dates[date] = []

            length:int = len(line_split)

            for index in range(1, length):
                # store stock's ticker into list, when stock has patent in current date
                if line_split[index] != '0.0' and line_split[index] != '0':
                    self.patent_dates[date].append(tickers[index - 1])

        self.market:Symbol = self.AddEquity('SPY', Resolution.Daily).Symbol

        # add market to prices dictionary
        self.prices[self.market.Value] = deque(maxlen=self.max_period)
        self.symbol_by_ticker:dict[str, Symbol] = {}

        self.month_counter:int = 0
        self.selection_flag:bool = False
        self.UniverseSettings.Resolution = Resolution.Daily
        self.AddUniverse(self.CoarseSelectionFunction, self.FineSelectionFunction)
        self.Schedule.On(self.DateRules.MonthStart(self.market), self.TimeRules.BeforeMarketClose(self.market), self.Selection)
```

```
    def OnSecuritiesChanged(self, changes):
        for security in changes.AddedSecurities:
            security.SetFeeModel(CustomFeeModel())
            security.SetLeverage(20)

    def CoarseSelectionFunction(self, coarse):
        # update daily prices
        for stock in coarse:
            ticker:str = stock.Symbol.Value

            if ticker in self.prices:
                self.symbol_by_ticker[ticker] = stock.Symbol
                self.prices[ticker].append((self.Time.date(), stock.AdjustedPrice))

        days_before:datetime.datetime = (self.Time - BDay(self.reaction_period_after_patent)).date()

        # check if there was any patent granted in d_period_after_patent days before todays date
        # market has to have price data ready
        if days_before in self.patent_dates and len(self.prices[self.market.Value]) == self.prices[self.market.Value].maxlen:
            if self.prices[self.market.Value][-self.d_period_after_patent][0] == days_before:
                # calculate market's return for last d_period_after_patent days
                market_return:float = self.prices[self.market.Value][-1][1] / self.prices[self.market.Value][-
self.d_period_after_patent][1] - 1

                tickers:list[str] = self.patent_dates[days_before]

                # calc market moves
                for ticker in tickers:
                    # if not self.prices[ticker].IsReady:
                    if len(self.prices[ticker]) != self.prices[ticker].maxlen:
                        continue

                    if self.prices[ticker][-self.d_period_after_patent][0] == days_before:
                        # calc stock's return for last d_period_after_patent days
                        stock_return:float = self.prices[ticker][-1][1] / self.prices[ticker][-self.d_period_after_patent][1] - 1

                        # calc excess market move value
                        market_move_value:float = stock_return - market_return

                        if ticker not in self.market_moves:
                            self.market_moves[ticker] = []
                        self.market_moves[ticker].append((days_before, market_move_value))

        # rebalance yearly
        if not self.selection_flag:
```

```
            return Universe.Unchanged

        # select stocks, which has at least one market move value
        return [x.Symbol for x in coarse if x.Symbol.Value in self.market_moves]

    def FineSelectionFunction(self, fine):
        fine:list[FineFundamental] = [x for x in fine if x.MarketCap != 0 and x.SecurityReference.ExchangeId in
self.required_exchanges and x.CompanyReference.IsREIT != 1]

        PMT:dict[FineFundamental, float] = {}   # stores stock's PMT value keyed by stock's object
        volatility:dict[Symbol, float] = {}     # stores volatility values for each symbol in current selection

        for stock in fine:
            symbol:Symbol = stock.Symbol
            ticker:str = symbol.Value
            market_cap:float = stock.MarketCap

            # fetch only market moves stored within cumulative period window
            sum_market_move:float = sum([x[1] for x in self.market_moves[ticker] if x[0] >= (self.Time -
relativedelta(months=self.m_cumulative_period)).date()])

            # in case there isn't last_CMP use formula: CMP = MP / (g + gama), otherwise use formula: # CMP = (1 - gama) *
last_CMP + MP
            curr_CMP_value:float = 0.85 * self.CMPs[ticker] + sum_market_move if ticker in self.CMPs else sum_market_move /
(0.20 + 0.15)

            # store new current CMP value keyed by stock's ticker
            self.CMPs[ticker] = curr_CMP_value

            # calc stock's PMT value
            PMT_value:float = curr_CMP_value / market_cap

            # store stock's PMT value keyed by stock's object
            PMT[stock] = PMT_value

            # volatility calculation - self.d_volatility_period
            daily_prices:np.ndarray = np.array([x[1] for x in self.prices[ticker]][-self.d_volatility_period:])
            daily_returns:np.ndarray =  daily_prices[1:] / daily_prices[:-1] - 1
            volatility[symbol] = np.std(daily_returns) * np.sqrt(252) # annualized volatility

        # make sure, there are enough stocks for selection
        if len(PMT) < self.quantile:
            return Universe.Unchanged

        # make percentile selection
```

```
        quantile:int = int(len(PMT) / self.quantile)
        sorted_by_PMT:list[FineFundamental] = [x[0] for x in sorted(PMT.items(), key=lambda item: item[1])]

        # long highest tertile
        long = sorted_by_PMT[-quantile:]

        # short lowest tertile
        short = sorted_by_PMT[:quantile]

        # portfolio weighting
        # calculate weights for long and short portfolio part
        if self.portfolio_weighting == PortfolioWeighting.EQUALLY_WEIGHTED:
            long_c:int = len(long)
            short_c:int = len(short)
            for stock in long:
                self.weights[stock.Symbol] = 1 / long_c
            for stock in short:
                self.weights[stock.Symbol] = -1 / short_c

        elif self.portfolio_weighting == PortfolioWeighting.VALUE_WEIGHTED:
            total_long_cap:float = sum([x.MarketCap for x in long])
            for stock in long:
                self.weights[stock.Symbol] = stock.MarketCap / total_long_cap

            total_short_cap:float = sum([x.MarketCap for x in short])
            for stock in short:
                self.weights[stock.Symbol] = -stock.MarketCap / total_short_cap

        elif self.portfolio_weighting == PortfolioWeighting.INVERSE_VOLATILITY_WEIGHTED:
            total_inv_volatility_long:float = sum( [1/volatility[stock.Symbol] for stock in long] )
            total_inv_volatility_short:float = sum( [1/volatility[stock.Symbol] for stock in short] )

            for stock in long:
                self.weights[stock.Symbol] = (1 / volatility[stock.Symbol]) / total_inv_volatility_long

            for stock in short:
                self.weights[stock.Symbol] = -(1 / volatility[stock.Symbol]) / total_inv_volatility_short

        # return stocks symbols
        return list(self.weights.keys())

    def OnData(self, data):
        # wait for selection flag to be set
        if not self.selection_flag:
            return
```

```
        self.selection_flag = False

        # trade execution
        invested = [x.Key for x in self.Portfolio if x.Value.Invested]
        for symbol in invested:
            if symbol not in self.weights:
                self.Liquidate(symbol)

        # rebalance/open new trades
        for symbol, w in self.weights.items():
            self.SetHoldings(symbol, w)

        self.weights.clear()

    def Selection(self):
        # wait for self.m_cumulative_period months to elapse from the start of the algorithm before first selection. It gives the chance to self.market_moves to potentially fill up.
        if self.Time.date() < (self.StartDate + relativedelta(months=self.m_cumulative_period)).date():
            return

        # rebalance once a rebalance period
        if self.month_counter % self.m_rebalance_period == 0:
            self.selection_flag = True
        self.month_counter += 1

# Custom fee model
class CustomFeeModel():
    def GetOrderFee(self, parameters):
        fee = parameters.Security.Price * parameters.Order.AbsoluteQuantity * 0.00005
        return OrderFee(CashAmount(fee, "USD"))
```