Market Timing and Big Data

Deniz Yasti
École Polytechnique Fédérale de Lausanne
deniz.yasti@epfl.ch

Léopold Coupet
École Polytechnique Fédérale de Lausanne
leopold.coupet@epfl.ch

Amir Ghali
École Polytechnique Fédérale de Lausanne
amir.ghali@epfl.ch

***Abstract*—This project consist in the implementation of most widely used trading strategies on a large financial dataset: the Russell 2000 index. After detailed exploration, the dataset will be cleaned. Upon formatting the data, financial performance of the *SMA*, *MACD* and momentum strategies will be analyzed.**

***Index Terms*—trading, *SMA*, *MACD*, momentum**

I. Introduction

By definition, a trading strategy is a fixed plan that is designed to achieve a profitable return by going long or short in markets. The main reasons that a properly researched trading strategy helps are its verifiability, quantifiability, consistency, and objectivity. [1]

Finance and trading rely on accurate inputs into business decision-making models. When it comes to existing financial models and analysis, it's fair to surmise that these rely heavily on the volume and accuracy of data collated. So, the more data that is produced, the more accurate the models become, enabling everyone from finance offices to investors to make more informed (and ultimately profitable) decisions. [2]

Big data impacts in many ways how financial trading transactions are carried out. It helps to make quicker and more accurate trades, thus reducing risk while maximizing the profitability of trading strategies. However, it is noteworthy that big data analytics cannot perfectly predict market scenarios all the time. It has imperfections such as incompleteness of data patterns. In the overall, however, big data analytics presents far more benefits than disadvantages to financial trading. That is why it is increasingly becoming an inevitable necessity for financial institutions. [3]

As this report will contain a large part of processing, we will expose the best practices we learnt throughout this course to use for this type of project. Then, we will detail the data used and the preprocessing tools. This will allow us to implement widely used trading strategies such as the momentum, the simple moving average and the moving average convergence divergence approaches. The aim of this report is to compare the strategies in terms of performance of cumulative portfolio value.

II. Best practices

Before diving into the pre-proccesing of a large dataset, we must keep in mind the most important practices seen during the course:

- Clear and meaningful folders and files names.
- Data check: Explore the data from its source manually and by doing some prints, to observe what type of data we are dealing with, the columns names, the dates, the available prices..
- Check for missing data.
- Remove unwanted data.
- Avoid using consuming methods like "for loops" when possible.
- Keep an eye on the run-time to be able to compare the methods we used.
- Intermediate savings of the results: this step is very useful as it allows us to run the pre-processing part of the notebook only once. Afterwards, a simple use of the python commands: *pickle.dump()*, *to_pickle()*, *to_csv()* to save, and *read_pickle()*, *read_csv()* to load would be sufficient.
- Data improvement: We saved the tickers separately when needed, and we merged (.join() function) them

into one dataframe later, so we can use stocks both for individual or pair-study, and also to compare all of them.

- Used breakpoints and debugging when we couldn't find where the error/inconsistency comes from.
- Dynamic programming and general functions, to have a generic code that allows for easier change of variables and values if needed.

III. Data preprocessing

A. Presentation of the dataset

The Russell 2000 index, created in 1984 by the Frank Russell Company, is a stock market index comprised of 2000 small-capitalization companies. It is made up of the bottom two-thirds of the Russell 3000 index, a larger index of 3000 publicly traded companies that represents nearly 98 percent of the investable U.S. stock market. The index is market-cap weighted and used as frequent benchmark for small-cap investors. [4] We chose to work with this index as it is trustworthy (the published prices), real, and provides us with enough data.

Our dataset consists of the historical data (as Date, Open, High, Low, Close and Adjusted Close prices and Volume) of each of the Russell 2000 companies, which we downloaded from the yahoo finance website, by their ticker names. [5]

To get the ticker names, we used the "USMembershiplist" [6] pdf document containing a table of the 2000 companies forming the Russell 2000 index along with their ticker names, where we manually select everything, copy paste into a text editor, and do some manipulations (like using the 'find' tool to search for the tickers) to extract them, and save these into a local text file.

The overall data is the 2002 tickers (each having the previous attributes) in the columns, throughout the last 10 years (from 11.01.2011 to 11.01.2021) in the rows, resulting in a total of 12,012 columns and 2,518 rows of data.

B. Data preprocessing

First we extract the ticker's names from the local text file and do some preprocessing to read and then save these tickers properly, after having a closer look at what we are dealing with (by looking at the file itself and by doing some prints).
In fact we read the file line by line, specifying how the names were separated, and we added each ticker to a list containing the full tickers. Some tickers had endings (like '.A' and '.B' ..) because of which they were not recognized in the yahoo finance data, so we had to remove these endings. We saved the final list of tickers in one string, separated by spaces, for easier further use (namely as input for the yahoo finance download method).

We downloaded the historical data using "yf.download()" function, "yf" being a short for "yahoo finance" library, a Python module to get stock data from Yahoo! Finance. For this we specified as input to the function the tickers for which we want the historical data, the 10 year period with an interval of 1 day (hence we get the daily prices) and grouped by the ticker name.
We chose a period of 10 years because we thought less is not enough, we may not be able to see patterns and significant changes in a 5 years interval for instance. And with a period of more than 10 years, we couldn't find many tickers having data for the exact same period to be able to compare them afterwards.

Our data is now represented as tickers in the columns, each ticker having the attributes [Date, Open, High, Low, Close, Adj Close and Volume] under it, and the Date column. Afterwards we look for empty tickers and remove them (Tickers with no available data in our specified period), set the Date attribute as our index, and save locally the non-empty tickers along with their data individually.

Our data is further transformed into time-series data, where we only keep the Adjusted Close Price and remove the other attributes, then a function "df_stocks()" joins these individual time-series-stocks together in a single dataframe, also indexed by Date.
We decided to work with the Adjusted closing price as it is a calculation adjustment made to a stock's closing price, making it more complex and accurate than the latter. The adjustment made to the closing price displays the true price of the stock because outside factors (Such as dividends, stock splits..) could have altered the true price.

Finally we decided to work with the tickers for which we have all of the 10 years data, so we can compare them fairly, so we removed the others, and we also dropped the stocks containing negative and null prices (keeping them can lead to biases in the results). This has finally left us with a total of 984 stocks and their Adjusted Close Prices for the past 10 years.

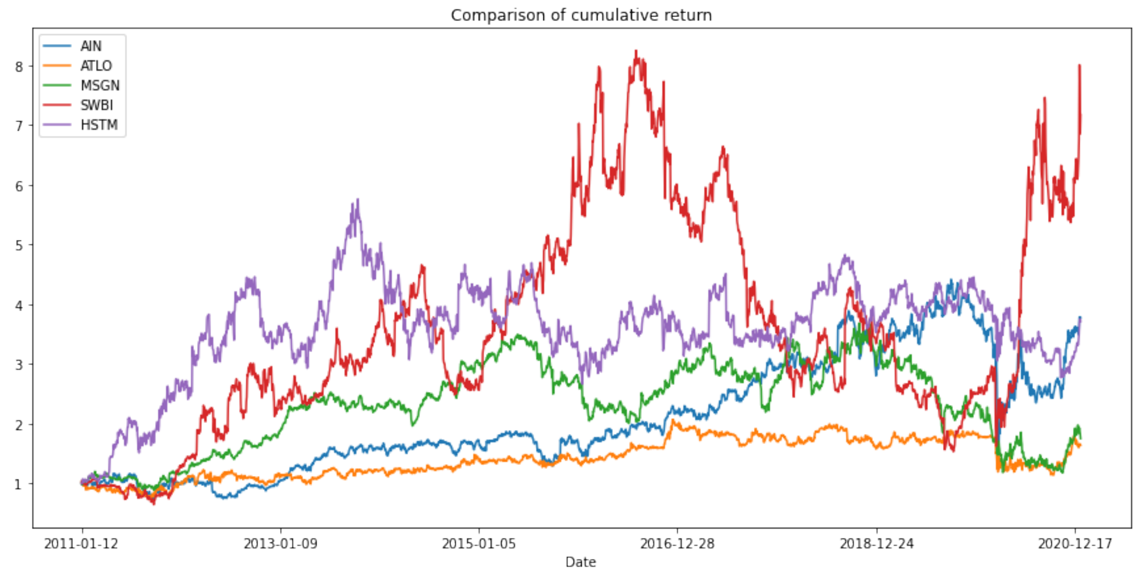

Fig. 1. Daily returns over 10 years

This is what the head of our final data table looks like:

	CEVA	CLW	UUUU	OPI	AXL	APPS	NSP	MAC
Date								
2011-01-03	21.17	40.625	44.0	46.989	13.52	1.4	11.450	27.235
2011-01-04	21.29	41.095	44.0	46.169	14.09	1.4	11.474	26.680
2011-01-05	22.80	40.905	53.0	46.605	14.35	1.4	11.612	26.765

Fig. 2. Date as index, tickers in the columns

Now we are ready to tackle the strategies!

IV. Trading strategies

After obtaining a clean dataset we plotted the daily returns over 10 years of 5 stocks. The figure 1 shows that stocks exhibit different behaviors during the years. Thus, it is not easy to form a portfolio of rising stocks. That is why, we will implement different trading strategies in order to try to capture the best companies possible.

A. Buy and hold with a Momentum strategy

Our first trading strategy follows a momentum theory. The assumption that is being made here is that if a stock had good performances over a particular period, it is more likely to perform well in a future given period. In other words, this strategy bets that best stocks will continue to outperform the market during a certain amount of time. At each time t, the model computes the return over a certain momentum period for each stock. Then at each period, we can create a ranking based on this return and identify the best and worst stock, according to the momentum theory. As the ranking changes rather quickly, we have set a rebalance period, that permits to invest in the best stock, during a specific time horizon. Therefore, the ranking is computed every rebalance period. It is worth noting that we preferred selecting stock with regard to their ranking, and not to their actual return, not to take into account the systematic risk of the market. Therefore, we could invest in a stock, even if they have a negative yearly return.

In order to get our final dataframe of signal, we have to shift our data. Indeed as we get the results of the signal at the end of the day, we can't invest on a signal that we don't have yet. Besides, as we work on close-to-close data, we can't invest the next day, as the next close price comes at the end of the day. This is why all our signal dataframe "df long" is shifted of 2 days. Consequently, the different periods used to calculate different signal are rather long-term period, so this shift does not impact too much the strategies.

Once we have our data frame called 'df long'

Fig. 3. Comparison between a momentum strategy and the benchmark

	AIN	ATLO	MSGN	SWBI	HSTM	NWBI	PRGS	CIA	YORW	APT	...	PFS	LORL	AMN	SB	HLX	TG	IIIN	SIG	AMSWA	ANIP
Date																					
2012-01-11	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2012-01-12	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2012-01-13	0	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	1	0
2012-01-17	0	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	1	0
2012-01-18	0	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	1	0

Fig. 4. Buying (1), and no position(0) states generated using the Momentum strategy

(Figure 4), that tells us which stock to invest in and when, we can build a portfolio. This portfolio is a simulated portfolio that will follow the daily returns of its underlying holdings. To reduce the risk, we have chosen to build the portfolio equally weighted, and not to weights stocks regarding their ranking. Besides, it was proved in the financial literature that momentum strategy works better with small cap stocks, so it makes more sense to give them the same importance as the large stocks.

Going more into details, at each time t, we add the returns of each stock we decided to invest in. As log-return are not asset additive, we use the simple return to do this. Once we have the total return of the portfolio at each time t, we compute the cumulative log-returns that permits to follow the evolution of the portfolio. It is worth noting that as we compare portfolio over a rather long period of time (10 years) we really need to use the log return, to compensate for the exponential growth of the portfolio value over the years. Indeed, as evolution works with cumulative return, 1% growth will generate more and more value as the portfolio increases. The logarithm permits to give more stationarity to the portfolio value.

For this strategy, we have chosen a momentum period of one year, a rebalance period of 132 days (6 months), and a number of stocks to invest in of 50. These parameters were manually selected. Indeed, the whole process to build a portfolio computationally takes an enormous time to run, so a parameter tuning over a train-test set, or any cross-validation method was out of reach for simple computers. Besides, the objective of our project is to compare passive and active trading strategy, we do not seek the best financial performance. Therefore, for this strategy and all the other to come, we have chosen the parameter according to the usual academic way.

Many momentum portfolio strategies, and generally speaking, factor-based portfolio, usually use the long-short strategy. Based on the ranking of the method, the most common way to build a portfolio is to long the ”best” stock, and short the ”worst” ones. In our buy-and-hold strategy, we only long the stocks. Indeed, we wanted to keep the strategy simple, and then do not make any short selling. Besides, we did not take into account any transaction costs, as they would not be relevant enough in this particular strategy. In fact, the period during we hold the stock is rather a long period (6 months).

For this study, our benchmark will consist in an equal weighted portfolio of all stocks. An alternative way is to compute weights with regard to the market capitalization of the stocks but this would lead for large companies to have a greater significance than the smaller ones, and thus obstruct our analysis. Besides as explained in the following section, the implemented momentum portfolio is chosen to be equal weighted which made the choice of an equal weighted portfolio even more consistent.

We can observe in the figure 3 that the momentum performed well, especially in bull market. As the Sharpe Ratio takes also into account the volatility, it is a bit lower for the momentum portfolio. Indeed the higher return is compensated with a higher volatility. It is consistent with the momentum theory, that has a focus on "growth" stocks, known to have a high volatility and a high performance.

In the next section, we will see if simple trading strategies based on technical analysis, will outperform first the benchmark, then our momentum portfolio that only consist on a buy and hold principle.

B. Simple Moving Average (SMA) strategy

Market timing is an important topic that have been studied a lot in the financial literature. Buy and sell stocks at the right time is one of the major concerns of fund managers.

The *SMA* method, which requires the use of technical analysis, is a quite well known trading strategy. A simple moving average of n days is simply the rolling average of a stock computed with the last n days. The main idea behind this approach is to use simple moving average curves and use them to get an understanding on the trend of a stock. In contrary to a momentum strategy, this is a mean-reverting method, that believes that after an important rise or decline, the stock price goes back to its long-term mean. There are multiple variations involving *SMA* curves. In fact, some use one *SMA* curve, other multiple *SMA* curves with different time frames. However, as the core idea is the same between all these *SMA* strategies, we decided to use an *SMA* strategy involving two *SMA* curves:

- SMA_{short} refers to a simple moving average with a short time horizon
- SMA_{long} refers to a simple moving average with a long time horizon

It should be noted that generating an *SMA* curve is quite easy in python using the talib library: *talib.SMA(prices, time horizon)*. The first difficulty in implementing this strategy is the correct choice of the short and long time horizons. In fact, if they are not chosen specifically for a given circumstance, it can lead to a very negative result. In our project, the rebalance period is chosen to be 132 days as explained previously. That is why, the long term horizon for this strategy is chosen to be 132 days. Moreover, as the actual trading would occur two days after the optimal stocks have been selected, the short time horizon should be very large compared to 2. This is what led us to choose 60 days.

The aim of the strategy is to generate a buying signal if the SMA_{short} crosses over the SMA_{long} curve. This is a logical result since this crossover is a clear indication of an increase in stock prices. Similarily, a selling signal would be generated if the SMA_{long} crosses over the SMA_{short} curve.

We wanted to implement this strategy to all the preprocessed stocks as well as those selected using the momentum strategy. This would allow a further comparison for of the strategies. To this end, we generated a function called *SMA Position*. This function's steps for each stock is as follows:

- Compute short time horizon *SMA*: SMA_{short}
- Compute long time horizon *SMA*: SMA_{long}
- Compute the difference of the curves to detect crossovers: SMA_{short}-SMA_{long}
- If the difference is positive: Assign a buying signal (i.e 1)
- If the difference is negative: Assign a selling signal (i.e -1)

,

This generates a DataFrame that indicates not only when to buy a stock as in section IV-A but also when to sell a stock buy a -1. For the *SMA* strategy on all stocks, this will generate a buying or selling state for each stock and for each date. For the *SMA* strategy combined with the momentum strategy, we will obtain a dataframe containing also 0 value that indicate that the stock was not selected at this time. Anyway, the resulting dataframes should be shifted twice as explained previously. The results are shown in figure 6.

The next step is to compute the value of the portfolio for each case. This process is the same as explained in the section IV-A. The cumulated value of the portfolio is plotted in the figure 5. The *SMA* strategy combined with a momentum approach seems to perform slightly better

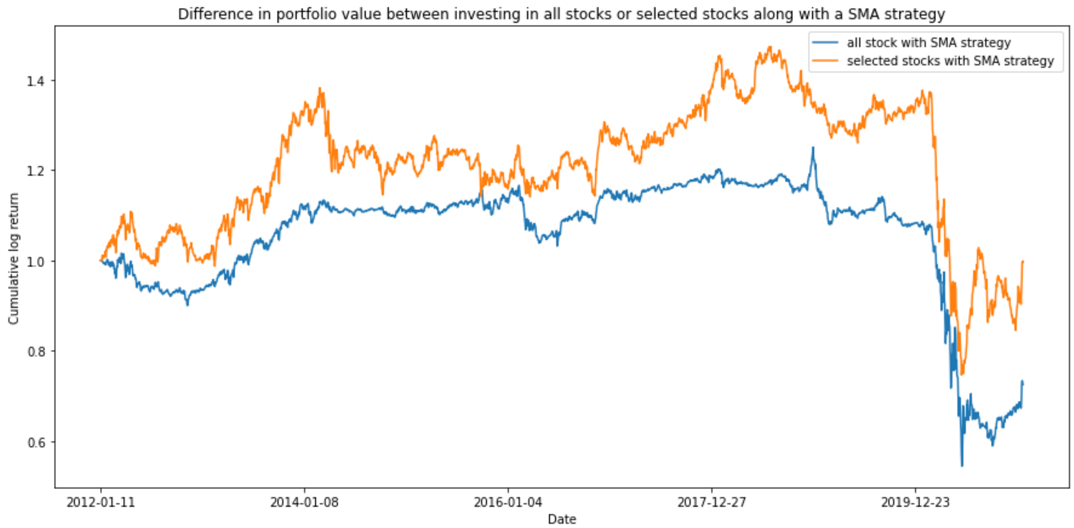

Fig. 5. Cumulated portfolio value generated using the *SMA* strategy

	AIN	ATLO	MSGN	SWBI	HSTM	NWBI	PRGS	CIA	YORW	APT	...	PFS	LORL	AMN	SB	HLX	TG	IIIN	SIG	AMSWA	ANIP
Date																					
2012-01-11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2012-01-12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2012-01-13	0.0	0.0	0.0	0.0	1.0	0.0	-0.0	0.0	0.0	0.0	...	0.0	0.0	-0.0	-0.0	0.0	0.0	-0.0	0.0	1.0	-0.0

3 rows × 982 columns

	AIN	ATLO	MSGN	SWBI	HSTM	NWBI	PRGS	CIA	YORW	APT	...	PFS	LORL	AMN	SB	HLX	TG	IIIN	SIG	AMSWA	ANIP
Date																					
2012-01-11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2012-01-12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2012-01-13	1.0	1.0	1.0	1.0	1.0	1.0	-1.0	1.0	1.0	1.0	...	1.0	1.0	-1.0	-1.0	1.0	1.0	-1.0	1.0	1.0	-1.0

Fig. 6. Buying (1), selling(-1) and no position(0) states generated using the *SMA* strategy

than the *SMA* strategy applied on all stocks. In order to comment on the financial performance, two metrics are computed. The total log-return for the *SMA* strategy on selected stocks is -0.18 % while the total log return for the *SMA* strategy performed on all stocks is -27.47 %. Moreover, the sharpe ratio of the *SMA* strategy on selected stocks is 0 while the sharpe ratio for the *SMA* strategy on all stocks is of -0.27.

C. Moving Average Convergence Divergence (MACD)

MACD is another approach widely used by technical analysts. As our first passive strategy, it follows a momentum model, unlike the mean-reverting *SMA*. It makes use of the exponential moving averages (EMA). Likewise *SMA*, there are many variants of the *MACD* strategy. We will use a classical *MACD* method. We use the following command from the talib library: *talib.MACD(prices,fastperiod,slowperiod,signalperiod).* This will generate three sets of values: *MACD*, signal and histogram. After computing the EMA with a medium and long time horizons, the *MACD* is obtained by subtracting the long time horizon EMA from the short time horizon EMA. The signal line represents an EMA with a short time horizon. The histogram is obtained by simply subtracting the signal line from the *MACD* values. The strategy relies on the sign of the histogram. Indeed, it dictates that a positive value in the histogram is an indicator of a positive trend in the stock and thus a buying signal must be generated. Similarly, when the histogram value is negative, a selling signal must be generated.

Again, the choice of the time horizons is a complexity of this method. It is important to emphasize on the fact that the time horizons chosen for the *SMA* and *MACD* strategies are different. This is a natural choice since the internal computations of the two strategies are different. A hyper parameter tuning could have been performed to have the parameters. However, as explained previously, as our dataset is heavy, this would require too much computational power. As the close prices are given each day and the trading can only occur two days later, the short time horizon can lie within a week. For the long time horizon, we wanted to choose a number that is around one fourth of the rebalance period. This led to the choice of long time horizon of 35 days and a short time horizon of 5 days.

We implemented this strategy to all the preprocessed stocks as well as the selected stocks using the momentum portfolio for the same reasons as in section IV-B. For this purpose, we defined a new function called

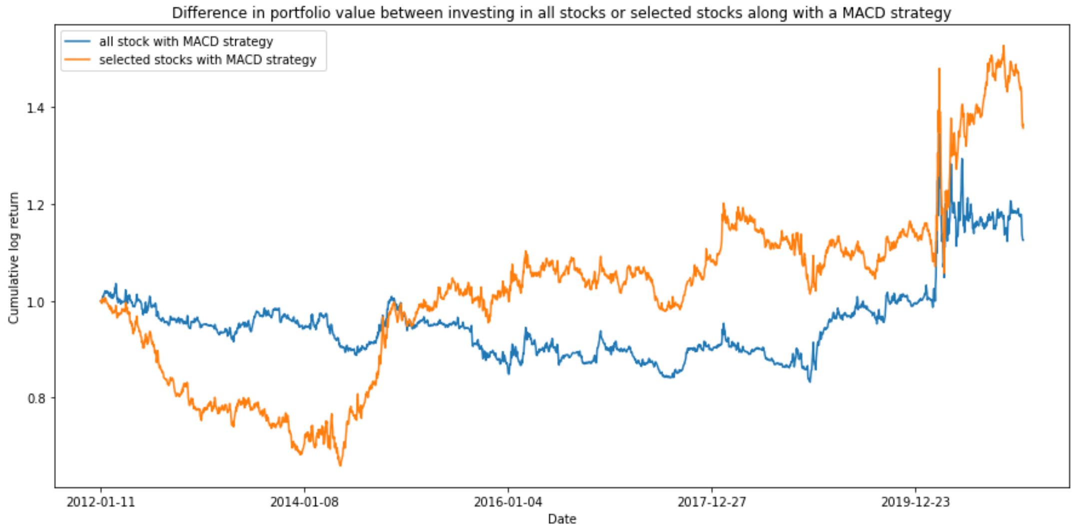

Fig. 7. Cumulative portfolio value generated using the *MACD* strategy

$MACD_{position}$. This function, based on the sign of the histogram, returns a buy (1), sell(-1) or no position (0) signal. After being shift of two days, the obtained dataframes are illustrated in figure 8.

The cumulative return of the two portfolios are shown in figure 7. There is no obvious pattern between the two portfolios according the the *MACD* strategy. Two financial metrics are computed in order to have a better understanding of their respective performances. The total log return for the *MACD* strategy on selected stocks is 35.70 % while the total log return for the *MACD* strategy performed on all stocks is 12.55 %. Moreover, the sharpe ratio of the *MACD* strategy on selected stocks is 0.24 while the sharpe ratio for the *MACD* strategy on all stocks is of 0.10 .

	AIN	ATLO	MSGN	SWBI	HSTM	NWBI	PRGS	CIA	YORW	APT	...	PFS	LORL	AMN	SB	HLX	TG	IIIN	SIG	AMSWA	ANIP
Date																					
2012-01-11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2012-01-12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2012-01-13	-0.0	0.0	-0.0	-0.0	-1.0	0.0	-0.0	-0.0	-0.0	0.0	...	0.0	0.0	-0.0	0.0	0.0	-0.0	0.0	-0.0	1.0	0.0

3 rows × 982 columns

	AIN	ATLO	MSGN	SWBI	HSTM	NWBI	PRGS	CIA	YORW	APT	...	PFS	LORL	AMN	SB	HLX	TG	IIIN	SIG	AMSWA	ANIP
Date																					
2012-01-11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2012-01-12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2012-01-13	-1.0	1.0	-1.0	-1.0	-1.0	1.0	-1.0	-1.0	-1.0	1.0	...	1.0	1.0	-1.0	1.0	1.0	-1.0	1.0	-1.0	1.0	1.0

Fig. 8. Buying (1), selling(-1) and no position(0) states generated using the *MACD* strategy

V. Strategies' performance comparison

In order to have a better understanding of the difference between the *SMA* and *MACD* strategies, we visualize both of the buy and sell processes with a randomly chosen stock. We decided to plot, for a given stock, its stock value along with its *SMA* and *MACD* curves during its first selection period. For this to work, we created a function called *First occurrence*. This function has two outputs. First, it returns the first index where the stock has been selected, denoted *ind1*. As the rebalance period is 132 days, it is sure that the stock is selected for at least 131 more days. That is why, we check if the stock is still selected every 132 days. This way we can find the index *ind2* which corresponds to the last value where the stock has been selected for this period. In other words, the first time period where the stock has been selected occurred from *ind1* to *ind2*.

We define another function *graph* that, given a ticker and the output of the *First occurrence* function, ouputs two graphs. The figure 11 is the buying and selling points using an *SMA* strategy along with the values of the simple moving average for the short and long time horizons. The figure 10 is the buying and selling points using an *MACD* strategy along with the values of the *MACD* and signal line. The resulting graphs are shown below.

Putting all the strategies together in the same plot, we observe that for a given strategy, the returns we got on the selected stocks are always better than the benchmark. Apart from that, this benchmark and the buy and hold momentum strategy were very similar and had significantly lower returns than the active

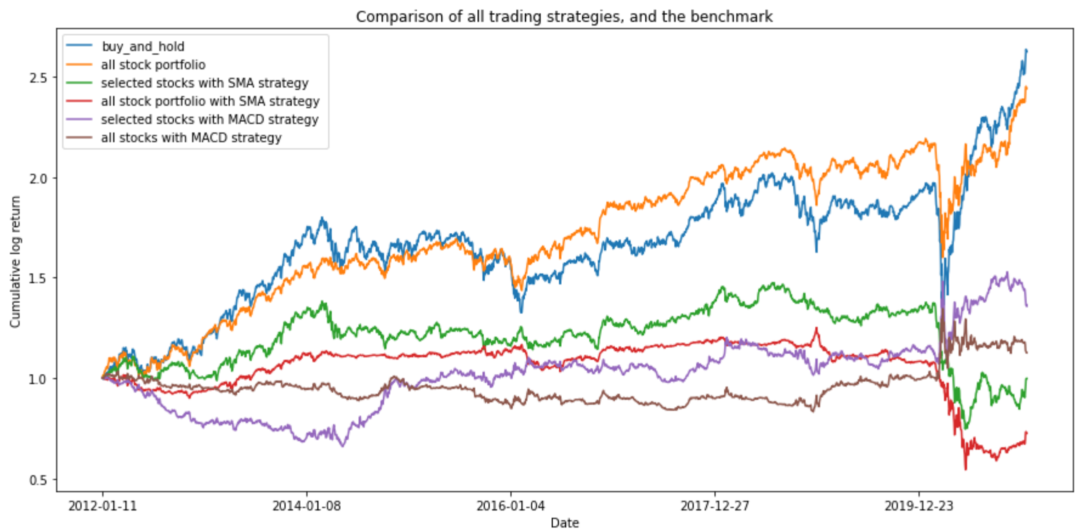

Fig. 9. Cumulative portfolio value generated for all strategies

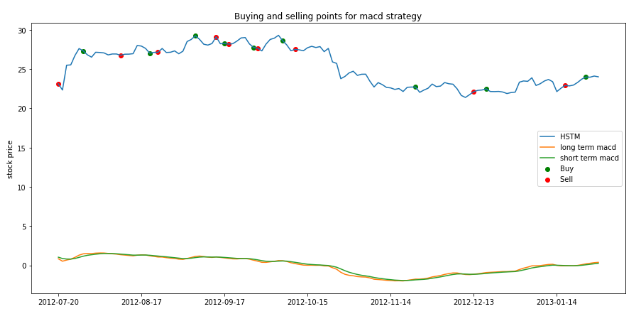

Fig. 10. *MACD*: buying and selling points example

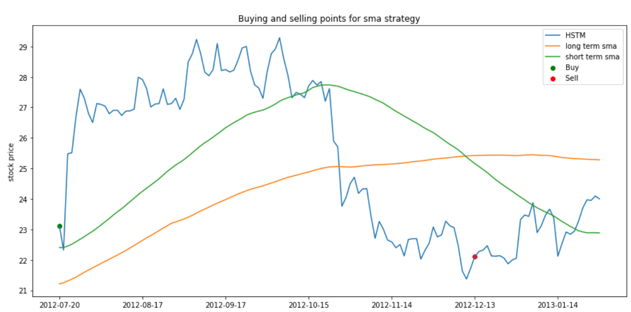

Fig. 11. *SMA*: buying and selling points example

trading strategies. The *MACD* strategy was by far the best one, providing the highest return since the start date, to average a return after 10 years nearly 8 times better than the buy and hold strategy and 4 times better than the *SMA*. Associating a momentum ranking, and a *MACD* timing strategy clearly outperformed the market. In this dataset, the mean-reverting strategy really under-performed the momentum ones.

Moreover, we computed the sharpe ratios as well as the total log returns to have a better understanding on the performances of each strategy. The tables I and II offer a summary of the results obtained during the simulations.

TABLE I
TOTAL LOG-RETURNS AND THE SHARPE RATIO FOR THE STRATEGIES PERFORMED ON ALL STOCKS

	Log return	Sharpe ratio
Benchmark	142.45 %	0.73
SMA	-27.47 %	-0.27
MACD	12.55 %	0.10

TABLE II
TOTAL LOG-RETURNS AND THE SHARPE RATIO FOR THE STRATEGIES PERFORMED ON SELECTED STOCKS

	Log return	Sharpe ratio
Benchmark	142.45 %	0.73
Buy and hold	162.25 %	0.68
SMA	-0.18 %	0
MACD	35.70 %	0.24

VI. CONCLUSION

Our results clearly showed that Technical Analysis was not as successful as predicted. And using it along with a momentum selection of stock did not particularly enhanced the financial performance. Indeed, our *SMA* and *MACD* methods did not particularly showed a great result in buy and sell timing, even along with a prior momentum selection. Furthermore, we know that we lost a lot of accuracy neglecting all type of transaction costs. Indeed in active strategies, the shorter the periods of rolling calibration are, the more transactions occur. In the real world transactions costs are really an issue as they drastically lower the global performance. The lack of a parameter tuning in both of the strategies definitely decrease our financial results. Besides, as TA-Lib offers a large amount of different technical indicators, we could have optimized our timing strategy, searching for the ones that lead to the best results. It is worth noting that our data was focus on the USA region, and then that the results do not infer a global underperformance of our particular active trading strategies.

REFERENCES

[1] https://en.wikipedia.org/wiki/Trading_strategy#:~:text=In%20finance%2C%20a%20trading%20strategy,quantifiability%2C%20consistency%2C%20and%20objectivity.

[2] https://www.bloomberg.com/professional/blog/3-ways-big-data-changing-financial-trading/

[3] https://www.SMArtdatacollective.com/3-key-ways-big-data-is-changing-financial-trading/

[4] https://www.investopedia.com/terms/r/russell2000.asp

[5] https://finance.yahoo.com/quote/%5ERUT?p=ˆRUT&.tsrc=fin-srch

[6] https://content.ftserussell.com/sites/default/files/ru2000_membershiplist_20200629.pdf

[7] Hazim S El-Baz, Ibrahim Al Awadhi, Assia Lasfer, SMA and MACD combinations for stock investment decisions in frontier markets: evidence from Dubai financial market
Link: https://www.researchgate.net/publication/264815041_SMA_and_MACD_combinations_for_stock_investment_decisions_in_frontier_markets_evidence_from_Dubai_financial_market