Active Investing Post COVID-19: Quantitative Predicting and Trading Strategy for U.S. Sector ETFs

Team Member: Pengcheng Xu, Jinhao Sun, Quanzhen Zhang

April 30, 2020

§ 1 Introduction

The global economy is badly slashed by the COVID-19 pandemic. Especially, with the increasing number of new cases and the continuous shutdown of economic activities, the U.S. market is facing a tremendous downhill since 2008. The climbing unemployment filings and the unprecedented negative oil futures leave the investors and consumers with uncertainty. S&P500 index has declined by about 30% from the previous level. Multiple industries such as transportation, hospitality, retail have a big part of revenue been wiped out. For the U.S. equity market, several times circuit breakers have been triggered, and the VIX index climbs and is extremely volatile since the beginning of 2020.

Many arguments surrounding the post-impact of COIVD-19 are out there. Passive investing will not be a curveball but will continue to be implemented prevalently. Nevertheless, we stand on the side believing that the U.S. market will demonstrate a significant advancement with upward impulse, even though the timing and extent are currently unclear. Under such circumstances, active investing can play a role in capturing greater market volatility.

§ 2 Research Objective

The ultimate objective is to ride the near-term market volatility post-COVID-19 for a maximum possible extent and outperform the benchmark. Due to the perspective of cost-saving as well as the perception of the overall U.S. economy's recovery, we aim to find an equilibrium and the best practice of constructing our portfolio. Moreover, recognizing that the long-term fundamental investing would only capture little of the near-term volatility, we ought to develop a better trading strategy with higher frequency to tackle this issue, i.e. data-driven quantitative trading algorithms by machine learning techniques, and perform backtesting to validate our strategy. We will specifically explore the utilization of different machine learning tools and technical indicators to optimize our strategy.

§ 3 Literature Review

This section mainly covers the research conducted that better promotes the soundness of our studies. We probe several topical areas regarding the research objectives, including portfolio construction, drivers of market volatility, strategy comparison, implementation of machine/deep learning, and value at risk.

§ 3.1 U.S. Sector ETFs

Equity indices bring a convenient way to comprehensively track the economic health. The SPDR S&P 500 ETF tracks the S&P 500 Index, which is tradeable by investors throughout the day. Diversification and low-cost characteristics make them appealing to investors. ETFs are traded the same as individual stock with features of short sale and leverage. In terms of the Global Industry Classification Standard (GICS), 11 ETFs track the major sector benchmarks.[1] Also, compared with mutual funds, ETFs have higher daily liquidity and lower fees.

§ 3.2 Drivers of Market Volatility

The current global health crisis evidently contributes as the biggest driver toward market volatility. At a deeper level, this pandemic acts as a stone triggering the waves all over the place. All the waves aggregate together altering the beats of the market pulse. These waves, the "True Drivers", include but not limited to the movements of oil price, bond price, interest rate spread, commodity price, foreign exchange, consumer confidence, and new orders of consumer goods and materials.[2] Not all the drivers provide the equivalent impact for market volatility, but they would be the potential indicators.

§ 3.3 Fundamental vs Quantitative Investing

One of the main differentiators is that a fundamental investor searches for specific companies, based on their historical performances as well as the prospective financial indications; in contrast, a quantitative investor spends the majority of time determining the characteristics of candidates for screening.[3] A fundamental investor as a value investor usually has a longer holding period and less trading frequency than a quantitative investor. Notably, often the quantitative investing combines with statistical, computational, and algorithmic tools to generate meaningful insights, to better characterize the inputs and prompt the trading executions.

§ 3.4 Machine Learning and Deep Learning

By definition, machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn and identify patterns and make decisions with minimal human intervention.[4] Deep learning is a subset of machine learning that has networks capable of learning unsupervised data; also referred to as deep neural networks. The application

[1] Global Industry Classification Standard (GICS®)

[2] Stock Market Volatility: Identifying Major Drivers and the Nature of Their Impact, *by Stefan Mittnik, Nikolay Robinzonov, Martin Spindler*

[3] Equity Valuation and Portfolio Management, Fundamental vs. Quantitative Investor, *by Frank J. Fabozzi, Harry M. Markowitz*

[4] Machine Learning: What It Is and Why It Matters

of these techniques bourgeons in the world of investment. Hedges funds such as Bridgewater Associates and Two Sigma Investments do have investment approaches of utilizing machine learning and deep learning algorithms to actively manage the investments, with stated annualized returns above 35% over 20 years.[5] BlackRock, as one of the world's largest investment management companies, offers the Aladdin Risk Platform – an operating system automatically monitoring over 2,000 risk factors per day and testing portfolio performance.[6]

§ 4 Data and Methodology

§ 4.1 Portfolio Construction

Given that our portfolio construction objective is to determine an equilibrium to address both perspectives in terms of cost-saving and overall economy's recovery, we selected 10 U.S. Sector ETFs which closely track benchmark indices. That being said, the transaction costs of trading Sector ETFs are dramatically lower than trading individual stocks, while capturing the general market movements on a large horizontal scale. On the other hand, the reason for not selecting S&P 500 (SPY) ETF as a stand-alone holding is that it would misalign our ultimate objective to actively ride the market volatility for a maximum possible extent. Instead, we use SPY as a benchmark to compare the performance of our strategy.

§ 4.2 Volatility Drivers and Indicators Selection

Data acts as the backbone of our predicting and trading strategies. We identified 54 market volatility drivers out there, including 30 currency pairs, 8 bond indices and spreads, 8 commodity prices, 7 energy, and metal prices, and the investors' sentiment index. Understandably, market events such as this pandemic, supply & demand, and corporate fundamentals drive the movements of these "drivers"; however, these "drivers" would drive backward the movements of the 10 U.S. Sector ETFs.

Besides the drivers identified above, we also made use of 140 technical indicators as the other part of data inputs and manipulated them into our model. Had the forward-looking perspective, some of the indicators are the momentum indicators, i.e. MACD, RSI, Williams %R, Elliott Wave Oscillator, etc. In total, our rough data inputs include 194 features that are employed to predict market directional volatility; by the way, these 194 features do not have the same impact on the volatility prediction model, some of which would largely impact or only impact little. Hence, penalization and elimination techniques are applied as a filter. This will be thoroughly illustrated in the next section *§ 5 Strategy Analysis and Result Discussion*.

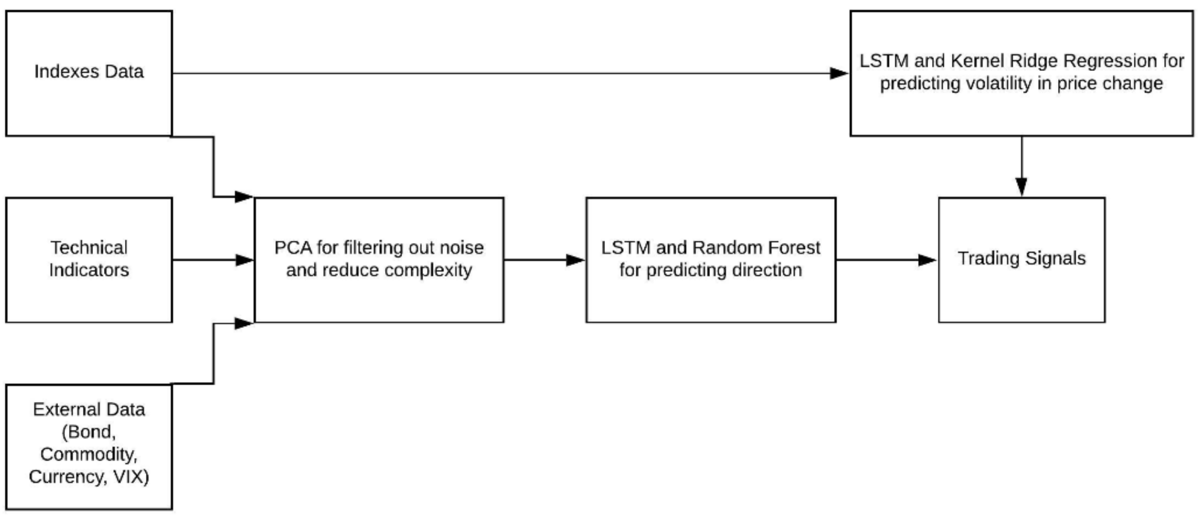
---

[5] Deep Learning Algorithms: The Future of Financial Investment, *by Harry Chiang*

[6] Machine Learning in Investment Management and Asset Management – Current Applications, *by Raghav Bharadwaj*

§ 4.3 Methodology Introduction

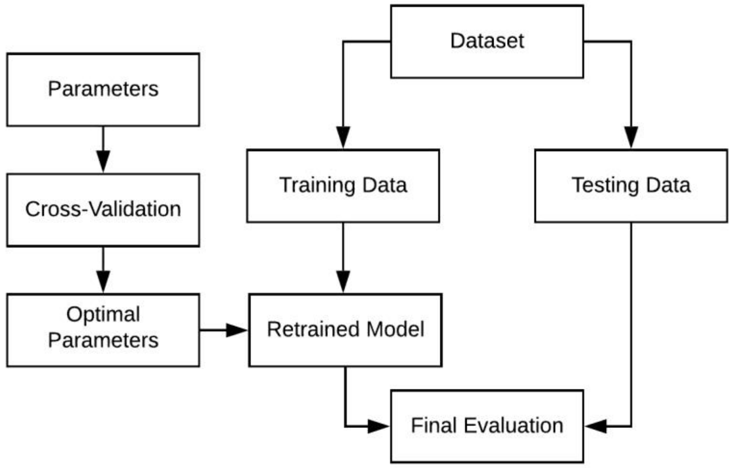
The brief overview of the strategies implemented includes two parts: predicting the next-day price volatility by classifying the historical days with price movement greater than 0.7%, and predicting the next-day price movement direction. Trades of the Sector ETFs will be executed on the days that have volatility greater than the threshold, and take long/short positions based on the price movement directions. The methodology flowchart is plotted below for better representation. No trade will be placed, while the same trading signal occurs as the previous day. Close out the positions, while the opposite trading signal alarms. This section will introduce the quantitative methodologies utilized to realize our strategy. Implementation in detail and analysis will be demonstrated in *Section 5*.



§ 4.3.1 Backtesting (Cross-Validation)

Backtesting measures the feasibility of trading strategies by examining the model performance by using historical data. A good backtesting result validates the model's soundness and boosts investors' confidence for prospective utilization. Cross-Validation is one of the techniques to test the effectiveness of machine learning models, which requires to keep aside a portion of historical data that does not use to train the model; in other words, it is the Train-Test Split Approach.[7] By analyzing the performance of the prediction model in the testing set, we test if the model is overfitted or underfitted, and readjust based upon the result. The flowchart below serves as a straightforward explanation of the flows.

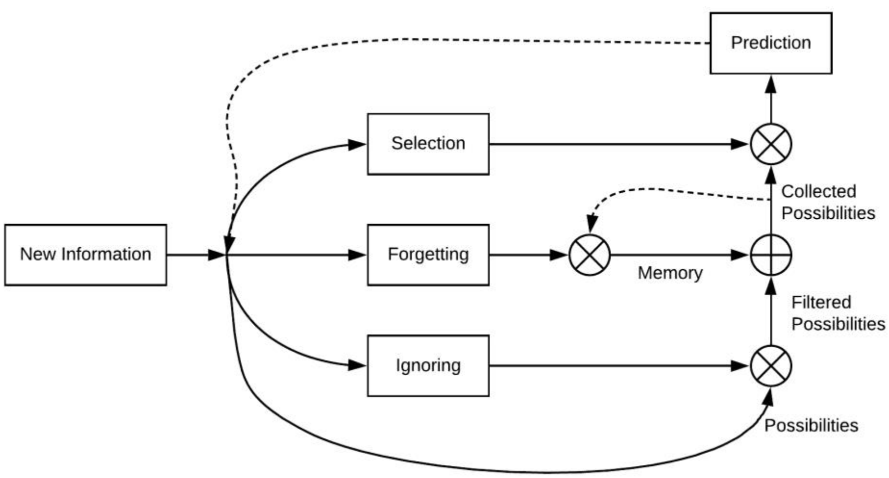
[7] Why and How to Cross Validate a Model, *by Sanjay.M*



[8]

§ 4.3.2 Long Short-Term Memory Network

Accuracy improvement regarding our predicting algorithms can be realized by Long Short-Term Memory Neural Network (LSTM) to train the model. It is a recurrent network belongs to the family of deep learning. The LSTM network is able to update the current state with information in the past and train the model recurrently for better predicting.[9] We adopt it to experiment with the model's accuracy in the testing set. The flowchart below is used as a good representation of the mechanics of the LSTM network.



[10]

§ 4.3.3 Principal Component Analysis (PCA)

[8] Cross-Validation: Evaluating Estimator Performance
[9] Improving Trading Technical Analysis with TensorFlow Long Short-Term Memory (LSTM) Neural Network, *by Chenjie Sang, Massimo Di Pierro*
[10] Understanding LSTM Networks, *by Christopher Olah*

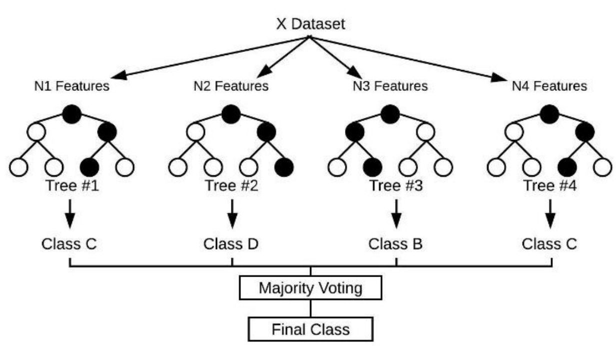
One of the most useful techniques for avoiding overfitting the model is PCA; it reduces feature dimensions as well as the computational complexity. PCA is especially productive to be applied to our strategy, as there are 194 total numbers of original predicting features employed as inputs. Some unimportant features will be eliminated, and only keep the ones with strong correlations remained. In another aspect, PCA provides feature extraction. Numbers of compounded independent variables are created, each of which is a combination of all the original features.

§ 4.3.4 Kernel Ridge Regression (KRR)

By definition, it is a non-parametric form of ridge regression to learn a function by the respective kernel k by minimizing a squared loss with a squared regularization term. The weight vectors in KRR can be expressed in the form of $\alpha = (K + \tau \mathrm{I})^{-1} y$, which K is the kernel matrix, and α is vector weights. The KRR function is evaluated as $f(x) + \sum_{i=1}^{N} \alpha_i k(x, x_i)$.[11] Inheriting the characteristics of ridge regression, KRR penalizes the predictors for regularization; we implement this property of KRR to forecast the next-day volatility of price movement.

§ 4.3.5 Random Forest Classifier

As the method of classification, Random Forest aggregates several decision tree classifiers by bootstrapping the original dataset with replacement to create a new bootstrapped dataset. Each decision tree is constructed based on the hierarchy of Gini Impurity; the feature with the lowest Gini Impurity is put on the top node. Embedded the logic of the Greedy Algorithm, decision trees follow the approach of making the locally optimal choices and then finding the global optimum. The majority of classes as output through the random forest is voted to determine the final class. We utilize the random forest to classify the next-day price movement directions as binary variables.



12

[11] Kernel Ridge Regression - Shogun 6.1.4
[12] Applying Random Forest (Classification) - Machine Learning Algorithm From Scratch With Real Datasets, *by Abilash R*

§ 5 Strategy Analysis and Result Discussion

§ 5.1 Logic Of The Strategy

The strategy is composed of three parts: 1, volatility prediction; 2, price movement prediction; 3 strategy execution.

But before going into details of the three parts, it should be made explicit that the fund for the strategy is divided into 10 parts: each part trading on an index tracking one S&P 500 sub-sector via ETF. The communication service sector index (XLC) was left out in the process, since the index started in 2018, and there was no sufficient historical data to train strategy.

In the first part, a model is trained on historical daily returns of each index for volatility prediction, expressed in terms of absolute value of daily return, and generates a 1-day forecast of volatility.

In the second part, a model is trained on historical daily returns above a volatility threshold of each index for return sign prediction, and generates a 1-day forecast of return sign, i.e. market direction.

In the third part, based on predictions given by the first two parts: if predicted volatility is greater than the threshold, the direction predicted by the model in the second part will determine the positions the strategy takes; if the model predicts the market to go downside tomorrow, the strategy will take long position, and short position if upside predicted. All 10 indices are traded in this fashion.

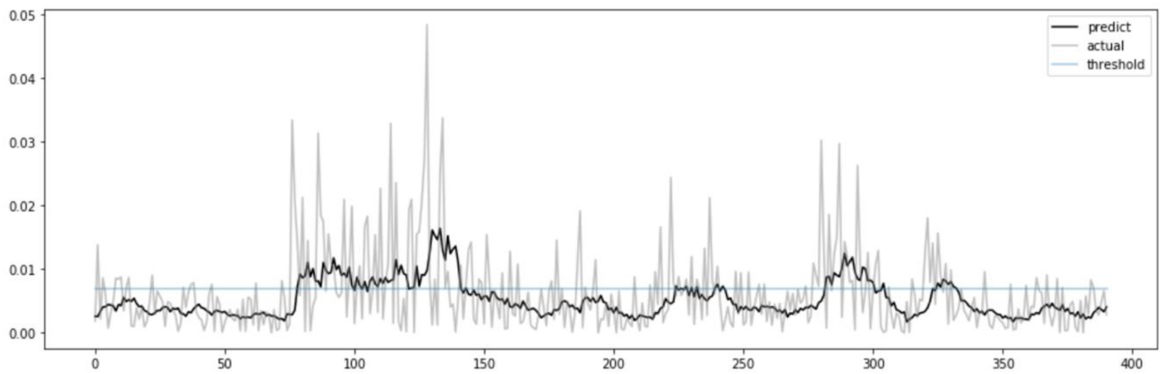
§ 5.2 Motivation For The Strategy

When it comes to predicting the market, it is common knowledge that its movement is characterized by randomness and defies most statistically modelling methods. Even super fancy machine learning/neural network methods do not work well with the market movement. As a result, we break the prediction into volatility and conditional movement forecasts, and wish to accomplish to beat the market in this fashion. Volatility tends to cluster, i.e. highly volatile market movements cluster in a period while the market stays quite in other times. Predicting volatility is a less formidable task than predicting the price itself.

Meanwhile, hopefully the market movements during volatile periods are less likely random noise and thus better described by technical indicators and macroeconomic factors. We hypothesize that the volatile market movements can be better predicted than less volatile ones. So conditional on the 'volatile signal' the model in the first part generates, meaning 1-day market volatility will exceed a certain threshold (usually between 0.5% and 1%), the model will generate a prediction for the market movement. Moreover, by feeding all technical indicators provided by TA-Lib and several macro-economic factors to the price movement prediction model, hopefully it will generate some accurate forecasts.

§ 5.2 Inside Volatility Prediction Model

We tried 6 models on S&P 500 historical data while determining the optimal model to predict volatility. Some of them differ from one another in terms of modelling methodologies, others in terms of features. The first model used LSTM method and 14 volatility lags, absolute returns in this case, as features. The training period is from January 2007 to June 2018, and the test period goes from June 2018 to January 2020. A plot of predicted volatility, real volatility and an arbitrarily set threshold 0.7% is presented below.



The predicted values seem dampened from the real values, and though in really volatile periods, they go beyond the threshold and mimic the real market volatility, they often fail to predict volatile days in quiet periods. If 0.7% is set as the threshold and the accuracy is determined by whether both real and predicted volatility go beyond the threshold, the model produces an accuracy of roughly 70% and AOC value of 63%. Other threshold values, such as 0.5% or 0.8% produce even worse accuracy. The results are not impressive enough.

So we proceeded to the second model, which uses binary values determined by if market volatility exceeds the threshold as y variable and a similar LSTM model, and had an accuracy of 69%, more or less the same as that of the first model.

We moved to the third model, which takes 1-day lag values of the following variables as input: 14-day rolling standard deviation, volume difference from the previous trading day and the absolute return itself and a similar LSTM model. The model generated an accuracy of 69%. The result was a bit frustrating.

We decided to add one more feature to what we had done in the third model, which was the difference between the absolute return and 14-day rolling standard deviation and model returned a similar accuracy. Only later we came to the idea that the difference, which is a linear combination of the two features used in the third model, was a total waste when it was added to the third model.

So we moved to the fifth model, which takes 7-day lag values of the variables used in the third model as input. Again to our disappointment, the accuracy was 68%.

The sixth model took 28-day lag values, however, the results did not improve either. and finally, we decided to use the Kernel Ridge Regression method to have a taste of a different modelling method. Through a greedy search for optimal parameters, the model generated an accuracy of 58%.

At this point, we decided that we would not move forward to other models. By comparing plots generated by difference models, we found that all models more or less face the same problems as the first model: The predicted values seem dampened from the real values, and though in really volatile periods, they go beyond the threshold and mimic the real market volatility, they often fail to predict volatile days in quiet periods. All these models are good at guessing high volatility during high volatility periods, and low volatility during low volatility periods, which do not help provide extra insights into the market volatility. In other words, they cannot foretell accurately if the market will switch gear to high volatility/low volatility beforehand.

For simplicity, we decided to use 1-day lag values of the following variables as input: 14-day rolling standard deviation, volume difference from the previous trading day and the absolute return itself with a LSTM model as the volatility prediction model among all, since all produced similar results and the simplest among them stood as the best one.

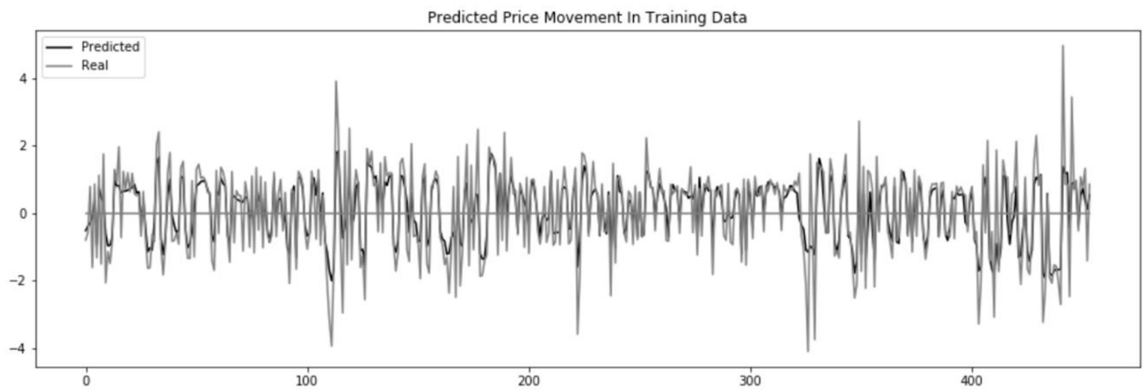
§ 5.3 Inside Market Movement Prediction Model

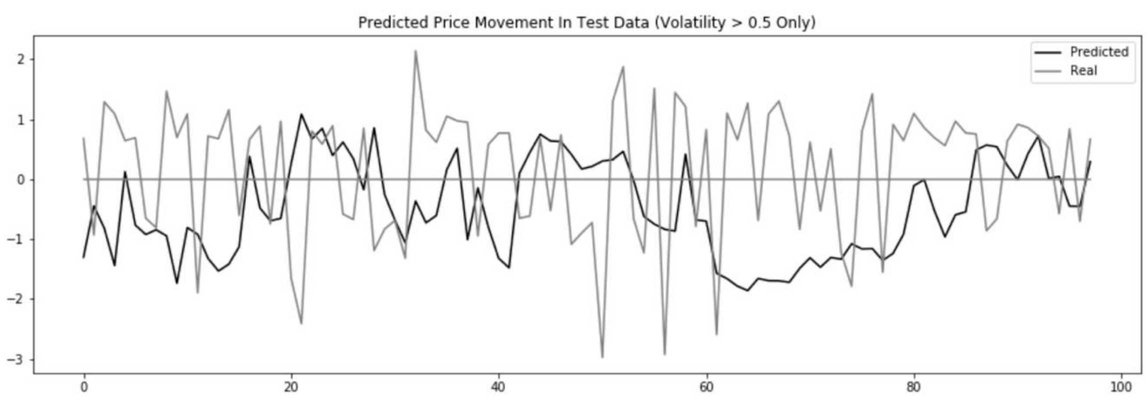
We explored a total of 7 models in this section again using S&P 500 historical data. As mentioned earlier, we used a total number of 140 technical indicators and 54 macro-economic factors as features. In order to improve computing efficiency, PCA was used to collapse all these features to 60, while capturing more than 95% variability in the data.

In the first model, we used 60-day lags of 60 PCs as input while training the model on historical day return data greater than 0.5% only. The second model used binary output as the y variable with the same LSTM modelling method.

The third model took 1-day lag of 60 PCs as input and the other conditions remained the same. The fourth model was a random forest model fitting the same features used in the third model.

The fifth model tried to predict market movement 5-days ahead using the LSTM model. The last two took 5-day lags of 60 PCs as input while differentiating with one another in volatility threshold.

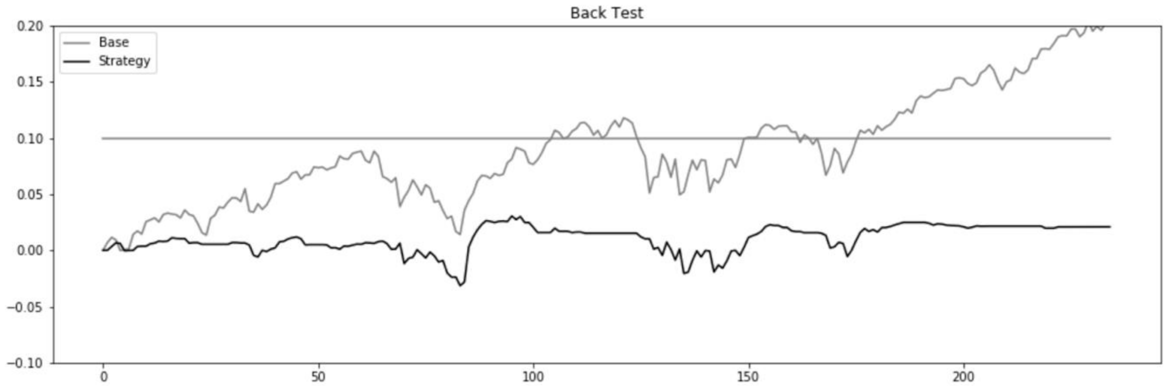




None of these results produced anything impressive, so to follow the rule of simplicity, a 5-day lags LSTM model is used.

§ 5.3 Result And Analysis

After these two models generated predictions for each index tracking one of the 10 sub sectors in the testing period, the values were used in the backtest to execute the trading positions. However, unfortunately, the result does not look great at all.



The cumulative return of the strategy was roughly 2%, while S&P 500 made a bullish rally during this period. There are two reasons for the disappointing results. Firstly, setting a threshold of volatility to decide when the strategy shall be executed actually limits the time when positions are taken. In other words, since predicted volatility for the most part exceeds the threshold only during very volatile time, positions are open in a short period, and for the rest time, it sits on cash. Secondly, models in neither parts seem to predict the future well. To be fair, they are slightly better than random guessing, however, in order to beat the market, slightly better than guessing are far from necessary.

Looking ahead, we should be exploring better market movement prediction models and features, or we should dive into the world of high frequency trading (HFT) and take advantage of the low latency which algorithmic trading is characterized with. HFT do not require predicting the future, nevertheless, they are technologically challenging in other respects.

Appendix

Market Volatility Drivers (54)	
Currency Pairs	**Bond Index & Spread**
GBPUSD	US High Yield
EURUSD	US Investment Grade
NZDUSD	Emerging Investment Grade
USDCHF	Emerging High Yield OAS
USDCAD	Emerging Investment Grade OAS
USDJPY	US High Yield OAS
USDCNY	US Investment Grade OAS
USDARS	US 10 Year -2 Year Gov Spread
USDBRL	
USDCLP	**Investors' Sentiments**
USDCOP	VIX
USDCZK	
USDDKK	**Commodity**
USDHKD	Corn
USDHUF	Coffee
USDIDR	Cocoa
USDILS	Live Cattle
USDINR	Cotton
USDKRW	Soybean
USDMXN	Sugar
USDMYR	Wheat
USDNOK	
USDPEN	**Energy & Metal**
USDPLN	Crude Oil
USDRUB	Natural Gas
USDSEK	Aluminum
USDSGD	Copper
USDTRY	Gold
USDTWD	Nickel
USDZAR BGN	Silver

Technical Indicators (140)				
AD	CDLLONGLINE	AROONOSC	TRIMA	CDLEVENINGDOJISTAR_5
ADD	CDLMARUBOZU	ADOSC	WMA	CDLEVENINGSTAR_5
ATAN	CDLMATCHINGLOW	APO	OBV	CDLMORNINGDOJISTAR_5
AVGPRICE	CDLONNECK	MFI	NATR	CDLMORNINGSTAR_5
BOP	CDLPIERCING	MINUS_DI	TRANGE	BBANDS_UPPERBAND
CDL3INSIDE	CDLRICKSHAWMAN	MINUS_DM	HT_DCPERIOD	BBANDS_MIDDLEBAND
CDL3LINESTRIKE	CDLSEPARATINGLINES	MOM	HT_TRENDMODE	BBANDS_LOWERBAND
CDL3OUTSIDE	CDLSHOOTINGSTAR	PLUS_DI	HT_DCPHASE	AROONDOWN
CDL3WHITESOLDIERS	CDLSHORTLINE	PLUS_DM	LINEARREG	AROONUP
CDLADVANCEBLOCK	CDLSPINNINGTOP	PRO	LINEARREG_ANGLE	HT_PHASOR_INPHASE
CDLBELTHOLD	CDLSTALLEDPATTERN	ROC	LINEARREG_INTERCEPT	HT_PHASOR_QUADRATURE
CDLCLOSINGMARUBOZU	CDLSTICKSANDWICH	ROCP	LINEARREG_SLOPE	HT_SINE_SINE
CDLCOUNTERATTACK	CDLTAKURI	ROCR	STDDEV	MACD_MACD
CDLDOJI	CDLTASUKIGAP	ROCR100	TYPPRICE	MACD_MACDSIGNAL
CDLDOJISTAR	CDLTHRUSTING	RSI	WCLPRICE	MACD_MACDHIS
CDLDRAGONFLYDOJI	CDLTRISTAR	TRIX	SQRT	MACDEXT_MACD
CDLENGULFING	CDLUNIQUE3RIVER	ULTOSC	LOG10	MACDEXT_MACDSIGNAL
CDLGAPSIDESIDEWHITE	CDLUPSIDEGAP2CROWS	WILLR	CDLDARKCLOUDCOVER_0	MACDEXT_MACDHIS
CDLGRAVESTONEDOJI	CDLXSIDEGAP3METHODS	DEMA	CDLEVENINGDOJISTAR_0	MACDFIX_MACD
CDLHAMMER	CEIL	EMA	CDLEVENINGSTAR_0	MACDFIX_MACDSIGNAL
CDLHANGINGMAN	COS	HT_TRENDLINE	CDLMORNINGDOJISTAR_0	MACDFIX_MACDHIS
CDLHARAMI	ATR	KAMA	CDLMORNINGSTAR_0	STOCH_SLOWK
CDLHARAMICROSS	BETA	MA	CDLDARKCLOUDCOVER_3	STOCH_SLOWD
CDLHIGHWAVE	CCI	MIDPOINT	CDLEVENINGDOJISTAR_3	STOCHF_FASTK
CDLHIKKAKE	CMO	MIDPRICE	CDLEVENINGSTAR_3	STOCHF_FASTD
CDLHOMINGPIGEON	CORREL	SMA	CDLMORNINGDOJISTAR_3	STOCHRSI_FASTK
CDLINVERTEDHAMMER	ADX	T3	CDLMORNINGSTAR_3	STOCHRSI_FASTD
CDLLONGLEGGEDDOJI	ADXR	TEMA	CDLDARKCLOUDCOVER_5	EWO

References

3.1. Cross-validation: evaluating estimator performance¶. (n.d.). Retrieved from https://scikit-learn.org/stable/modules/cross_validation.html

Bharadwaj, R. (2020, April 3). Machine Learning in Investment Management and Asset Management – Current Applications. Retrieved from https://emerj.com/ai-sector-overviews/machine-learning-in-investment-management-and-asset-management/

Chiang, H. (2019, December 26). Deep Learning Algorithms: The Future of Financial Investment? Retrieved from https://iknowfirst.com/rsar-deep-learning-algorithms-future-financial-investment

Fabozzi, F. J., & Markowitz, H. M. (n.d.). Equity Valuation and Portfolio Management. Retrieved from https://www.oreilly.com/library/view/equity-valuation-and/9780470929919/chap1-sec02.html

GICS - Global Industry Classification Standard. (2018, September 28). Retrieved from https://www.msci.com/gics

Kernel Ridge Regression¶. (n.d.). Retrieved from https://www.shogun-toolbox.org/examples/latest/examples/regression/kernel_ridge_regression.html#

Machine Learning: What it is and why it matters. (n.d.). Retrieved from https://www.sas.com/en_us/insights/analytics/machine-learning.html

Maklin, C. (2019, July 21). LSTM Recurrent Neural Network Keras Example. Retrieved from https://towardsdatascience.com/machine-learning-recurrent-neural-networks-and-long-short-term-memory-lstm-python-keras-example-86001ceaaebc

Mittnik, S., Robinzonov, N., & Spindler, M. (2015, April 28). Stock market volatility: Identifying major drivers and the nature of their impact. Retrieved from https://www.sciencedirect.com/science/article/pii/S0378426615000795

R, A. (2018, July 31). APPLYING RANDOM FOREST (CLASSIFICATION) - MACHINE LEARNING ALGORITHM FROM SCRATCH WITH REAL... Retrieved from https://medium.com/@ar.ingenious/applying-random-forest-classification-machine-learning-algorithm-from-scratch-with-real-24ff198a1c57

Sang, C., & Pierro, M. D. (2018, November 14). Improving trading technical analysis with TensorFlow Long Short-Term Memory (LSTM) Neural Network. Retrieved from https://www.sciencedirect.com/science/article/pii/S2405918818300539

Sanjay.M. (2018, November 13). Why and how to Cross Validate a Model? Retrieved from https://towardsdatascience.com/why-and-how-to-cross-validate-a-model-d6424b45261f