Certificate in Quantitative Finance

January 2018 Cohort

Final Project

July 12, 2018

Contents

OPTIMISED MODEL BASED ON GARCH AND SVR

July 12, 2018

1 Introduction

Historically, ARMA-GARCH model has been adopted by researchers to make predictions about electricity prices, or to forecast exchange rates. In the recent years, with the rise of machine learning, an increasing number of techniques related are being adopted to predict financial instruments prices more accurately. In this paper, ARMA-GARCH model is tested in the first place. Then two machine learning methods called Support Vector Classification (SVC) and Support Vector Regression (SVR) are respectively combined with GARCH to predict the return of index ETF products. As far as the test results can tell, the SVR-GARCH hybrid model proves to be the best one. The implementation is carried out using Python solely throughout the project.

2 Datasets

2.1 Stock Index ETF

For the sake of simplicity, only 11 financial time series are selected here. The focus here is on stock index ETF, so to speak, S&P 500 ETF (SPY) and the other 10 index ETFs with the top trading volume in the United States are put into use. All the prices are daily close price (subscription price).

2.2 Data Source

All the datasets are gathered and downloaded from Bloomberg Terminal, with the help of Spreadsheet Builder.

3 Methodology

3.1 ARMA-GARCH

The Autoregressive–Moving-Average (ARMA) model introduced by Peter Whittle(1951) and the Generalized ARCH (GARCH) model introduced by Bollerslev(1986) are generally accepted for measuring volatility in financial models.

ARMA models have an unconditionally non-random and constant variance, which typically serves well in effectively representing homoskedastic data. The GARCH models feature variable variance that is non-random when conditioning in the past. ARMA and GARCH can be combined to model the dynamics of stock indices and their volatilities.

3.1.1 Definition

The general linear autoregressive moving average (ARMA (p, q)) model for conditional mean is expressed as

$$y_t = \mu + \sum_{i=1}^{p} \rho_i * y_{t-i} + \sum_{j=1}^{q} \theta_j * \epsilon_{t-j} + \epsilon_t \tag{1}$$

where y_t is the time series needed to be modelled, μ is a constant, p is the number of autoregressive orders, q is the number of moving average orders, ρ_i is autoregressive coefficients, θ_j is moving average coefficients and ϵ_t is the error.

In the last formula, the term $\sum_{i=1}^{p} \rho_i * y_{t-i} + \sum_{j=1}^{q} \theta_j * \epsilon_{t-j}$ is the deterministic component that presents the forecast of the current state as the functions of past observations and errors. The term of error ϵ_t is the random component, which is commonly assumed to be zero mean and constant variance.

However, for some practical time series, the error terms ϵ_t do not satisfy the homoscedastic assumption of constant variance. The time-varying variance, which depends on the observations of the immediate past, is called conditional variance. The conditional variance of ϵ_t, σ_t^2, is defined as

$$\sigma_t^2 = Var_{t-1}(\epsilon_t) = E_{t-1}(\epsilon_t^2) \tag{2}$$

The generalized autoregressive conditional heteroscedasticity (GARCH (p, q)) prediction model for the conditional variance of ϵ_t is

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \beta_i * \sigma_{t-i}^2 + \sum_{j=1}^{q} \alpha_j * \epsilon_{t-j}^2 \tag{3}$$

with constraints $\sum_{i=1}^{p} \beta_i * + \sum_{j=1}^{q} \alpha_j < 1$, $\alpha_0 > 0$, $\beta_i \geq 0 (i = 1, 2, \ldots, p)$, $\alpha_j \geq 0 (j = 1, 2, \ldots, q)$.

So, the GARCH-ARMA model can be summarized in the following equations

$$y_t = \mu + \sum_{i=1}^{p} \rho_i * y_{t-i} + \sum_{j=1}^{q} \theta_j * \epsilon_{t-j} + \epsilon_t, \epsilon_t \sim N(0, \sigma_t^2) \tag{4}$$

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \beta_i * \sigma_{t-i}^2 + \sum_{j=1}^{q} \alpha_j * \epsilon_{t-j}^2 \tag{5}$$

$$\epsilon_t = \sigma_t Z_t, Z_t \sim N(0, 1) \tag{6}$$

3.1.2 Pros and Cons of Model

The simplicity and explicitness of ARMA-GARCH model structures have lots of advantages. It is shown that under the same confidence level, the ARMA-GARCH model can predict the more accurate confidence intervals compared to the traditional ARMA model. ARMA-GARCH model can capture volatility clustering effect of rates and hence is an appropriate model to forecast it.

Although ARMA-GARCH models can predict stock indices and their volatility, they assume that the time series have the homoscedastic property, and thus, ignore the heteroskedasticity of data, an essential feature of data, in time series forecasting. For the parameter estimation in ARMA–GARCH model, the quasi-maximum likelihood estimator based on Gaussian density (Gaussian-QMLE) is widely used due to its tractability and excellent performances. However, when the error distribution of ARMA–GARCH model is either skewed or leptokurtic, Gaussian–QMLE does not perform successfully.

3.2 Support Vector Machine

Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

3.2.1 Definition

More formally, a support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks like outliers detection. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

Whereas the original problem may be stated in a finite dimensional space, it often happens that the sets to discriminate are not linearly separable in that space. For this reason, it is proposed that the original finite-dimensional space be mapped into a much higher-dimensional space, presumably making the separation easier in that space. To keep the computational load reasonable, the mappings used by SVM schemes are designed to ensure that dot products may be computed easily in terms of the variables in the original space, by defining them in terms of a kernel function $K(x, y)$ selected to suit the problem. The hyperplanes in the higher-dimensional space are defined as the set of points whose dot product with a vector in that space is constant. The vectors defining the hyperplanes can be chosen to be linear combinations with parameters α_i of images of feature vectors x_i that occur in the data base. With this choice of a hyperplane, the points x in the feature space that are mapped into the hyperplane are defined by the relation: $\sum_i \alpha_i k(x_i, x) = constant$.

If $K(x, y)$ becomes small as y grows further away from x, each term in the sum measures the degree of closeness of the test point x to the corresponding data base point x_i. In this way, the sum of kernels above can be used to measure the relative nearness of each test point to the data points originating in one or the other of the sets to be discriminated. Note the fact that the set of points x mapped into any hyperplane can be quite convoluted as a result, allowing much more complex discrimination between sets which are not convex at all in the original space.

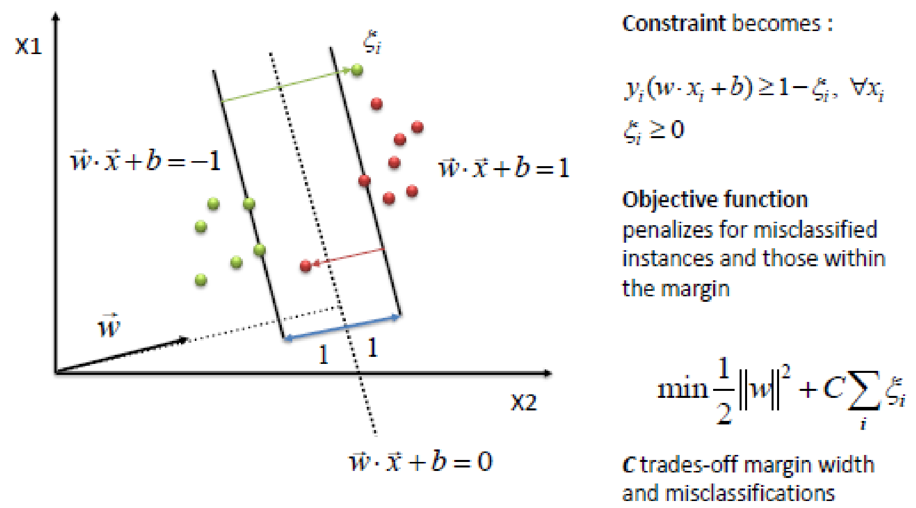


Figure 1: Support Vector Machine for Classification

3.2.2 Pros and Cons of Model

On one hand, SVM does not assume that there is a probability density function over the return series but it adjusts the parameters relying on the empirical risk minimization inductive principle. By comparing with other models, such as GARCH, which performs well when the data is drawn from a Gaussian distribution, but if the probability density function is not Gaussian, then SVM will give a better result.

On the other hand, SVM is time-consuming and computational demanding mainly because SVM is a binary classifier. Also, compared with logistic regression and neural networks, SVM has high algorithmic complexity and extensive memory requirements when it comes to large-scale tasks.

3.3 Support Vector Regression

Support Vector Machine can also be used as a regression method, maintaining all the main features that characterize the algorithm (maximal margin). The Support Vector Regression (SVR) uses the same principles as the SVM for classification, with only a few minor differences. First of all, because output is a real number it becomes very difficult to predict the information at hand, which has infinite possibilities. In the case of regression, a margin of tolerance (epsilon) is set in approximation to the SVM which would have already requested from the problem. But besides this fact, there is also a more complicated reason, the algorithm is more complicated therefore to be taken in consideration. However, the main idea is always the same: to minimize error, individualizing the hyperplane which maximizes the margin, keeping in mind that part of the error is tolerated.

3.3.1 Definition

Given training set $T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_l, y_l)\}$ where $x_i \in R^d$ and $y_i \in R$, the goal of SVR is to find the prediction function $f(x) = w^T\phi(x)$ (the bias term is dropped by preprocessing the data with zero mean). The formulation of SVR can be expressed as

$$\min_{w,\xi(*)} \frac{1}{2}||w||_2^2 + Ce^T(\xi + \xi^*) \tag{7}$$

s.t.

$$\langle\phi(x_i), w\rangle - y_i \leq \epsilon + \xi_i \tag{8}$$

$$y_i - \langle\phi(x_i), w\rangle \leq \epsilon + \xi_i \tag{9}$$

$$\xi_i, \xi_i^* \geq 0, i = 1, 2, \cdots, l. \tag{10}$$

In most cases, formula (7) can be solved easily in the following dual formulation.

$$\min_{\hat{a}} \frac{C}{2}\hat{a}^T H\hat{a} + g^T\hat{a} \tag{11}$$

s.t.

$$0 \leq \hat{a} \leq e \tag{12}$$

Here, $H = \begin{bmatrix} K & -K \\ -K & K \end{bmatrix}$, $K \in R^{l\times l}$ is the kernel function, $K_{ij} = K(x_i, x_j)$, $i, j \in \{1, 2, \cdots, l\}$, and $g = \begin{bmatrix} \epsilon e - Y \\ \epsilon e + Y \end{bmatrix}$, $\hat{a} = \frac{1}{C}\begin{bmatrix} \alpha_* \\ \alpha \end{bmatrix}$, where α^* and α are Lagrange multipliers.

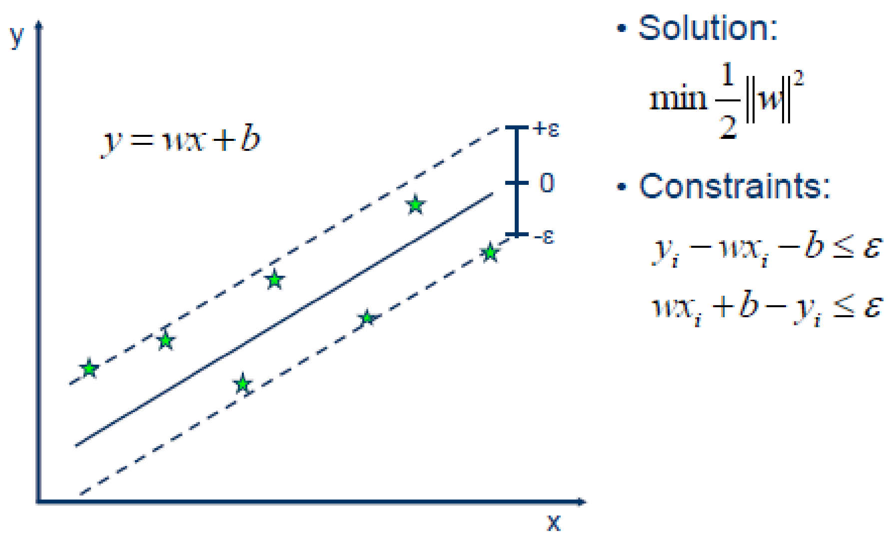


Figure 2: Support Vector Machine for Regression

3.3.2 Pros and Cons of Model

The main difference between SVC and SVR lies in the loss function and the main advantage of SVR is the ability to generate nonlinear decision boundaries through linear classifiers while having a simple geometric interpretation. The cost function in SVR is formulated by equality instead of inequality constraints and an LS loss function. This reformulation significantly simplifies the problem in such a way that the solution is estimated by solving a simple linear system instead of the QP problem.

For a given set of training data, the SVR finds a linear function in which the training data is considered either in the input space itself or mapped into the feature space via a kernel mapping such that the regressor is made as flat as possible while fitting the training data. SVR perfectly keeps SVM's inherent merits, which seeks two parallel hyperplanes based on the SRM principle and an ϵ-insensitive loss function.

However, one of its potential drawbacks is that it is not robust to outliers due to the LS loss function. In real-world applications, outliers may arise due to various reasons, such as mechanical faults, fraudulent behavior, instrument error or human error. In order to decrease the influence of outliers, many methods have been proposed to improve the robustness of SVR for nonlinear regression applications.

3.4 Trading Strategy and Backtesting Results

3.4.1 ARMA-GARCH

The first try is given to the ARMA-GARCH model. Nevertheless, the exact value of the index is difficult to predict, so only the trend is expected to be found.

To begin with, the moving average is applied to manage the data. The rolling window is set to be 5 days, of which the reasons are as follows. First, if the data of a long period is applied, the value of the result will approach 0 very easily because the long-term expectation usually converges to 0. Thus, the data for a long period is not suitable for this research. Second, ror the rolling window of 5 days, 10 days and 20 days, the 5-day rolling window is the best. Then the parameters p and q are determined based on Bayes'Theorem. However, when the 5-day rolling window is adopted, there is a great probability that the time series will fluctuate and the process of moving average will be irreversible. So the strategy should be changed to avoid these two situations. The data of average return for 5 days will be used as the predicted value. In the meantime, due to the small quantity of 5-day data, the maximum number of parameters used in ARMA model and GARCH model are both 3.

Additionally, before putting GARCH model into practice, it can be found that the price decreases when the return fluctuation is large. Therefore, ARMA model is used to calculate the predicted value of the return. Besides, GARCH is applied in volatility prediction. Then, if the predicted value of volatility is larger than the current volatility, then the prediction of a downward trend could be made and the predicted value of the return next day equals ARMA model minus 2 times the result of GARCH model. If the predicted value of volatility is lower than the present volatility, a downward trend is more likely. Then the predicted value for next day equals ARMA model plus 8 times the result of GARCH model. It should be noted here that the number 2 is a value chosen after comparing results given by different by other coefficients. The number 8 is a value chosen to make the model more sensitive to the bullish market. These two numbers can be dynamically changed according to real practice.

As for the trading strategy, if the predicted return for the next day is above 0, the investor can long the stock. On the contrary, if the value is below 0, the investor can short the stock. It should be noticed here that for simplicity, the transaction costs and other fees are not involved.

The backtesting result can be seen in the below picture. Please note that the red line represents the strategy result and the blue the "buy and hold" throughout the project.

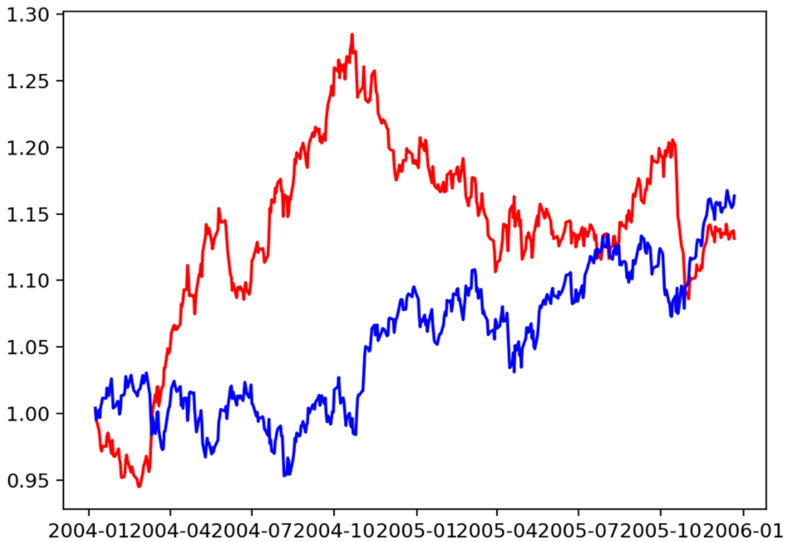


Figure 3: ARMA-GARCH hybrid model simple return (Data: SPY)

Apparently, the result is still far from perfect, as the strategy outperforms the market only during the bearish period, and when it comes to the bullish market, it stops being effective. The reasons should be categorized into the following points. First, even if GARCH does a great job in predicting volatility, the ARMA part cannot give a relatively accurate prediction of the return. Second, there is no dynamic adjustment about the buying and selling signals. Third, the ETF return is only closely related to the recent data, which means it is not so sensitive to the less recent data or, so to speak, the less recent data contains too much useless information. In sum, it is by no means a good model and should be abandoned.

3.4.2 SVM-GARCH

Then the attention is shifted towards the Support Vector Machine method, which is obviously one of the most popular machine learning techniques these days. It should be noted here only six factors have been used in the model: open price, close price, high price, low price 5-day rolling window and volatility. The kernel function is set to be RBF throughout the paper.

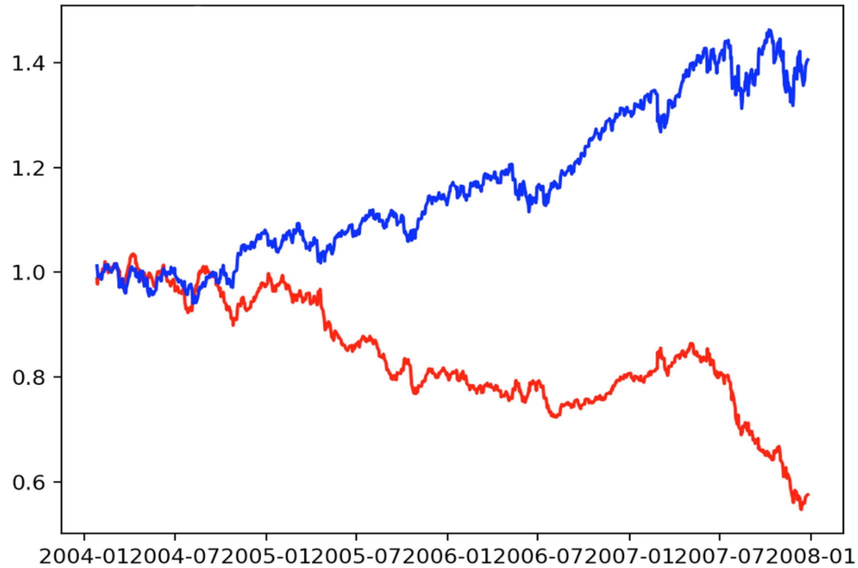


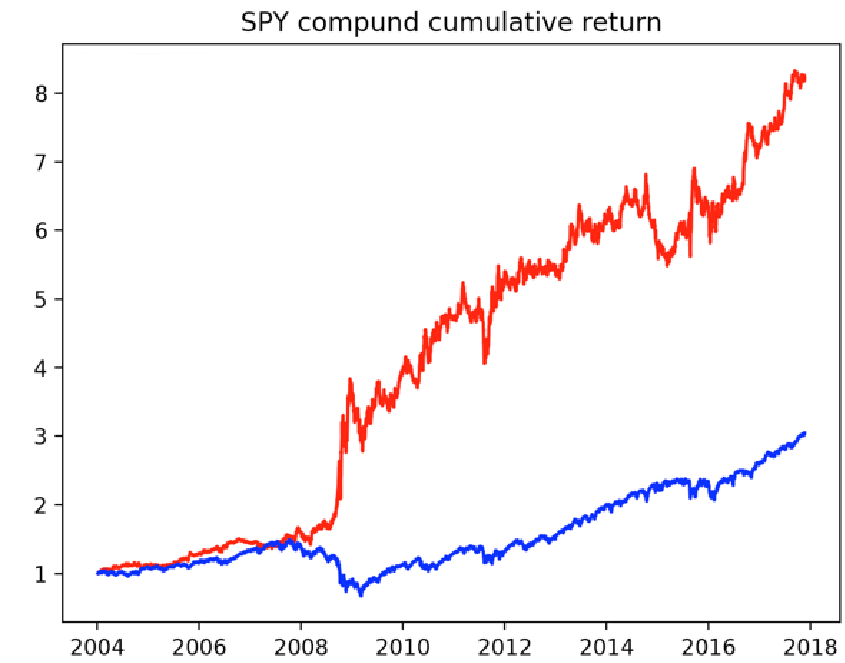
Figure 4: SVM-GARCH hybrid model simple return (Data: SPY)

Even during the prosperity before 2008, the SVM-GARCH fails to outperform the benchmark. It demonstrates a striking opposite direction of the benchmark, which c can be quoted as one of worst cases.

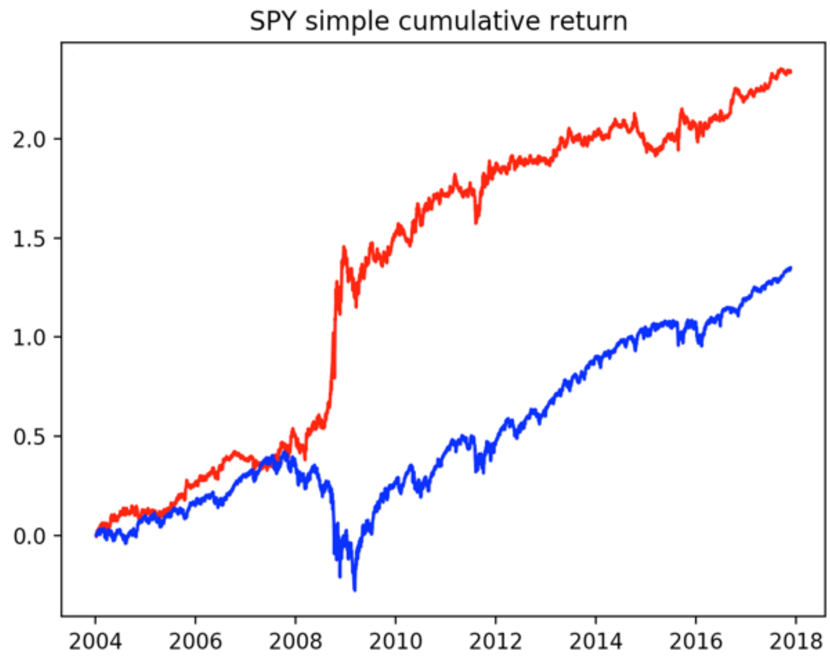
A few reasons can account for the utter failure. To begin with, technically speaking, SVM only works when provided with a huge number of features. However, due to the the limited features I can gather, it is impossible. Second, even given sufficient factors, the feature engineering of SVM tends to be time-consuming and highly complicated. Based on the limited time frame to complete the project, it is not likely to be put into practice for the time being. Besides, the multicollinearity among the four prices can also lead to inaccurate output.

3.4.3 SVR-GARCH

Now that it is no longer realistic to apply SVM or Support Vector Machine for Classification in my research for various reasons, another branch of support vector technique is taken into serious consideration. To be more specific, GARCH is tried to be combined with Support Vector Machine for Regression (with RBF kernel function) this time. The initial experiment was performed on SPY again to make a comparison with the previous result. No optimisation is applied in the model at the first attempt, and only two features, the 5-day-rolling window and volatility, are chosen for the model. The parameters are all default values.



(a) compound return



(b) simple return

Figure 5: SVR-GARCH hybrid model before optimisation (Data: SPY)

At the first glance, it appears much better than its SVM counterpart, in both cumulative return and simple return. However, anyone who takes a closer look will find that the strategy only outperforms the benchmark after the year of 2008 and lose its advantage during the 2008 global financial crisis and the bullish time before 2008. In addition, the lines are frightfully volatile compared with the benchmark.

3.4.4 Trading Strategy Optimisation

Trading execution:

$$BBC = 2, \quad sup\{BBC\} = 4, \quad inf\{BBC\} = 0$$

if $\quad P_{predict} < P_t$, and $Vol_{predict} \times f(BBC) > g(BBC, Vol_{last5days})$:

$$\text{Sell}: \ -1 \times h(BBC)$$

$$\text{where } f(BBC) = [1 - 0.1 \times (5 - BBC)],$$

$$g(BBC, Vol_{last5days}) = Vol_i, \quad i = BBC \in [0, 4], Vol_0 > Vol_1 > ... > Vol_5$$

$$h(BBC) = BBC \times 0.2 \times (1 + \frac{BBC}{10}), \quad inf\{h(BBC)\} = 0.2, \quad sup\{h(BBC)\} = 1$$

else:

$$\text{Buy}: \ +1 \text{ (full position)}$$

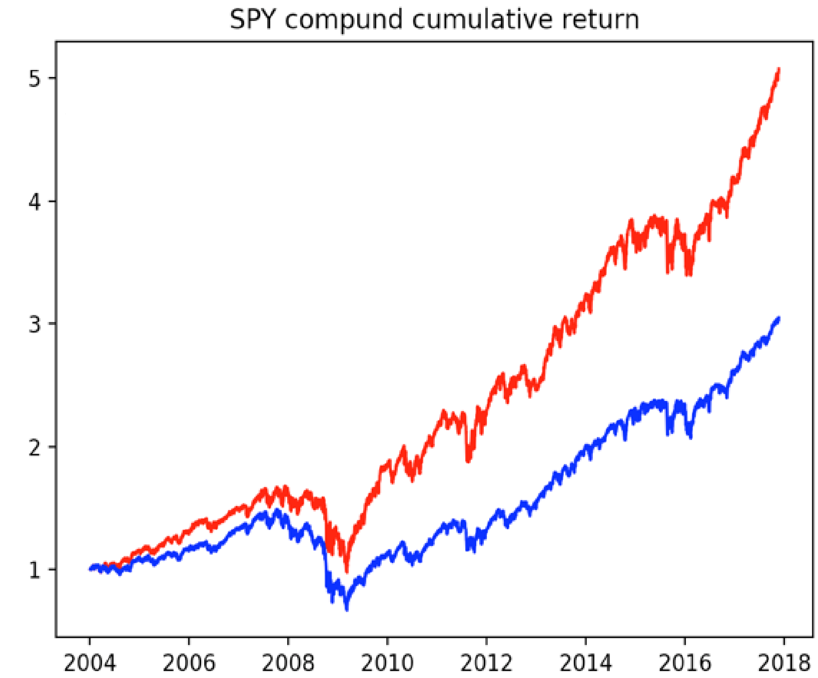
BBC Adjustment:
if $\quad P_{predict} > P_{t+1}$:

$$BBC = BBC + 2$$

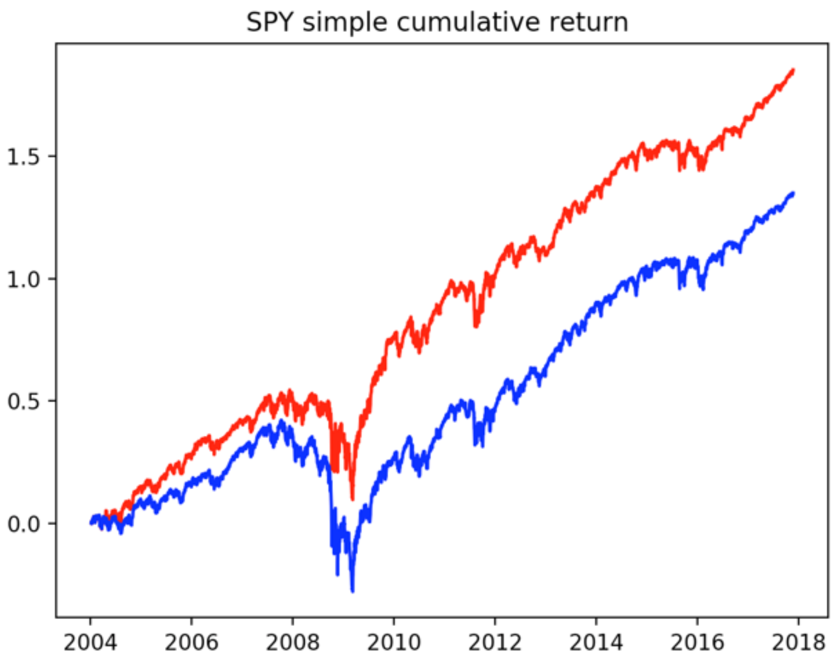
else:

$$BBC = BBC - 2$$

To address the aboved problem or to optimise the hybrid model, something called Bear-Bull-Coefficient (BBC) is introduced into the SVR-GARCH hybrid model. It is a coefficient which ranges from 0 to 4, with 0 represents the strongest market, 4 the weakest and 2 the initial value. The coefficient is dynamically changed according to the market performance, Further technical details about the idea can be found in the part of Trading Strategies. And then, after optimisation, the backtesting results show a picture which is completely different.



(a) compound return



(b) simple return

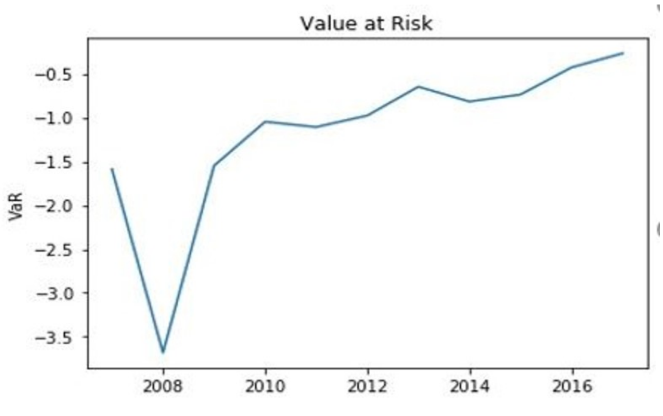
Figure 6: SVR-GARCH hybrid model after optimisation (Data: SPY)

Then another nine index ETFs with the highest trading volume are tested in the model. Note here Value at Risk has been calculated for the nine ETFs for better comparison.

(a) compound return

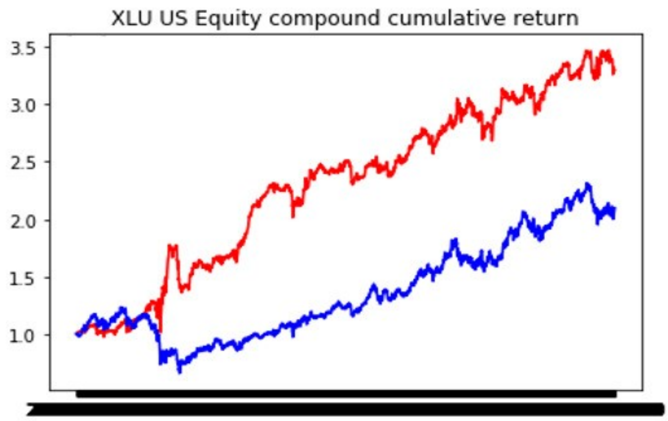
(b) simple return

Figure 7: SVR-GARCH hybrid model after optimisation (Data: USO)



(a) Value at Risk

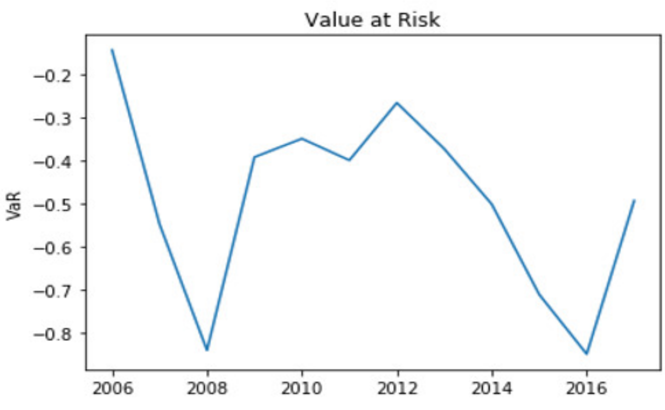
Figure 8: Value at Risk after optimisation (Data: USO)



(a) compound return

(b) simple return

Figure 9: SVR-GARCH hybrid model after optimisation (Data: XLU)



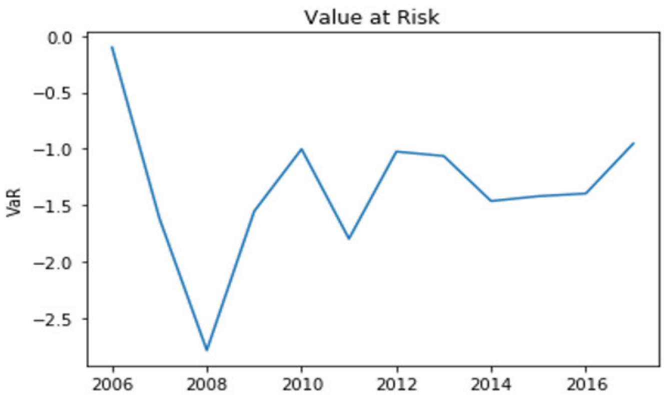
(a) Value at Risk

Figure 10: Value at Risk after optimisation (Data: XLU)

(a) compound return

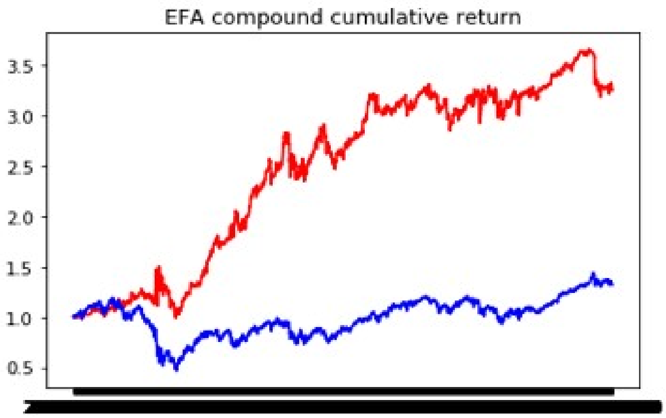
(b) simple return

Figure 11: SVR-GARCH hybrid model after optimisation (Data: XLE)

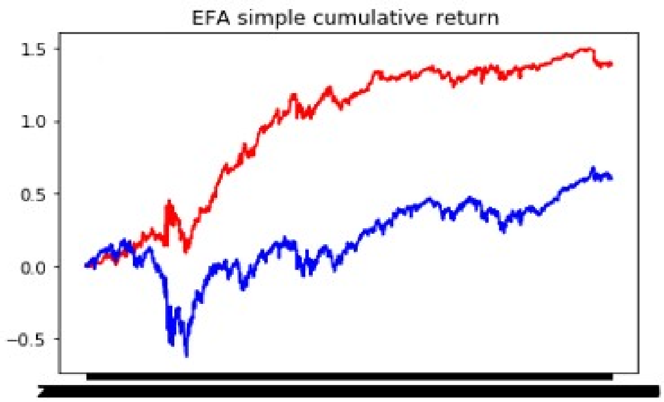


(a) Value at Risk

Figure 12: Value at Risk after optimisation (Data: XLE)

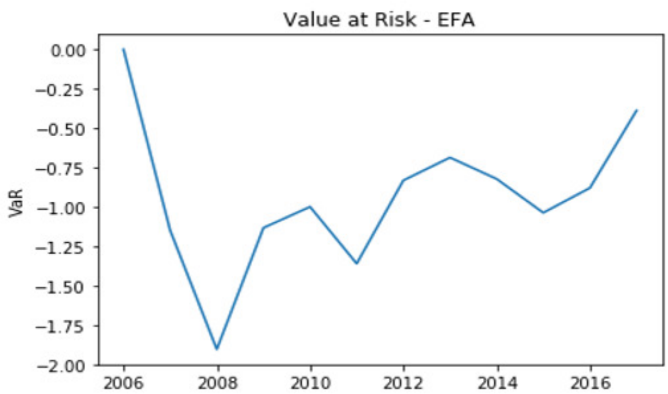


(a) compound return



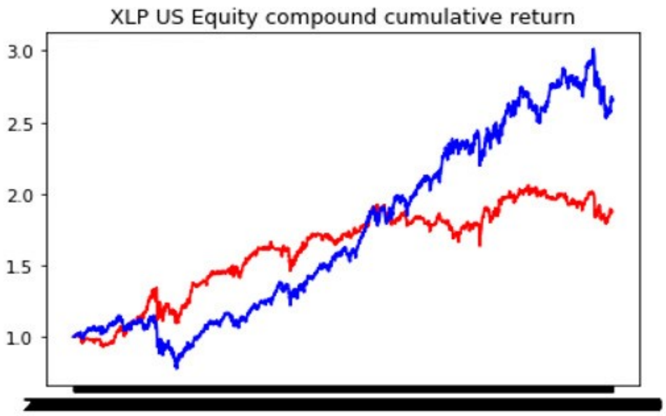
(b) simple return

Figure 13: SVR-GARCH hybrid model after optimisation (Data: EFA)

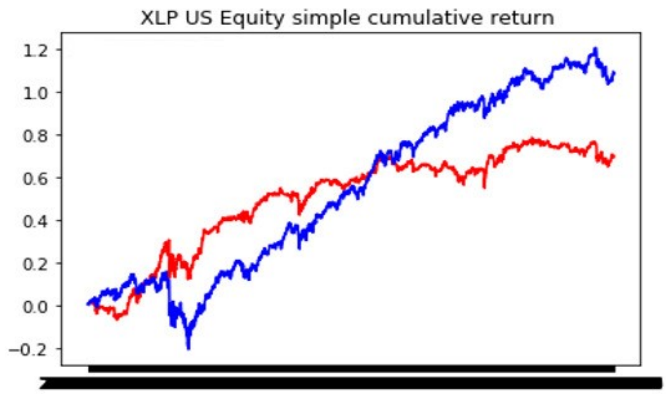


(a) Value at Risk

Figure 14: Value at Risk after optimisation (Data: EFA)

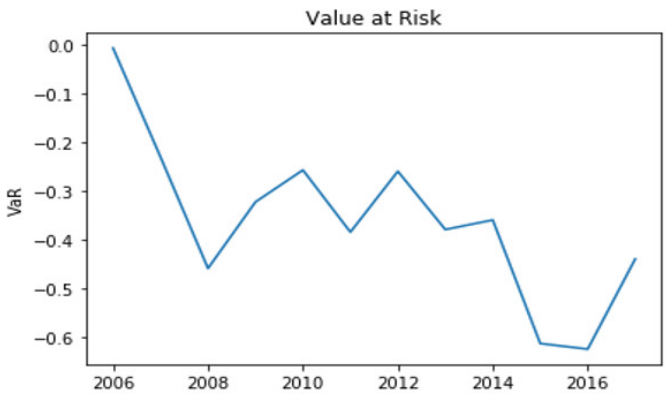


(a) compound return



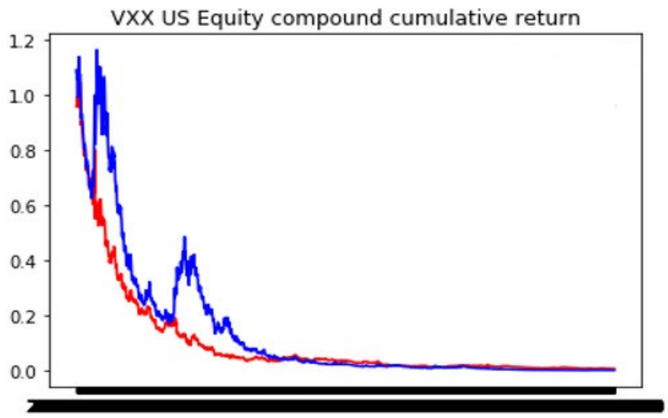
(b) simple return

Figure 15: SVR-GARCH hybrid model after optimisation (Data: XLP)



(a) Value at Risk

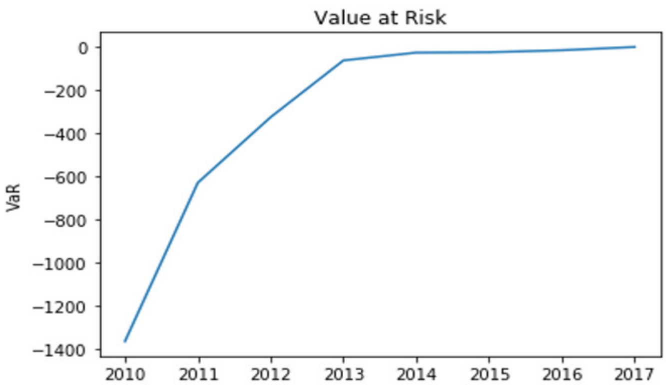
Figure 16: Value at Risk after optimisation (Data: XLP)



(a) compound return

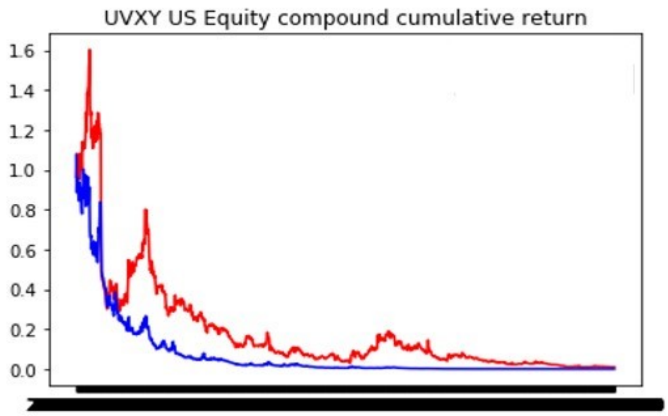
(b) simple return

Figure 17: SVR-GARCH hybrid model after optimisation (Data: VXX)



(a) Value at Risk

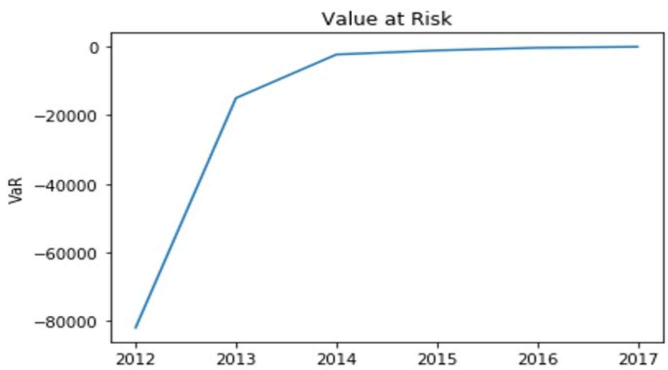
Figure 18: Value at Risk after optimisation (Data: VXX)



(a) compound return

(b) simple return

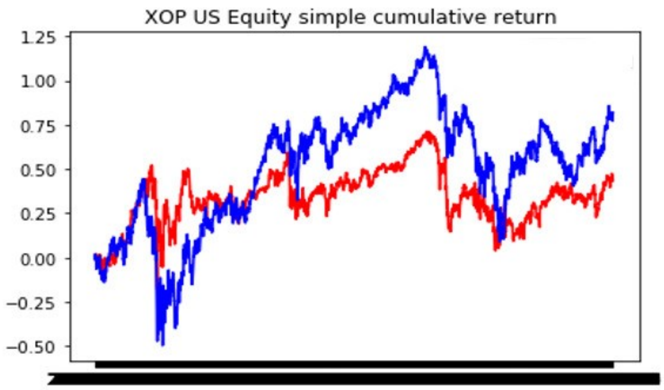
Figure 19: SVR-GARCH hybrid model after optimisation (Data: UVXY)



(a) Value at Risk

Figure 20: Value at Risk after optimisation (Data: UVXY)

(a) compound return

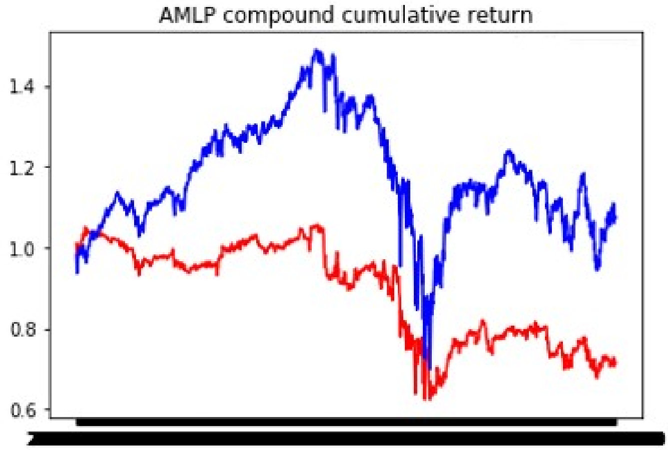


(b) simple return

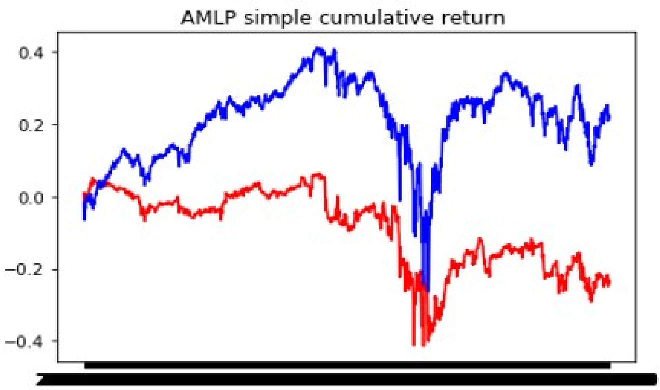
Figure 21: SVR-GARCH hybrid model after optimisation (Data: XOP)

(a) Value at Risk

Figure 22: Value at Risk after optimisation (Data: XOP)

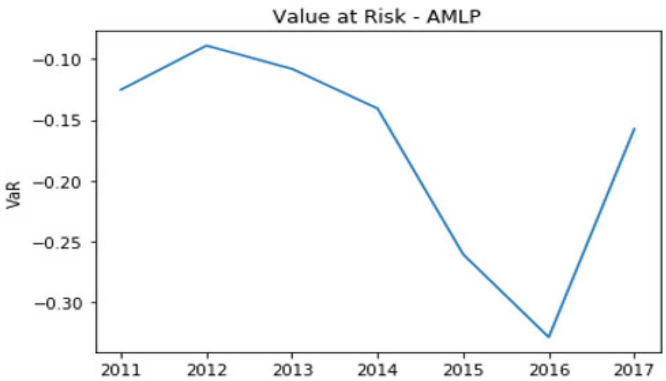


(a) compound return



(b) simple return

Figure 23: SVR-GARCH hybrid model after optimisation (Data: AMLP)



(a) Value at Risk

Figure 24: Value at Risk after optimisation (Data: AMLP)

It can be clearly observed that the backtesting results are not ideal for all the index ETFs. Even if the cumulative return of USO, XLU, XLE and EFA appear to be much better than "buy and hold", the advantage is not so obviously during the Global Financial Crisis, or in the bullish market. Besides, the strategy loses its effect when the price demonstrates tremendous volatility. As can be noticed in the plots of XLP, VXX, UVXY, XOP and AMLP. In the meantime, risk management can be a very difficult job based on the extremely high value at risk. The most typical cases are VXX and UVXY.

ETF / INDICATOR	SPY	USO	XLU	XLF	XLE	EFA
Annual Return	13.360%	1.773%	0.109%	14.284%	11.492%	12.008%
Max Drawdown	-44.889%	-89.310%	-25.700%	-71.760%	-45.380%	-36.910%
Sharpe Ratio	0.598	0.006	0.614	0.420	0.404	0.366
Information Ratio	0.556	0.298	0.206	0.246	0.160	0.213
Winning Odds	54.993%	47.630%	49.420%	50.150%	49.280%	48.920%

Table 1: Five main indicators of test results after optimisation for SPY, USO, XLU, XLF, XLE and EFA

ETF / INDICATOR	XLP	VXX	UVXY	XOP	AMLP
Annual Return	5.9429%	-45.9432%	-25.7815%	2.9400%	-3.1910%
Max Drawdown	-18.450%	-415.660%	-266.180%	-69.730%	-22.850%
Sharpe Ratio	0.319	-0.986	-0.287	0.043	0.988
Information Ratio	-0.274	0.213	0.753	-0.099	0.374
Winning Odds	47.710%	45.120%	45.860%	48.540%	50.850%

Table 2: Five main indicators of test results after optimisation for XLP, VXX, UVXY, XOP and AMLP

The two tables above present the five main indicators of backtesting results after optimisation. The eleven ETFs are divided into two group for comparison. Table 1 consists of those produce positive annual return, Sharpe Ratio and information ratio, whilst Table 2 gives a relatively unsatisfactory result of the strategy.

They only further support the conclusion that the strategy is not so stable across all the index ETFs. Even it demonstrates comparatively better performance in Table 1, the max drawdown is surprisingly high and way too risky.

3.4.5 Risk Management

Estimating the risk of loss to an algorithmic trading strategy, or portfolio of strategies, is of extreme importance for long-term capital growth. Many techniques for risk management have been developed for use in institutional settings. One technique in particular, known as Value at Risk or VaR, is put into use in the risk management of the strategy. The VaR equation is expressed as below:

$$VaR_{\alpha}(x) = \inf\left(x \in \mathbb{R} : \mathbb{F}_x(x) > 1 - \alpha\right)$$

VaR provides an estimate, under a given degree of confidence, of the size of a loss from a portfolio over a given time period. A "portfolio" can refer to a single strategy, a group of strategies, a trader's book, a prop desk, a hedge fund or an entire investment bank. The "given time period" will be chosen to reflect one that would lead to a minimal market impact if a portfolio were to be liquidated.

The application of VaR in the project has addressed the worst loss at the 95% confidence level for all the exchange-traded funds discussed. Beginning with the daily returns over a year, the 95th percentile worst loss is recorded for each year. This provides the movement of the value at risk for each year for each ETF over its operation period.

It is observed that the VaR behaves almost similarly across the ETFs. This shows that ETFs usually do not vary much with their riskiness, so it is plausible that the choice across the ETFs be based majorly on returns.

4 Conclusion

It is the very first time of mine to apply machine learning techniques into practice and combine them with time series models. The P&L obtained from the strategy can be tremendously outstanding but still far from atable and perfect. It should be pointed out that the validity of results for real trading is debatable due to the following reasons:

(1) Transaction costs and other fees were never taken into consideration. However, in real world, if the positions taken are very high, they can have a huge impact on the P&L.It should not have been ignored in the project.

(2) Further research about Support Vector Machine should be conducted in the future. Careful feature selection will be the main focus if provided with a sufficient number of factors.

(3) Due to the limited time, the strategy was not backtested over a longer timeline.

(4) No work has been done about the parameter tuning to further optimise the results. Probably the results will be more elegant and stable after it.

(5) The future work will involve testing on other ETF products, individual stocks and commodities to see whether something more promising and interesting could come into the view.

References

Cortes, Corinna, and Vladimir Vapnik, 1995. Support-vector networks. Machine Learning, 20(3), 273–297.

Chen, S and KIHO, J, 2007. Support Vector Regression Based GARCH Model with Application to Forecasting Volatility of Financial Returns, 1-2

Steve R. Gunn, 1998. Support Vector Machines for Classification and Regression, 1.

Manish Kumar, 2002. Forcasting Stock Index Movement/ A Comparison of Support Vector Machines and Random Forest, 1-2, 5.

Altaf Hossain, Faisal Zaman, M. Nasser and M. Mufakhkharul. IslamComparison of GARCH, Neural Network and Support Vector Machine in Financial Time Series Prediction, 4.

Phichhang Ou and Hengshan Wang, 2010. Financial Volatility Forecasting by Least Square Support Vector Machine/ Based on GARCH, EGARCH and GJR Models: Evidence from ASEAN Stock Markets. International Journal of Economics and Finance, 2(1), 51-56.

Yi Yang, Yao Dong, Yanhua Chen and Caihong Li, 2014. Intelligent Optimized Combined Model Based on GARCH and SVM for Forecasting Electricity Price of New South Wales, Australia. Hindawi, 2014, 1-8.

Mergani Khairalla, Xu-Ning and Nashat T. AL-Jallad, 2017. Hybrid Forecasting Scheme for Financial Time-Series Data using Neural Network and Statistical Methods. International Journal of Advanced Computer Science and Applications, 8(9), 319-320, 326.

Fernando Perez-Cruz, Julio A Afonso-Rodriguez and Javier Giner, 2003. Estimating GARCH models using support vector machines. Quantitative Finance, 3(2003), 1-2, 9.

R. F. Engle. 1982. Autoregressive conditional heteroskedasticity with estimates of the variance of United Kingdom inflation, Econometrica, 50, 987–1007.

T. Bollerslev, 1986. Generalized autoregressive conditional heteroskedasticity, Journal of Econometrics, 31(3), 307–327.

W.C. Wong, F. Yip, and L. Xu. 1998. Financial Prediction by Finite Mixture GARCH Model," Proceeding of Fifth International Conference on Neural Information Processing, 1351–1354.

Michael Nwogugu, 2005. Further critique of GARCH/ARMA/VAR/EVT Stochastic-Volatility models and related approaches, Applied Mathematics and Computation,182(2), 1735-1748.

Radha, S. and Thenmozhi, M. 2006. Forecasting Short Term Interest Rates Using Arma, Arma-Garch and Arma-Egarch Models. Indian Institute of Capital Markets 9th Capital Markets Conference Paper.

A.A.P. Santos, N.C.A. Costa Jr., and L.S. Coelho. 2007. Computational intelligence approaches and linear models in case studies of forecasting exchange rates, Expert Syst. Appl. 33(4), 816–823.

Jeongcheol Ha, Taewook Lee. 2011. NM–QELE for ARMA–GARCH models with non-Gaussian innovations, Statistics and Probability Letters, 81(6), 694-703.

Heping Liu, Jing Shi. 2013. Applying ARMA–GARCH approaches to forecasting short-term electricity prices, Energy Economics, 37,152-166.

Xie, H., Li, D., Xiong, L. 2016. Exploring the regional variance using ARMA-GARCH models. Water Resources Management, 30(10), 3507-3518.

Neha, C. R., Tiwari, P. 2016. Forecasting volatility spillover of information technology sector stocks in india: An application of ARMA and GARCH model. International Journal of Business Analytics and Intelligence, 4(2), 36-44.

Dr.Alonso Pena, Value at Risk and Expected Shortfall

Dr.Stephen Taylor, Volatility Models: The ARCH Framework

https://www.csie.ntu.edu.tw/~cjlin/papers/linear-svr.pdf

http://www.saedsayad.com/data_mining_map.htm