AMS518 Project

Portfolio replication of Taiwan 50 index using stochastic volatility with heavy-tailed distribution as rebalancing signal

Chi-Sheng Lo
Student ID: 114031563

Abstract
This project aims to achieve portfolio replication of Taiwan's TW50 index with mixed integer linear programming (MILP) and to examine its post-rebalancing performance right after an unexpected volatility shock. In this MILP, the objective is to minimize the maximum absolute difference between tracking portfolio and index returns subject to the constraints of cardinality, buy-in, budget, and transaction costs. The volatility model is estimated by a modified stochastic volatility model that incorporates not only non-constant volatility and serial dependence, but also heavy-tailed distribution. Additionally, a series of sub-sample analysis is provided to compare the post-rebalancing performance at different periods with both in-sample and out-of-sample backtestings. The sub-sample analysis is based on the rebalancing timing triggered by the shock of volatility jump. Since the tracking error between the in-sample and the out-of-sample is consistently narrow, we can conclude that the model is suitable for tracking the TW50 with strong potential to outstrip the index. Most importantly, active rebalancing using volatility shock as signal is proven to outperform the passive portfolio.

Keywords: Mixed-integer linear programming, index tracking, state-space, Bayesian method, stochastic volatility, heavy-tailed distribution

JEL classification: C11, C22, C61

1. Introduction

1.1 Objective

In this project, I will examine whether jump in volatility is a good signal for readjusting the allocation of a tracking portfolio which is supposedly a passive strategy that tries to mirror the performance of an index. Since Bloom (2009) showed that unexpected shock is highly correlated with economic activity including the financial market, our long-only portfolio should make adjustment in accordance with the latest regime switch triggered by the volatility shock. Furthermore, since financial markets tend to exhibit fat-tailed distribution, I will use a volatility model that best fits this phenomenon.

The formulation of my tracking portfolio will be based on the mixed-inter linear programming (MILP) that aims to minimize the maximum absolute difference between tracking portfolio and index returns subject to the constraints of cardinality (one-third of index members), buy-in, budget, and transaction costs.

Altogether, I essentially combine the cutting-edge topics of econometrics and operations research into this project.

1.2 Dataset

My sample spans from Jan 3rd, 2011 to Oct 4th, 2021 which has ten years (2641 daily prices) of data. The index that I track is the FTSE TWSE Taiwan 50 index (TW50). The TW50 is composed of 50 largest companies by the market capitalization among all the 948 listed companies. In fact, it covers 70 percent of the entire listed stocks space by market capitalization. Thereby, it is the most narrowed defined index targeting at the most representative "blue-chips" firms in Taiwan. The entire data including the index and all stocks time series will be collected from CMoney which is a financial data provider in Taiwan. The overview about the TW50 index is shown below.

Table 1. Overview of TW50

Index Universe	Listed companies of Taiwan Stock Exchange
Weighting Method	Free Float Market Capitalization
Base Date	2002.04.30
Launch Date	2002.10.29
Base Value	5,000
Calculation Frequency	Every 5 seconds
Number of Constituents	50
Periodic Review	March, June, September, December

1.3 Computational tool

The portfolio replication is executed in the Portfolio Safeguard (PSG) for R language. The stochastic volatility with heavy-tailed distribution is estimated in the MATLAB.

2. Model formulation

2.1 Volatility estimation: Evolution from SSM → SVM → HTSVM

2.1.1 State-space model (SSM)

We should discuss the SSM first since the SVM is part of the state space model (SSM) family pioneered by Kalman (1960). The most conventional one is the linear SSM which has been commonly used for times series estimation and forecasting. The SSM can be simply expressed as:

$y_t = \alpha_t + \varepsilon_t$, $\varepsilon_t \sim N(0, \sigma^2)$ (1)

$\alpha_t = \alpha_{t-1} + \mu_t$, $\mu_t \sim N(0, \sigma^2 q^2)$ (2)

Equation (1) is the first level of the SSM and is called the measurement equation which contains the observation and where the observable y_t is the sum of unobserved α and the error term. The subscript t is a time series notation; for example, t and t – 1 represent the current period and previous one period, respectively. Equation (2) is the second level of SSM where α_t is the sum of its previous one period and the error term. Equation (2) can be either called the state or transition equation that captures the evolution of the state. It is called the state of the system because the future will depend only on the current state and future input. Both error terms in equations (1) and (2) are normally distributed as well as mutually and serially independent. The variance $\sigma^2 > 0$ and the signal-to-noise ratio $q \geq 0$ are fixed and unknown.

2.1.2 Stochastic volatility model (SVM)

On the contrary, the SVM, introduced by Taylor (1986), is a nonlinear state space model where the measurement equation is nonlinear. The SVM is especially designed for financial time series that are subject to volatility clustering. One major feature is that the SVM can accommodate time-varying volatility that breaks the assumption of constant variance in the standard volatility model. The SVM can be described as:

$y_t = \mu + \varepsilon_t^y$, $\varepsilon_t^y \sim N(0, e^{h_t})$ (3)

$h_t = \mu_h + \phi_h(h_{t-1} - \mu_h) + \varepsilon_t^h$, $\varepsilon_t^h \sim N(0, w_h^2)$ (4)

Additionally, the log-volatility h_t follows a stationary AR(1) process with $|\phi_h| < 1$ and unconditional mean μ_h.

2.1.3 SVM model with heavy-tailed error distribution (HTSVM)

The HTSVM is an extension of SVM. Chan and Hsiao (2013) stressed that since Gaussian distribution used by the conventional volatility model has lack of mass for more extreme values, there is a need for adding a new weapon which can fully capture heavy-tailed distribution. Watanabe and Asai (2001) also showed that HTSVM can fit the financial data much better than its counterparts such as the normal and the generalized error distributions. Now, consider the HTSVM:

$y_t = \mu + e^{\frac{1}{2}h_t}\lambda_t^{\frac{1}{2}}\varepsilon_t$, $\varepsilon_t \sim N(0, 1)$ (5)

$h_t = \mu_h + \phi_h(h_{t-1} - \mu_h) + \varsigma_t$, $\varsigma_t \sim (0, \sigma_h^2)$, $h_1 \sim N\left(\mu_h, \sigma_h^2/(1-\Phi_h^2)\right)$ (6)

Where ε_t, ς_t, *and* λ_t are independent. The model is subject to $|\phi_h| < 1$. Since HTSVM adopts the student's t distribution where if $(\lambda_t|v) \sim IG(v/2, v/2)$, then $\tilde{\varepsilon}_t = \lambda_t^{\frac{1}{2}}\varepsilon_t$ has a standard student's t distribution.

The HTSVM must also allow for persistence through an MA(1) error process in which:

$y_t = \mu + u_t$ (7)

$u_t = \varepsilon_t + \psi\varepsilon_{t-1}, \varepsilon_t \sim \mathrm{N}(0, \lambda_t e^{h_t})$ (8)

Subject to $\varepsilon_0 = 0$ and $|\psi| < 1$

Notations:

u: average daily return

μ_h: unconditional mean (expected value)

y: observation

h_t: log volatility

Φ_h: First-order autoregression coefficient

ψ: Moving average coefficient

σ_h^2: Variance

λ: scale mixture variable

v: degree of freedom parameter

2.2 Index tracking

In this index tracking formulation based upon MILP, the main objective is to minimize the maximum absolute error between the tracking portfolio and the index. The linear programming problem can be express as:

$\min_{\vec{x}} \varepsilon_{MAX}(\vec{x}) = \min_{\vec{x}} \max_{1 \le t \le T} |L_t(\vec{x})|$ (9)

Where $\varepsilon_{MAX}(\vec{x})$ is the maximum absolute error and thereby $\min_{\vec{x}} \varepsilon_{MAX}(\vec{x})$ is the minimization of maximum absolute error. $L_t(\vec{x}) = \theta_{t0} - \sum_{i=1}^{N} \theta_{ti} X_i$.

Subject to

Cardinality constraint (restricts the number of assets in the rebalanced portfolio):

$\sum_{i=1}^{N} \delta(x_i) \le \mathrm{K}$ (10)

Buy-in constraint (all non-zero positions $\ge \sigma$):

$\sum_{i=1}^{N} \beta_\sigma^+(x_i) \le 0$ (11)

Rebalance portfolio + transaction cost constraint:

$\sum_{i=1}^{N} x_i + \sum_{i=1}^{N} \partial_i \left|x_i - x_i^0\right| + \mathrm{A}\sum_{i=1}^{N} \delta\left(\left|x_i - x_i^0\right|\right) \le \mathrm{C}$ (12)

Total transaction cost constraint:

$\sum_{i=1}^{N} \partial_i \left|x_i - x_i^0\right| + \mathrm{A}\sum_{i=1}^{N} \delta\left(\left|x_i - x_i^0\right|\right) \le \gamma\mathrm{C}$ (13)

$x_i \ge 0, \mathrm{I} = 1, \ldots, \mathrm{N}$ (14)

Where variable cost: $\sum_{i=1}^{N} \partial_i \left|x_i - x_i^0\right|$; fixed cost: $\mathrm{A}\sum_{i=1}^{N} \delta\left(\left|x_i - x_i^0\right|\right)$

Notations:

a_i = variable cost coefficient

N = number of assets

T = time period

X_i^0 = initial value of asset I in tracking portfolio

C = $\sum_{i=1}^{N} X_i^0$ = total value of current tracking portfolio

δ = indicator function

Y_t = price of index at period t = 0,…, T.

V_{it} = price of asset I at period t = 0,…, T, I = 1,…, I.

$\vec{x}$ = the vector of values of assets in the rebalanced tracking portfolio
γ = limit on fraction of current portfolio value to be spent on transaction costs, $0 < \gamma < 1$
$L_t(\vec{x})$ = underperformance of the portfolio at time t
$\varepsilon_{MAX}(x_i)$ = Maximum Absolute Error

3. Description of solution methods

3.1 Solution method for HTSVM

The entire numerical solution for HTSVM requires the Bayesian method that utilizes a branch of Markov Chain Monte Carlo (MCMC) namely the Metropolis-Hasting algorithm (MHA) which is estimated in MATLAB. The burn-in (iterations) and loop for the MHA of MCMC are set at 1000 and 5000, respectively, which are arbitrary. Like Monte Carlo integration and Gibbs sampling, the MHA produces a sequence of draws $\theta^{(r)}$ for r = 1, …, R with the property $\hat{g} = \frac{\sum_{r=1}^{R} g(\theta^{(r)})}{R}$ converges to E[g(θ)|y] as R goes to infinity. It involves drawing from a convenient density related to the importance function, referring to as the candidate generating density. Candidate draws of $\theta^{(r)}$ are either accepted or rejected with a certain probability referred to as an acceptance probability. If they are rejected, then $\theta^{(r)}$ is set to $\theta^{(r-1)}$.

Furthermore, the objective of inference in Bayesian analysis is the posterior p(θ|y); in this project, the posterior means, standard deviation, and quantiles of model parameters will be generated from MHA. For instance, p(β, h|y) = p($\beta|h, y$)p(h|y). We will first draw h from p(h|y) and then draw β from p($\beta|h, y$) to obtain a draw from p(β, h|y).

The posterior draws are:
1. p($u|y, h, \lambda, v,\ u_h, \Phi_h, \sigma_h^2$) = p($u|y, h, \lambda$)
2. p($h|y, \lambda, u, v, u_h, \Phi_h, \sigma_h^2$) = p(h|y, λ, u, u_h, σ_h^2)
3. p($\lambda|y, h, u,\ u_h, \Phi_h, \sigma_h^2$) = $\prod_{t=1}^{T}(\lambda_t|y_t, h_t, u, v$)
4. p($v|y, h, \lambda,\ u, u_h, \Phi_h, \sigma_h^2$) = p($v|\lambda$)
5. p($\sigma_h^2|y, h, \lambda,\ u, v, u_h, \Phi_h$) = p($\sigma_h^2|h, u_h, \Phi_h$)
6. p($u_h|y, h, , \lambda,\ u, v, \Phi_h, \sigma_h^2$) = p($u_h|h, \Phi_h, \sigma_h^2$)
7. p($\Phi_h|y, h, , \lambda,\ u, v, u_h, \sigma_h^2$) = p($\Phi_h|h, u_h, \sigma_h^2$)

Prior parameters are set to make prior means E(μ) = 0, E(ψ) = 0, E(μ_h) = 0, E(Φ_h) = 0.95, and E(σ_h^2) = 0.02. Estimation is done by combining the samplers from above. Thereby, let z = H_ψ^{-1}y, then $(z|h, \lambda, \psi, u) \sim \mathrm{N}\left(uH_\psi^{-1}1, \sum_z\right)$ where: u, v, u_h, Φ_h, and σ_h^2 are independent prior distribution and λ is a scale mixture variable. The MATLAB code is shown in section 7.2 of appendix.

3.2 Solution method for index tracking

The problem statement discussed in section 2.2 is formulated with the PSG nonlinear functions developed by Professor Stan Uryasev. The PSG is an optimization package for solving nonlinear and mixed-integer nonlinear optimization problems and can be executed in several programming environments such as C++, MATLAB, Run-File for Windows, Monitor for Ubuntu, and R. Among them, I select R as my main platform. Eventually, I will obtain the optimal weights for the selected stocks determined by my problem statements with PSG and then I will calculate the tracking portfolio based on the optimal weights on excel spreadsheet. In this problem, I assume

that I can only afford to allocate one-third of the entire TW50 index stocks; therefore, the cardinality constraint is set at 17. Please read below for code and modification.

Table 2. Main function in PSG

Function name	String in PSG Code
Maximum	max_risk
Mean Absolute Error	meanabs_err
Buyin Positive	buyin_pos
Polynomial Absolute	polynom_abs
Cardinality	cardn

Maximum = $\max\{|L_j(\vec{x})|\}$, Mean Absolute Error = $\frac{1}{J}\sum_{j=1}^{J}|L_j(\vec{x})|$, Polynomial Absolute = $\sum_{i=0}^{I} a_i|(\vec{x},\vec{\theta_j})|$, Polynomial Absolute = $\sum_{i=0}^{I}\alpha_i\,|x_i - x_i^0|$, Cardinality = $\sum_{i=0}^{I}\delta\,(x_i)$

PSG code for the problem statement:

```
minimize
  max_risk(matrix_inmmax)
Constraint: <=17
  cardn_pos(0.01, matrix_ksi)
Constraint: <= 0
  buyin_pos(0.01, matrix_ksibuy)
Constraint: <= 20000000
  linear(matrix_ksi)
  +variable(trcost)
Constraint: <= 76000
  variable(trcost)
Constraint: <= 0
  -variable(trcost)
  +0.01*polynom_abs(matrix_ksipol)
  +100*cardn_pos(0.01, matrix_ksipol)
  +100*cardn_neg(0.01, matrix_ksipol)
Box: >= 0
Solver: precision=7, stages =30
```

Total transaction cost

The transaction cost is totally based on the actual setting and regulation in Taiwan.
Total transaction cost = commission + stock transaction tax. Note that brokerage commission is 0.1424% and tax is 0.3% of the total value of stocks traded. Hence, the total transaction constraint is set at around $76000 (In my calculation, the actual number is $75225).

Other adjustments in PSG

kpol = $Shares_{under\ budget\ constraint}|x_i - x_i^0|$ and KB (KBUY) = $1000000; however, the total budget constraint is set at 20000000 (In my calculation, the actual number is $17000000).

Multi-period, in-sample, and out-of-sample analysis

There will be one full-sample and three sub-sample analyses divided by the volatility shock. Furthermore, there will be in-sample and out-of-sample analysis each one as well.

4. Results

4.1 Volatility estimation

4.1.1 HTSVM

Figure 1 is the direct estimation from MATLAB. Moreover, figure 2 has two panels that illustrates the model comparison between Gaussian distribution and Student's t distribution. The left panel is the estimate of marginal density $p(\psi|y)$ which assumes Gaussian distribution and the right panel is the estimate of $p(\nu|y)$ which reflects the pertinence to a Student's t distribution. Accordingly, the right panel is much more peaked and skewed than the left panel.

Figure 3 illustrates the comparison between TAIEX VIX vs TW50 SV (HTSVM). The TAIEX VIX (also can be called TAIWAN VIX). As the name suggest, it is the implied volatility index of the TAIEX. The calculation follows the same procedure of CBOE's VIX methodology. The VIX is an index for implied volatility which is the volatility traded by the market. The reason for showing the VIX is to check whether the stochastic volatility can track the market volatility closely. Some will say why not just use VIX instead of any realized volatility models like the HTSVM or a conventional stochastic volatility. The answer is that VIX can be highly overvalued during huge market shock since people will be hurried to cut the loss by buying back short positions in either call or puts. Therefore, this result in unreasonably high risk premium in option pricing temporarily.

Statistically, the correlation (Pearson method) between VIX and HTSVM is around 0.85 which approaches a perfectly positive relationship. To examine whether there exists a strong relation in the long run. I also run a Johansen cointegration test proposed by Johansen (1991); the result shows that we can reject the null hypothesis that there is no cointegration (see appendix 7.3).

Figure 1. HTSVM time series graph output of HTSVM from MATLAB

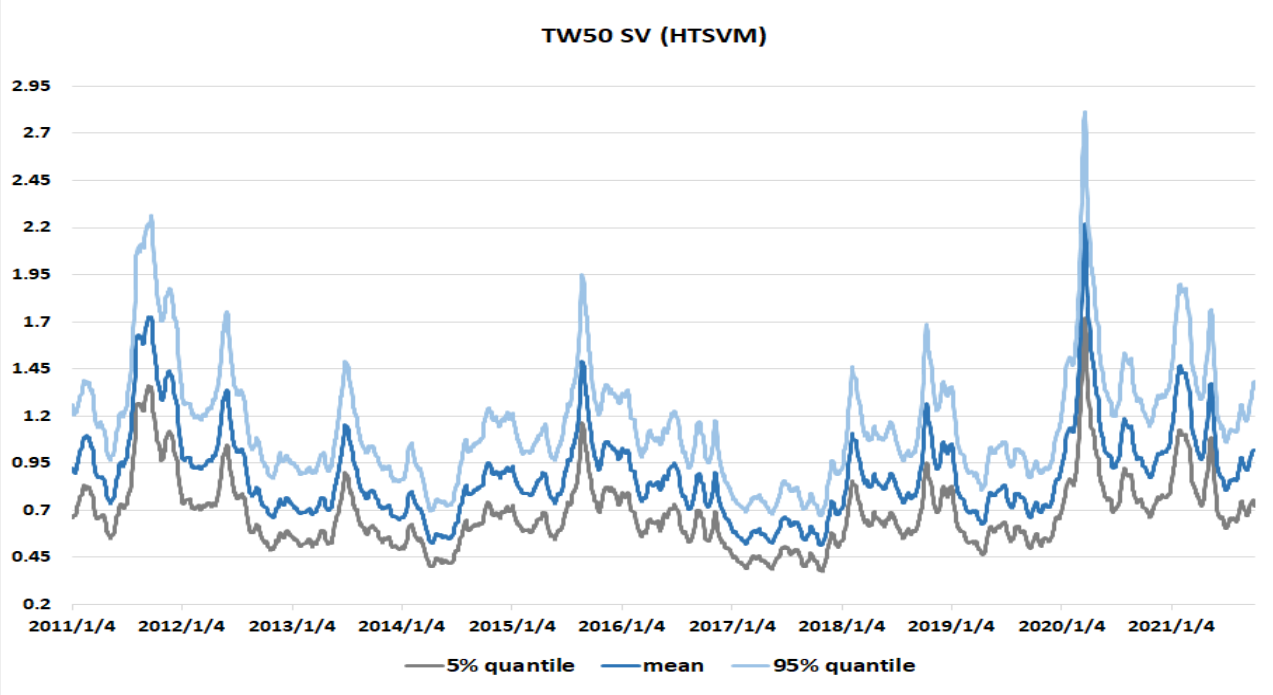


Figure 2. Left: marginal density of p(ψ|y) and Right: marginal density of p(v|y)

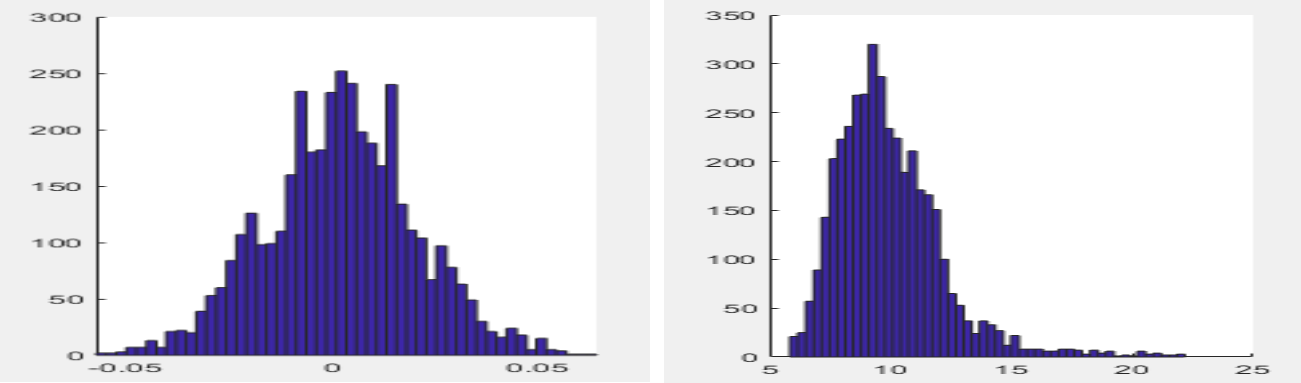
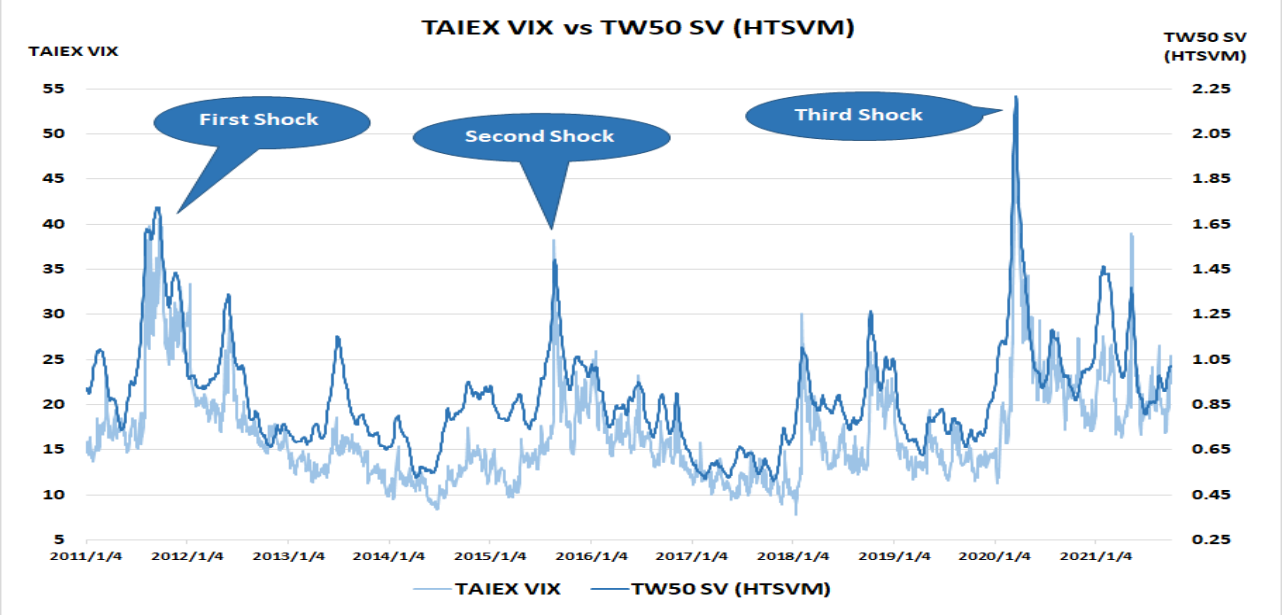


Figure 3. TAIEX VIX vs TW50 SV (HTSVM)



4.1.2 Volatility of HTSVM as signal for portfolio rebalancing

After the cointegration validation with VIX, I can then safely use HTSVM as signal. The long-run mean of my HTSVM from 2011 to 2021 is 0.891. The standard deviation (SD) is 0.245. If HTSVM is two SDs above the long-run mean, then I initiate the rebalance. The two SDs above the mean is around 1.38; hence, this is the signal for rebalancing. Consequently, there are three shocks (table 3) and three parts of sub-samples (table 4). The period R0 includes both quiet period and the first shock. The reason is that the optimal weight from this sample period can be more representative to reflect the true market which should go through a few stages in the business cycle and can then be used for the out-of-sample for R1.

Table 3. Three shocks that match the above criteria:

First shock	September, 2011
Second shock	August, 2015

Third shock	March, 2020

Table 4. Three parts of sub-sample analysis based on the volatility shock

R0	01.03.2011 ~ 08.25.2015
R1	08.26.2015 ~ 03.20.2020
R2	03.23.2020 ~ 10.04.2021

4.2 Index tracking

4.2.1 Full sample (Passive)

Figure 4 illustrates the cumulative return curve of both the tracking portfolio and the TW50. The base is set at 100 for both from day 1 of the time series. The tracking portfolio has beaten the TW50 most of the time. Figure 5 shows the time series of 5-day rolling tracking errors (TE), the TE oscillates around the long-run mean at 0.01 until January 2020. Since then, it is also the period where the tracking portfolio begins to outperform the TW50 by a substantial margin. My way of calculating the tracking error is simply the standard deviation of the returns between the tracking portfolio and the TW50. Another common measure is to look at the tracking portfolio's excess return (ER) against the index. A positive ER means that the tracking portfolio outperforms the index, and vice versa. Table 5 shows that the ER is 1.394 and the overall TE is 0.014. This means that the tracking portfolio can stick with the index while having the premium. However, this is just the in-sample performance which, in most cases, will perform worse during out-of-sample and live market.

Table 6 lists the comparison of average beta between the TW50 and the tracking portfolio. The beta represents the sensitivity to the systematic risk in a portfolio. It is a quantitative measure to see how our individual stock or the entire portfolio moves if the index changes by one percent. Usually, if a stock has beta greater than 1, it is considered an aggressive stock and if the stock has beta lower than 1, it is considered a defensive stock. The average beta of TW50 stocks is about 0.32 and the average beta of the tracking portfolio is about 0.38. Overall, this result is very reasonable as the denominator of the later one is much smaller. Both portfolios are quite neutral to the market risk since both of them are below 0.5.

Next, we should look at how the tracking portfolio is formed in terms of member stocks, figure 6 illustrates the weight by stock; as we can see, Taiwan Semiconductor Manufacturing Corporation is the largest component which occupies 14.14% of total weight. As a global chip giant, it can be a very moderately offensive stock.

As for the weight of tracking portfolio by industry sectors, figure 7 shows that technology is the largest followed by financial service, transportation, retails, telecom, and chemicals. If look at the weight by sectors, it is more evenly distributed across telecom, financial service, and technology since telecom, financial service, and technology each consists of at least 20 percent of total weight.

Figure 4. Full sample analysis

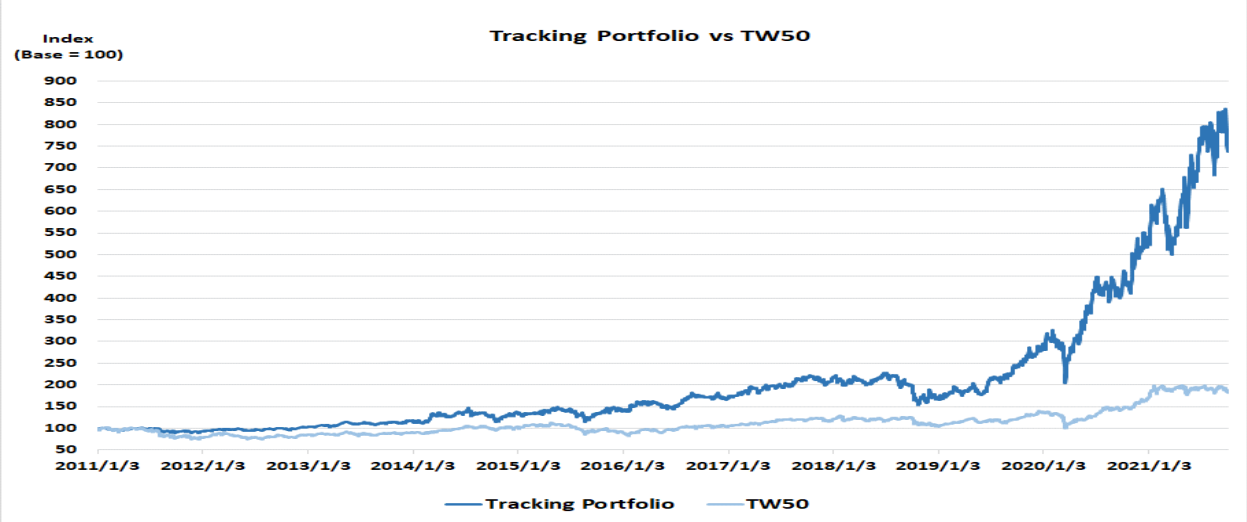


Figure 5. 5-day rolling tracking errors

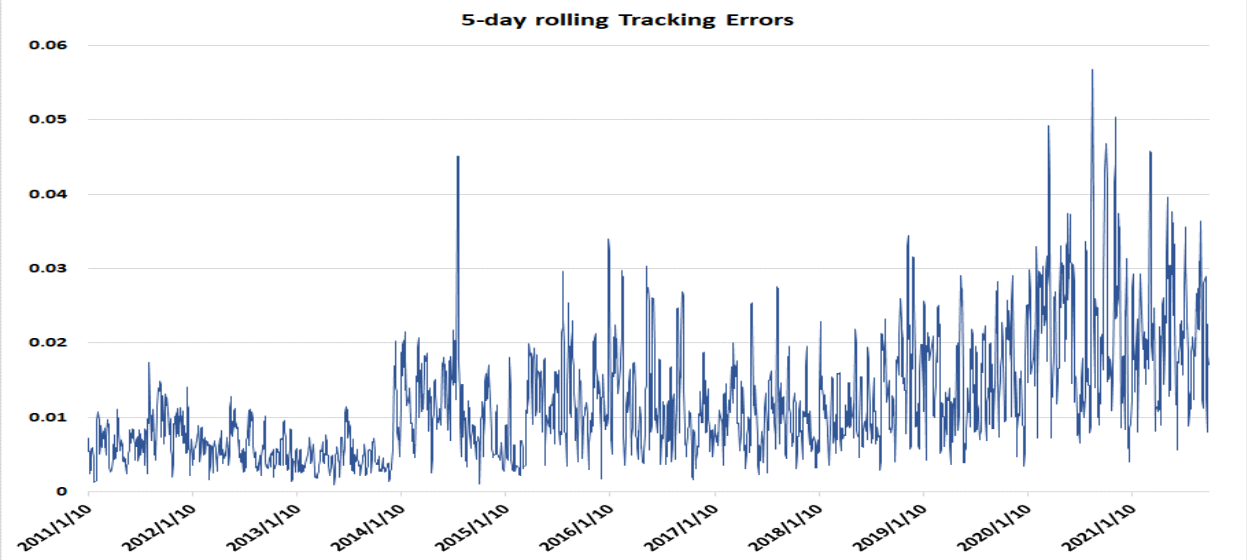


Table 5. TE and ER of full-sample analysis

TE	ER
0.014	1.394

TE: $\sqrt{Var\left(Return_{trackign\ portfolio} - Return_{TW50}\right)}$

ER: $\text{Log}\left(\frac{Price\ of\ trackign\ portfolio_{last\ day}}{Price\ of\ tracking\ portfolio_{first\ day}}\right) - \text{Log}\left(\frac{Price\ of\ TW50_{last\ day}}{Price\ of\ TW50_{first\ day}}\right)$

Table 6. Beta comparison between the tracking portfolio and TW50

	Average Beta
TW50 stocks	0.320
Tracking Portfolio stocks	0.326

Figure 6. Weight by stock

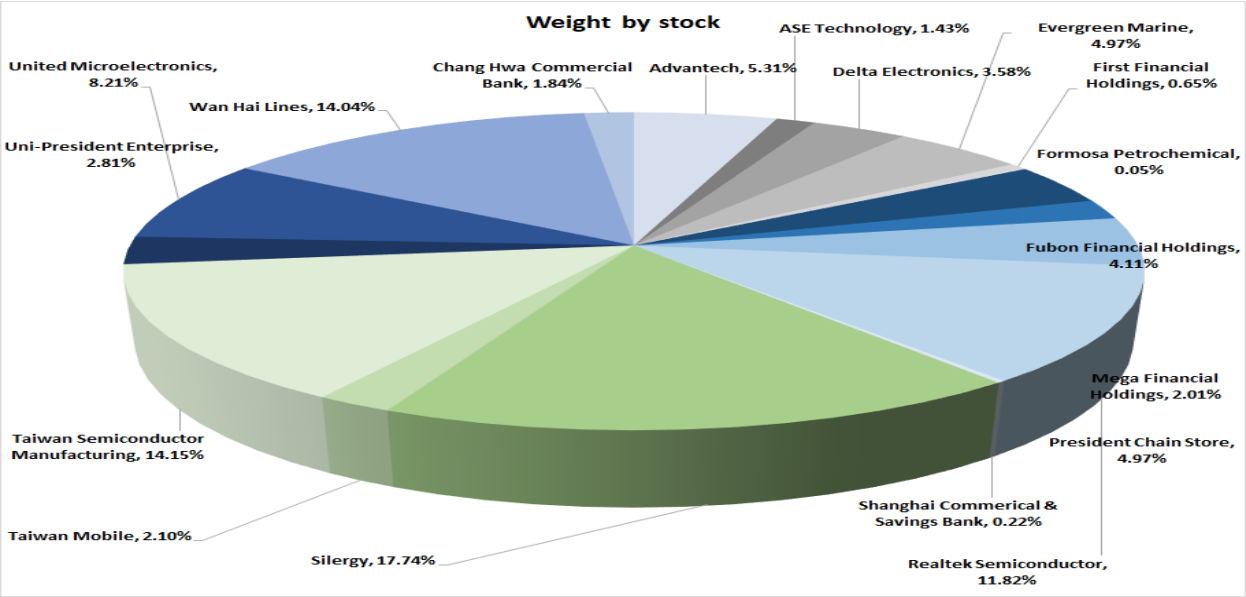
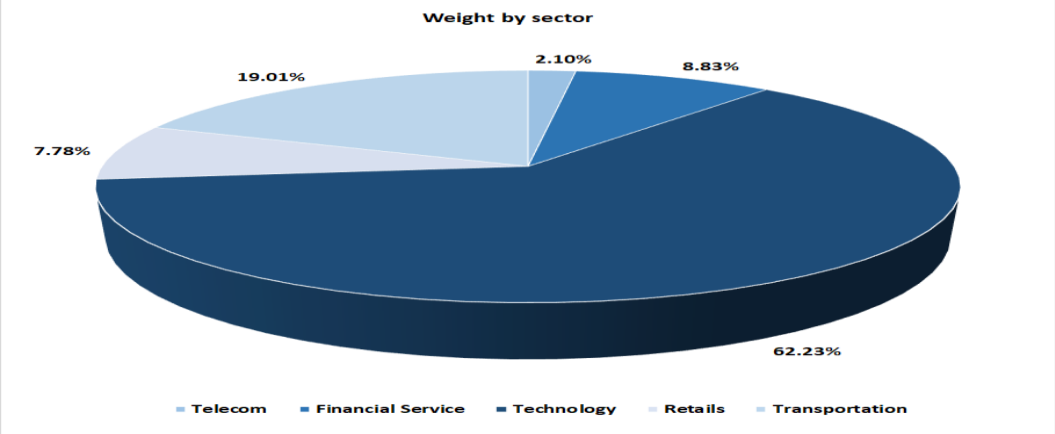


Figure 7. Weight by sector



4.2.2 Splits the full-sample (FS) into in-sample (IS) and out of-sample (OS)

Nonetheless, the result from section 4.2.1 could be due to over-fitting which is very likely to fail in reflecting the actual performance. In this section, I cut the full-sample into IS and OS. The IS contains the first eighty percent of data and the OS contains the last twenty percent of data from the FS.

Therefore, the IS starts from Jan 3rd, 2011 to August 2nd, 2019 and the OS starts from August 5th, 2019 to October 4th, 2021. In addition, the weight of the OS comes from the weight of the IS. Figure 8 shows that in the IS, its tracking portfolio outperforms the index while in the OS, the overall result is reversed. From table 7, we can see that ER of IS is 0.857 and the ER of OS is a negative 0.881, even though both have very close TE.

Figure 8. In-sample (LHS) vs Out-of-Sample (RHS)

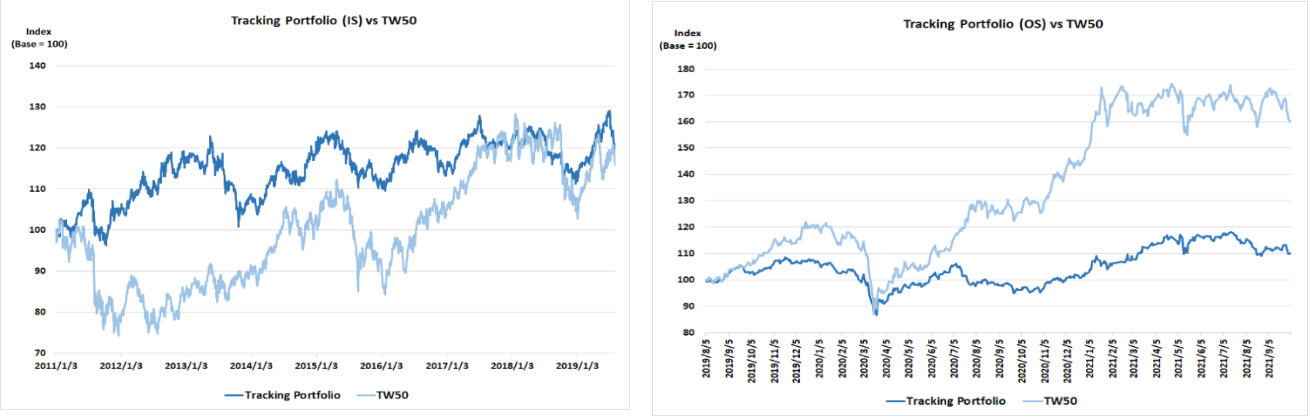


Table 7. Performance comparison of IS and OS of FS

	TE	ER
In-Sample	0.020	0.857
Out-Of-Sample	0.015	−0.881

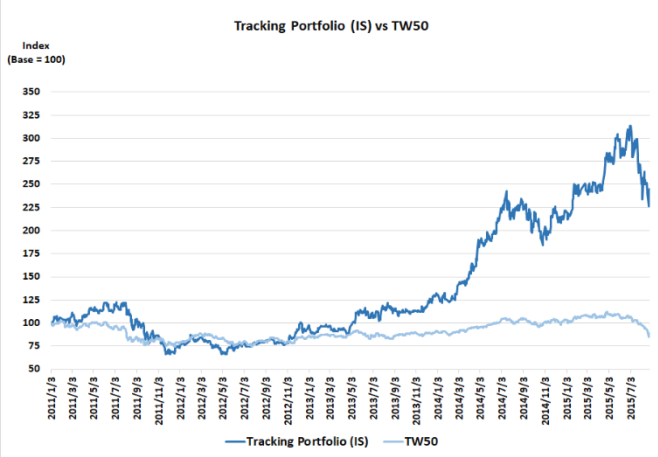
4.2.3 Sub-sample analysis: three periods separated by the volatility shocks with the in-sample (IS) and the out-of-sample (OS) analysis

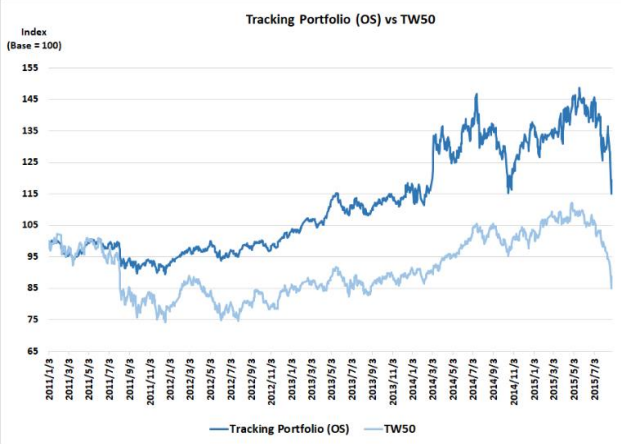
In this section, I will discuss the outcomes about the TE and ER from the IS and OS in three separate periods: R0, R1, and R2. The purpose of doing so is to reexamine the index tracking formulation from section 2.2. Before we go on to live trading, it is necessary to do several kinds of backtesting such as plain vanilla out-of-sample, random data simulation, and rolling-window. In this section, my method is a mixture of rolling window and time weighted average. The rebalancing decision is based on the volatility shock identified and summarized in table 3 and the volatility shock is based on the volatility estimation of HTSVM introduced in section 2.1.3. Also, as mentioned in section 4.1.2, there is one exception which is the period R0. It contains both tranquil (low volatility) and fear (high volatility) periods. Since it experiences a more complete economic (business) cycle, the learning process from the model can be more representative to all kinds of environment and we can compare it with R1 and R2 which either have short period of time or unlike to experience a complete economic cycle. Also, we can examine the comparison between large and small sample.

Rebalance Zero (R0)

R0 has 1145 days of return data. The OS of RO uses the weight from the full-sample analysis. Consequently, the ER of IS is about 3.4 times that of OS. The TE of IS is also twice as large as that of OS.

Figure 9. In-sample (LHS) vs Out-of-Sample (RHS) for full-sample during R0

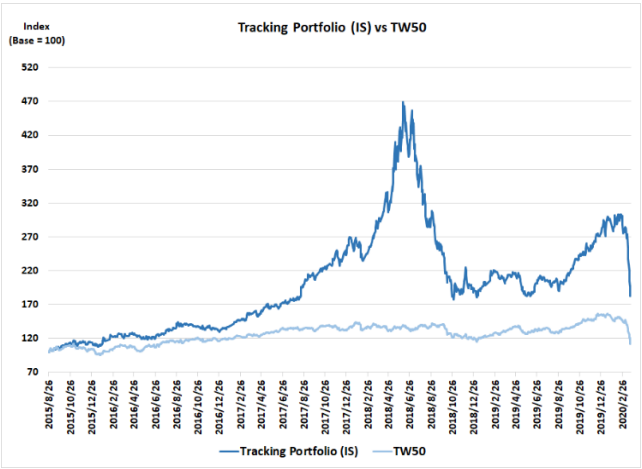


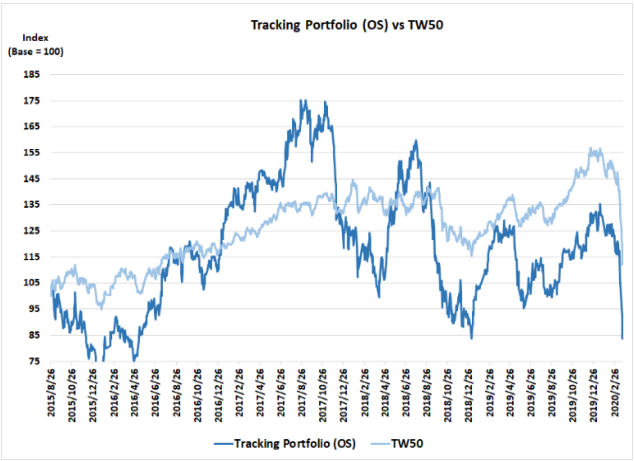


Rebalance One (R1)

R1 has 1117 days of return data. The OS of R1 adopts the weight from the IS of R0. The ER of IS continues to outperform that of OS during R1; moreover, the TE of IS is marginally smaller than that of OS.

Figure 10. In-sample (LHS) vs Out-of-sample (RHS) for R0 during R1

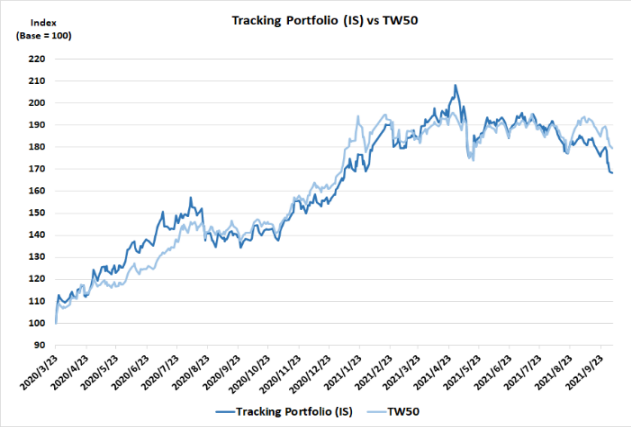


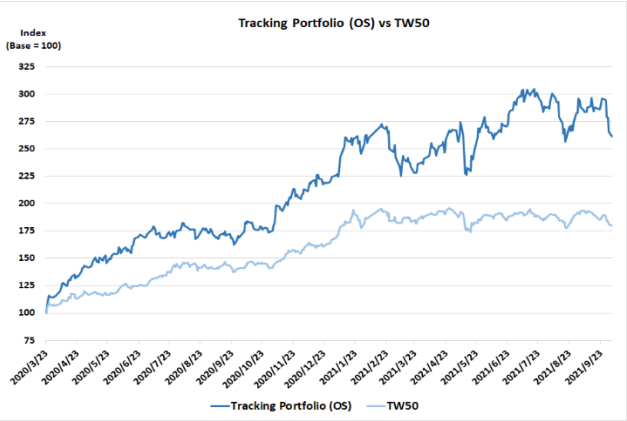


Rebalance Two (R2)

R2 has 377 days of return data. The OS of R2 adopts the weight from the IS of R1. The negative ER of IS indicates that the tracking portfolio underperforms the TW50 during R2. The ER of IS turns out to be negative while the ER of OS has strong positive result in R2. Also, the TE of IS is about half of that of OS.

Figure 11. In-sample (LHS) vs Out-of-sample (RHS) for R1 during R2





Weighted Average

Table 8 shows that by looking at the weighted average, both TE and ER of in-sample (IS) and out-of-sample (OS) are very close to each other. The ER of in-sample is much higher than that of out-of-sample. Since the in-sample models may have issue in overfitting, this result is expected. However, surprisingly, the TE of out-of-sample is slightly lower than that of in-sample. Additionally, table 9 shows that for ER, both IS and OS are moderately correlated and for TE, both IS and OS are highly correlated. This suggests that relationship between IS and OS tracking performances is positive and quite significant.

Table 8. Weighted average of TE and ER from both in-sample and out-of-sample

	In-Sample		Out-Of-Sample	
	TE	**ER**	**TE**	**ER**
R0	0.018	1.014	0.009	0.296
R1	0.016	0.498	0.020	−0.274
R2	0.009	-0.065	0.016	0.375
Weighted Average	**0.016**	**0.642**	**0.015**	**0.066**

Where the weight is calculated as $\frac{\text{Ri}}{\text{R0+R1+R2}}$. Therefore, the weights for R0, R1, R2 are 0.434, 0.423, 0.143, respectively.

Table 9. Pearson correlation

	ER (In-Sample)	TE (In-Sample)
ER (Out-Of-Sample)	0.310	
TE (Out-Of-Sample)		0.612

5. Conclusion

5.1 Summary

This project conducts research on the index tracking of TW50 index listed in Taiwan. The model formulation is built with MILP in which the objective is to minimize the tracking error between tracking portfolio and the TW50 subject to constraints of cardinality, budget, and real-world transaction costs. As expected, the in-sample has much stronger ER than out-of-sample. The tracking errors from in-sample have proven to be tracked closely with the ones from out-of-sample and consistent throughout both full-sample and sub-sample analysis.

Another primary focus of this project is to investigate whether volatility jump (2SDs > mean) suggests a good timing for portfolio rebalancing on the TW50 index. The volatility estimation from HTSVM is invented to explain the financial market behaviors such as non-constant volatility, leptokurtic distribution, and serial dependence. As a consequence, the head-to-head comparison in table 10 shows that the rebalancing via volatility shock generates a relatively superior performance.

Table 10. Comparison between full-sample (passive) vs rebalancing after volatility shock

	In-Sample		Out-Of-Sample	
	TE	**ER**	**TE**	**ER**
Full-Sample (passive)	0.020	0.857	0.015	−0.881
Rebalancing by volatility shock	0.016	0.642	0.015	0.066

5.2 Potential extension

As for future extension, the linear programming problem for index tracking can also include the objective of maximization of excess return. For instance, Li *et al.* (2011) had done so by including both minimization of tracking error and maximization of excess return. A number of researches, such as Beasley *et al.* (2003) as well as Dose and Cincotti (2005), improvised a single objective that is a weighted average of several objectives.

6. References

Beasley, JE, Meade, N & Chung TJ 2003, 'An evolutionary heuristic for the index tracking problem', *European Journal of Operational Research*. vol. 148: 3. pp. 621-643.

Bloom, N 2009, 'The impact of uncertainty shocks', *Econometrica*, vol. 77: 3, pp. 623-685.

Chan, JCC & Hsiao, CYL 2013, 'Estimation on stochastic volatility models with heavy tails and serial dependence', *Bayesian Inferences in the Social Sciences*, Chichester, UK: Wiley.

Dose, C & Cincotti, S 2005, 'Clustering of financial time series with application to index and enhanced index tracking portfolio', *Physica A: Statistical Mechanics and its Applications*, vol. 18: 4, pp. 145-151.

Johansen, S 1991, 'Estimation and hypothesis testing of cointegration vectors in Gaussian vector autoregressive models', *Econometrica*, vol. 59: 6, pp. 1551-1580.

Kalman, R 1960, 'A new approach to linear filtering and prediction problems', *ASME Journal of Basic Engineering*, vol. 92, pp. 35-45.

Li, Q, Sun, L & Bao, L 2011, 'Enhanced index tracking based on multi-objective immune algorithm', *Expert Systems with Applications*, vol. 38, pp. 6101-6106.

Mezali, H & Beasley, JE 2014, 'Index tracking with fixed and variable transaction cost', *Optimization Letters*, vol. 8, pp. 61-80.

Taylor, SJ 1986, *Modelling financial time series*, Chichester, UK: Wiley.

Watanabe, T & Asai, M 2001, 'Stochastic volatility models with heavy-tailed distributions: A Bayesian analysis', Available from: https://www.math.chuo-u.ac.jp/~sugiyama/15/15-02.pdf/ [Accessed 28th September, 2021].

7. Appendix

7.1 List of TW50 index members and tickers

Company	Ticker
Advantech	2395
Airtac International Group	1590
ASE Technology	3711
Asia Cement	1102
Asus	2357
AU Optronics	2409
Cathay Financial Holdings	2882
Chailease Holding	5871
Chang Hwa Commercial Bank	2801
China Steel	2002
Chunghwa Telecom	2412
CTBC Financial Holdings	2891
Delta Electronics	2308

E. Sun Financial Holdings	2884
Evergreen Marine	2603
Far Eastern New Century Corp.	1402
Far EasTone Telecommnication	4904
Feng Tay Enterprise	9910
First Financial Holdings	2892
Formosa Chemicals & Fibre	1326
Formosa Petrochemical	6505
Formosa Plastic Corp	1301
Fubon Financial Holdings	2881
Hon Hai Precision Industry	2317
Hotai Motor	2207
Hua Nan Financial Holdings	2880

Largan Precision	3008
MediaTek	2454
Mega Financial Holdings	2886
Nan Ya Plastics	1303
Nan Ya Printed Circuit Board	8046
Nanya Technology	2408
Novatek Microelectronics	3034
Pegatron	4938
President Chain Store	2912
Quanta Computer	2382
Realtek Semiconductor	2379

Shanghai Commerical & Savings Bank	5876
Silergy	6415
Taishin Financial Holdings	2887
Taiwan Cement	1101
Taiwan Coooperative Financial Holdings	5880
Taiwan Mobile	3045
Taiwan Semiconductor Manufacturing	2330
Uni-President Enterprise	1216
United Microelectronics	2303
Wan Hai Lines	2615
Yageo	2327
Yang Ming Marine Transport	2609

Yuanta Financial Holdings	2885

7.2 MATLAB code for HTVSM developed by Chan and Hsiao (2013)

```
clear; clc;
nloop = 5000;
burnin = 1000;
load 'TW50.csv';
y = TW50;
T = length(y);

%% prior
invVmu = 1/5;
phih0 = .95; invVphih = 1;
muh0 = 0; invVmuh = 1/5;
invVpsi = 1;
nuh = 10; Sh = .02*(nuh-1);
nuub = 50;  % upper bound for nu

disp('Starting MCMC.... ');
disp(' ' );
start_time = clock;

% initialize the Markov chain
sigh2 = .05;
phih = .95;
muh = 1;
psi_p = 0;
nu = 5;
lam = 1./gamrnd(nu/2,2/nu,T,1);
mu = mean(y);
h = log(var(y)*.8)*ones(T,1);
Hpsi = speye(T) + sparse(2:T,1:(T-1),psi_p*ones(1,T-1),T,T);
f = @(x) fMA1(x,y-mu,h);
psihat = fminbnd(@(x) -f(x),-.99,.99);
psi_p = psihat;

% initialize for storage
store_theta = zeros(nloop - burnin,6); % store [psi_p nu mu muh phih sigh2]
store_exph = zeros(nloop - burnin,T);   % store exp(h_t/2)

%% compute a few things outside the loop
psipri = @(x) log(normpdf(x,0,sqrt(1/invVpsi)) ...
   /(normcdf(sqrt(invVpsi))-normcdf(-sqrt(invVpsi))));
nugrid = linspace(.1,nuub,201)';
countnu = 0;
countpsi = 0;

rand('state', sum(100*clock) ); randn('state', sum(200*clock) );

for loop = 1:nloop
      %% sample mu
   Sigy = Hpsi*sparse(1:T,1:T,exp(h).*lam)*Hpsi';
   Dmu = 1/(invVmu + ones(1,T)*(Sigy\ones(T,1)));
   muhat = Dmu*(ones(1,T)*(Sigy\y));
```

```
mu = muhat + sqrt(Dmu)*randn;
   %% sample h
Ystar = log(((Hpsi\(y-mu)).^2 + .0001);
[h muh phih sigh2] = SV(Ystar,h,muh,phih,sigh2,[muh0 invVmuh ...
   phih0 invVphih nuh Sh]);
   %% sample psi_p
f = @(x) fMA1(x,y-mu,h+log(lam)) + psipri(x);
psihat = fminsearch(@(x) -f(x),psihat); % find the mode
sqVpsic = .05; Vpsic = sqVpsic^2;
psic = psihat + sqVpsic*randn;
if abs(psic)<.9999
   alpMH = f(psic) - f(psi_p) + ...
      -.5*(psi_p-psihat)^2/Vpsic + .5*(psic-psihat)^2/Vpsic;
else
   alpMH = -inf;
end
if alpMH>log(rand)
   psi_p = psic;
   Hpsi = speye(T) + sparse(2:T,1:(T-1),psi_p*ones(1,T-1),T,T);
   countpsi = countpsi + 1;
end
   %% sample lam
temp1 = (Hpsi\(y-mu)).^2./exp(h)/2;
lam = 1./gamrnd((nu+1)/2,1./(nu/2+temp1));
   %% sample nu
sum1 = sum(log(lam));
sum2 = sum(1./lam);
fnu = @(x) T*(x/2.*log(x/2)-gammaln(x/2)) - (x/2+1)*sum1 - x/2*sum2;
f1 = @(x) T/2*(log(x/2)+1-psi(x/2)) - .5*(sum1+sum2);
f2 = @(x) T/(2*x) - T/4*psi(1,x/2);
S = 1;
nut = nu;
while abs(S) > 10^(-5)     % stopping criteria
   S = f1(nut);
   Knu = -f2(nut);        % infomation matrix
   nut = nut + Knu\S;
end
sqrtDnu = sqrt(1/Knu);
nuc = nut + sqrtDnu*randn;
if nuc < nuub
   alp = exp(fnu(nuc)-fnu(nu)) ...
      * normpdf(nu,nut,sqrtDnu)/normpdf(nuc,nut,sqrtDnu);
   if alp > rand
      nu = nuc;
      countnu = countnu+1;
   end
end
if ( mod( loop, 2000 ) ==0 )
   disp(  [ num2str( loop ) ' loops... ' ] )
end

if loop>burnin
   i = loop-burnin;
   store_exph(i,:) = exp(h/2)';
   store_theta(i,:) = [psi_p nu mu' muh phih sigh2];
```

```
   end
end

disp( ['MCMC takes '  num2str( etime( clock, start_time) ) ' seconds' ] );
disp(' ' );

thetahat = mean(store_theta)';
exphhat = mean(store_exph)'; %this the volatility I want
exphlb = quantile(store_exph,.05)';
exphub = quantile(store_exph,.95)';
tid = linspace(2011,2021,T)';
figure; plot(tid, [exphhat exphlb exphub]);
box off; xlim([2011 2021]);

figure;
subplot(1,2,1); hist(store_theta(:,1),50); box off;
subplot(1,2,2); hist(store_theta(:,2),50); box off;
```

7.3 Johansen cointegration test

```
library(urca)
library(readxl)
volcompare <- read_excel("volcompare.xlsx")
View(volcompare)
vix=volcompare$VIX
sv=volcompare$HTVSM
Time=volcompare$Date
corr=cor(vix, sv, method= c("pearson"))
corr
plot(vix,sv, main="Scatterplot", xlab="VIX", ylab="HTVSM ", pch=19)
abline(lm(vix~sv), col="red") # regression line (y~x)
lines(lowess(vix,sv), col="blue") # lowess line (x,y)
#Johansen cointegration
jotest=ca.jo(data.frame(vix, sv), type="trace", K=2, ecdet="none", spec="longrun")
summary(jotest)
#We can reject the null hypothesis and conclude that there is no cointegration.
```

7.4 Instruction on where to locate main corresponding computational files

Main database

TW50 databank: Taiwan 50 members.xlsx

Main models

HTSVM: MATLAB Stochastic Volatility file → HTSVM.m
Full-sample: Full-sample TW50.R
Full-sample IS: FSIS.R
Full-sample OS: FSOS.R
R0: R0.R, R1: R1.R, and R2: R2.R

Portfolio construction

Full-sample: Full sample.xlsx, Full-sample IS: Full sample_IS.xlsx, and Full-sample OS: Full sample_OS.xlsx
R0: Rebalance Zero.xlsx
R1: Rebalance One.xlsx
R2: Rebalance Two.xlsx