Portfolio Optimization in Emerging Markets Using ARMA-EGARCH Models

LI Chung Yat, 20422979
TSE Cheuk Lun, 20745333
WONG Tsz Wing, 20189826

1 Introduction

Portfolio management is widely used by investors in order to minimize portfolio risk, increase risk-adjusted return and achieve long-term investment plans. To construct or build an investment portfolio, three main processes are required to follow. First, investors need to understand their risk appetite, investment objectives, and investment horizon before constructing their own investment portfolio, which is called the planning step. The next step will be the execution. After considering the investment constraints and objectives, investors can determine the asset allocation strategy for their investment portfolio, such as asset classes and the allocation portion of each asset class. Asset classes are ranging from cash, fixed-income and publicly traded equities to gold, commodities and alternatives. The feedback step will be the last step of the portfolio management process. Investors monitor the movement of the asset prices, rebalance their portfolio periodically based on the returns of each asset class and the change of the investment preferences to reallocate the capital to different asset classes and, most importantly, evaluate the performance of the portfolio against the performance of the benchmark.

The objective of this report was to optimize a portfolio that consists of multiple emerging market indices. MSCI Brazil, MSCI Taiwan, MSCI Qatar, MSCI Korea, and MSCI China A are the components of the portfolio.

Specifically, this report attempt to:

i. observe the characteristics and the performance of multiple emerging markets indices;

ii. apply a time series model (i.e. ARMA-eGARCH model) in the training data set and forecast the returns comparing to the testing data;

iii. establish the portfolio mixed which the portfolio return is optimized.

2 Theory

From the definition of MSCI (2021), markets can be classified into three groups - developed markets, emerging markets and frontier markets - based on economic development, requirements of size and liquidity, and market accessibility. And, emerging markets are the focus in this paper.

2.1 MSCI Index Calculation Methodology

In general, there are two approaches to compute MSCI Indices, which are the Price Adjustment Factor (abbreviated PAF) and the Index divisors. PAF is the most common method for index calculation. Indices are calculated on a daily basis and some of them are calculated on a real-time basis. The price of the indices is capturing the market performance of the components weighted in terms of the market capitalization size. The index level is calculated by using the movement of the market performance comparing to the prior index level, which the equation is showing as the following:

$$PriceIndexLevelUSD_t = PriceIndexLevelUSD_{t-1} * \frac{IndexAdjustedMarketCapUSD_t}{IndexInitialMarketCapUSD_t} \quad (1)$$

where:

$$IndexAdjustedMarketCapUSD_t =$$

$$\sum_{s \in I,t} \frac{EndOfDayNumberOfShares_{t-1} * PricePerShare_t * InclusionFactor_t * PAF_t}{FXRate_t} \quad (2)$$

$$IndexInitialMarketCapUSD_t =$$

$$\sum_{s \in I,t} \frac{EndOfDayNumberOfShares_{t-1} * PricePerShare_{t-1} * InclusionFactor_t}{FXRate_{t-1}} \quad (3)$$

$InclusionFactor_t$ is one or more than one of the factors of domestic inclusion, foreign inclusion, value inclusion, growth inclusion and index inclusion;

PAF_t is the price adjustment factor of selected security at time t;

$FXRate_t$ is the foreign exchange rate of the market price currency of the selected security against USD at time t.

2.2 ARCH Model

ARCH model is suggested by Engle (1982), which is the first model for volatility modeling. ARCH(m) model:

$$a_t = \sigma_t \epsilon_t, \qquad \sigma_t^2 = \alpha_0 + \sum_{i=1}^{m} \alpha_i a_{t-i}^2 \quad (4)$$

where $\{\epsilon_t\}$ is a series of independent and identically distributed random variables, which the mean is 0 and the variance is 1. To modify the equation (4), the ARCH model can be written as:

$$a_{i,t} = \sqrt{h_{i,t}} \epsilon_t \quad (5)$$

where $h_{i,t} = V[a_{i,t} | a_{i,t-1}, ..., a_{i,t-x}] = \alpha_0 + \sum_{l=1}^{x} \alpha_{i,l} a_{i,t-l}^2$, $\alpha_{i,l}$ is the coefficients of ARCH and $\epsilon_t \sim$ N(0,1) for index i.

2.3 GARCH Model

GARCH model is a volatility model built by Bollersev (1986), which is the extension of the ARCH model. Assume at time t the innovation $a_t = r_t - \mu_t$ for a log return series r_t, a_t follows a GARCH(m,s) model if

$$a_t = \sigma_t \epsilon_t, \qquad \sigma_t^2 = \alpha_0 + \sum_{i=1}^{m} \alpha_i a_{t-i}^2 + \sum_{j=1}^{s} \beta_j \sigma_{t-j}^2 \tag{6}$$

where $\{\epsilon_t\}$is a series of independent and identically distributed random variables, which the mean is 0 and the variance is 1, $\alpha_0 > 0$, $\alpha_i \geq 0$, $\beta_j \geq 0$, and $\sum_{i=1}^{max(m,s)} (\alpha_i + \beta_i) < 1$. $\alpha_i = 0$and $\beta_j = 0$ if and only if $i > m$ and $j > s$ respectively. The constraints ensure the conditional variance is positive and stationary. After modifying the equation (6), the GARCH model can be described as:

$$a_{i,t} = \sqrt{h_{i,t}}\epsilon_t \tag{7}$$

where $h_{i,t} = V[a_{i,t}|a_{i,t-1},..., a_{i,t-x}] = \alpha_0 + \sum_{l=1}^{x} \alpha_{i,l} a_{i,t-l}^2 + \sum_{m=1}^{y} \beta_{i,m} h_{i,t-m}$, $\beta_{i,m}$ is the coefficients of GARCH and $\epsilon_t \sim$ N(0,1) for index i.

2.4 EGARCH Model

In general, ARCH models usually neglect the serial correlation which an ARMA model can be useful to handle this weakness. And, for ARCH models or GARCH models, one of the limitations is that the shocks have the same effect on the volatility regardless of the direction of the shocks. To eliminate this restriction, the eGARCH model is proposed by Nelson (1991). The eGARCH(m,s) model:

$$ln(\sigma_t^2) = \alpha_0 + \frac{1+\beta_1 B+\cdots+\beta_{s-1} B^{s-1}}{1-\alpha_1 B-\cdots-\alpha_m B^m} g(\epsilon_{t-1}) \tag{8}$$

where α_0 is a constant, B is the lag operator. The equation (8) can be re-written as:

$$a_{i,t} = \sqrt{h_{i,t}}\epsilon_t \tag{9}$$

where $ln(h_{i,t}) = \alpha_0 + \sum_{l=1}^{x} \alpha_{i,l} g(\epsilon_{t-l}) + \sum_{m=1}^{y} \beta_{i,m} ln(h_{i,t-m})$, $g(\epsilon_t) = \theta\epsilon_t + \gamma[|\epsilon_t| - E(|\epsilon_t|)]$and $\epsilon_t \sim$ N(0,1) for index i.

2.5 ARMA-GARCH Model

ARMA-GARCH model is an ARMA model with $a_{i,t}$ to be the white noise with variance represented by the eGARCH model. The ARMA-eGARCH model is written as:

$$r_{i,t} = \mu + \sum_{j=1}^{p} \varphi_{i,j} r_{i,t-j} - \sum_{k=0}^{q} \theta_{i,k} a_{i,t-k} \quad (10)$$

where $a_{i,t} \sim N(0, h_{i,t})$ for index i.

3 Methodology

3.1 Data Source

Emerging market indices are chosen for their potential growth. However, there are also risks associated with the emerging market, to name a few, political, legal, economic, and currency exchange risks. Hence, we chose multiple emerging market indices with stable environments and copious growth potential to diversify the risks as mentioned.

The MSCI Emerging Markets Index measures the performance of large (market capitalization value of over $10 billion) and mid-cap (market capitalization value of between $2 and $10 billion) companies in emerging markets. Index returns are based on the assumption of reinvestment, do not include fees, expenses or sales charges and cannot be invested directly.

The data of emerging markets indices were retrieved from MSCI Inc. Data is collected with the period from 2017-01-01 to 2021-04-30.

Below is a brief overview of the selected indices from the MSCI EM INDEX

3.1.1 MSCI China.A

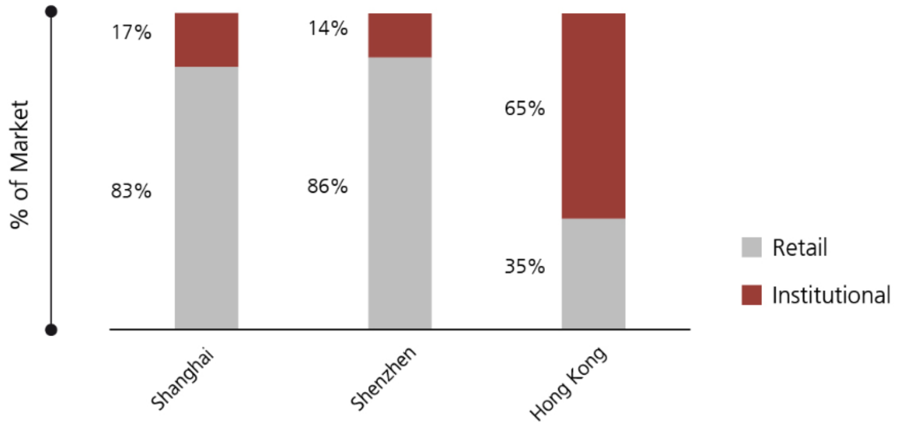

Figure 1: Shanghai, Shenzhen & Hong Kong Stock Exchanges (% of Market), 2018

The chart above (UBS, 2020) shows that retail investors account for over 80% of China's A-share market turnover. Retail investors of this market have shorter holding periods as compared to institutional investors, which implies a higher volatility level, creates opportunities for active strategies. The MSCI China A index underweights the low size companies. It spiked between Jan 15 and Feb 16, with promising growth, as shown in the chart below.

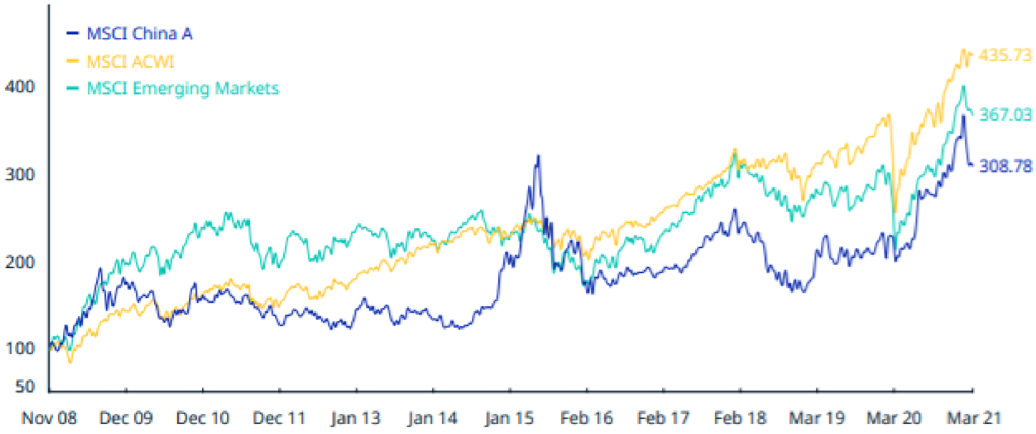

Figure 2: Cumulative Index Performance - Gross Returns (USD) chart (MSCI, 2021)

The MSCI China A Index consists of 470 constituents that are listed on the Shanghai and Shenzhen exchanges and accessible through "Stock Connect". There is no one dominant sector within the index. The top three weighted sectors are Financials, Consumer Staples and Industrials.

3.1.2 MSCI Taiwan

Information Technology is the leading force of Taiwan's market, occupying more than half of the weights of the MSCI Taiwan Index. Moreover, although there is only one Taiwanese company on the top 10 constituents list, TAIWAN SEMICONDUCTOR MFG boasts the highest free float-adjusted market capitalization of all, thanks to the increasing demand for semiconductors. The MSCI Taiwan Index underweights low size companies and overweights rising stocks, follows the MSCI ACWI IMI and MSCI closely but drastically outperformed recently since Dec 19 as shown in the chart below.

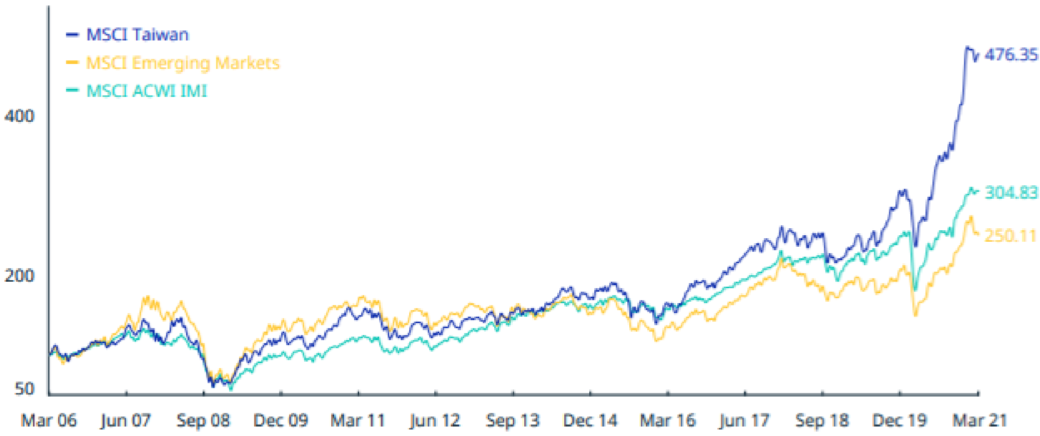

Figure 3: Cumulative Index Performance - Gross Returns (USD) chart (MSCI, 2021)

The MSCI Taiwan Index covers around 85% of the free float-adjusted market capitalization in Taiwan, consisting of 87 constituents. The index weighs heavily on the Information Technology sector, which takes up a whopping 74.83%.

3.1.3 MSCI Korea

Similar to Taiwan, the Information Technology sector also plays a major role. Noticeably, SAMSUNG ELECTRONICS CO alone weighted 31.99% in the MSCI Korea index, ranked 4th among the top 10 constituents of the MSCI EM Index. Unlike Taiwan, the Information Technology sector takes up only 48.26%, with the rest evenly distributed among other sectors. The MSCI Korea Index also overweights high momentum stocks and underweights low size companies like the MSCI Taiwan Index. It does not deviate from the MSCI EM Index and the MSCI ACWI IMI Index very much as shown in the chart below.

Figure 4: Cumulative Index Performance - Gross Returns (USD) chart (MSCI, 2021)

The MSCI Korea Index covers about 85% of the Korean equity universe with 106 constituents.

3.1.4 MSCI Brazil

Brazil has a strong Materials Sector, weights closely behind the Financials sector in the index. The MSCI Brazil Index underweights the lower-risk stocks but also the rising stocks more than the MSCI EM Index. It has high volatility, overperformed from Mar 06 to Dec 14, and then surpassed by the MSCI EM Index and the MSCI ACWI Index as shown in the chart below.

Figure 5: Cumulative Index Performance - Gross Returns (USD) chart (MSCI, 2021)

The MSCI Brazil Index covers about 85% of the Brazilian equity universe with 53 constituents.

3.1.5 MSCI Qatar

Although the GDP of Qatar is the least among the selected indices, its weight in the MSCI EM index has been steadily increasing over the years (Xu, W. & MSCI.). The index is dominated by the Financials sector, occupying two-third of the total sector weights. Compared to the MSCI EM Index, it overweights lower-risk substantially and underweights rising stocks. It outperformed the MSCI EM Index and the MSCI ACWI Index between Sep 13 and Mar 16 but generally performed similarly or worse during some periods as shown in the chart below.

Figure 6: Cumulative Index Performance - Gross Returns (USD) chart (MSCI, 2021)

The MSCI Qatar Index covers approximately 85% of the free float-adjusted market capitalization in Qatar with 12 constituents.

3.2 Model Selection

In the model selection process, we tried to find a model which can fit the model on most of the indices. We use the below tests and criteria to select the model which can apply to all the indices, with the training data from 2017-01-01 to 2020-12-31.

3.2.1. Test for the mean

From the one-sample test, t-ratio for all indices testing $H_0: \mu = 0$versus $H_a: \mu \neq 0$ with significant p-value. Thus, the zero mean model is chosen for our model to build the econometric model. Even if the null hypothesis test is rejected, the$\mu$is so minimal in which it is neglectable.

3.2.2. Test for ARCH effect

Since the p-value of all indices are significant, all ARCH effects are properly captured by the ARMA(2,2)-eGARCH(1,1) model.

3.2.3. Test for the distribution

The log-likelihood should be maximized when we consider the model. For the results, student t distribution has a larger log-likelihood than the normal distribution for all indices. Therefore, the student t distribution is considered in our model.

3.2.4. BIC

Bayesian information criterion (BIC) is used to select our model. Although not all the indices have the minimum BIC for the ARMA(2,2)-eGARCH(1,1) model, the performance is decent.

Indices	p-value for t-test on mu	p-value for ARCH LM test	BIC
MSCI Brazil	0.106684	0.7912	-5.1688
MSCI Taiwan	0.000907	0.6371	-6.3519
MSCI Qatar	0.868678	0.729	-6.5126
MSCI Korea	0.442543	0.6152	-5.8649
MSCI China.A	0.009536	0.4319	-6.1143

Table 1: Test statistics in Model Selection

3.3 Rolling window

Data fed into the ARMA-GARCH model is on a rolling window basis. This implies that we refit the model whenever 1 day has passed. Mathematically speaking, we add the most up-to-date information into the model as of Filtration t to do prediction on Filtration (t + 1).

3.4 Forecasting

Because of the rolling window feature, we only need to focus on making a precise 1-day forward prediction. After refitting the model with the updated data at Filtration t, we will perform 100 simulations on the indexes price for time (t + 1). By taking the arithmetic average on these simulations, this means the value will be deemed as the predicted price at the time (t + 1).

3.5 Trading Strategy

To keep the strategy as simple as possible, we will be using a long-only portfolio with daily rebalancing. Assuming the indexe's price is only continuous over the actual trading hours of exchanges, the close price at time t is the same as the open price at the time (t + 1). The model will take long positions of the corresponding index opening price and unwind all the positions at market close.

3.6 Portfolio Weighting

The portfolio weights of the indexes are assigned based on the predicted return of the corresponding index. The model will assign zero weight on the index if the predicted return is smaller than or equal to zero. Else, the weights of each index will be calculated as the weighted average of the predicted return.

3.7 Trading flow

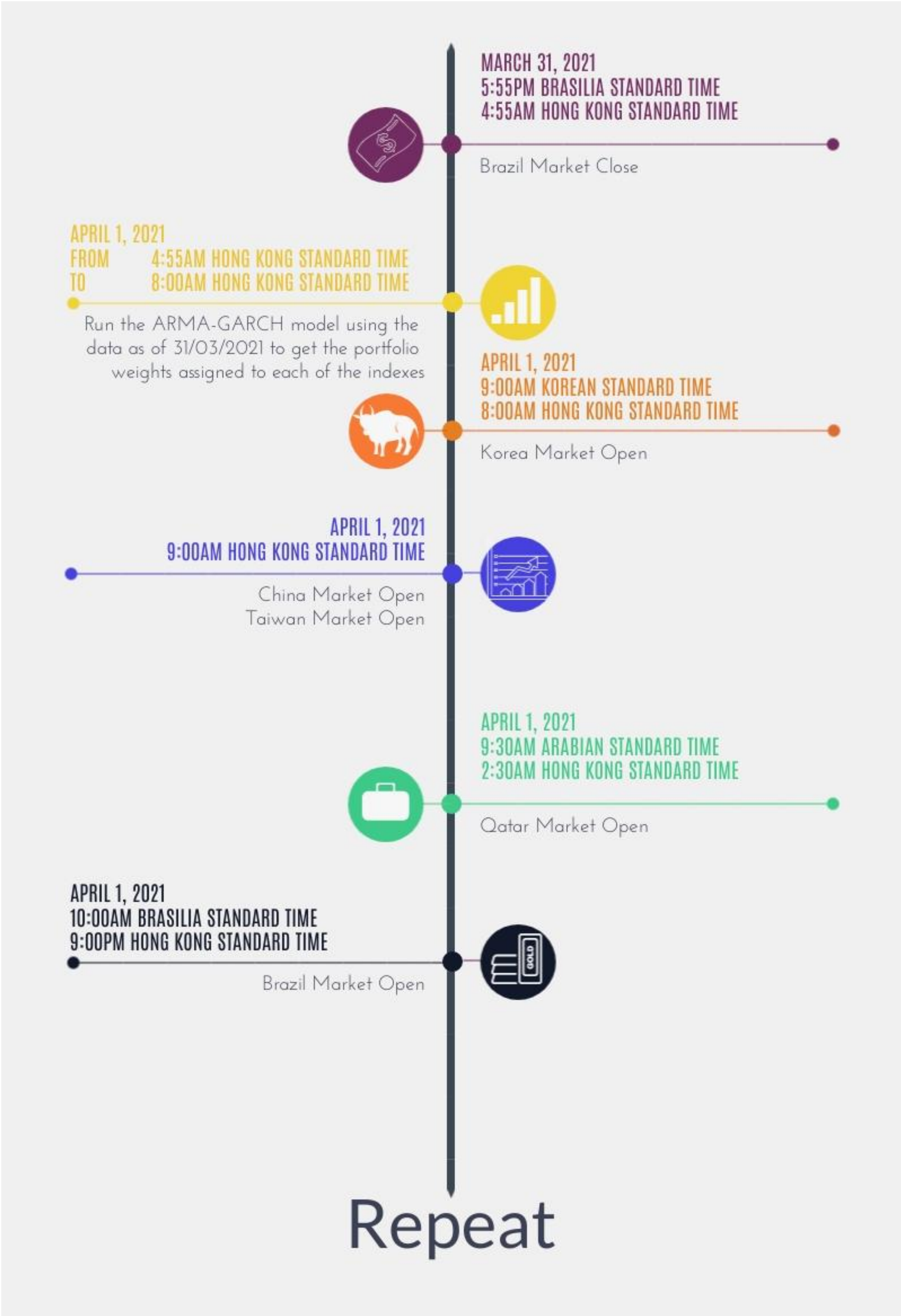

Figure 7: Trading Flow in 1 day

4 Results & Discussions

4.1 Model Prediction: Price

Indexes price

<u>Plots</u>

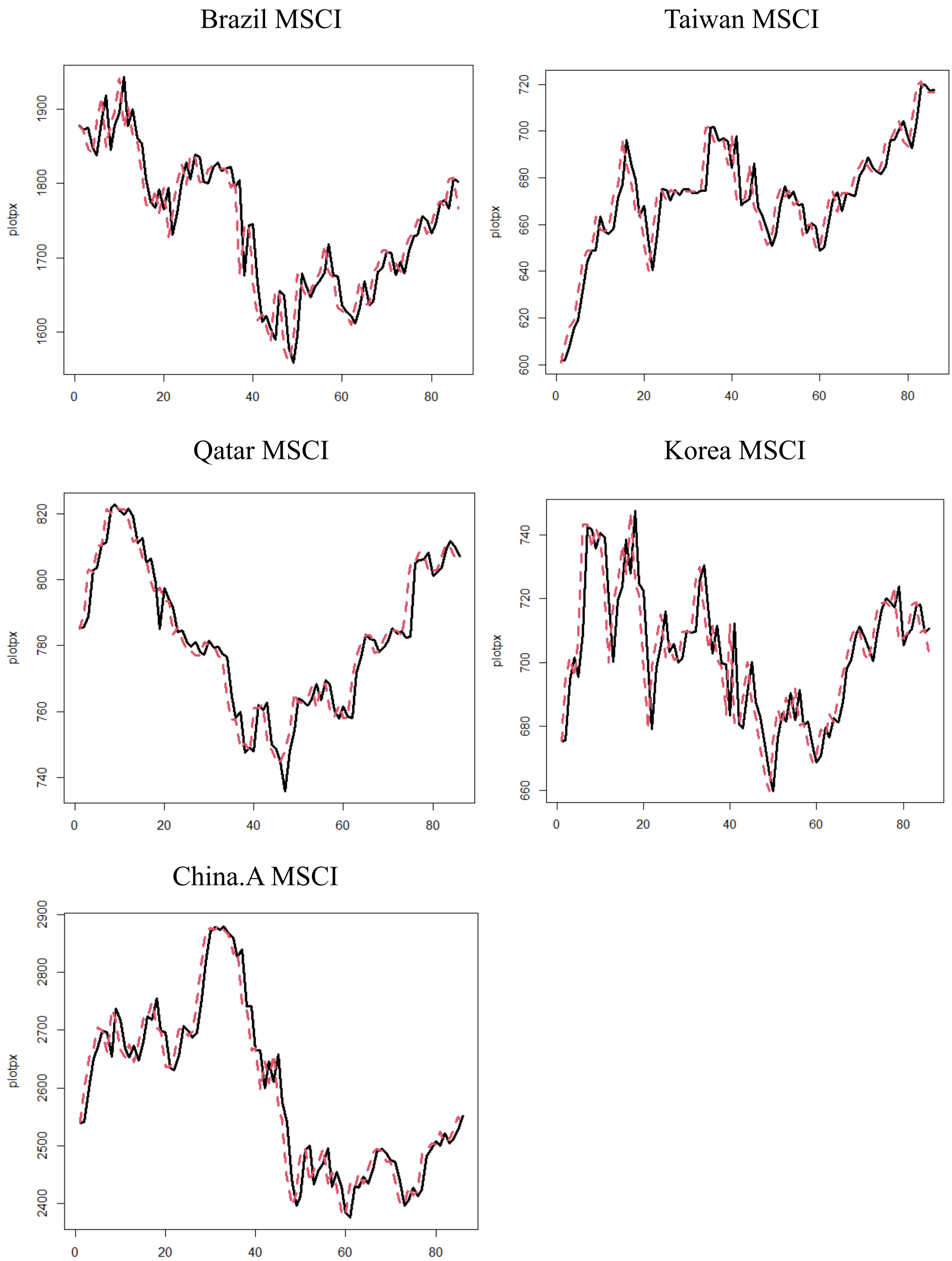

Figure 7: Indexes Actual price in red and Model predicted price in black

It is also observable that the predicted price exhibits a certain extent of a 1-day lag effect. It is because of the rolling window technique we added to the model. However, this dramatically increases the model's ability to make forecasts that track the actual index price well, since this effect allows the model to readjust the forecast promptly. In theory, our model will have good performance in a low volatility environment as it can catch the short-term index price momentum.

4.2 Model Prediction: Return

Indexes return

Plots

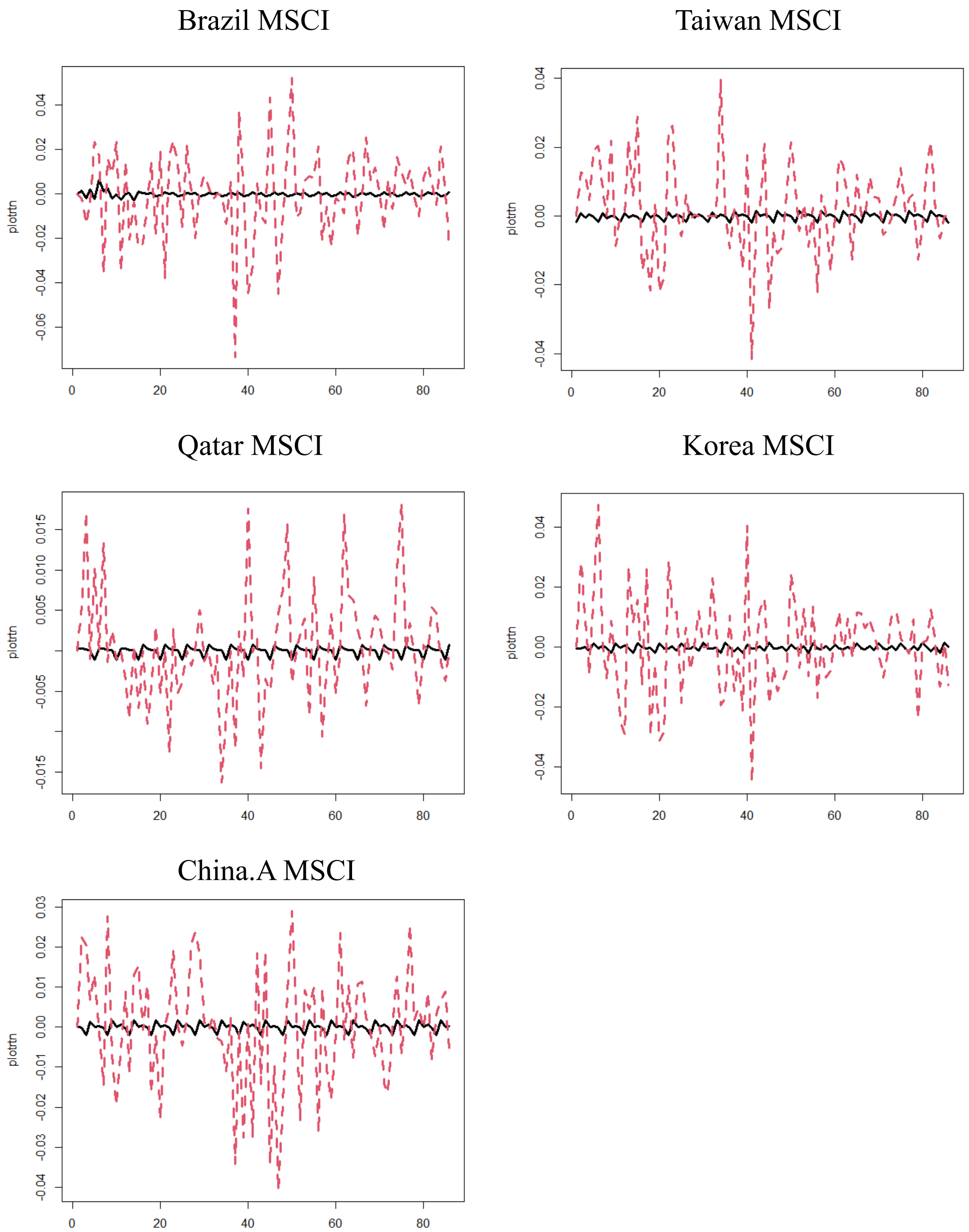

Figure 7: Indexes Actual return in red and Model predicted return in black

Despite the fact that the predicted return, in general, has a very small magnitude because the predicted price at t is close to the actual t-1 price, as mentioned, the rolling window effect, is just an indicator for the model to make investment decisions and assign corresponding weights on the indexes. As long as the predicted return can accurately predict the movement of the indexes' price, the model is going to make a profit in the long run.

4.3 Model Performance: Accuracy

Model Statistics

	Brazil	Taiwan	Qatar	Korea	China
Accuracy	0.4302326	0.5232558	0.4418605	0.6046512	0.4883721
Precision	0.4054054	0.5789474	0.3970588	0.6571429	0.5250000
Recall	0.3571429	0.4680851	0.7941176	0.5111111	0.4565217
F Score	0.3797468	0.5176471	0.5294118	0.5750000	0.4883721

Table 2: Model Statistics Table

4.3.1. Precision

The precision is the indicator that reflects the model's ability to make a correct decision given that the model decided to engage in a trade. This statistics brings more financial interpretation to the model compared to the others. As there will be no profit and loss incurred if the model did not engage in a trade, precision is a more informative factor to be evaluated on when profitability is the final goal of the model.

The model in general can make precise investment decisions. However, the performance in different markets is quite diverse. The performance in Taiwan and Korean stock market is outstanding, having over 55% precision while the performance in Brazil and Qatar is roughly 40%.

4.3.2. Recall

The recall reflects the model's ability to catch the profitable scenario. It evaluates the probability of the model to engage in trades given that the actual return is positive.

It is noticeable that the recall for the Qatar index is around 80%. This means the model can catch 80% of the profitable scenario. However, due to the low precision in the Qatar index, it is believed that the model engaged in Qatar trades too frequently and resulted in a high recall. Although the recall for all the indexes is poor, it is not a huge drawback for the model because we are not aiming to catch all the profitable trades but to make a profit.

4.3.3. F Score

F Score is the statistic that considers the tradeoff of precision and recall. It helps us to evaluate the model's overall performance when a model is over-precise or over-aggressive, like the Qatar index in our model. Even though our model is engaged in too many Qatar trades and resulted in low precision, the F Score pointed out that the model performance in Qatar is better than Taiwan after balancing out the precision and recall in the model.

4.3.4. Accuracy

Accuracy examines the model's chance to "do the right thing", buy-in when the price will increase, and pass when the price will drop or stay at the time (t + 1). This indicator can give us a general sense of how well the model is performing but cannot bring too much value to the model from a financial perspective.

4.4 Model Performance: Cumulative Return

Cumulative return

Black line: Cumulative return for ARMA(2,2)-eGARCH(1,1) model
Red line: Cumulative return for control model using time (t - 1) price as predicted price at time t

Plots

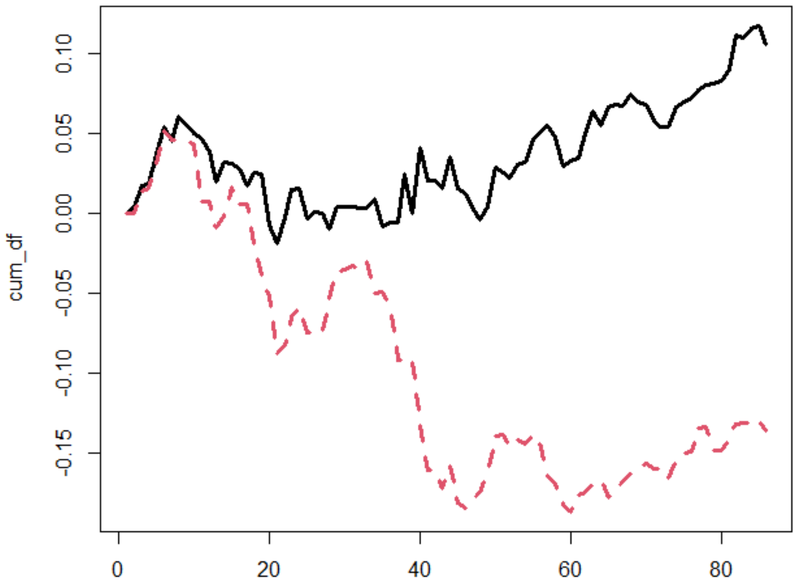

Figure 7: Control cumulative return in red and Model cumulative return in black

The cumulative return also showed that the performance of the ARMA(2,2)-eGARCH(1,1) model has a better prediction power than a martingale model that uses stock price at the time (t - 1) as the best estimate of the stock price at time t. This is clear evidence that the 1-day lag effect we spotted in the price prediction plots is not simply a 1-day lag price.

The cumulative return plot also validates the hypothesis we raised earlier that it catches short-term price momentum. The AR(2) part in the model is believed to be the contributor behind this as the autocorrelation effect in the model captures the 1-day and 2-day price lags to do prediction. The chart also suggested that our model has a strong comovement behavior as the control model's return in the bullish market. As the control model is basically "longing a 1-day lag momentum" (it takes a long position if the

1-day lag return is positive), it means our model is also taking similar approaches, in other words, longing short-term momentum as it is dominated by the autocorrelation effect in a bull market.

What is more, the model can withstand a falling market in February and March credit to the implementation of MA(2) and eGARCH(1,1) in our model. Remarkably, eGARCH can incorporate the asymmetries in stock return volatilities. In the actual stock markets, the bear market tends to have higher volatility, and this effect amplifies in emerging markets. The high volatility environment captured by eGARCH(1,1) during the setback period will then bring a more significant averaging impact on the index price forecast. This implies that the moving average effect in the model will be weaker in a bullish market and stronger in a bearish market. Price is predicted to go down in a high volatility environment.

5 Conclusions

5.1 Summary

The application of ARMA(2,2)-eGARCH(1,1) in index price forecasting for emerging markets is promising. The portfolio can survive the highly volatile environment in emerging markets. The rolling window features reinforced the adaptiveness of the model in a high volatility environment and the AR part granted the ability to capture these short-term momentums in the markets. On the other hand, the model is capable of surviving in market setback periods, like the February and March 2021 data we performed backtesting on, the model is hit during the period but it survives and yields an outstanding performance when the markets recover credit to the MA part and eGARCH part in the model.

5.2 Future developments

During the model construction and selection process, we manually tested various models for each of the indices to find out the model which can fit the model on most of the indices and optimize the portfolio outcome. This is a time-consuming process, especially when we need to deal with 5 indices in this report. There is an existing auto.arima function for finding the best ARIMA model. Hence, to cope with this constraint, an autofit ARMA-GARCH function should be built or created in R, which can improve the efficiency of the model selection process.

Besides that, a more sophisticated trading strategy will be needed. Given the fact that when we calculate the portfolio return, we solely consider the price movement of different indices without taking into account the associated cost induced during the transaction of each trade, for example, broker fees and exchange fees. Especially, our trading strategy is using daily rebalancing of the portfolio, there will be a huge amount of transaction cost when market fictions are added into the return calculation. The model needs to acquire the ability to identify buy-in and unwind opportunities, e.g. by doing 2-day forecasts such that the model will not unwind the position if it predicts the index price will keep increasing over 2 days.

Lastly, the model performance in Brazil and Qatar markets is not ideal, resulting in low precision. Changing or widening the scope is believed to improve the performance if the ARMA(2,2)-eGARCH(1,1) model is kept.

Reference

Bollerslev, T. (1986). *Generalised autoregressive conditional heteroskedasticity.* Journal of Econometrics, 31(3): 307-327.

Engle, R. F. (1982). *Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation*. Econometrica, 50(4), 987–1008. doi:10.2307/1912773

MSCI. *MSCI Index Calculation Methodology*. MSCI, 2021.

MSCI. (2021). MSCI China A Index (USD). https://www.msci.com/documents/10199/35cd04e7-a7f2-41de-9316-e92db9f17cbc

MSCI. (2021). MSCI Taiwan Index (USD). https://www.msci.com/documents/10199/6f36d84d-425d-4e1f-8d56-e65c455ebda1

MSCI. (2021). MSCI Korea Index (USD). https://www.msci.com/documents/10199/e8fc2a89-b809-4088-a807-4b9d9ec04abc

MSCI. (2021). MSCI Brazil Index (USD). https://www.msci.com/documents/10199/efef068e-d252-4e75-8142-2c8b8f441759

MSCI. (2021). MSCI Qatar Index (USD). https://www.msci.com/documents/10199/cd4c3955-b178-4ff5-9d08-a8f73229328f#:~:text=The%20MSCI%20Qatar%20Index%20is,adjusted%20market%20capitalization%20in%20Qatar.&text=The%20MSCI%20Qatar%20Index%20was%20launched%20on%20Jan%2023%2C%202006.

Xu, W. & MSCI. (n.d.). The Rise of Emerging Markets and Asia - Equity-Market Size May Not Reflect the Underlying Economy. MSCI. https://www.msci.com/insights-gallery/emerging-markets

Nelson, D. B. (1991). *Conditional heteroskedasticity in asset returns: A new approach.* Econometrica, 59(2), 347–370. doi:10.2307/2938260

UBS. (2020, August). China A-Shares Top 18 FAQs. https://www.ubs.com/global/en/investment-bank/in-focus/spotlight-on-china/_jcr_content/mainpar/toplevelgrid_6367126/col2/actionbutton_copy.1542577760.file/bGluay9wYXRoPS9jb250ZW50L2RhbS91YnMvZ2xvYmFsL2ludmVzdG1lbnRfYmFuay9jaGluYS1hLXNoYXJlcy10b3AtZmFxcy5wZGY=/china-a-shares-top-faqs.pdf

R Code

Main.R

```r
library(rugarch)
library(zoo)
library(dplyr)
library(caroline)

HInd <- read.csv(file = 'historyIndex.csv')
HInd <- mutate(HInd, Date= as.Date(Date, format= "%d/%m/%Y"))
train <- subset(HInd, Date < as.Date("2021-01-01"))
test <- subset(HInd, Date >= as.Date("2021-01-01"))

weightcaldf <- data.frame(row.names = test$Date)
actrtndf <- data.frame(row.names = test$Date)
# Model specification

for (nindex in (2:length(HInd))) #length(HInd)
{
 Ind <- HInd[,c(1,nindex)]
 Ind$rtn <- c("null", diff(log(Ind[,2])))
 Ind <- Ind[-1,]
 row.names(Ind) <- NULL
 CurInd <- subset(Ind, Date < as.Date("2021-01-01"))
 Curtest <- subset(Ind, Date >= as.Date("2021-01-01"))
 # Simulation
 nsim = length(Curtest[,2]) ### number of forecast to make
 msim = 100  ### number of path
 meanpxlist <- list()
 rtn_list <- list()
 for (nday in (1:nsim))
 {
  print(nday)
```

```
  if (nday == 1)
  {
    datainscope <- CurInd
  } else
  {
    datainscope <- rbind(CurInd, Curtest[1:nday-1,])
  }
  # Auto Fit ARMA GARCH
  sim_spec <- ugarchspec(
    mean.model     = list(armaOrder = c(2, 2), include.mean = FALSE),
    variance.model = list(model="eGARCH", garchOrder=c(1,1)),
    distribution.model = "sstd"
  )
  sim_model <- ugarchfit(spec = sim_spec, data = datainscope$rtn,  solver.control = list(tol = 1e-12))
  setfixed(sim_spec) <- as.list(coef(sim_model))
  # Forecast
  forc <- ugarchforecast(fitORspec = sim_spec, data = datainscope$rtn, n.ahead = 1)
  simul <- ugarchpath(spec=sim_spec, m.sim=msim, n.sim=1,rseed=sample(1:1000, 1))
  # output px
  last_px <- datainscope[,2][length(datainscope[,2])]
  px <- last_px * apply(fitted(simul), 2, 'cumsum') + last_px
  meanpx <- data.frame(Means=mean(px))
  meanpxlist <- append(meanpxlist,meanpx)
  # output return
  rtn_n <- log(meanpx/last_px)
  rtn_list <- append(rtn_list,rtn_n)
}

meanpxdf <- t(data.frame(meanpxlist))
rtndf <- t(data.frame(rtn_list))
plotpx <- data.frame(predicted_px = meanpxdf, Actual_px = Curtest[,2])
row.names(plotpx) <- as.Date(Curtest$Date)
```

```
  plotrtn <- data.frame(predicted_rtn = rtndf, Actual_rtn = Curtest[,3])
  row.names(plotrtn) <- as.Date(Curtest$Date)
  # adjusted predicted return
  adjusted_p_rtn <- plotrtn$predicted_rtn
  adjusted_p_rtn[adjusted_p_rtn<0] <- 0
  weightcaldf[colnames(Curtest)[2]] <- adjusted_p_rtn
  # Actual return
  actrtndf[colnames(Curtest)[2]] <- plotrtn$Actual_rtn
  # Graph
  par(mfrow=c(1,1))
  matplot(plotpx, type = 'l' , lwd=3)
  matplot(plotrtn, type = 'l' , lwd=3)
}

# Weight calculation
weightcaldf$Sum <- rowSums(data.frame(sapply(weightcaldf, function(x) as.numeric(x))))
weights <- data.frame(row.names = Curtest$Date)
wgted_rtn <- data.frame(row.names = Curtest$Date)
for (col_ind in 1:ncol(weightcaldf)-1)
{
  weights[colnames(weightcaldf)[col_ind]] <- weightcaldf[,col_ind]/weightcaldf[,length(weightcaldf)]
}
weights[is.na(weights)] <- 0
# Weighted Return
actrtndf <- data.frame(sapply(actrtndf, function(x) as.numeric(x)), row.names = Curtest$Date)
for (col_ind in 1:ncol(weights))
{
  wgted_rtn[colnames(weights)[col_ind]] <- weights[colnames(weights)[col_ind]][,1] *
actrtndf[colnames(actrtndf)[col_ind]][,1]
}
wgted_rtn$Sum <- 0
wgted_rtn$Sum <- rowSums(data.frame(sapply(wgted_rtn, function(x) as.numeric(x))))
# Cumulative return
```

```
prod <- 1
cum_df <- data.frame(row.names = Curtest$Date)
cum_list <- list()
for (ws in wgted_rtn$Sum + 1){
  prod <- prod * ws
  cum_list <- append(cum_list,prod-1)
}
cum_df$rtn <- cum_list

# Control (lag-1)
Ctl <- rbind(train[length(train$Date),],train[length(train$Date),], test[1:length(test$Date)-1,])
Ctl$Date <- c(as.Date("2020-12-31"), Curtest$Date)
Ctl_rtn <- data.frame(row.names = Ctl$Date)
Ctl_wgted_rtn <- data.frame(row.names = Curtest$Date)
for (nindex in (2:length(Ctl)))
{
  Ctl_rtn[colnames(Ctl)[nindex]] <- c("null", as.vector(diff(unlist(log(Ctl[colnames(Ctl)[nindex]])))))
}
Ctl_rtn <- Ctl_rtn[-1,]
Ctl_rtn <- data.frame(sapply(Ctl_rtn, function(x) as.numeric(x)))
Ctl_rtn[Ctl_rtn < 0] <- 0
Ctl_rtn$Sum <- rowSums(Ctl_rtn)
Ctl_weights <- data.frame(row.names = Curtest$Date)
for (col_ind in 1:ncol(Ctl_rtn)-1)
{
  Ctl_weights[colnames(Ctl_rtn)[col_ind]] <- Ctl_rtn[,col_ind]/Ctl_rtn[,length(Ctl_rtn)]
}
Ctl_weights[is.na(Ctl_weights)] <- 0
for (col_ind in 1:ncol(Ctl_weights))
{
  Ctl_wgted_rtn[colnames(Ctl_weights)[col_ind]] <- Ctl_weights[colnames(Ctl_weights)[col_ind]][,1] *
actrtndf[colnames(actrtndf)[col_ind]][,1]
}
```

```
Ctl_wgted_rtn$Sum <- 0
Ctl_wgted_rtn$Sum <- rowSums(data.frame(sapply(Ctl_wgted_rtn, function(x) as.numeric(x))))
# Cumulative return
prod <- 1
cum_list <- list()
for (ws in Ctl_wgted_rtn$Sum + 1){
  prod <- prod * ws
  cum_list <- append(cum_list,prod-1)
}
cum_df$lag1rtn <- cum_list
matplot(cum_df, type = 'l' , lwd=3)

# Accuracy
accdf <- data.frame(row.names = Curtest$Date)
pre_acc <- weightcaldf/abs(weightcaldf)
pre_acc[is.na(pre_acc)] <- -1
pre_acc <- pre_acc[,c(-length(pre_acc))]
act_acc <- actrtndf/abs(actrtndf)
act_acc[is.na(act_acc)] <- -1
for (col_ind in 1:ncol(pre_acc))
{
  accdf[colnames(pre_acc)[col_ind]] <- pre_acc[,col_ind] * act_acc[,col_ind]
}
accdf[accdf < 0] <- 0
acctable <- data.frame(row.names = "Average Accuracy")
for (col_ind in 1:ncol(accdf))
{
  acctable[colnames(accdf)[col_ind]] <- mean(unlist(accdf[colnames(accdf)[col_ind]]))
}
# Precision
prectable <- data.frame(row.names = "Average Precision")
for (col_ind in 1:ncol(pre_acc))
```

```
{
  pre_pos <- pre_acc[colnames(pre_acc)[col_ind]][pre_acc[colnames(pre_acc)[col_ind]] > 0]
  map_act <- act_acc[colnames(pre_acc)[col_ind]][pre_acc[colnames(pre_acc)[col_ind]] > 0]
  preclist <- pre_pos*map_act
  preclist[preclist < 0] <- 0
  prectable[colnames(pre_acc)[col_ind]] <- mean(preclist)
}
# Recall
recalltable <- data.frame(row.names = "Average Recall")
for (col_ind in 1:ncol(act_acc))
{
  act_pos <- act_acc[colnames(act_acc)[col_ind]][act_acc[colnames(act_acc)[col_ind]] > 0]
  map_pred <- pre_acc[colnames(act_acc)[col_ind]][act_acc[colnames(act_acc)[col_ind]] > 0]
  recalllist <- act_pos*map_pred
  recalllist[recalllist < 0] <- 0
  recalltable[colnames(act_acc)[col_ind]] <- mean(recalllist)
}
# F
statmtx <- rbind(acctable, prectable, recalltable)
f <- 2*(statmtx[2,]*statmtx[3,])/(statmtx[2,]+statmtx[3,])
rownames(f) <- "F Score"
statmtx <- rbind(statmtx, f)
statmtx
```

Model_Selection.R

```
HInd <- read.csv(file = 'historyIndex.csv')
HInd <- mutate(HInd, Date= as.Date(Date, format= "%d/%m/%Y"))
test <- subset(HInd, Date >= as.Date("2021-01-01"))
library(fGarch)
library(forecast)
library(doParallel)
library(rugarch)
Bra <- HInd[1:1044,2]
BraReturns = diff(log(Bra))
###Brazil###
sim_spec <- ugarchspec(
  mean.model     = list(armaOrder = c(2, 2), include.mean = FALSE),
  variance.model = list(model="eGARCH", garchOrder=c(1,1)),
  distribution.model = "sstd"
)
sim_model <- ugarchfit(spec = sim_spec, data = BraReturns)
sim_model
###Taiwan###
TW <- HInd[1:1044,3]
TWReturns = diff(log(TW))
sim_spec <- ugarchspec(
  mean.model     = list(armaOrder = c(2, 2), include.mean = FALSE),
  variance.model = list(model="eGARCH", garchOrder=c(1,1)),
  distribution.model = "sstd"
)
sim_model <- ugarchfit(spec = sim_spec, data = TWReturns)
sim_model
###QATAR###
QT <- HInd[1:1044,4]
QTReturns = diff(log(QT))
sim_spec <- ugarchspec(
```

```
  mean.model     = list(armaOrder = c(2, 2), include.mean = FALSE),
  variance.model = list(model="eGARCH", garchOrder=c(1,1)),
  distribution.model = "sstd"
)
sim_model <- ugarchfit(spec = sim_spec, data = QTReturns)
sim_model
###Korea###
KOR <- HInd[1:1044,5]
KORReturns = diff(log(KOR))
sim_spec <- ugarchspec(
  mean.model     = list(armaOrder = c(2, 1), include.mean = FALSE),
  variance.model = list(model="eGARCH", garchOrder=c(1,1)),
  distribution.model = "sstd"
)
sim_model <- ugarchfit(spec = sim_spec, data = KORReturns)
sim_model
###China###
CHI <- HInd[1:1044,6]
CHIReturns = diff(log(CHI))
sim_spec <- ugarchspec(
  mean.model     = list(armaOrder = c(2, 2), include.mean = FALSE),
  variance.model = list(model="eGARCH", garchOrder=c(1,1)),
  distribution.model = "sstd"
)
sim_model <- ugarchfit(spec = sim_spec, data = CHIReturns)
sim_model
```