QWIM Project
Applications of machine learning for empirical asset pricing and risk premia forecasting

Author

Yu-Ting Chen
Yihua Guo
Raye Shen

Professors

Cristian Homescu
Agnes Tourin

New York University
Tandon School of Engineering
Department of Finance and Risk Engineering

Contents

I Abstract **1**

II Introduction **1**

III Literature Review **2**

IV Methodology **3**
- Data and Data pre-processing 4
- Univariate Forecasting Method 5
  - Naive Forecaster 6
  - Exponential Smoothing 6
  - Theta Forecaster 7
  - Reduced Regression: KNN Regression 8
  - Reduced Regression: Random Forest 9
  - Pipeline Model 10
- Multivariate Forecasting Method 10
  - Vector AutoRegression (VAR) 11
  - ESRNN 12
- Portfolio Optimization 16
  - Mean Variance: quadratic risk utility 16
  - Max Diversification 17
  - Minimum Variance 18
  - Hierarchical Risk Parity (HRP) 18
  - Black Litterman 19
  - Critical Line Algorithm (CLA) 19

V Analysis and Result **20**
- Prediction Interval 20
- Forecasting Values Comparison 22
- Forecasting Metrics Comparison 27
- Portfolio Optimization Comparison 30

VI Summary and Conclusion **45**

VII Appendix **47**
- Graphs for Optimization Result (2020-01-03 - 2020-07-17) 47
- Charts for Optimization Result (1996-03-19 - 2020-07-17) 49
- Charts for Optimization Result (1996-03-19 - 2020-01-02) 51

VIII Reference **54**

I Abstract

Machine learning has gained enormous popularity for asset pricing and forecasting in recent research studies. This paper presents a comprehensive analysis of some of the "most promising" machine learning methods demonstrated in literature. In this study, eight selected forecasting methods will be applied to five time series, and the predicted time series will be analyzed based on eight portfolio optimizers. By conducting the performance measurement, we identify the best performing models and trace their advantages and disadvantages.

II Introduction

A lot of literature have investigated cumulative stock market returns or ETF/Index returns predictability using machine learning techniques. The future looms largely on machine learning and AI application in numerous aspects, especially for the finance field. In the past, most of the researches utilized plethora of predictors to predict the return, such as the intraday price, trading volume of the day, interest rate and other macroeconomic factors, which needs further research such as feature engineering to determine whether these factors truly contribute to the predictions of the return for ETF/Index. These studies mostly arose from the M4 competition. Different from classical machine learning or deep learning methods, these methods are specially designed for time series data. In this paper, we will use only the time series itself to predict its future values. The predictor should be the lagged return, and the lag is decided specifically to algorithm's internal computations.

We used eight forecasting methods - Naive forecaster, Exponential Smoothing, Theta forecaster, KNN regression, Random Forest, Pipeline Model, Vector Autoregression, and ES-RNN. We analyzed the predictive ability of the lagged return of ETF and other major indices across various industries. Since classical machine learning has drawbacks on capturing the fluctuations in time series data itself, simple classical methods such as Exponential Smoothing and ARIMA are expected to perform better in such cases. Nevertheless, the classical methods rely on some assumptions. To begin with, these methods cannot handle missing values in the dataset. Secondly, the data must have linear relationships, which implies the trend component must have downward or upward trends overall. Next, most of the models work on only univariate forecasting, which means a majority of classical methods used here cannot be applied with the model that has exogenous predictors. On the other hand, deep learning methods such as Multi-layer perceptrons(MLP), RNN, and CNN have their own merits when conducting the time series data forecasting. For MLP, it can directly handle the missing inputs in the data set and it accepts multivariate input data; CNN will implement feature engineering internally on the raw data; LSTM can not only handle multivariate input, but it is also robustness to noise and possesses feature engineering functions. Moreover, the LSTM model has an internal mapping function as well, which it can map the input to the output from time to time. Therefore, aside from the classical methods using for time series, we also manipulated the hybrid model ESRNN, which is the combination of exponential smoothing and Recurrent neural network.

For the out-sample evaluations, there are three aspects to compare. First, the prediction return and the actual return value comparison. This part was shown by the graphs and some simple metrics comparisons. The next part combined the previous forecasting methods with seven different optimization methods. With these optimizers, returns of each ETF and Index were assigned with a variety of weights. The forecasting performance was again compared with the return and these weights. Eventually, in this case, we concluded than classical methods and hybrid models performed better that classical machine learning. The rest of the paper was arranged as the following: section III summarized some methods and metrics in literature. Section IV explained the methodologies we used in detail. This section began with the five ETF/Indices as our input data, and we graphed the data distribution after transformation. We then illustrated the theories and formulas behind the forecasting models, separated by univariate and multivariate models. The last part of section IV demonstrates the optimization methods. Section V presents a close look at the predicted results and compares the models' forecasting metrics. Then for each optimizer, three portfolios constructed from optimization methods are analyzed and compared based on portfolio metrics. Section VI summarized the analysis and provided recommendations for further research. The appendix includes comparison graphs for optimization results and charts for portfolio metrics evaluations.

III Literature Review

Literature is composed of three types: most of them use a list of Firm Characteristics(FC) as the predictor variable and perform feature selection on FC; a small portion of papers, on the other hand, use the time series itself or the lagged return as the predictors; last but not least, a very few papers served as summaries for other literature that have used machine learning methods. Although in this paper we only focused on time series prediction with itself being the predictor, we will provide some examinations on all methods used by researches.

Rapach, D E., et al. (2019) [48] used the monthly lagged industry return data from 19060 to 2016, applied an OLS post-LASSO method and achieved an average return of 7.33% per annum and annualized Sharpe Ratio of 0.65%. There are some papers also utilized LASSO models, for example, Freyberger, J. et al.(2018) [20] and Audrino, F(2017) [34] both used LASSO to select features from an FC set as large as 62 characteristics including size, book-to-market, beta, and other prominent variables. Weigand, A.(2019) [49] provides a comprehensive summary of popular machine learning methods.

Deep learning techniques are also widely used among researchers. Feng, G, et al.(2018) [12] compared three deep learning architectures of different number of neurons with Ridge, PLS, Lasso, ElasNet; Messmer, M(2017) [43] built a feedforward neural networks (DFN) based on a set of 68 firm characteristics (FC). Mayster, B.(2017) [34] compared DNN, Random Forest, SVM on (A) prior prices, (B) prior volume, (C) dummies for days of the week and months of the year, and (ABC) all information sets combined. An interesting method Mayster, B.(2017) [34] used to measure the forecasters' performance is **Gain Criteria**, where it is calculated by creating random data from a uniform distribution bounded by 0 and 1 and replacing the original input data with these simulated data in the modeling process. Then $Gain^{ETF}$ = *Actual Data Test Score*^{ETF} - *Simulated Data Test Score*^{ETF}.

According to the author of M4[38], the M4 competition is the continuation of three previous competitions started more than 45 years ago whose purpose was to learn how to improve forecasting accuracy, and how such learning can be applied to advance the theory and practice of forecasting. The purpose of M4 was to replicate the results of the previous ones and extend them into three directions: First significantly increase the number of series, second **include Machine Learning (ML) forecasting methods**, and third evaluate both point forecasts and prediction intervals. M4 could outplay the traditional machine learning methods by incorporating a "hybrid" approach that utilizes both statistical and machine learning features. The M4 model has 10% higher sMAPE than the benchmark model and could achieve amazing success in specifying the 95% prediction intervals correctly.

As for the model performance, most literature used MSE and out-of-sample R^2.

Summarized below are some common conclusions celebrated by most researches:

- The majority of FC plays a minor role in the variation of these predictions.
- Allowing for nonlinearities substantially improves predictions.
- The distance between nonlinear methods and the benchmark widens when predicting portfolio returns
- The economic gains from machine learning forecasts are large.
- **Machine learning methods on their own do not identify deep fundamental associations among asset prices and conditioning variables.**

IV Methodology

In this article, two different forecasting directions will be introduced, first one is **Univariate Forecasting**, second one is **Multivariate Forecasting**. Some models related to the M4 Competition will be introduced.

As for the Univariate Forecasting, **Sktime** package in **Python** will be introduced. Sktime is a new scikit-learn compatible Python library with a unified interface for machine learning with time series. This paper uses Sktime to both replicate and extend key results from the M4 forecasting study. In particular, it further investigates the potential of simple off-the-shelf machine learning approaches for univariate forecasting[40]. Six different statistical models combined with Machine Learning regression models will be tested, including **Naive Forecaster, Exponential Smoothing, Theta Forecaster, KNN Regression, Random Forest, and Pipeline model**.

In the Multivariate Forecasting, two different models will be introduced: **VAR** and **ESRNN**. Compared to Univariate Forecasting, it considers the patterns between each time series and evaluates features in batches.

After implementing the forecasting models in **Python**, different forecasting metrics such as **sMAPE, MAE, MASE, RMSE** will be defined to evaluate the forecasting performances of different models in the **Analysis and Result** section. The forecasting metrics will be primarily calculated outside of the training period for the forecasting methods, and metrics evaluation on each series will be presented in this article.

Besides, this article would incorporate analysis of benchmark portfolios selected from the most common "optimal portfolio" types used in the industry and academia. Several different **portfolio optimization** methods will be applied to calculate the optimal portfolios within each forecasting methods talked above. **portfolio metrics** would be calculated over the 3 timelines. Combining all the methodology discussed above, this article would compare:

- Standalone forecasting performance (using forecasting metrics)
- Forecasting performance within the context of optimal portfolios (using portfolio metrics)

Data and Data pre-processing

Risk Premium Data are retrieved from Bloomberg within the time range of 1996/3/19 to 2020/7/27. Five equal-length ETF/Index series are picked as examples in this project.

1. iShares Russell Top 200 ETF
2. MSCI Japan Index
3. S&P 500 Information Technology Index
4. S&P 500 Health Care Index
5. S&P 500 Industrials Index

Since most of the forecasting models require all positive values, risk premium + 1 will be used as input.

Next, 13-week Treasury bill rate is treated as risk free rate, risk premium is calculated as $rp_i = r_i - r_f + 1$ on each day during 1996/3/19 to 2020/7/27 to make sure that all the values are positive.

After checking that there are no missing values in the five series, we will define the Train/Test sets in this way: keep 80% early data as training data and the rest of 20% data as test data. Each point in the series represents a day. After splitting the training and test data (y_{train}, y_{test}), the forecasting horizon y_{pred} pass to our forecasting algorithm should have the same length and time horizon with y_{test} .

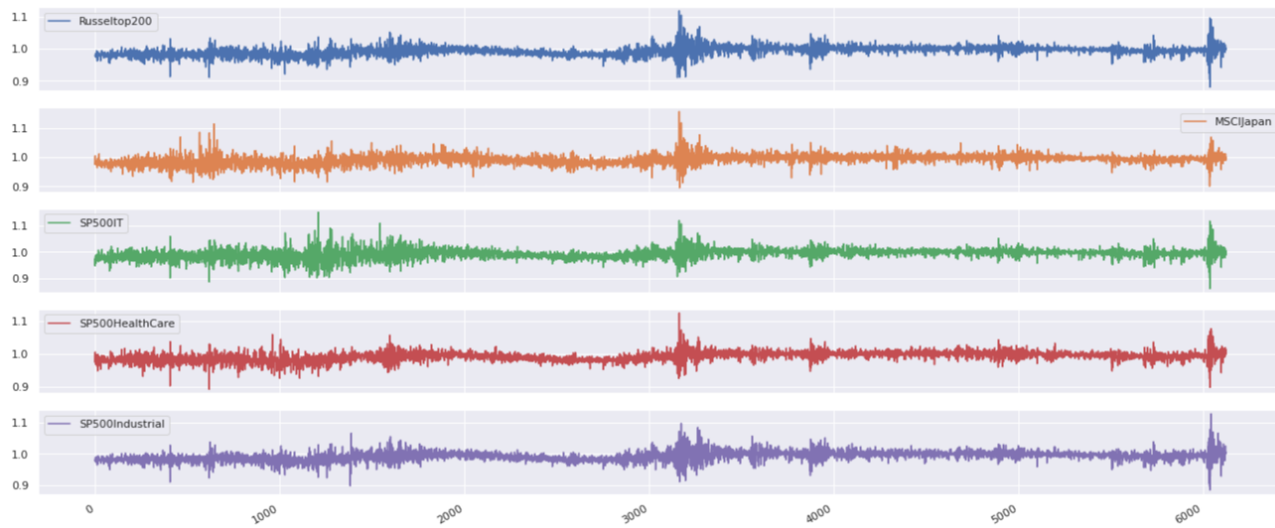

Figure 1: Five ETF/Index Risk Premium Data

Univariate Forecasting Method

In this paper, all models are tested by using **Sktime in Python**, which is an open-source machine learning toolbox for forecasting that allows to easily build, tune and evaluate composite machine learning models and is compatible with one of the major machine learning toolboxes (scikit-learn). For the practioner's problem, classical univariate forecasting problem can be considered with discrete time points. The task is to use the observations $y = (y(t_1)...y(t_T))$ of a single time series observed up to time point t_T to find a forecaster $\hat{f}$ which can make accurate temporal forward predictions $\hat{y}(h_j) = \hat{f}(h_j)$ for the given time points $h_1...h_H$ of the forecasting horizon.[40]

Like in scikit-learn, in order to make forecasts, need to first specify (or build) a model, then fit it to the training data, and finally call predict to generate forecasts for the given forecasting horizon. **Sktime** comes with several forecasting algorithms (or forecasters) and tools for composite model building. All forecaster share a common interface. Forecasters are trained on a single series of data and make forecasts for the provided forecasting horizon.

In order to train models and make forecasts, an Augmented Dickey–Fuller test is conducted to check stationary. The following table shows the result from the ADF test. All series have p-value < 0.5 to 0.6, meaning that the risk premia series are relatively stationary.

Augmented Dickey-Fuller Test	ADF Statistic	p-value	Critical(1%)	Critical(5%)	Critical(10%)
Russeltop200	-2.738239	0.067652	-3.432	-2.862	-2.567
MSCIJAPAN	-3.173556	0.021567	-3.432	-2.862	-2.567
SP500IT	-4.106062	0.000946	-3.432	-2.862	-2.567
SP500HealthCare	-2.839443	0.052857	-3.432	-2.862	-2.567
SP500Industrial	-3.249027	0.017317	-3.432	-2.862	-2.567

Table 1: Augmented Dickey-Fuller Test Result

Naive Forecaster

NaiveForecaster is a forecaster that makes forecasts using simple strategies. Naïve forecasts are the most cost-effective forecasting model and provide a benchmark against which more sophisticated models can be compared. This forecasting method is only suitable for time series data.[23] Using the naïve approach, forecasts are produced that are equal to the last observed value. This method works quite well for economic and financial time series, which often have patterns that are difficult to reliably and accurately predict.

In this article, time series is believed to have seasonality, the seasonal naïve approach may be more appropriate to be applied. So seasonal naive forecaster predicts the last value observed in the same season:

$$\hat{y}_{T+h|T} = y_T \tag{1}$$

Exponential Smoothing

With the exponential smoothing method, past observations are weighted. Since the weights decaying exponentially as time past, the more recent observations will possess higher weight compared to the older observations. The method chosen is the Holt-Winters' multiplicative method[27]. This method assumes that the seasonal variations are changing proportionally to the level of the series. In this case, the seasonal component will be represented by percentages, and the time series data will be adjusted seasonally by dividing through the seasonal component [31]. In the model shown below, the seasonality is adjusted by setting $m = 252$, which means that within each year, the seasonal component will sum up to approximately 252.

$$
\begin{aligned}
\hat{y}_{t+h|t} &= (l_t + hb_t)s_{t+h-m(k+1)} \\
l_t &= \alpha \frac{y_t}{s_{t-m}} + (1-\alpha)(l_{t-1} + b_{t-1}) \\
b_t &= \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1} \\
s_t &= \gamma \frac{y_t}{(l_{t-1} + b_{t-1})} + (1-\gamma)s_{t-m}
\end{aligned} \tag{2}
$$

which embedded with three smoothing components:

- l_t : Indicates the level.
- b_t : Indicates the trend.
- s_t : Indicates the seasonal components.

with their corresponding coefficients - α, β^* and γ.

Aside from the multiplicative Holt-Winters' method, there are also other kinds of exponential smoothing methods that can be applied, such as the additive method and linear method.

Theta Forecaster

The "Theta method" of forecasting was introduced by Assimakopoulos and Nikolopoulos (2000). The method performed particularly well in the **M3-competition** (Makridakis and Hibon, 2000) and is, therefore, of interest to forecast practitioners. However the description of the method is quite complicated, Hyndman and Baki [30] found that it can be expressed much more simply than in A&N, which is similar to simple exponential smoothing (SES) with drift. By using this similarity, appropriate prediction intervals for the method based on a state-space model underlying SES with drift.

As for the mathematical expression, let $X_1, ..., X_n$ denote the observed univariate time series. From this series, A&N created a new series $Y_{1,\theta}, ..., Y_{n,\theta}$ such that for t = 3,...,n:

$$Y''_{t,\theta} = \theta X''_t \tag{3}$$

where X''_t denotes the second difference of X_t and $Y''_{t,\theta}$ denotes the second difference of $Y_{t,\theta}$. We note that 3 is a second-order difference equation and has the solution:

$$Y_{t,\theta} = a_\theta + b_\theta(t-1) + \theta X_t \tag{4}$$

where $a_\theta and b_\theta$ are constants and t=1,...,n. Thus, $Y_{t,\theta}$ is equivalent to a linear function of X_t with a linear trend added.

Forecasts from the Theta method are obtained by a weighted average of forecasts of $Y_{t,\theta}$ for different values of θ. In this case, the h-step ahead forecast based on observations $X_1, ..., X_n$ us given by:

$$X_n\hat{(}h) = \frac{1}{2}[\hat{Y}_{n,0}(h) + \hat{Y}_{n,2}(h)] \tag{5}$$

where $\hat{Y}_{n,0}(h)$ is obtained by extrapolating the linear part of 4 and $\hat{Y}_{n,2}(h)$ is obtained using SES on the series $Y_{t,2}$. Hence:

$$\hat{Y}_{n,0}(h) = \hat{a}_0 + \hat{b}_0(n + h - 1) \tag{6}$$

$$\hat{Y}_{n,2}(h) = \alpha \sum_{i=0}^{n-1} (1-\alpha)^i Y_{n-i,2} + (1-\alpha)^n Y_{1,2} \tag{7}$$

where α is the smoothing parameter for the SES, note that SES forecasts are equivalent for all h.

Reduced Regression: KNN Regression

Consider that combining the Machine Learning algorithm in univariate time series forecasting. Sktime provides a meta-estimator for this approach, which is: "modular and compatible with scikit-learn, so that we can easily apply any scikit-learn regressor to solve our forecasting problem." [40] Because of the meta-forecaster for reduction to time series regression, it adapts the scikit-learn's estimator interface to that of a forecaster, making sure that we can tune and properly evaluate our model.

In this section and next section, KNN and Random Forest Model with tuning will be introduced in this paper. Here the KNN regressor will be used as a regressor by using **ReducedRegressionForecaster** in Sktime.

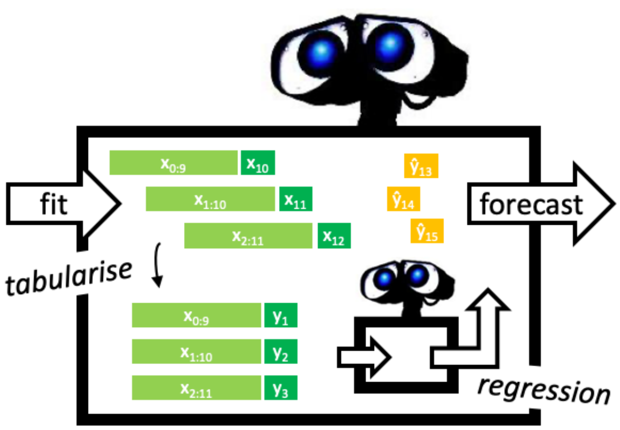

Figure 2: Reduction: from forecasting to regression

As for the KNN regression, KNN is a simple algorithm that can store the available cases and predict the numerical target based on a distance function. The most common way to find this distance is the Euclidean distance as shown below:

$$d(q,p) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \tag{8}$$

KNN regression is to calculate the average of the numerical target of the K nearest neighbors. In this paper, K nearest neighbors = 1 after tuning the hyperparameters.

Reduced Regression: Random Forest

Random Forest regressor is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the prediction accuracy. Since a forest averages the prediction of a set of m trees with individual weight functions W_j, a common mathematical expression can be shown below, where x_i will change to the time series itself.

$$\hat{y} = \frac{1}{m}\sum_{j=1}^{m}\sum_{i=1}^{m} W_j(x_i, x^{'})y_i = \sum_{i=1}^{n}(\frac{1}{m}\sum_{j=1}^{m} W_j(x_i, x^{'}))y_i \tag{9}$$

Following the previous section of talking about how many machine learning strategies are expressible as meta-estimators, reduction approaches, and model selection routines like grid-search and cross-validation. **ReducedRegressionForecaster** is a meta-estimator with explicit, tunable hyperparameters for determining the window length and the method for iterating over multiple output periods of the forecasting horizon [41]. In this section, the regressor to be used will change to RandomForestRegressor, and in the **ReducedRegressionForecaster**, **window length** is the hyper-parameter which we may want to optimize. In this paper, a sequence of numbers from 10 to 25 have been tested. Note that **Window Length** is used by Sktime as a temporal time series splitter, similar to the cross-validation splitter in scikit-learn.

Window length Tuning Result		
Series	**window length**	**Best hyperparameter**
Russeltop200	10,15,20,25	25
MSCIJapan	10,15,20,25	25
SP500IT	10,15,20,25	20
SP500HealthCare	10,15,20,25	25
SP500Industrial	10,15,20,25	25

Table 2: ReducedRegression: Random Forest window length Tuning Result

Pipeline Model

Up to now, different statistical methods and machine learning methods have been tested. Note that the reduction regression approach above does not take any seasonal or trend into account, but a pipeline model can combine the **De-trending, De-seasonalization, Tuning** and **Regression** together. Sktime provides the Pipeline class that allows us to apply data transformation before fitting, and apply the inverse transformation to the predicted values. In this section, KNN will be used again as the machine learning regression techniques.

Detrending: Sktime provides a generic detrender, transforms an input time series, returning a detrended time series in the same domain as the input series[41], for example, polynomial detrending (used in this paper). Detrending transformers keep track of the time index seen in fitting, so that trends are correctly computed over new time indices when transforming new data.

Deseasonalization: Same methodology as talked in **Detrending**, it returns a seasonal detrend time series.[41]

Tunning: Again, the hyperparameters of components of the pipeline will be optimized:

Window length Tuning Result		
Series	**window length**	**Best hyperparameter**
Russeltop200	10,15,20,25	25
MSCIJapan	10,15,20,25	20
SP500IT	10,15,20,25	25
SP500HealthCare	10,15,20,25	25
SP500Industrial	10,15,20,25	20

Table 3: Reduced Regression: KNN window length Tuning Result

By Combining all the methodologies and features above, the pipeline model using KNN regressor can be created.

Multivariate Forecasting Method

So far, Six statistical and machine learning combined univariate time series models have been discussed above, which are implemented by **Sktime in Python**. Actually, univariate time series modeling is the most commonly used forecasting approach, however, univariate forecasting assumes that each series are forecasted by itself and there are no other factors' effects. In this article, five ETFs/Index data are selected and our purpose is to forecast all of them. **Multivariate time series ofrecasting** considers the correlations and patterns between each series, and output the results simultaneously. However, it doesn't mean that multivariate time series forecasting must perform better than univariate time series forecasting. In the following sections, two multivariate time series approaches will be introduced, **Vector AutoRegression and ESRNN** models.

Vector AutoRegression (VAR)

A most common used statistical multivariate time series forecasting technique is called **Vector AutoRegression (VAR)**. VAR is a stochastic model that captures the linear correlations among multiple series, and it is an extension of univariate autoregressive model (AR), which can forecast multiple variables instead of one. Each variable is based on its own lagged values, other variables lagged values, and error term.

Consider a set of k variables (in this project: k = 5) over the same sample period (t = 1,...,T) as a linear function of their past values. The variables are collected in a k-vector. $y_{i,t}$ denotes the observation at time t of the i-th variable. **VAR(p)** can be expressed below:

$$y_t = c + A_1 y_{t-1} + A_2 y_{t-2} + ... + A_p y_{t-p} + e_t \tag{10}$$

where c is a k-vector of constants(intercepts), p denotes the number of lags, A_i is a time-invariant $k \times k$-matrix and e_t is a k-vector of error terms satisfying

- $E(e_t) = 0$: every error term has mean zero
- $E(e_t e_t^{'}) = \Omega$: the covariance matrix of error term is Ω
- $E(e_t e_{t-k}^{'}) = 0$: for any non-zero k, there is no serial correlation in individual error terms.

This project will dive a bit deep into the process just like a typical machine learning model implementation. The following is the process of implementing the VAR Forecasting method.

1. **Test for causality**: To see that if there is a correlation between the variables, **Grangers causality test** can be conducted, which shows an association between the variables. The table below shows that the p-value of variable_x causing variable_y, if p-value is smaller than 0.05, then variable_x causes variable_y. Hence, we can see that there are correlations between these five series.

ETFs/Index	Russeltop200$_x$	MSCIJapan$_x$	SP500IT$_x$	SP500HealthCare$_x$	SP500Industrial$_x$
Russeltop200$_y$	1.0000	0.2388	0.0247	0.0442	0.0104
MSCIJapan$_y$	0.0001	1.0000	0.1202	0.0013	0.0033
SP500IT$_y$	0.0000	0.0029	1.0000	0.0006	0.0007
SP500HealthCare$_y$	0.0000	0.0000	0.0001	1.0000	0.0000
SP500Industrial$_y$	0.0014	0.0047	0.1142	0.0065	1.0000

Table 4: Grangers Causality Test Result.

2. **Make strictly stationary - taking first difference**: Although in the previous section the stationary test (ADF test) has been done, we still apply first differencing on training set make all five series stationary.

3. **Modelling and Forecasting**: By fitting VAR model, it automatically figures out the best number of lags to be used. After setting the forecasting interval, five time series will be computed simultaneously.

4. **Inverting**: Since the first difference has been taken to fit the model, the forecasting output needs to be transformed into the original one.

Notice that although VAR is the most commonly used multivariate time series forecasting model, VAR model has some drawbacks when forecasting the financial time series. VAR model only captures the linear relationship, and it didn't consider the conditional heteroskedasticity or seasonality (in simple VAR model). The following section will introduce the ESRNN method, which is a hybrid model combining both statistical methods and deep learning.

ESRNN

As shown in figure 3, ESRNN represents a hybrid method of exponential smoothing and recurrent neural networks [53]. This method applies both Holt-Winters' multiplicative method and LSTM. Data are being accessed with hierarchical manner, the exponential smoothing is responsible for capturing the main components of the individual series, for instance, the level and seasonality part. On the other hand, LSTM focuses on both capturing the non-linearity portion of the time series data and cross-learning.

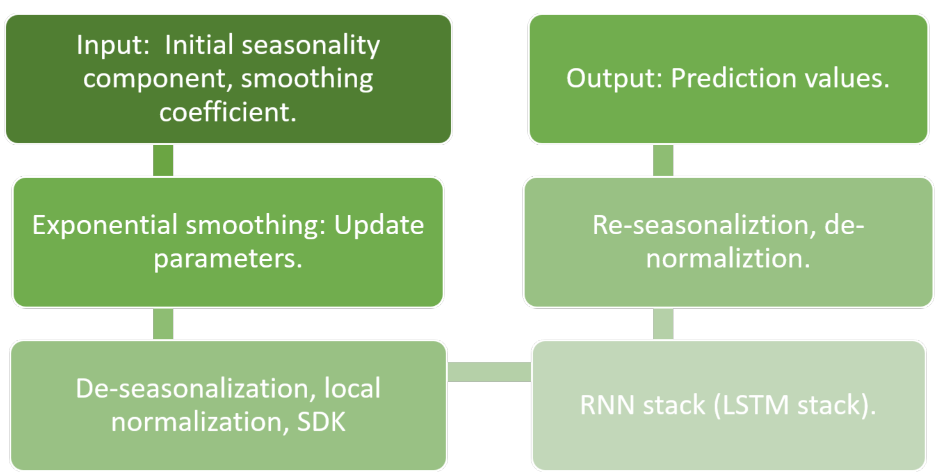

Figure 3: The ESRNN process map (Note that the RNN and the classical Holts-Winters parameters are co-trained)

Since the input contains numerous time series, and all of them may contain various seasonality, the pre-processing layers will conduct deseasonalization and adaptive normalization,which apply exponential smoothing methods with stochastic gradient descent(SGD) [57]. Owing that the model doesn't consider local linear trend, the b_t in equation 2 will then be replace by the LSTM component and be rewrite as the following:

$$\begin{aligned}\hat{y}_{t+h|t} &= (l_t + h \times LSTM(X_t))s_{t+h-m(k+1)} \\ x_i &= \frac{y_i}{l_t s_i}\end{aligned} \tag{11}$$

where X_t is a vector of normalized, de-seasonalized features that contains scalar component x_t in equation 11. At each time step, the values in the input and output windows will be normalized by dividing them by the last value of the level in the input window. Furthermore, if exists seasonality, the data will need to be also divided by the relevant event seasonality component [53]. In this model, the daily data is used, which implies the seasonality is set as 7. Eventually, the $log()$ function is applied in order to mitigate the influence brought by the outliers in the dataset.

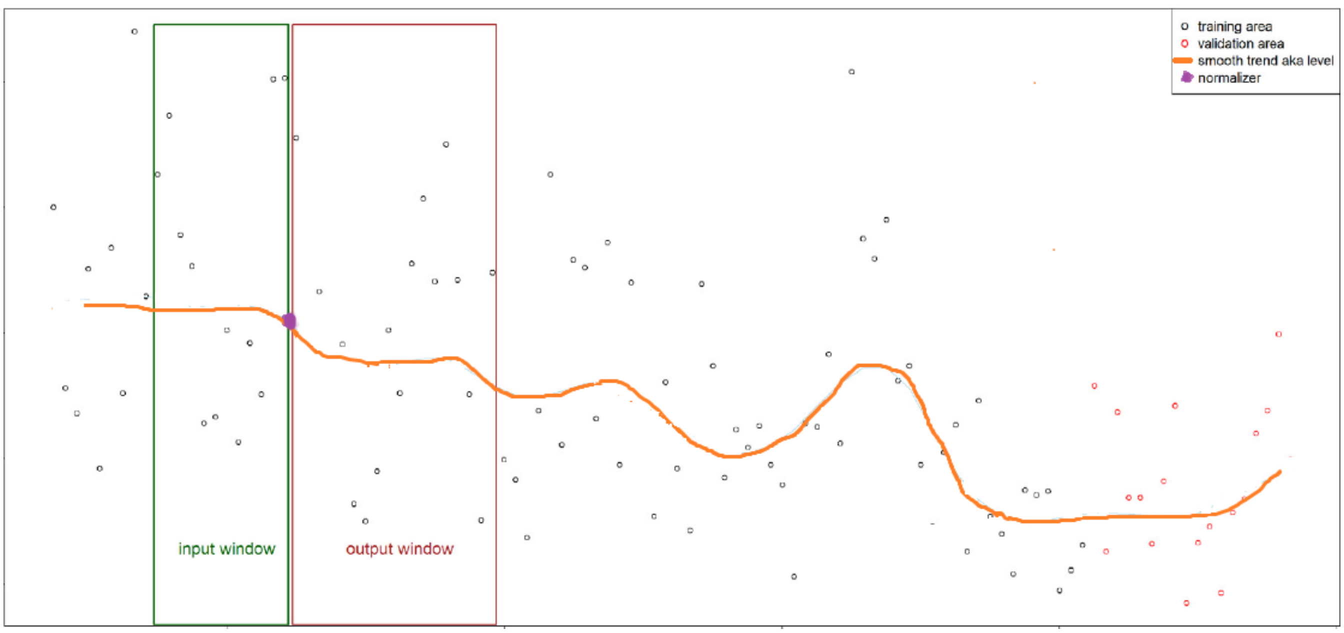

Figure 4: Pre-processing a random daily time series with rolling window.

In the deep learning layer, the dilated LSTM stacks are applied. The general advantage of LSTM algorithm is that it possesses high level of computational efficiency and allows the network to remember information from earlier time instances. The LSTM cell is constructed by 4 gates [54] as shown in Figure 5 :

- **Forget Gate**: Applies the sigmoid function(logistic function) and output a vector which contains values between 0 and 1.

$$f_t = \sigma(W_f \cdot [h_t, x_t] + b_f) \tag{12}$$

$$\sigma(X) = \frac{1}{1 + e^{-X}} \tag{13}$$

- **Input Gate and Input Candidate Gate**: Two gates will work together in order to render a new cell state C_t, which will be passed into the next time step. The input gate uses the sigmoid function as activation function, whereas the input candidate gate applies the hyperbolic tangent function, which will yield outputs between value -1 and 1. Two gates will generate i_t and $\tilde{C}_t$. In this case, i_t is responsible to select the features in $\tilde{C}_t$ that will be in new state of C_t in the later phase.

$$i_t = \sigma(W_t \cdot [h_{t-1}, x_t] + b_i) \tag{14}$$

$$\tilde{C}_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{15}$$

$$tanh(X) = \frac{e^X - e^{-X}}{e^X + e^{-X}} \tag{16}$$

- **Output Gate**: This gate decides which features to be selected, and generate output h_t in the end by combining o_t and tanh-applied C_t.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{17}$$

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \tag{18}$$

$$h_t = o_t \times tanh(C_t) \tag{19}$$

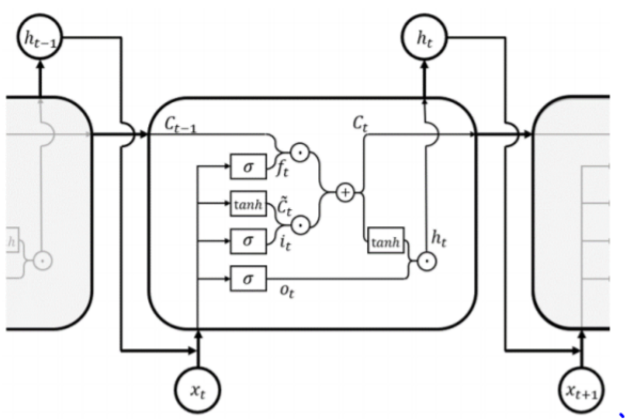

Figure 5: Traditional LSTM cell.

Different from the traditional LSTM, the dilated LSTMs utilize part of their input the hidden state from previous, but not always the latest steps [53]. In this case, the dilation is set to be 1,7,14. Traditionally, LSTM hidden weights are the inputs to the next cell in the layer, nonetheless, with the dilation number of 7 in the second layer, it means that the hidden weights and the bias weights are forwarded seven cells forward and so on and so forth.

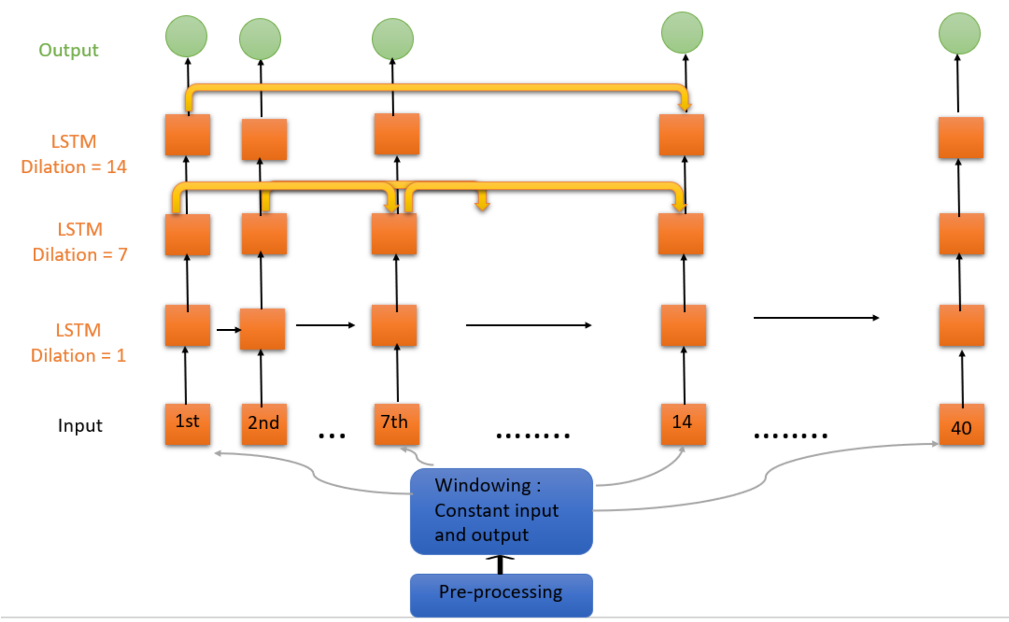

Figure 6: NN Architecture.

The choice of hyper-parameters is decided by the backtesting of the algorithm, the data is tested on the validation (removed) area after every epoch, which is the last horizon-number of choice in each time series. The following hyper-parameters are being considered:

- **Initial Learning Rate**: After observing the behaviors of the validation errors during each training epoch, the learning rate will reduce with a certain rate set at the beginning. This is also the size of the stochastic gradient descent steps.
- **Learning Multipliers**: This is the multiplier for per-series parameters smoothing and initial seasonality of the learning rate.
- **Learning decay**: This is the size of decay for the learning rate in a certain time period.
- **Level variability penalty**: This is the multiplier that only applies to the seasonal model. It controls the strength of the penalization to the wigglines of the level vector, which induces smoothness in the output.
- **Input Size**: This will decide how many data points will be used as input per training.
- **Step Size**: This controls the period for each learning rate decay.

ESRNN Hyperparameters Tuning Result		
Hyperparameters	**Choices**	**Best hyperparameter**
Initial Learning Rate	0.0007, 0.0003, 0.0001	0.0007
Learning Multiplier	0.2,0.3,0.5	0.2
Learning Decay	0.2,0.5	0.5
Level Variability Penalty	50,100	100
Input Size	10,15,20,40	40
Step Size	12,15,20	12

Table 5: ESRNN Hyperparameters Tuning Result

The last step in this algorithm will be reseasonalization and denormalization, which is conducted because of the previous step of deseasonaliztion and normalization.

Portfolio Optimization

So far, statistical combined machine learning methods for asset risk premium forecasting have been discussed above. This QWIM project is going to introduce **Portfolio Optimization Analysis**, by incorporating portfolio optimization analysis, this project can have a portfolio perspective to compare different forecasting methods by looking at **Portfolio Metrics** within each forecasting methods.

In this section, several portfolio optimization methods that are commonly used in the industry and in academia will be introduced, including **Mean Variance, Minimum Variance, Maximum diversification, hierarchical risk parity, Black-Litterman, and CLA (robust version)**. List of methods' codes are implemented by **PyPortfolioOpt, MLFinLab** packages in Python. Different portfolio Metrics and how such comparison of portfolio metrics can be used within context of a QWIM project based on forecasting will be discussed later in the next section.

Mean Variance: quadratic risk utility

Mean Variance Analysis, or called **Modern Portfolio Theory**, is assembling a portfolio of assets that maximize expected return for a given level of risk. Economist **Harry Markowitz** introduced MPT in 1952, and this theory assumes that all investors are risk averse. Mean Variance theory compares the expected return of a portfolio with the variance within this portfolio, so efficient frontier is commonly used in mean variance analysis.

In this mean variance analysis, a objective function called **quadratic risk utility** will be used. The objective function consists of both the portfolio return and risk, which is going to minimize the portfolio risk correspondingly maximize the return.

$$\begin{aligned} &\min_{w} \lambda w^T \Sigma w - \mu^T w \\ &s.t \sum_{i=1}^{n} w_i = 1 \\ &w_i \geq 0, i = 1, ..., N \end{aligned} \tag{20}$$

where

- **w**: Represents the weight of the portfolio assets.
- μ: Represents the expected return for assets.
- Σ: Refers the covariance matrix for assets.
- λ: Refers the risk-aversion parameter, setting by default value

Max Diversification

Maximum diversification portfolio tries to diversify the holdings across as many as possible. Diversification within a portfolio can be quantified using the diversification ratio as defined below, which measures how much the volatility is reduced relative to a scenario where all assets are perfectly correlated.[13] By maximizing the diversification ratio, a most diversified portfolio can be constructed for a given scenario. The diversification ratio D of a portfolio is defined as:

$$D = \frac{w^T \sigma}{\sqrt{w^T \Sigma w}} \tag{21}$$

where σ is the vector of volatilities and Σ is the covariance matrix. The denominator represents the volatility of the portfolio and numerator represents the weighted average volatility of the assets. More diversification within the portfolio leads to a higher diversification ratio. The mathematical expression can be shown below[13]:

$$\begin{aligned} &\max_{w} D \\ &s.t \sum_{i=1}^{n} w_i = 1 \\ &w_i \geq 0, i = 1, ..., N \end{aligned} \tag{22}$$

where

- **w**: Represents the weight of the portfolio assets.
- **D**: Represents the diversification ratio

Minimum Variance

Just like the name, minimum variance portfolio is a collection of securities that combine to minimize the portfolio volatility, where volatility is a risk measure that describe how the asset price goes up and down. The objective is to generate a portfolio with the least variance, as can be described below:

$$\begin{aligned} &\max_{w} w^T \Sigma w \\ &s.t \sum_{i=1}^{n} w_i = 1 \\ &w_i \geq 0, i = 1, ..., N \end{aligned} \tag{23}$$

where

- **w**: Represents the weight of the portfolio assets.
- Σ: Refers the covariance matrix for assets.

Hierarchical Risk Parity (HRP)

Hierarchical Risk Parity implements the hierarchical clustering models in allocation shown below.

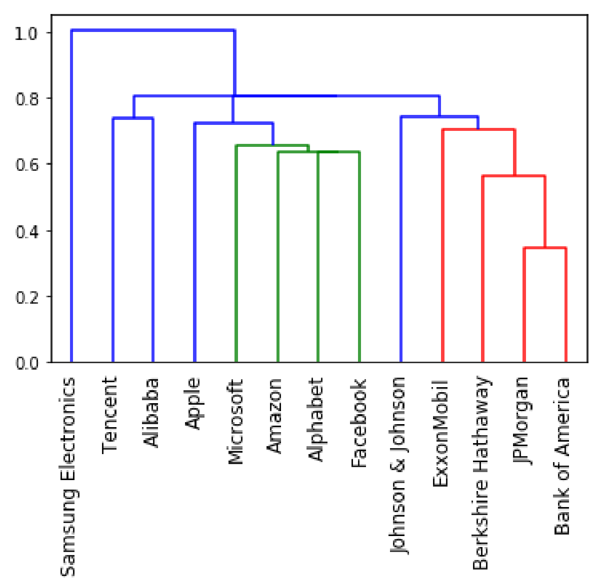

Figure 7: Hierarchical Clustering Allocation

Hierarchical Risk Parity(HRP) works by implementing the following steps:

1. Calculate the **distance matrix** based on the correlation of a set of assets.

2. Cluster the assets into a tree-like hierarchical clustering using the distance matrix calculated in step 1.

3. Form portfolios with the **minimum variance** for every two assets in tree branches

4. Repeat step 3 for all levels and combine the mini-portfolios at each node

Black-Litterman

The Black-Litterman(BL) model takes a Bayesian approach to asset allocation. The Black-Litterman formula is given below:

$$E(R) = [(\tau\Sigma)^{-1} + P^T\Omega^{-1}P]^{-1}[(\tau\Sigma)^{-1}\Pi + P^T\Sigma^{-1}Q] \tag{24}$$

- $E(R)$ is a Nx1 vector of **expected returns**, where N is the number of assets.
- Q is a Kx1 vector of **views**.
- P is the KxN **picking matrix** which maps views corresponding to the assets.
- Ω is the KxK **uncertainty matrix** of views.
- Π is the Nx1 vector of **prior expected returns**.
- Σ is the NxN **covariance matrix** of asset returns.
- τ is a scalar tuning constant.

The optimized weights are given as:

$$w = (\delta\Sigma)^{-1}E(R) \tag{25}$$

While building the Black-Litterman optimizer, the annualized average return for each portfolio is utilized as an `absolute_views` input for the optimizer.

Critical Line Algorithm (CLA)

The Critical line algorithm is to optimize general quadratic functions subject to linear inequality constraints. In other words, each weight of an asset in the portfolio can have different lower and upper bounds[44]. The optimization objective will be the same as in the classical mean-variance optimization, and the entire form will be shown as follow:

$$\begin{aligned} &\min_{w} w^T \Sigma w \\ &s.t \sum_{i=1}^{n} w_i = 1 \\ &w_i \leq u_i \\ &w_i \geq l_i \end{aligned} \tag{26}$$

where

- **w**: Represents the weight of the portfolio assets.
- Σ: Refers the covariance matrix for assets.
- u_i: Implies the upper bound of asset i.
- l_i: Indicates the lower bound of asset i.

The weight that are bounded are named bounded weights, and others that are not bounded are refer as free weights. By definition, a turning point is defined as a constrained minimum variance portfolio that its vicinity other constrained minimum variance portfolios contain different free assets[1]. The solution w^* will generate a set of weights that satisfies that condition of turning point. In this case, two types of optimization functions are used, which are maximum sharpe ratio and minimum variance.

V Analysis and Result

Above all, different forecasting algorithms and portfolio optimization methodologies have been discussed in the previous section. In this section, different **Metrics** will be used to evaluate the forecasting performance of each model, including **Forecasting Metrics** for standalone forecasting model performance and **Portfolio Metrics** for forecasting performance within the context of optimal portfolios. Hence, this project is going to have an **overall analysis** and evaluation to determine the "best forecasting methods" among these forecasting models.

Prediction Interval

Consider the statistical concept confidence interval, prediction interval is just a type of it used with predictions in regression analysis; it is a range of values that the model predicts with a percentage of confidence. For example, a 95% prediction interval means that there is a 95% chance future values will fall into that range. Below is the mathematical formula of 95% prediction interval for h-step forecast distribution.

$$\hat{y}_{T+h} \pm 1.96\hat{\sigma}_h \tag{27}$$

As an example, the theta forecaster 95% prediction interval will be presented below:

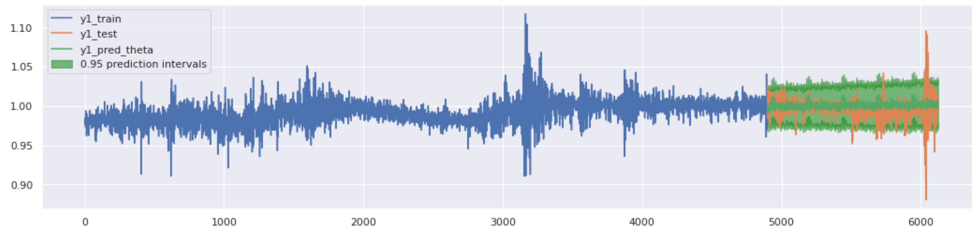

Figure 8: 95% Theta Forecasting interval for Russell Top200 ETF.

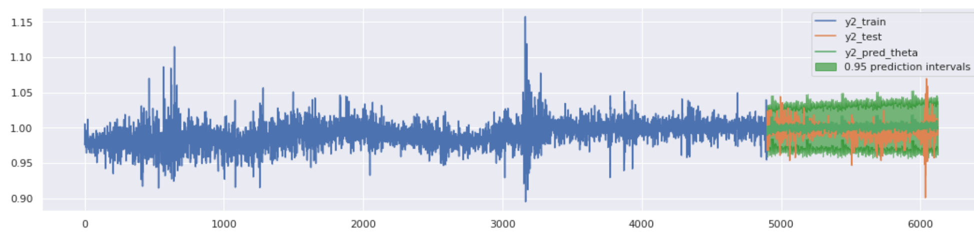

Figure 9: 95% Theta Forecasting interval for MSCI JAPAN Index.

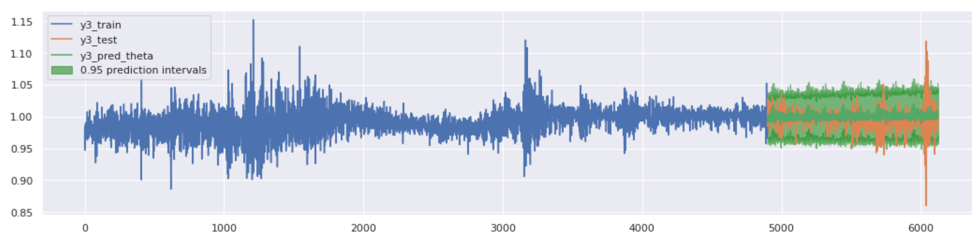

Figure 10: 95% Theta Forecasting interval for S&P500 IT Index.

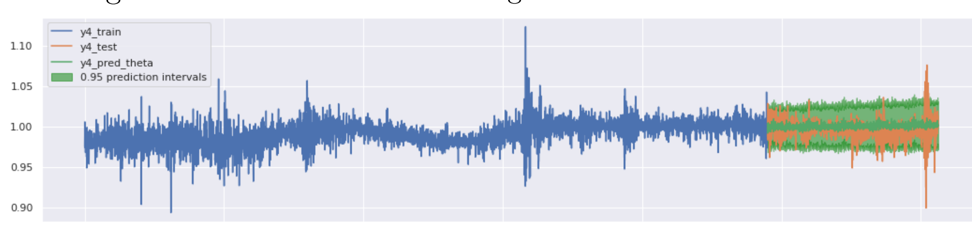

Figure 11: 95% Theta Forecasting interval for S&P500 HealthCare Index.

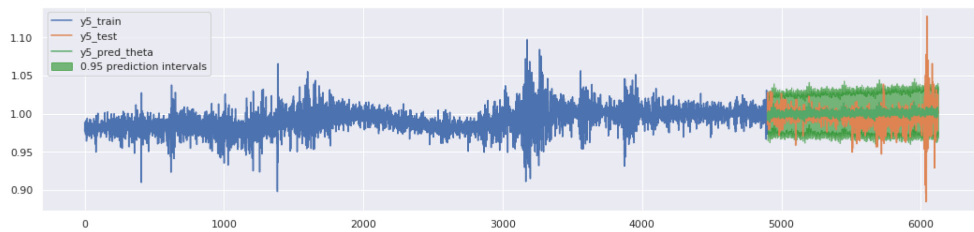

Figure 12: 95% Theta Forecasting interval for S&P500 Industrial Index.

Forecasting Values Comparison

Note that for every graph, two different colors are used, **Red Line: Actual Value; Blue Line: Predicted Value**. Every figure can be split into 5 sub-graphs, which represent different ETF and indexes - iShare Ruussel Top 200 ETF; MSCI Japan Index; S&P500 IT index; S&P500 Healthcare index and S&P500 Industrial index.

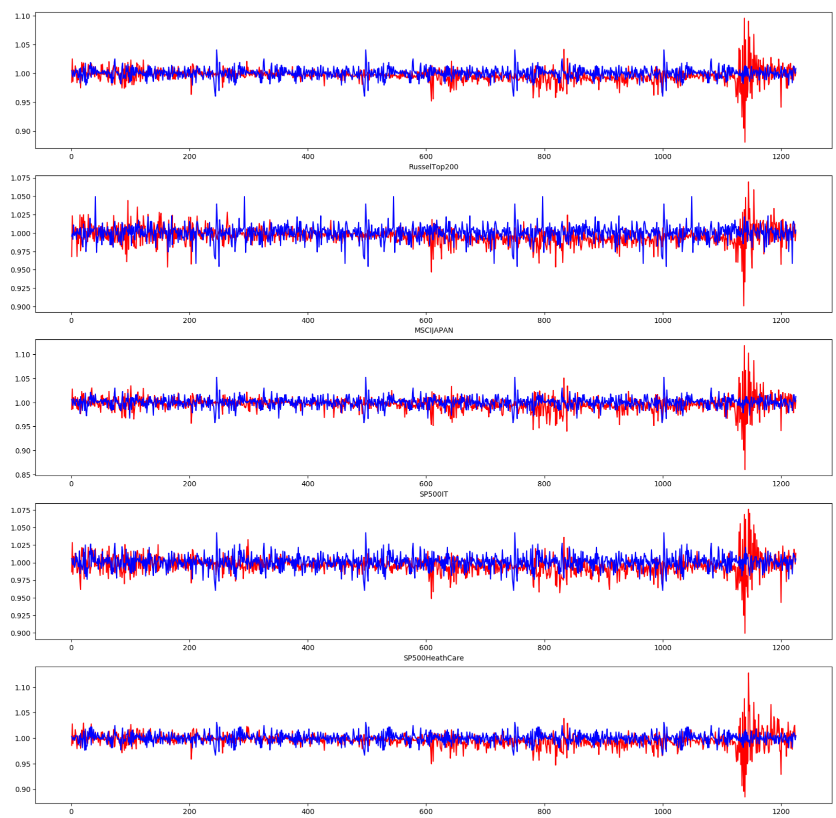

Figure 13: Actual returns and returns generated by naive forecaster.

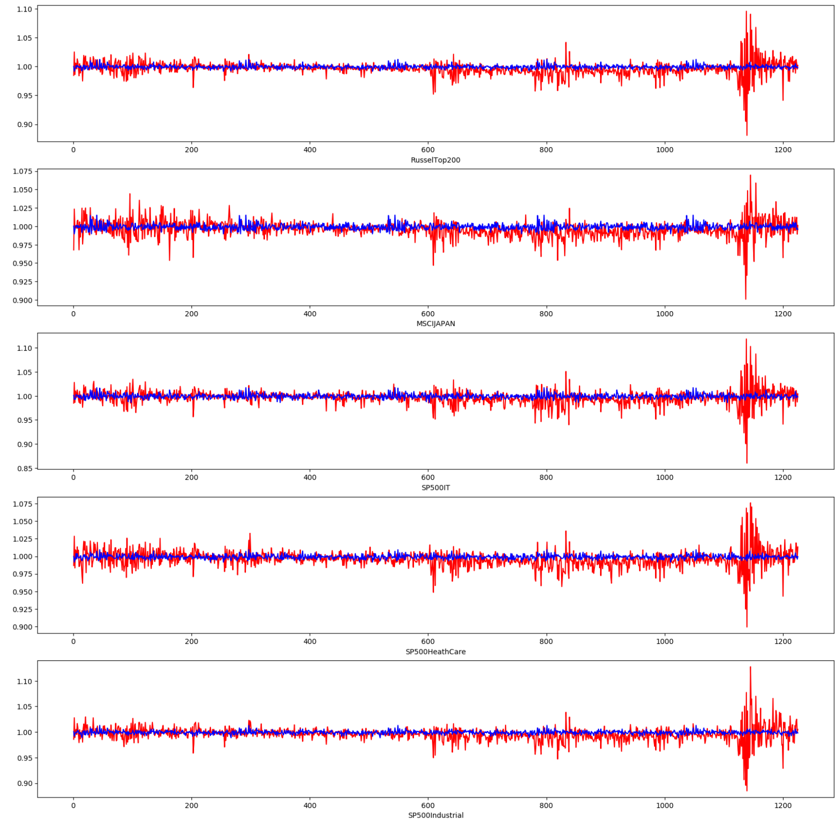

Figure 14: Actual returns and returns generated by exponential smoothing forecaster.

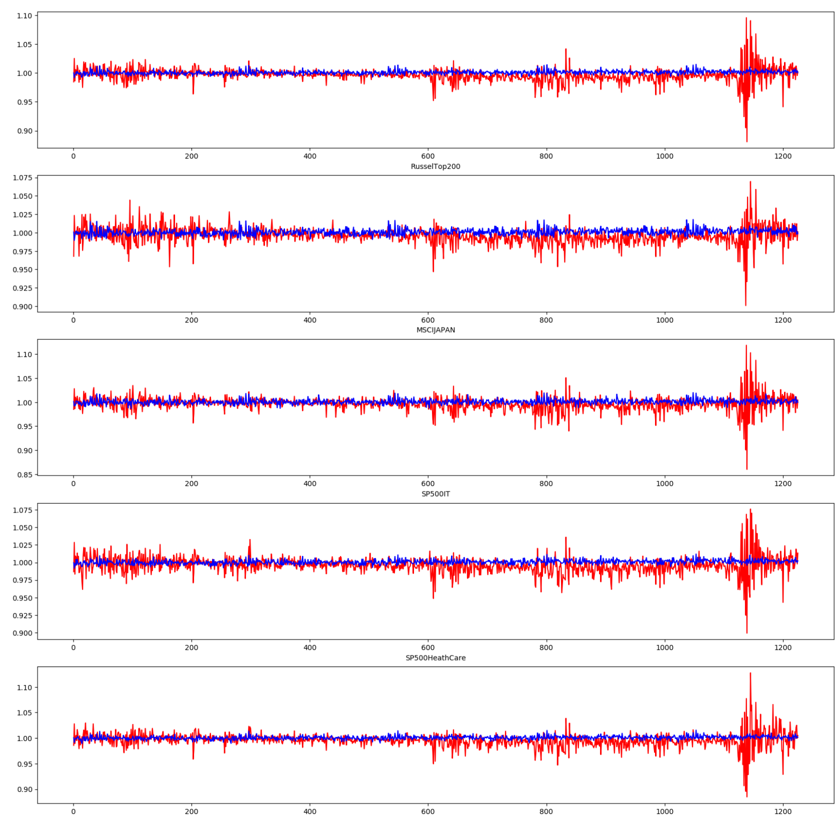

Figure 15: Actual returns and returns generated by theta forecaster.

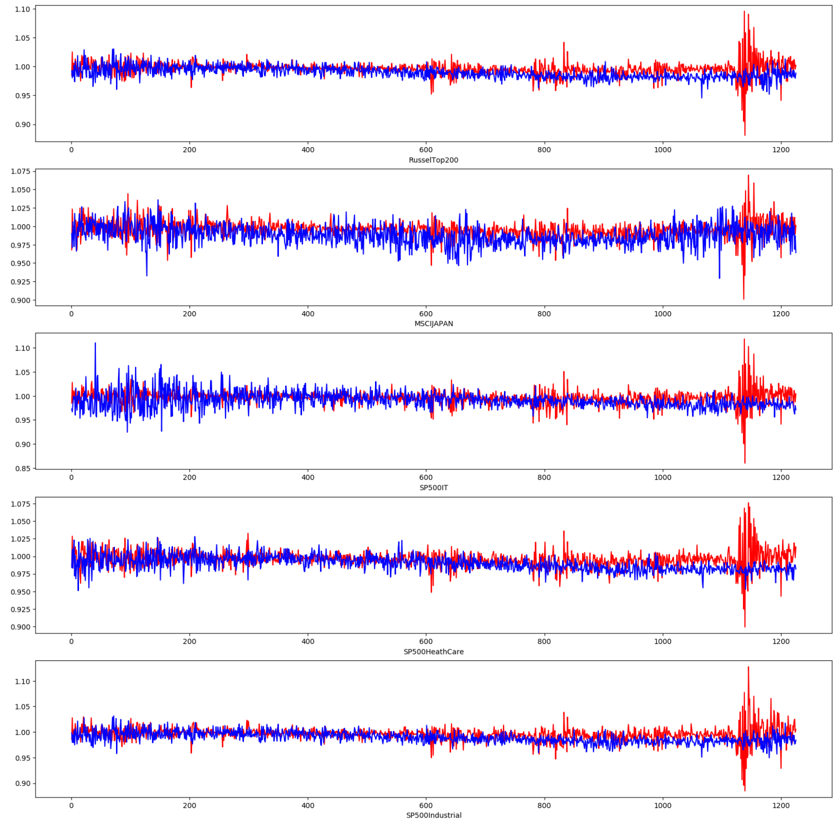

Figure 16: Actual returns and returns generated by KNN forecaster.

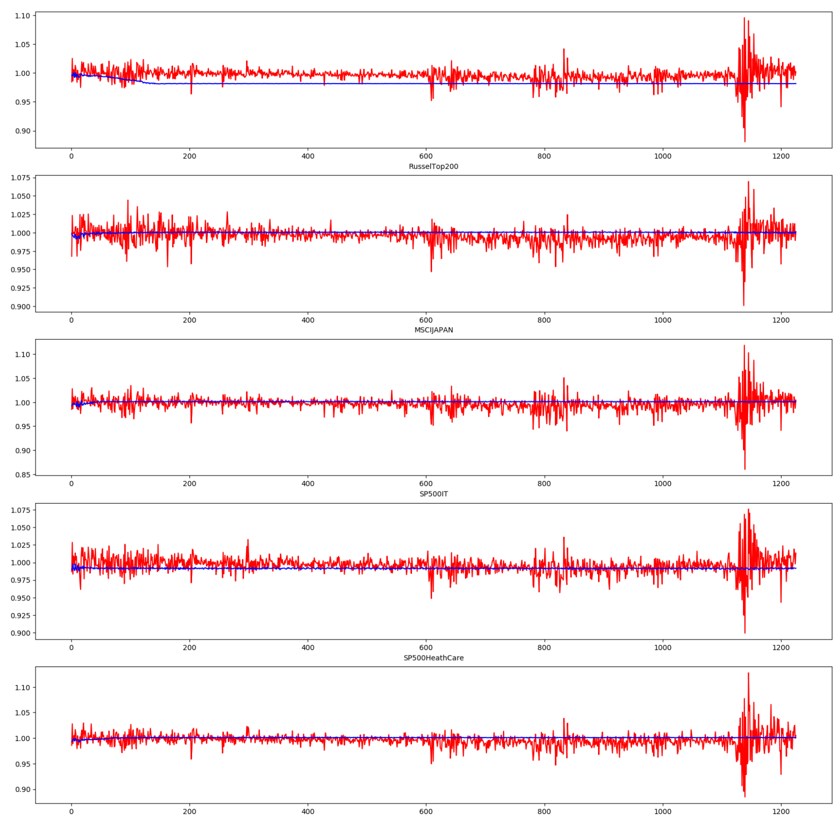

Figure 17: Actual returns and returns generated by random forest forecaster

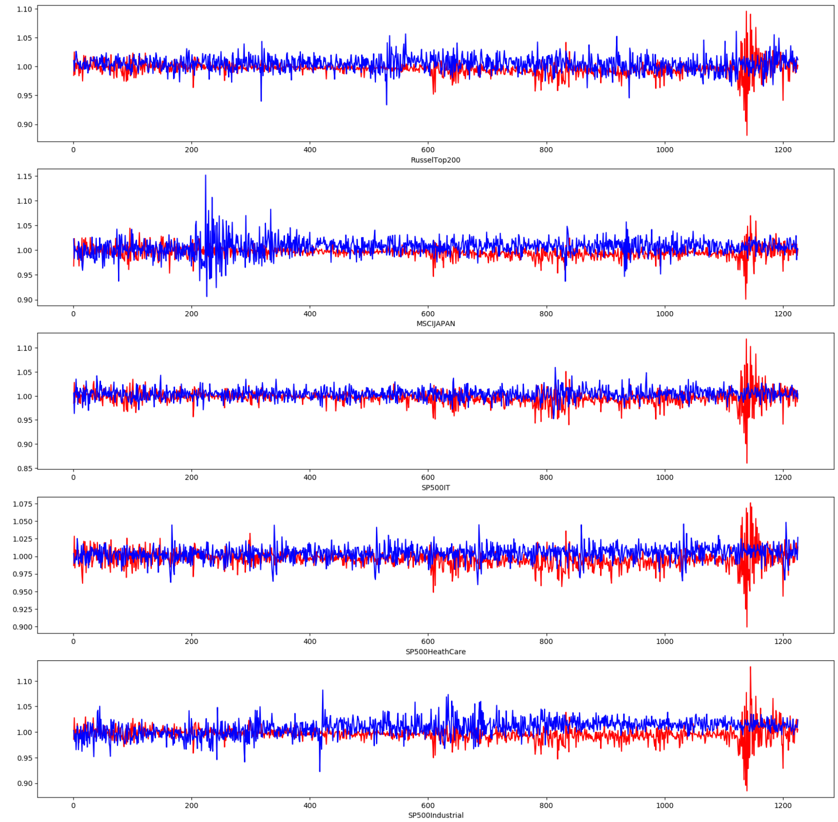

Figure 18: Actual returns and returns generated by pipeline forecaster.

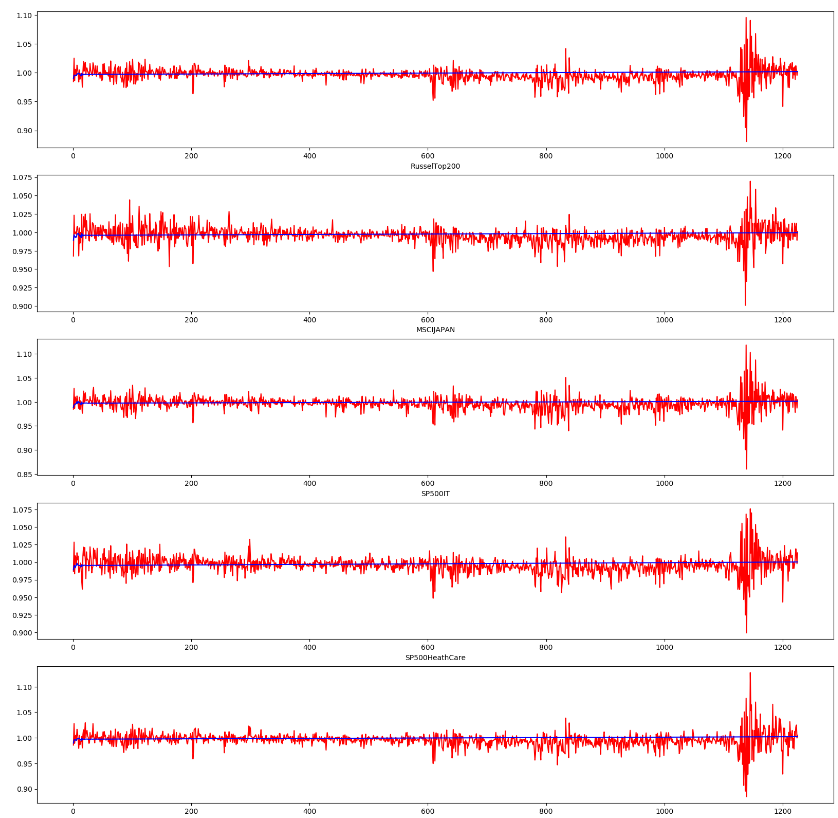

Figure 19: Actual returns and returns generated by VAR forecaster.

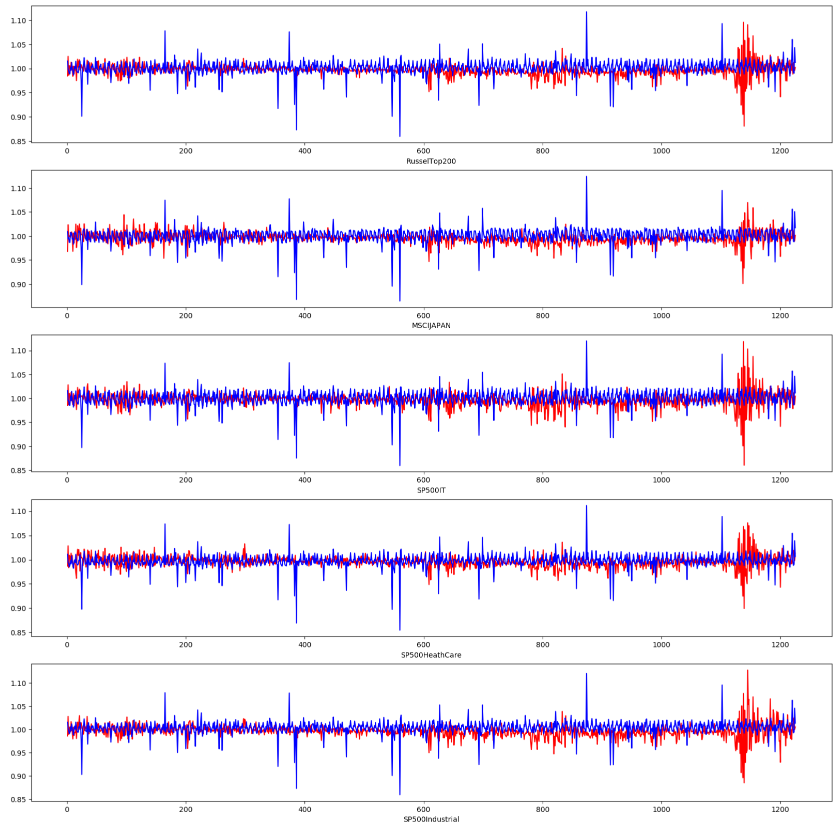

Figure 20: Actual returns and returns generated by ESRNN forecaster.

Figure 13 to figure 20 show the comparison of actual values and the predict values for each data set. For univariate forecaster, naive forecaster, KNN forecaster and pipeline forecaster seem to capture the fluctuation of the return better than the others. Moreover, random forest forecaster's predict return seems to be a flat line, which underperforms all the other univariate methods in detecting the movement or volatility of the financial market. Note that as discussed above in the **Methodology** section, naive forecaster sometimes performs well in the financial market. Exponential Smoothing and Theta Method perform well in M4 Competition, it seems that ES and Theta Method also can generate relatively good results in forecasted values for stationary time series.

For multivariate forecasting methods, ESRNN has a better ability to capture the fluctuation of the data, whereas VAR has similar performance to random forest forecaster which yields a relatively "flat" result. As discussed in the previous section, VAR used the linear relationship to get the forecasted result, that can be a reason why VAR model can't get the volatility of the series. However, the forecasting metrics(in the next section) show a good result for VAR model because both actual values and predicted values are close to 0. Although other models capture the movement and volatility for the series, the difference between actual values and predicted values can be large.

Forecasting Metrics Comparison

In this section, four forecasting matrics will be used in order to compare different forecasting models, as talked in Methodology section, they are **sMAPE, MAE, MASE, RMSE** respectively:

1. **sMAPE**: Refers to Symmetric mean absolute percentage error, which is an accuracy measure based on percentage error. It is a most commonly used metric in forecasting problem, also used in M4 Competition.

$$sMAPE = \frac{100\%}{n} \sum_{i=1}^{n} \frac{\mid y_i - \hat{y}_i \mid}{\mid y_i \mid + \mid \hat{y}_i \mid} \tag{28}$$

2. **MAE**: Refers to Mean Absolute Error, which is a measure of errors between paired observations expressing the same pheonomenon. It is one of a number of ways of comparing forecasts with their eventual outcomes.

$$MAE = \frac{\sum_{i=1}^{n} \mid y_i - \hat{y}_i \mid}{n} \tag{29}$$

3. **MASE** (For seasonal time series): Refers to Mean Absolute Scaled Error, it is a well-established alternatives of MAE. MASE is a measure of the accuracy of forecast, and has favorable properties when compared to other methods for calculating forecast errors, for example, scale invariance and seasonality.

$$MASE = \frac{\mid y_i - \hat{y}_i \mid}{\frac{1}{n-M} \times \sum_{i=M+1}^{n} y_i - y_{i-M}} \tag{30}$$

4. **RMSE**: Refers to Root Mean Square error, which is a frequently used measure of the differences between values. Note that RMSE is always non negative, and is sensitive to outliers.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{31}$$

where

- y_i: The actual value.
- $\hat{y}_i$: The forecast value.
- M: The number of seasonality.
- n: The end of the time period.

The following tables shows the forecasting metrics results of each financial series:

Models	sMAPE	MAE	MASE	RMSE
Naive	1.130078	0.011270	0.891957	0.015825
Exponential Smoothing	0.855517	0.008521	0.674396	0.013210
Theta	0.923761	0.009205	0.728513	0.013815
KNN	1.218096	0.012088	0.956729	0.016684
RRF	1.508392	0.014955	1.183644	0.018011
TTF	1.483697	0.014844	1.174846	0.019939
VAR	0.840514	0.008370	0.662430	0.013219
ESRNN	1.343055	0.013382	1.059130	0.019908

Table 6: Russeltop200 Model Performances.

Models	sMAPE	MAE	MASE	RMSE
Naive	1.239396	0.012357	0.750911	0.016358
Exponential Smoothing	0.912218	0.009087	0.552182	0.012587
Theta	0.988624	0.009853	0.598736	0.013326
KNN	1.483229	0.014693	0.892851	0.018980
RRF	0.877370	0.008736	0.530886	0.012241
TTF	1.883624	0.018878	1.147143	0.024299
VAR	0.832790	0.008291	0.503801	0.011883
ESRNN	1.399355	0.013937	0.846896	0.019707

Table 7: MSCIJapan Model Performances.

Models	sMAPE	MAE	MASE	RMSE
Naive	1.380751	0.013768	0.752274	0.019105
Exponential Smoothing	1.077967	0.010738	0.586691	0.016163
Theta	1.125376	0.011214	0.612750	0.016657
KNN	1.735453	0.017248	0.942441	0.023511
RRF	1.057052	0.010527	0.575195	0.016130
TTF	1.501963	0.015009	0.820100	0.020417
VAR	1.013469	0.010091	0.551366	0.015778
ESRNN	1.390818	0.013825	0.755415	0.021045

Table 8: SP500IT Model Performances.

Models	sMAPE	MAE	MASE	RMSE
Naive	1.234934	0.012324	1.001679	0.016780
Exponential Smoothing	0.900484	0.008971	0.729153	0.013059
Theta	0.965683	0.009624	0.782239	0.013645
KNN	1.242334	0.012338	1.002834	0.016663
RRF	1.505860	0.014935	1.213982	0.017932
TTF	1.420379	0.014196	1.153914	0.018549
VAR	0.868904	0.008654	0.703408	0.012787
ESRNN	1.280031	0.012729	1.034611	0.019216

Table 9: SP500HealthCare Model Performances.

Models	sMAPE	MAE	MASE	RMSE
Naive	1.250476	0.012466	0.903853	0.017474
Exponential Smoothing	0.966088	0.009618	0.697366	0.014963
Theta	1.031074	0.010269	0.744588	0.015554
KNN	1.321659	0.013119	0.951234	0.018124
RRF	0.893495	0.008891	0.644702	0.014338
TTF	2.001530	0.020065	1.454903	0.025869
VAR	0.971147	0.009667	0.700965	0.015063
ESRNN	1.494085	0.014894	1.079934	0.021406

Table 10: SP500Industrial Model Performances.

Forecasting Models Metrics Analysis: From the tables for five different risk premium series above,it can be concluded that:

1. **iShares Russel Top 200 ETF**: It can be seen that **Vector Autoregression model, Exponential Smoothing and Theta Method** have the best prediction accuracy than other models. Notice that sMAPE of VAR model is only 0.84 and MASE is only 0.66, followed by Exponential Smoothing, Theta method, Naive method, KNN, ESRNN, Pipeline and random forest model. However top models such as VAR do not show strong volatility of the series, the forecasted values are quite steady.

2. **MSCI Japan Index**: Similarly, **Vector Autoregression model, Random Forest Regression, Exponential Smoothing** relatively outperform other models, followed by Theta Method, Naive Method, ESRNN, KNN and Pipeline model. All of the top three models yield sMAPE smaller than 1 and MASE smaller than 0.55, especially the MASE for VAR model is only 0.5038

3. **S&P500 IT Index**: Again, **Vector Autoregression model, Random Forest Regression, Exponential Smoothing** have relatively better results than other models, followed by Theta, Naive Method, ESRNN, KNN and Piprline model. Compared to MAE and RMSE, the difference in sMAPE and MASE of each model are relatively higher and obvious than MAE and RMSE. From Table 8, it can be seen that sMAPE and MASE of VAR model are only 1.01 and 0.55 respectively. It has the same model rank as in MSCI JAPAN Index

4. **S&P500 HealthCare Index**: Similarly, **Vector Autoregression model, Exponential Smoothing and Theta Method** relatively outperform than other models, followed by Naive, KNN, ESRNN, Pipeline and Random Forest models. It is no doubt that VAR model yields the lowest sMAPE, MAE, MASE and RMSE again, with values of 0.868904, 0.008654, 0.703408, 0.012787 respectively.

5. **S&P500 Industrial Index**: Finally, **Random Forest model, Exponential Smoothing and Vector Autoregression model** have the best prediction accuracy than other models, followed by Theta, Naive, KNN, ESRNN and Pipeline models respectively. This Random Forest model yields the lowest error score, where sMAPE and MASE are only 0.89 and 0.64.

To conclude, it can be seen that **Vector Autoregression and Exponential Smoothing** always yield better result among all the five ETF/Index risk premium series. The results of Exponential Smoothing and Theta Methods are pretty similar because they share similar mathematical methodology. Naive Forecaster, Machine learning methods such as KNN, Pipeline and ESRNN models yield relatively good result, and the rank of these methods are quite steady. Random Forest regression outperforms in some series, while it can the lowest rank in other series.

However, notice that all five risk premium series are quite stationary, and all the forecasted values are quite small, this can be a reason why Vector Autoregression and Exponential Smoothing can compute better results than other models, while KNN, Naive or ESRNN models capture a movement(volatility) of the risk premium series. Hence, this paper will combine the portfolio optimization perspective and use risk metric to compare the forecasting models.

Portfolio Optimization Comparison

Although four different forecasting metrics give an intuitive result for each model, another perspective that combining portfolio optimization should be considered. A comparison of forecasting performance by using portfolio metrics can be conducted, and these metrics will be computed within each optimal portfolio where optimal weighted are computed based on different forecasting models.

Note that the forecasted return is obtained from the forecasted values of the risk premium for each series, the returns are calculated by **return = risk premium + risk free rate**

The process of comparing forecasting models using portfolio optimization:

1. **Create Benchmark and Enhanced portfolios**: Setting one Benchmark portolio which used all historical data and eight enhanced portfolios, each enhanced portfolio represents a forecasting method. So in total, there are **1+8** portfolios to be constructed.
2. **Setting portfolio construction time**: In this project, assuming that **2020-01-02** is the portfolio construction date.
3. **Split three timelines for comparison**: In order to have a side-by-side comparisons of portfolio metric, three different time periods will be splitted:
   - **New Set**: From the date of portfolio construction (2020-01-03) to the last date of the dataset (2020-07-17).
   - **Whole Set**: From the starting date of dataset (1996-03-19) to last date of the dataset (2020-07-17)
   - **Hist Set**: From the starting date of dataset (1996-03-19) to the portfolio construction date (2020-01-02).

4. **Calculate Expected Return and Covariance**:
   - For Benchmark portfolio: Vector of expected means and expected covariance matrix are calculated from actual historical data(**Hist**) available at 2020-01-02.
   - For Enhanced portfolio: Vector of expected means and expected covariance matrix are calculated from forecasted values available at 2020-01-02 computed by each forecasting models trained on training set.
5. **Calculate optimal weights**: Using different portfolio optimization algorithm get optimal weights for all Benchmark portfolio and Enhanced portfolios. Note that **equal weight** can be considered as another "Benchmark" in metric analysis.
6. **Calculate portfolio metrics based on three timelines**: After getting different optimal weights from the previous step, portfolio metrics can be calculated across three timelines within each portfolios.

Following are the portfolio metrics that used in this article, assuming annual risk-free rate = 0.02:

1. **Information Ratio**: IR is a measurement of portfolio returns beyond the returns of a benchmark. It measures a portfolio manager's ability to generate excess returns relative to benchmark by incorporating a tracking error.
$$IR = \frac{Portfolio\, Return - Benchmark\, Return}{Tracking\, Error} \tag{32}$$
   - **Benchmark Return** = Return on fund used as benchmark, here it set to be risk-free rate.
   - **Tracking Error** = Standard deviation of difference between portfolio and benchmark return.
2. **Annual Return**: Refers to the return that portfolio provides over a period of time.
3. **Annual Volatility**: Refers to the standard deviation of the portfolio yearly returns, assuming that there are 252 trading days within one year, denoted as σ_{annual}
4. **Calmar Ratio**: It is a function of the fund's average compounded annual rate of return versus its maximum drawdown. The higher the calmar ratio, the better it performed on a risk adjusted basis during the given time frame.
$$CR = \frac{Average\, Annual\, Rate\, of\, Return}{Maximum\, Drawdown} \tag{33}$$
5. **Omega Ratio**: It is a risk-return performance measure of a portfolio, defined as the probability weighted ratio of gains versus losses for some threshold return target. Considered as an alternative to Sharpe Ratio.
$$\Omega(\theta) = \frac{\int_{\theta}^{\infty}[1 - F(r)]dr}{\int_{-\infty}\theta F(r)dr} \tag{34}$$

- **F**: cummulative probability distribution function of returns
- θ: target return threshold, set to 0

6. **Sharpe Ratio**: It is the average return earned in excess of the risk-free rate per unit of volatility. Sharpe ratio is widely used to help investors understand the return of an investment compared to its risk.

$$SR = \frac{R_p - R_f}{\sigma_p} \tag{35}$$

- R_p = return of portfolio
- R_f = risk free rate
- σ_p = standard deviation of portfolio

7. **Sortino Ratio**: It is a variation of the Sharpe Ratio that differentiates harmful volatility from total overall volatility by using the downside deviation. Hence everything else will be same as Sharpe Ratio, except for the denominator will change to the standard deviation of downside deviation.

8. **Tail Ratio**: It determines the ratio between the right (95%) and left tail (5%). For example, a tail ratio = 0.25 means that losses are four times as bad as profits, which can be simply considered as $\frac{Profit}{losses}$

9. **Mean Absolute Deviation(MAD)**: In a set of data, it determines the average distance of each value and the data mean.

$$MAD = \frac{1}{n}\sum_{i=1}^{n} \mid y_i - \bar{y} \mid \tag{36}$$

- y: Data value.
- $\bar{y}$: Mean of the data.

10. **Value at Risk**: This is a measurement that measures the potential loss in value of a risky asset or portfolio over a defined period for a given confidence interval. Here it is defines as 95% Value at Risk.

$$VaR_\alpha(X) = -inf_{t\in(0,T)}[X_t \in R : F_X(X_t) > \alpha] \tag{37}$$

For example, 95% VaR = 4% means that with 95% confidence that the worst daily loss will not exceed 4%

11. **Expected Shortfall(cVaR)**: This is an alternative measurement of VaR, which is more sensitive to the tail distribution. It is the expected loss if that worst-case threshold is ever crossed.

$$cVaR = \frac{1}{1-c}\int_{-1}^{VaR} xp(x)dx \tag{38}$$

- p(x)dx: The probability density of getting a return with value ”x”.

- x: The return value that is expected to get.
- c: The cut-off point on the distribution that is set to be the VaR breakpoint.
- VaR: The agreed-upon VaR level.

12. **Worst Scenario**: Refers to Worst Realization of a return series, also can be considered as the **maximum loss** during that period.

$$WR(X) = max(-X) \tag{39}$$

13. **Maximum Drawdown(MDD)**: This is an indicator of downside risk over a specified time period. It represent the maximum observed loss from a peak to a trough of a portfolio, before a new peak is attained. However, this doesn't measure the frequency of losses, which it doesn't provide the information about how long can the investors recover from the losses.

$$MDD = \frac{Trough\,Value\ -\ Peak\,Value}{Peak\,Value} \tag{40}$$

Portfolio Metrics Analysis within each Optimal Portfolio:

As discussed in **The process of comparing forecasting models using portfolio optimization**, three timelines of actual datasets have been split and used to calculate the portfolio metrics based on the optimal weights computed by each forecasting model within each portfolio optimization method. In this section, several Graphs and Metrics tables for **New Set: 2020-01-03 to 2020-07-17** will be presented and considered as an out-of-sample metrics evaluation, other tables will be presented in **Appendix**.

In addition, portfolio graphs of **return + 1** in the portfolio construction will be presented, followed by the **metrics results** within each portfolio optimization methods.

To begin with a detailed analysis with **Hierarchical Risk Parity**, Figure 27 in **Appendix** shows the results from the **Hierarchical Risk Parity** Optimizer. The predicted *returns* are scaled to $returns + 1$ for a clearer view. The **Pipeline portfolio** outperforms any other portfolios with the highest return. The worst performing portfolio here is **Random Forest**.

Metrics	Equal Weights	Baseline	Naive	ES	Theta	KNN	Random Forest	Pipeline	VAR	ESRNN
Information Ratio	0.011277	0.011876	0.013639	0.012871	0.013141	0.009425	0.003924	0.020164	0.014977	0.014093
Annual Return	0.008691	0.016564	0.026495	0.022394	0.024055	0.000484	-0.041105	0.068420	0.035753	0.029821
Annual Volatility	0.412686	0.395032	0.402193	0.397655	0.398091	0.397257	0.419938	0.420229	0.399472	0.400331
calmar ratio	0.027285	0.054369	0.086455	0.073524	0.079019	0.001560	-0.121520	0.221363	0.118653	0.098005
omega ratio	1.035433	1.037339	1.043089	1.040591	1.041469	1.029491	1.012004	1.064491	1.047406	1.044566
sharpe ratio	0.179024	0.188528	0.216514	0.204319	0.208611	0.149617	0.062295	0.320101	0.237748	0.223717
sortino ratio	0.313227	0.330367	0.367453	0.351518	0.357416	0.275274	0.151786	0.510908	0.397334	0.377979
tail ratio	0.946923	1.019028	0.977062	0.995214	0.993986	0.985971	0.950123	1.032420	0.977039	0.982975
Mean Absolute Deviation	0.016803	0.016099	0.016343	0.016175	0.016189	0.016207	0.017390	0.017015	0.016227	0.016266
Value at Risk	0.043948	0.039930	0.040789	0.040092	0.040089	0.041198	0.043124	0.042619	0.040696	0.040228
Expected Shortfall	0.066497	0.063351	0.064635	0.063882	0.063938	0.064095	0.066593	0.066434	0.064352	0.064278
Worst Scenario	0.107596	0.102916	0.105309	0.103846	0.104032	0.102907	0.106326	0.112475	0.104706	0.104839
maximum drawdown	0.357005	0.339084	0.340702	0.338609	0.338319	0.346759	0.386285	0.341654	0.333664	0.337812

Table 11: Hierarchical Risk Parity: Metrics Summary(2020/01/03 - 2020/07/17).

Table 11 shows the detailed metrics for HRP optimizer during 2020-01-03 to 2020-07-17. From the table, there are no big differences in Information Ratio(IR) for all models, but Pipeline model slightly outperforms other models with the highest IR of 0.02. This distinction is also reflected in Annual Return, as the Pipeline model has a maximum Annual Return of 6.842%, while all the other models have Annual Returns $< 4\%$. The Annual Volatility floats within $0.39 \sim 0.42$.

When considering calmar ratio, Pipeline outplayed other models again, which is not due to the difference in maximum drawdown, but mainly because the Pipeline model has the highest Annual Return. By looking at the maximum drawdown, there is no such predominance among all models, where most models have max drawdown in 33% to 36%. Here Random Forest has a maximum drawdown of 38.6%, making it again the worst-performing models with a negative annual return that worsened the numbers, also bringing the calmar ratio to negative.

Omega ratio is even, meaning that the probability of earning money to losing money is the same for all models. For Sharpe ratio and sortino ratio, the result is easily predicted and understood, as these two ratios are variations of Information Ratio, given that the risk free rate is fixed to 0.02 here.

The next few metrics are some different measurements for the risk. For the tail ratio, only two models have tail ratio > 1: the Baseline model and the Pipeline model. This shows that profits are generally smaller than losses in most models, and this is common considering the current financial situation. Even Pipeline model and Baseline model managed to gain a tail ratio > 1, they only obtained tail ratio slightly greater than 1. All models did not differ much in Mean Absolute Deviation, Value at Risk(VaR), Expected Shortfall(ES) and worst scenario. The average VaR and ES are around 4% and 6.5%, meaning that the maximum loss for the portfolio is 4% times the principal amount at 95% confidence.

Overall by looking at the metrics summary table for the HRP optimizer, Pipeline model seems to have the highest performance, which is already indicated in Figure 27(see Appendix).

Although the Pipeline model is the best and Random Forest seems to be the worst in measuring portfolio returns for **New Set**. It is not always the case when looking at **Whole Set**(see Table 21 in Appendix). For the portfolio constructed on 2020/1/2, surprisingly, **both** Pipeline model and Random Forest are the two best models over this period. These two models both have Information Ratio above 2.5%, where all other models have IR around $2.2\% \sim 2.3\%$. The two models also have the highest Annual Returns: Pipeline model has an Annual Return of 8.567% and Random Forest's is 7.946%. The Annual Return for all other models do not exceed 7.5%.

You may see that the numbers for portfolio over the whole period are generally larger than the portfolio over the prediction period, given the IR and Annual Return mentioned above. And this is different with Value at Risk, Expected Shortfall. The Value at Risk for all models is averaged at 1.8%, comparing to the 4% average VaR for prediction period; ES for all models in **Whole Set** are all below 2.9%, comparing to 6.5% ES for **New Set**. Given a higher return and lower risk, does it mean that the portfolio over the whole period is better than the portfolio over the prediction period? Probably. When looking at the maximum drawdown, the max drawdown for **Whole Set** is almost two times the max drawdown for **New Set**. All these differences could be explained by market movement, inaccurate prediction, but one possible explanation here could be, due to the long period **Whole Set** has existed, it experienced more extremes than **New Set** does, and given the overall market shows a growing tendency, the annual return is also higher as the **Whole Set** was established at a low starting price.

Another explanation would be due to the several circuit breakers happening in March this year, the **New Set** may not perform as well as before. This question could be resolved by solely looking at a portfolio built on historical prices before 2020: the **Hist Set**(1996/3 - 2020/1), and see if **Hist Set** outperforms both **Whole Set** and **New Set**. Please refer to Appendix: Table 28 for a detailed metrics table. From the table, **Hist Set** has metrics very close to that of **Whole Set**, but the Annual Returns are slightly higher for all models. This shows some validity of the explanation mentioned at the beginning of this paragraph. Further research could be done to investigate the impact of financial crisis happenings; even though this is not this paper's focuses on financial time series prediction and optimization, readers of interests are encouraged to conduct their own researches.

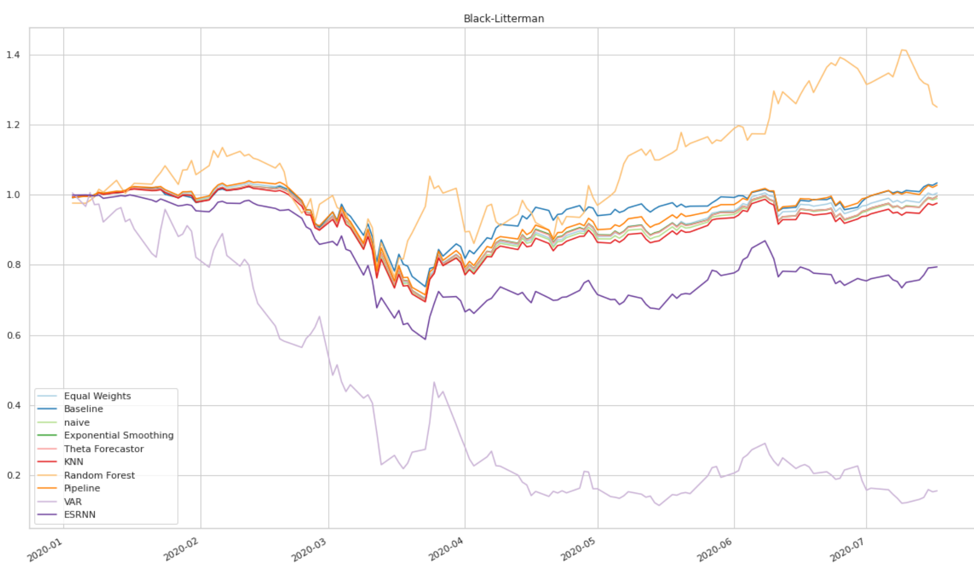

Figure 21: Returns from Black-Litterman Optimizer.

Figure 21 displayed a line chart for the returns from **Black-Litterman optimizer**. **Random Forest model** outplayed all other models for the most time and the VaR model is the worst performing model with returns far below other models. The gap between model performances is huge compared to other optimization methods.

Metrics	Equal Weights	Baseline	Naive	ES	Theta	KNN	Random Forest	Pipeline	VAR	ESRNN
Information Ratio	0.011277	0.018835	0.006539	0.008200	0.008083	0.002256	0.060173	0.017660	-0.082183	-0.049334
Annual Return	0.008691	0.063001	-0.019882	-0.008666	-0.009666	-0.044379	0.511522	0.050917	-0.968309	-0.347690
Annual Volatility	0.412686	0.384897	0.404957	0.402717	0.403667	0.398984	0.597459	0.418869	1.635959	0.444683
calmar ratio	0.027285	0.225139	-0.062220	-0.027334	-0.030399	-0.137931	1.569361	0.162648	-1.091443	-0.841986
omega ratio	1.035433	1.060238	1.020384	1.025547	1.025180	1.006956	1.175546	1.056211	0.800265	0.866324
sharpe ratio	0.179024	0.298995	0.103807	0.130173	0.128320	0.035807	0.955212	0.280338	-1.304610	-0.783154
sortino ratio	0.313227	0.492473	0.209558	0.247664	0.244903	0.117489	1.485537	0.455050	-1.828837	-1.006722
tail ratio	0.946923	1.140079	0.953251	1.020271	1.012320	0.904831	1.254711	1.027627	0.880904	0.981969
Mean Absolute Deviation	0.016803	0.015587	0.016517	0.016479	0.016516	0.016352	0.028067	0.016986	0.075874	0.019229
Value at Risk	0.043948	0.038679	0.043711	0.042869	0.043063	0.043581	0.057130	0.043304	0.175762	0.045587
Expected Shortfall	0.066497	0.059799	0.065909	0.064640	0.064850	0.065137	0.079753	0.066448	0.223696	0.070803
Worst Scenario	0.107596	0.102493	0.105070	0.104414	0.104541	0.101037	0.131351	0.111361	0.283885	0.103344
maximum drawdown	0.357005	0.304269	0.359209	0.356046	0.357378	0.364005	0.358087	0.347766	1.682561	0.508909

Table 12: Black-Litterman: Metrics Summary(2020/01 - 2020/07).

Table 12 lists the metrics calculated from Black-Litterman optimizer. From Figure 21, though it is clear that Random Forest will have the best metrics, it is still worthy to take a look at the numbers.

For Information Ratio, the Random Forest model has the highest IR of 6%, whereas all other models do not have IR exceeding 2%. Note that VAR model has a negative IR of -8.22%, and this is due to its also negative Annual Return. For all models, 6 our of 10 have negative Annual Returns. Random Forest model has an Annual Return as high as 51%, far above other models where their Annual Returns are no more than 9%. The VAR model has the highest annual volatility, which is also reflected in Figure 21, as the line goes more volatile for VAR than other models.

The Random Forest ranks first in calmar ratio, not because of a very low max drawdown but because of its high Annual Return. The maximum drawdown is very even among all models, except for VAR. VAR has an exceptionally high max drawdown of 168%, which could possibly be the shape dip happened during April on 21.

8 out of 10 models have omega ratio > 1, where Random Forest remains to be the highest. VAR and ESRNN have omega ratios < 1, meaning the chance of winning is smaller than the chance of losing money. Sharpe ratio for Random Forest is very close to 1, whereas Sharpe ratio for VAR is much smaller than -1. Sortino ratio could go more extreme than Sharpe ratio, as it usw the standard deviation of negative portfolio returns. The largest sortino ratio is from Random Forest and goes nearly to 150%, and in contrast, the smallest sortino ratio is with VAR and goes nearly to -200%. There is a huge gap in performance between these two models.

One thing worth noting is that VAR has the highest Value at Risk of 17.58%, where all other models, including Random Forest, has Value of Risk at around 5%. The situation is the same for the Expected Shortfall and Worst Scenario.

In a word, when considering the results from Black-Litterman optimizer, Random Forest outplayed all other models, but the difference is not very large. VAR model is the worst performing model by having a very negative Annual Return, large volatility, and high risk. Both the graph and the table have proved this conclusion.

Since the **New Set** is built from the predicted values, it is necessary to check the prediction accuracy and see if VAR model also performs badly throughout the whole period. See Table 22 and Table 29 for more information.

For the **Whole Set**, Random Forest is no longer the best model, with even a negative Annual Return. It also has high volatility, making it the second worst model among all models. VAR models has remained to be the worst performing model, with negative Annual Return, negative Information Ratio, large volatility, and highest Value at Risk.

For the **Hist Set**, the metrics are very close to that of **Whole Set**. Comparing three periods, VAR model has been consistent in being the worst performing model. Random Forest is the best performing model in **New Set**, but it becomes the second-worst performing model in **Whole Set** and **Hist Set**. It is possible that Random Forest model does not work well in a long period perspective.

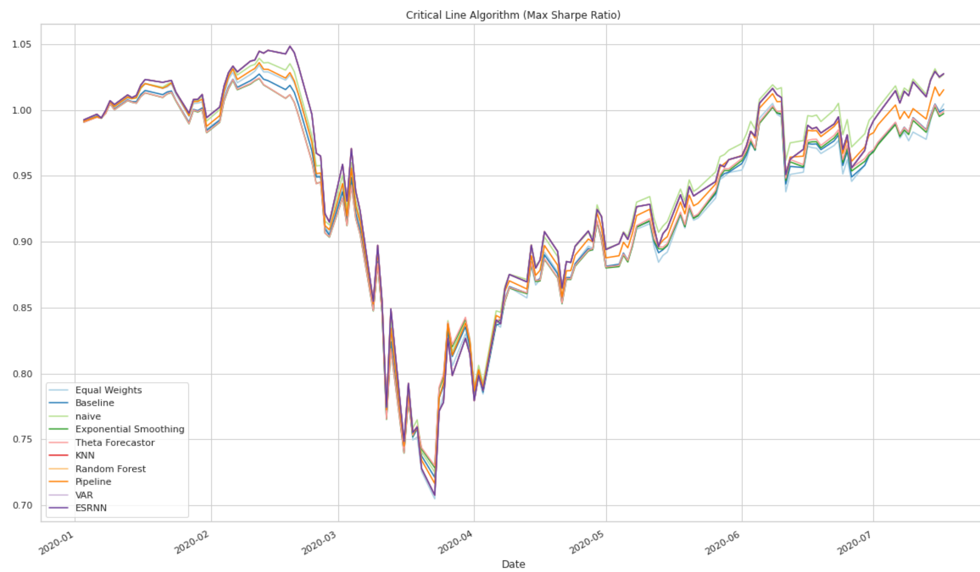

Figure 22: Returns from Critical Line Algorithm (Maximum Sharpe Ratio).

Figure 22 shows the cumulative return using critical line algorithm with maximum sharpe ratio as optimization function. Before mid May of 2020, the ESRNN forecasting method has generated the highest return compared to other methods. However, the naive forecaster performed better than ESRNN afterward. At July 2020, the cumulative return generated by these two forecasting methods are almost the same. In most of the time, most of the forecasting method outperformed the benchmark. Nonetheless, at mid-March 2020, when the Covid-19 outbreak in the United States, the financial market ushered in severe shock. The tendency of the financial market becomes harder to predict than usual. In such a situation, only three forecasting methods still outperformed the benchmark method - Naive, random forest, and exponential smoothing forecaster.

Metrics	Equal Weights	Baseline	Naive	ES	Theta	KNN	Random Forest	Pipeline	VAR	ESRNN
Information Ratio	0.011277	0.009201	0.017498	0.007754	0.008193	0.018222	0.018222	0.014187	0.018222	0.018222
Annual Return	0.008691	0.000968	0.050753	-0.004831	-0.002153	0.051853	0.051853	0.028910	0.051853	0.051853
Annual Volatility	0.412686	0.389008	0.411324	0.376987	0.376740	0.438427	0.438427	0.408294	0.438427	0.438427
calmar ratio	0.027285	0.003249	0.167814	-0.016739	-0.007488	0.159464	0.159464	0.093695	0.159464	0.159464
omega ratio	1.035433	1.028936	1.055823	1.024250	1.025630	1.058194	1.058194	1.044904	1.058194	1.058194
sharpe ratio	0.179024	0.146061	0.277779	0.123097	0.130061	0.289270	0.289270	0.225206	0.289270	0.289270
sortino ratio	0.313227	0.269017	0.448963	0.239340	0.248927	0.462182	0.462182	0.377470	0.462182	0.462182
tail ratio	0.946923	0.916137	0.930900	0.896198	0.896168	1.006682	1.006682	0.931599	1.006682	1.006682
Mean Absolute Deviation	0.016803	0.015788	0.016633	0.015353	0.015346	0.017739	0.017739	0.016571	0.017739	0.017739
Value at Risk	0.043948	0.041766	0.044091	0.040970	0.041017	0.045432	0.045432	0.041913	0.045432	0.045432
Expected Shortfall	0.066497	0.064323	0.066672	0.062874	0.062822	0.070002	0.070002	0.065907	0.070002	0.070002
Worst Scenario	0.107596	0.101292	0.109650	0.096598	0.096486	0.118530	0.118530	0.107209	0.118530	0.118530
maximum drawdown	0.357005	0.329894	0.332918	0.318275	0.316706	0.361796	0.361796	0.342802	0.361796	0.361796

Table 13: Critical Line Algorithm (Maximum Sharpe Ratio): Metrics Summary.

Table 13 shows the metrics summary for critical line optimizer with maximum sharpe ratio during 2020-01-03 to 2020-07-17. In this case, KNN, random forest, VAR and ESRNN forecasting method outperform others in information ratio. From the annual return aspect, exponential smoothing and theta forecasters have generated negative values. This can also be seen in figure 22, for more than half of the time, these two forecasters underperformed the benchmark. All of the methods have similar annual volatility, which is in the range between 0.37 and 0.43.

On the calmar ratio, sortino ratio and sharpe ratio aspects,KNN, random forest, VAR, ESRRNN outplay again with sharpe ratio of 0.289. This situation indicates they performed better on a risk-adjusted basis at the given period of time. Next, omega ratio has suggested that the above four forecasting methods have a higher probability to generate gain compared to other methods. Hence, these metrics imply that KNN, random forest, VAR and ESRNN forecasting methods outperformed others.

On the other hand, not of the method has a tail ratio greater than 1, some of them are in the range of 0.89 to 0.94. This implies that the loss is around 1.12 to 1.06 times as bad as to the profit. For Value-at-Risk, KNN, random forest, VAR and ESRNN all have a value of 4.5%, which are larger than other methods' VaR. This means that the maximum potential loss will be around 4.5% times the investment amount at 95% of confidence interval. The expected shortfall, worst scenario and maximum drawdown metrics also imply the same phenomenon. These four methods all have the highest MDD value as well, which is 0.361. This is suggesting that these four forecasting methods may bring a higher potential loss to the portfolio in such a given period of time.

Nevertheless, the situation changes a little bit in different time periods. Referring to table 30, which focuses on the time from 1996-03-19 to 2020-01-02, these four forecasting methods - KNN, random forest, VAR, ESRNN still have the most outstanding performance on numerous aspect, such as information ratio, annual return, calmar ratio and sharpe ratio. On the contrary, none of the value has tail ration greater than 1, which means that the loss is over 1 times as bad as the profit. Viewing the loss evaluation metrics during this time period, pipeline method performs the worst in the Value-at-Risk aspects; the Naive method has the highest value for both expected shortfall and maximum markdown, showing that under certain confidence, it may have a chance to generate the largest potential loss among all the methods. As for the worst scenario, exponential smoothing has the highest score. Although it seems that KNN, random forest, VAR and ESRNN still have better performance on return aspects during this time period, they didn't again perform the worst in the loss aspects.

When looking at the entire dataset in table 23 in the Appendix, KNN, random , VAR and ESRNN still outperform on the return indicators, such as information ratio, annual return. On the loss measurement aspects, the above four forecasting methods perform worst in the expected shortfall and worst scenario; pipeline methods have the highest value on Value-at-Risk; naive forecaster performs worst in the maximum markdown metric. This shows some uncertainty in the possible loss measurements.

In different time periods, **KNN**, **ESRNN**,**random forest** and **VAR** forecasting methods seems to outperform most of the time with CLA optimizer in the return aspects even if them possesses higher potential loss in the given time frame. Comparing three tables- table 13 , table 30 and table 23, contradiction can be found through the loss metrics.This may be due to the influence brought by the outbreak of COVID-19. Referring back to figure 22, one can observe the huge fluctuation of the return starting from March 2020. The outbreak of the virus not only affects people's daily lives all around the world but also change the relationships among different countries. All these factors will contribute to the movement of the financial market, and thus make it harder to predict compared to the past.

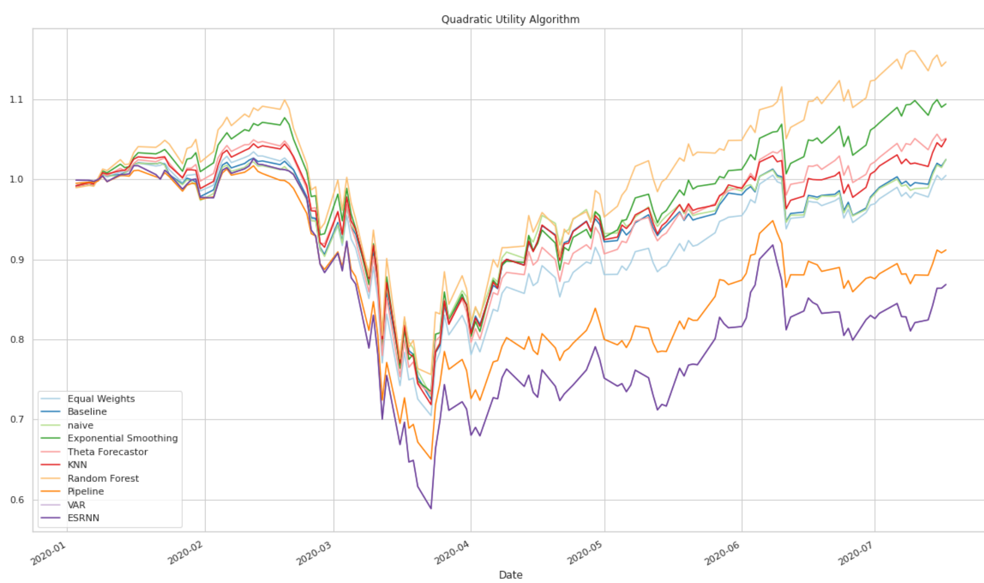

Figure 23: Returns from Quadratic Utility Algorithm.

Figure 23 shows the return using Mean Variance Analysis - Maximize return minimize volatility(refer to quadratic utility algorithm) as optimization function. The Random Forest forecasting method has generated the highest return compared to other methods. As discussed above, due to COVID-19 happened in March 2020, the U.S financial market suffered a huge loss, there are still some forecasting models outperformed the baseline portfolio, which are Random Forest model, Exponential Smoothing and KNN.

Metrics	Equal Weights	Baseline	Naive	ES	Theta	KNN	Random Forest	Pipeline	VAR	ESRNN
Information Ratio	0.011277	0.016333	0.016162	0.034413	0.024140	0.023820	0.045080	-0.015056	-0.018909	-0.018909
Annual Return	0.008691	0.045943	0.046198	0.181005	0.096699	0.094405	0.288158	-0.157596	-0.229796	-0.229796
Annual Volatility	0.412686	0.391618	0.382154	0.471814	0.427862	0.430428	0.501883	0.424239	0.507490	0.507490
calmar ratio	0.027285	0.156329	0.161614	0.569895	0.319624	0.302628	0.923389	-0.437477	-0.539088	-0.539088
omega ratio	1.035433	1.051539	1.050734	1.112614	1.077798	1.076348	1.148189	0.955773	0.945400	0.945400
sharpe ratio	0.179024	0.259284	0.256556	0.546293	0.383218	0.378127	0.715623	-0.239002	-0.300178	-0.300178
sortino ratio	0.313227	0.433253	0.434397	0.828166	0.596127	0.597310	1.079268	-0.261223	-0.357989	-0.357989
tail ratio	0.946923	1.087868	1.121287	0.993248	0.945936	1.122874	1.001948	0.980722	0.969423	0.969423
Mean Absolute Deviation	0.016803	0.015998	0.015693	0.018991	0.017284	0.017491	0.020463	0.017805	0.021561	0.021561
Value at Risk	0.043948	0.038001	0.040073	0.046451	0.044842	0.043202	0.047030	0.046168	0.050424	0.050424
Expected Shortfall	0.066497	0.061287	0.059486	0.072650	0.068295	0.066155	0.074891	0.069794	0.081109	0.081109
Worst Scenario	0.107596	0.103291	0.099334	0.130149	0.115515	0.115447	0.139149	0.100560	0.114453	0.114453
maximum drawdown	0.357005	0.323725	0.313999	0.345997	0.330766	0.344725	0.333760	0.421528	0.521263	0.521263

Table 14: Quadratic Utility Algorithm: Metrics Summary.

Table 14 gives a clear risk metric results regarding to Mean Variance optimal portfolio in **New Set**, with a comparison of the **Whole Set**: Table 20 and **Hist Set**: Table 27. In the **New Set**, it can be seen that **Random Forest** optimal portfolio generates the highest annual return and sharpe ratio compared to other enhanced portfolios and baseline portfolio, followed by **Exponential Smoothing**. On the calmar ratio aspect, Random Forest followed by Exponential Smoothing again have the best performance with values 0.923 and 0.569 respectively. On the contrary, VAR and ESRNN portfolios perform the worst, which computes negative value for calmar ratio of -0.53. This is because they generate negative annual return of -0.229 in this time period.

Notice that almost all the portfolios have omega ratio that are larger than 1, for the last three portfolio: Pipeline, VAR and ESRNN, they have omega ratios around 0.95. Especially for the sharpe ratio and information ratio of Random Forest portfolio, which are 0.71 and 0.045 respectively, followed by Exponential Smoothing portfolio. This indicate under the risk adjusted environment, Random Forest followed by Exponential Smoothing have tendency to generate higher return.

As for tail ratio, only baseline, naive, KNN and random forest portfolios are greater than 1. This implies that the loss for all portfolios are around 0.95 to 1.12 time as bad as the profit. On the other hand, VAR and ESRNN underperform all the other methods on loss measurements. They both have the highest values for Value at Risk, expected shortfall and maximum drawdown. This suggest that these two methods may have apotential to generate the greatest loss among various method during this time period. Note that random forest portfolio has the highest worst scenario, and there is no doubt that Random Forest portfolio has higher annual volatility.

By looking at the **Hist Set** and **Whole Set**, different phenomenon is observed. From **Hist Set**, **Random Forest and Exponential Smoothing** have outperformed in the annual return aspect. while KNN has the highest information ratio. This fact also reflect on the calmar ratio, sharpe ratio, omega ratio and sortino ratio, KNN outperforms than other methods, followed by ES and random forest. In the above indicators, we can see that there is a significant increase in annual return and sharpe ratio from **Hist to New Set** for random forest. On the loss measurement, random forest performs worst in Value at Risk, expected shortfall, worst scenario and maximum drawdown. This means that in this time period, random forest may possess the highest potential loss.

In the **Whole Set**, situation is similar to the **Hist Set**, **Random Forest and Exponential Smoothing** have outperformed in the annual return aspect. while KNN has the highest information ratio. This fact also reflect on the calmar ratio, sharpe ratio, omega ratio and sortino ratio, KNN outperforms than other methods, followed by ES and random forest. Also, random forest performs worst on the loss measurement, with the highest volatility, same as in **Hist Set**.

Above all, **Random Forest** followed by **Exponential Smoothing** seems to outperform from 2020-01-03 to 2020-07-17. However, using the **Whole Set and Hist Set**, random forest and ES can have a higher return, but random forest has relatively high risk and higher potential loss. From a longer period perspective, KNN can be the best portfolio, but in the out-of-sample case, random forest indeed performs very well. There are various reasons, for example, the COVID-19 situation as we talked above. Hence a conclusion can be derived: **KNN** is the best portfolio for Whole and Hist Set, but **Random Forest and Exponential Smoothing** perform best on New Set in Mean-Variance portfolio: quadratic utility algorithm.

Metrics	Equal Weights	Baseline	Naive	ES	Theta	KNN	Random Forest	Pipeline	VAR	ESRNN
Information Ratio	0.011277	0.019418	0.016349	0.016371	0.015777	0.017884	0.005629	0.014187	0.018222	0.018222
Annual Return	0.008691	0.064047	0.042443	0.043531	0.039598	0.055106	-0.016428	0.028910	0.051853	0.051853
Annual Volatility	0.412686	0.411189	0.414117	0.408375	0.407862	0.398332	0.373478	0.408294	0.438427	0.438427
calmar ratio	0.027285	0.213823	0.138197	0.143280	0.130123	0.188361	-0.056847	0.093695	0.159464	0.159464
omega ratio	1.035433	1.062084	1.052069	1.052123	1.050179	1.056999	1.017518	1.044904	1.058194	1.058194
sharpe ratio	0.179024	0.308246	0.259529	0.259874	0.250445	0.283907	0.089364	0.225206	0.289270	0.289270
sortino ratio	0.313227	0.492334	0.423099	0.425117	0.411863	0.460360	0.193773	0.377470	0.462182	0.462182
tail ratio	0.946923	0.959070	0.933153	0.976613	0.970035	0.987481	0.885890	0.931599	1.006682	1.006682
Mean Absolute Deviation	0.016803	0.016627	0.016752	0.016525	0.016509	0.016134	0.015240	0.016571	0.017739	0.017739
Value at Risk	0.043948	0.043925	0.044040	0.043057	0.043035	0.042583	0.040498	0.041913	0.045432	0.045432
Expected Shortfall	0.066497	0.066377	0.067004	0.065978	0.065988	0.064698	0.062492	0.065907	0.070002	0.070002
Worst Scenario	0.107596	0.109920	0.110557	0.108947	0.108691	0.105611	0.095838	0.107209	0.118530	0.118530
maximum drawdown	0.357005	0.328831	0.339157	0.335248	0.336017	0.320809	0.319226	0.342802	0.361796	0.361796

Table 15: Critical Line Algorithm (Minimum Volatility): Metrics Summary.

Table 15 shows the metrics summary for critical line optimizer with minimum volatility in **New Set**, with a comparison of the **Whole Set**: Table 24 and **Hist Set**: Table 31. In the **New Set**, it can be seen that Baseline portfolio has the highest annual return and sharpe ratio compared to other enhanced portfolios which are computed by forecasting models. On the calmar ratio aspect, KNN algorithm has the best performance with a value around 0.18, on the contrary, random forest performs the worst, which possesses a negative value of -0.05. This is because random forest also has a negative annual return in this time period.

In this case, all of the methods possess a similar omega ratio that is larger than 1, which means they all have similar probability to generate gain. However, differences among various forecasting methods still appear in the sharpe ratio and sortino ratio. In these two indicators, the benchmark apparently outplayed all the other forecasting methods. This indicates under the risk-adjusted environment, the benchmark method has a tendency to generate higher average returns.

As for tail ratio, only two methods have tail ratio greater than 1, which are VAR and ESRNN, which are both 1.006. This implies that the loss is around 0.99 times as bad as the profit. On the other hand, VAR and ESRNN underperform all the other methods on the loss measurements. They both have the highest values for Value-at-Risk, expected shortfall, worst scenario and maximum drawdown. This suggests that these two method may have the potential to generate the greatest loss among various methods during this time period.

By looking at the **Hist Set** and **Whole Set**, different phenomenon is observed. From the **Hist Set**, VAR and ESRNN have outperformed in the return aspect. They both have the highest information ratio since they also have the highest return. This fact also reflects on the calmar ratio, sharpe ratio, omega ratio and sortino ratio. In the above indicators, VAR and ESRNN win the predominance. On the loss measurement aspect, the benchmark perform worst in three indicators, which are Value-at-Risk, expected shortfall and maximum drawdown. This means that in this time period, the benchmark method may possess the highest potential loss.

In the **Whole Set**, situation is similar to the **Hist Set**. VAR and ESRNN again outperform others in the return measurements. The Baseline method performs the worst in Value-at-Risk and maximum drawdown indicators as well. However, in a longer period, the VAR and ESRNN method performs the worst on the expected shortfall and worst scenario indicators, which is one of the differences compared to the **Hist Set**.

In the **New Set**, it seems that the baseline method outplayed all the other forecasting methods. Nonetheless, using the **Whole Set** and **Hist Set**, ESRNN and VAR seem to perform the best among all the methods. This may be caused by the reason mentioned before - the uncertainties in the financial market. Financial market fluctuates enormously when global disasters occur or political operations change. This can be seen in figure 24 in the appendix, after the outbreak of COVID-19 at March, the predominance methods kept changing drastically afterward, which means that under such uncertain situation, none of the methods can perform as stable as usual.

Metrics	Equal Weights	Baseline	Naive	ES	Theta	KNN	Random Forest	Pipeline	VAR	ESRNN
Information Ratio	0.011277	0.015513	0.003076	0.021443	0.021862	0.007688	-0.004730	0.019603	0.023715	0.004473
Annual Return	0.008691	0.043126	-0.034227	0.080084	0.082784	-0.010938	-0.090814	0.064660	0.094952	-0.013373
Annual Volatility	0.412686	0.373799	0.383379	0.383820	0.385188	0.399154	0.410762	0.418273	0.392664	0.339286
calmar ratio	0.027285	0.153600	-0.111458	0.287536	0.296949	-0.034795	-0.265003	0.209285	0.340081	-0.049711
omega ratio	1.035433	1.048855	1.009478	1.068423	1.069874	1.023962	0.985727	1.062653	1.076126	1.013695
sharpe ratio	0.179024	0.246264	0.048837	0.340400	0.347051	0.122048	-0.075083	0.311187	0.376464	0.071003
sortino ratio	0.313227	0.416904	0.138244	0.549308	0.558313	0.236643	-0.036169	0.498928	0.595507	0.178777
tail ratio	0.946923	1.093586	0.889227	1.138967	1.138894	0.966976	0.905141	1.046505	1.072333	0.953071
Mean Absolute Deviation	0.016803	0.015276	0.015744	0.015612	0.015654	0.016309	0.017038	0.016938	0.015915	0.014049
Value at Risk	0.043948	0.037394	0.039306	0.038935	0.039200	0.042150	0.043214	0.042455	0.040512	0.033752
Expected Shortfall	0.066497	0.059823	0.062671	0.060841	0.061076	0.064551	0.066209	0.066104	0.062881	0.055408
Worst Scenario	0.107596	0.097170	0.096277	0.101454	0.101956	0.102845	0.100628	0.111775	0.104340	0.085367
maximum drawdown	0.357005	0.307961	0.344748	0.303432	0.303615	0.352832	0.394905	0.341777	0.303190	0.295395

Table 16: Maximum Diversification: Metrics Summary.

Table 16 shows the metrics summary for Maximum Diversification optimization method in **New Set**, with a comparison of the **Whole Set**: Table 18 and **Hist Set**: Table 25. In the **New Set**, it can be seen that **Vector Autoregression** optimal portfolio generates the highest annual return and sharpe ratio compared to other enhanced portfolios and baseline portfolio, followed by **Theta, Exponential Smoothing and Pipeline**. On the calmar ratio aspect, VAR followed by Theta again have the best performance with values 0.34 and 0.29 respectively. On the contrary, ESRNN portfolio performs the worst, which compute negative value for calmar ratio of -0.05. This is because they generate negative annual return of -0.013 in this time period.

Notice that almost all the portfolios have omega ratio that is larger than 1 except for the random forest portfolio, which has omega ratio of 0.9857. Especially for the sharpe ratio and information ratio of VAR portfolio, which are 0.3764 and 0.0237 respectively, followed by Theta, Exponential Smoothing and Pipeline portfolio. This indicates under the risk adjusted environment, VAR followed by Theta and ES have a tendency to generate higher returns.

As for tail ratio, Baseline, ES, Theta, Pipeline and VAR portfolios are greater than 1. This implies that the loss for all portfolios is around 0.88 to 1.13 time as bad as the profit. On the other hand, random forest underperforms all the other methods on loss measurements. It has the highest values for Value at Risk, Expected Shortfall and maximum drawdown. This suggests that this method may have the potential to generate the greatest loss among various methods during this time period. Note that VAR portfolio has the highest worst scenario, and Pipeline portfolio has higher annual volatility.

By looking at the **Hist Set** and **Whole Set**, different phenomenon is observed. From **Hist Set**, **Pipeline** has outperformed in the annual return aspect. and Pipeline has the highest information ratio. This fact also reflect on the calmar ratio, sharpe ratio, omega ratio and sortino ratio, Pipeline outperforms other methods, followed by Theta and Exponential Smoothing. In the above indicators, we can see that there is a slight increase in annual return and sharpe ratio from **Hist to New Set** for VAR. On the loss measurement, VAR performs worst in Value at Risk, expected shortfall, worst scenario and maximum drawdown. This means that in this time period, VAR may possess the highest potential loss.

In the **Whole Set**, situation is similar to the **Hist Set**, **Pipeline** has outperformed in the annual return aspect. Pipeline has the highest information ratio too. This fact also reflect on the calmar ratio, sharpe ratio, omega ratio and sortino ratio, Pipeline outperforms than other methods, followed by Theta and ES. Also, VAR performs worst on the loss measurement, with the highest volatility, same as in **Hist Set**.

Above all, **VAR** followed by **Theta, Exponential Smoothing and Pipeline** seems to outperform from 2020-01-03 to 2020-07-17. However, using the **Whole Set and Hist Set**, Pipeline can have a higher return, but VAR has a relatively high risk and higher potential loss. From a longer period perspective, Pipeline can be the best portfolio, but in the out-of-sample case, VAR indeed performs very well. There are various reasons, for example, the COVID-19 situation as we talked above. Hence a conclusion can be derived: **Pipeline** is the best portfolio for Whole and Hist Set, but **VAR** performs best on New Set in Maximum Diversification Optimization portfolios.

Metrics	Equal Weights	Baseline	Naive	ES	Theta	KNN	Random Forest	Pipeline	VAR	ESRNN
Information Ratio	0.011277	0.011070	0.009552	0.008471	0.008907	0.016718	-0.000365	0.023731	-0.017105	0.019701
Annual Return	0.008691	0.015258	-0.002841	-0.005137	-0.002906	0.048239	-0.071052	0.094207	-0.111504	0.068312
Annual Volatility	0.412686	0.377380	0.413012	0.396375	0.398381	0.392390	0.427278	0.419992	0.319787	0.391984
calmar ratio	0.027285	0.052349	-0.008915	-0.016473	-0.009295	0.164653	-0.202982	0.311250	-0.397848	0.238962
omega ratio	1.035433	1.034528	1.030009	1.026043	1.027381	1.053270	0.998896	1.076460	0.950808	1.061495
sharpe ratio	0.179024	0.175724	0.151637	0.134477	0.141393	0.265392	-0.005796	0.376721	-0.271536	0.312746
sortino ratio	0.313227	0.317292	0.273436	0.257366	0.266963	0.437933	0.056547	0.593377	-0.275807	0.521355
tail ratio	0.946923	1.060229	0.907012	0.960290	0.964463	1.091522	0.882018	1.108516	0.775304	1.216300
Mean Absolute Deviation	0.016803	0.015491	0.016780	0.016439	0.016535	0.015881	0.017792	0.016982	0.013675	0.016286
Value at Risk	0.043948	0.036517	0.044431	0.041112	0.041212	0.040325	0.047214	0.042318	0.033846	0.039095
Expected Shortfall	0.066497	0.059984	0.067315	0.062228	0.062421	0.062992	0.068004	0.065680	0.053802	0.059408
Worst Scenario	0.107596	0.097662	0.108392	0.101202	0.101730	0.104070	0.106110	0.113250	0.098047	0.099931
maximum drawdown	0.357005	0.322422	0.356416	0.349763	0.350679	0.322167	0.403821	0.332339	0.316449	0.312619

Table 17: Minimum Variance: Metrics Summary.

Table 17 shows the metrics summary for Minimum Variance optimization method in **New Set**, with a comparison of the **Whole Set**: Table 19 and **Hist Set**: Table 26. In the **New Set**, it can be seen that **Pipeline** optimal portfolio generates the highest annual return and sharpe ratio compared to other enhanced portfolios and baseline portfolio, followed by **ESRNN**. On the calmar ratio aspect, Pipeline followed by ESRNN again has the best performance with values 0.31 and 0.238 respectively. On the contrary, VAR portfolio performs the worst, which compute negative value for calmar ratio of -0.39. This is because they generate negative annual return of -0.11 in this time period.

Notice that almost all the portfolios have omega ratio that are larger than 1 except for the random forest and VAR portfolios, which have omega ratio of 0.998 and 0.95 respectively. Especially for the sharpe ratio and information ratio of Pipeline portfolio, which are 0.376 and 0.0237 respectively, followed by ESRNN portfolio. This indicates under the risk adjusted environment, Pipeline followed by ESRNN have a tendency to generate higher returns.

As for tail ratio, Baseline, KNN, Pipeline and ESRNN portfolios are greater than 1. This implies that the loss for all portfolios is around 0.88 to 1.21 time as bad as the profit. On the other hand, random forest underperforms all the other methods on loss measurements. It has the highest values for Value at Risk, Expected Shortfall and maximum drawdown. This suggests that this method may have the potential to generate the greatest loss among various method during this time period. Note that Pipeline portfolio has the highest worst scenario, and random forest portfolio has higher annual volatility.

By looking at the **Hist Set** and **Whole Set**, different phenomenon is observed. From **Hist Set**, **Pipeline** has outperformed in the annual return aspect. and Pipeline has the highest information ratio. This fact also reflects on the calmar ratio, sharpe ratio, omega ratio and sortino ratio, Pipeline outperforms than other methods, followed by ESRNN. In the above indicators, we can see that there is no big change in annual return and sharpe ratio from **Hist to New Set** for Pipeline. On the loss measurement, VAR performs worst in Value at Risk, expected shortfall, worst scenario and maximum drawdown. This means that in this time period, VAR may possess the highest potential loss.

In the **Whole Set**, situation is similar to the **Hist Set**, **Pipeline** has outperformed in the annual return aspect. Pipeline has the highest information ratio too. This fact also reflects on the calmar ratio, sharpe ratio, omega ratio and sortino ratio, Pipeline outperforms than other methods, followed by ESRNN. Also, VAR performs worst on the loss measurement, with the highest volatility of 23.6%, same as in **Hist Set**.

Above all, **Pipeline** followed by **ESRNN** seems to outperform from 2020-01-03 to 2020-07-17. In addition, using the **Whole Set and Hist Set**, Pipeline can also have a higher return, where VAR has a relatively high risk and higher potential loss. From both long period perspective and out-of-sample case, Pipeline followed by ESRNN portfolio can be the best-enhanced portfolio with the Minimum Variance Optimization method.

VI Summary and Conclusion

To summarize, the goal of this project is to do the asset pricing and risk premia forecasting by using different statistical combined machine learning model, and explore the relative top 2-3 models from both forecasting perspective and portfolio optimization perspective. **Eight** different statistical combined with machine learning models have been fitted within univariate or multivariate forecasting approaches. **Four** forecasting metrics have been evaluated for standalone model forecasting performance and **seven** portfolio optimization methods have been tested to output the optimal weights in order to construct the optimal portfolios. Several risk metrics are used to compare the forecasting methods portfolio (Enhanced Portfolio) performance in three timelines.

Above all, the top two to three forecasting models are obtained from an overall analysis that combines **Forecasting Values(Graphs), Forecasting Metrics and Portfolio Optimization**. A short statistical summary can be conducted:

Forecasting Graphs: As can be seen and discussed in the previous section, naive forecaster, KNN forecaster, pipeline forecaster and ESRNN seem to capture the fluctuation of the risk premium, VAR and Random Forest seem to predict flat curve, while Exponential Smoothing and Theta method results are quite steady, but can still capture the movement of each point. Considering each point movement, naive, KNN, pipeline and ESRNN can give the clients a better visualizing result.

Forecasting Metrics: There is no doubt that forecasting metrics give a direct and mathematical based result. By calculating the prediction accuracy for each forecasting model, it gives us the numerical result for forecasting accuracy. From the results and discussions above, **Vector Autoregression and Exponential Smoothing** always yield better result among all the five ETF/Index risk premium series. Since all risk premia series are quite stationary and close to zero, so the methods where the forecasted values are not that volatile can give good metrics results.

Portfolio Optimization: Note that by conducting different optimization methods, different objective functions and constraints are used. It can be concluded that probably the best forecasting portfolio (enhanced portfolio) within different optimization methods are not always the same. An overall summary will be conducted and **2 to 3 best portfolios** using forecasting values computed by each forecasting model will be selected below by comparing the portfolio metrics across **Hist Set, New Set and Whole Set**. The definition and description of each metric have been talked above.

The following is a summary of the best performance portfolios within each optimization method.

- Mean-Variance Optimization: Random Forest and Exponential Smoothing for New Set, KNN for Whole and Hist Set.
- Minimum Variance: Pipeline followed ESRNN for all New, Whole and Hist Set.
- Maximum Diversification: VAR followed by Theta, Exponential Smoothing and Pipeline for New Set, Pipeline for Whole and Hist Set.
- Hierarchical Risk Parity: Pipeline for all New, Whole and Hist Set.
- Black Litterman: Random Forest for New Set, Pipeline for Whole and Hist Set.
- CLA Max Sharpe Ratio: KNN, random Forest, VAR, ESRNN for all New, Whole and Hist Set.
- CLA Minimum Volatility: Baseline for New, VAR and ESRNN for Whole and Hist Set.

For the portfolio optimization perspective, **Pipeline** often outperforms than other enhanced and baseline portfolios in **Hist Set and Whole Set**. **Random Forest, VAR, Pipeline and ES** often perform well in **New Set**. Hence, we can select **Random Forest, VAR, Pipeline and Exponential Smoothing** as top forecasting models from portfolio optimization perspective. Above all, For the five ETF/Index series in this project, it can be concluded that **Vector Autoregression, Exponential Smoothing and Pipeline model** are the top forecasting models in different perspectives.

In this project, instead of using traditional forecasting approaches, many machine learning techniques have been included. From simple linear models to regression trees and neural networks, machine learning is designed to estimate complex nonlinear associations. Machine Learning enhanced the way of financial time series forecasting, especially when dealing with complex and multivariate time series forecasting. As for some further considerations and improvements, since in this article, we consider that forecasting only depends on the series themselves, what will happen if adding some other features? For example, trading volume at each day or some macroeconomic features. In addition, how to improve the running time can be another problem, in this project, there is no doubt that ESRNN method used a quite long time when tunning the parameters and output the forecasting result. Hence, adding other features and improve time efficiency can be further considerations and improvements for this project.

VII Appendix

Graphs for Optimization Result (2020-01-03 - 2020-07-17)

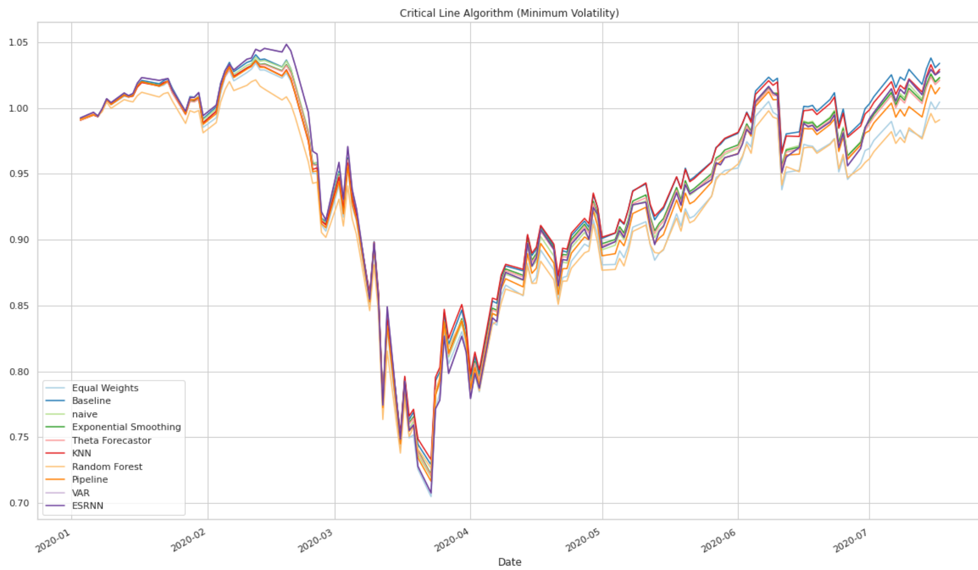

Figure 24: Returns from Critical Line Algorithm (Minimum Volatility).

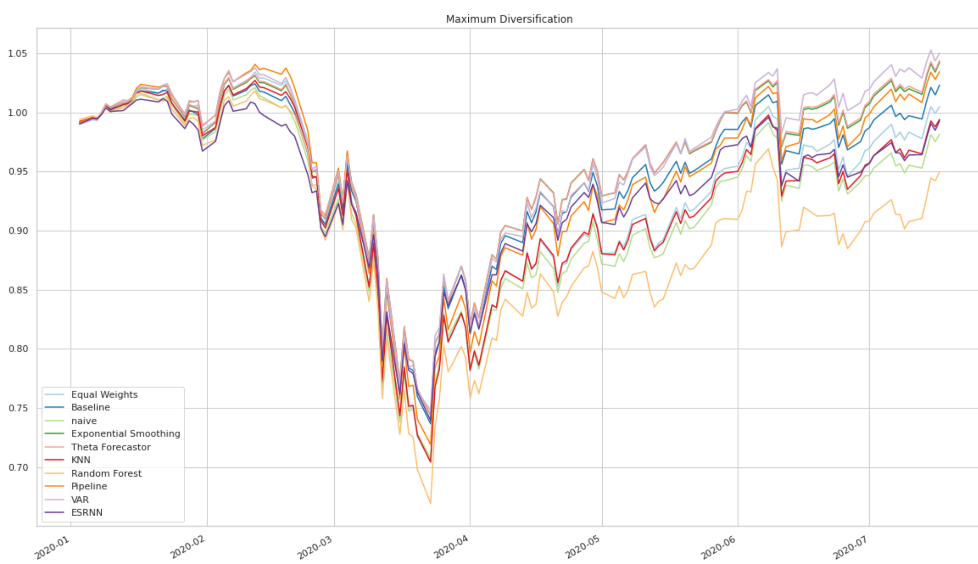

Figure 25: Returns from Max Diversification.

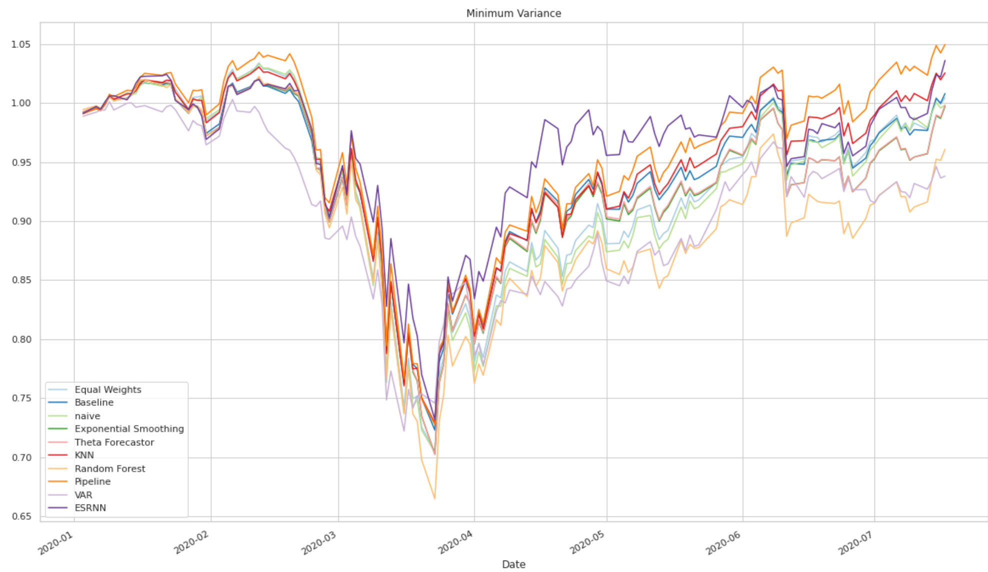

Figure 26: Returns from Minimum Variance.

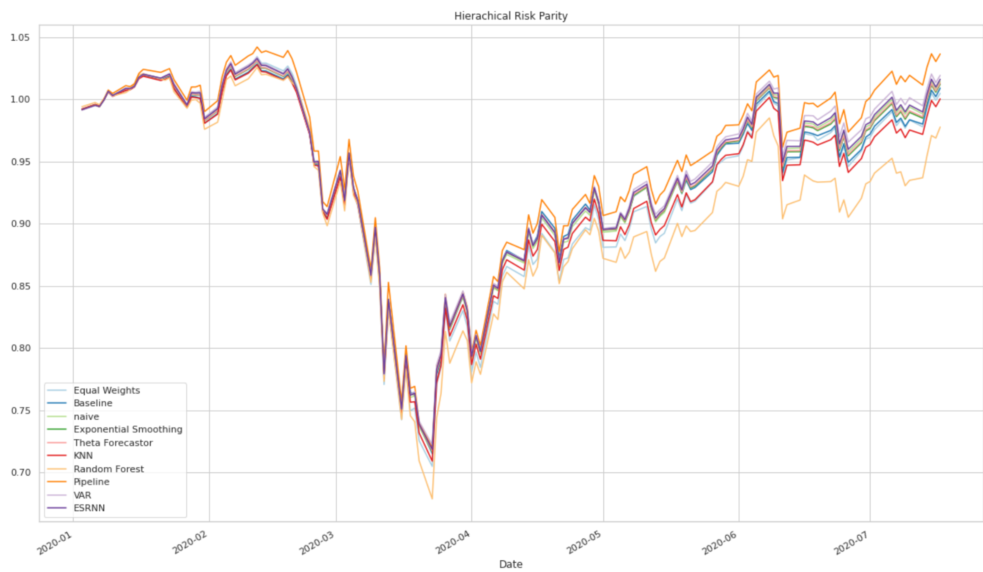

Figure 27: Returns from Hierarchical Risk Parity Optimizer.

Charts for Optimization Result (1996-03-19 - 2020-07-17)

Metrics	Equal Weights	Baseline	Naive	ES	Theta	KNN	Random Forest	Pipeline	VAR	ESRNN
Information Ratio	0.023168	0.022389	0.018527	0.024290	0.024320	0.022252	0.020982	0.026269	0.022844	0.015982
Annual Return	0.074986	0.070934	0.059343	0.077862	0.078171	0.070696	0.066335	0.085588	0.075536	0.050929
Annual Volatility	0.192826	0.184386	0.188352	0.188829	0.189829	0.185545	0.183340	0.194279	0.201754	0.182711
calmar ratio	0.141179	0.146824	0.113717	0.151078	0.149493	0.135877	0.123220	0.159227	0.126403	0.109921
omega ratio	1.070751	1.067013	1.055384	1.072898	1.072984	1.067743	1.063522	1.080642	1.068289	1.047081
sharpe ratio	0.367778	0.355418	0.294103	0.385598	0.386072	0.353239	0.333080	0.417002	0.362641	0.253708
sortino ratio	0.670225	0.661867	0.568822	0.702935	0.703050	0.653671	0.622483	0.741213	0.662796	0.517978
tail ratio	0.948165	0.966125	0.951376	0.974608	0.982218	0.955420	0.990561	0.958195	0.948367	0.962109
Mean Absolute Deviation	0.008221	0.008010	0.008148	0.008203	0.008247	0.007925	0.007860	0.008276	0.008780	0.007992
Value at Risk	0.018707	0.018207	0.018132	0.018595	0.018612	0.017913	0.017166	0.019018	0.020115	0.017705
Expected Shortfall	0.028560	0.027110	0.027744	0.027702	0.027833	0.027590	0.027555	0.028725	0.029392	0.026631
Worst Scenario	0.107596	0.097170	0.096277	0.101454	0.101956	0.102845	0.100628	0.111775	0.104340	0.085367
maximum drawdown	0.672542	0.599360	0.655334	0.663812	0.677639	0.653282	0.696618	0.692087	0.826320	0.543678

Table 18: Maximum Diversification: Metrics Summary.

Metrics	Equal Weights	Baseline	Naive	ES	Theta	KNN	Random Forest	Pipeline	VAR	ESRNN
Information Ratio	0.023168	0.023168	0.021961	0.025150	0.025508	0.024059	0.024527	0.027651	0.001314	0.027200
Annual Return	0.074986	0.072075	0.070644	0.078902	0.080214	0.076162	0.078220	0.090526	-0.002888	0.088264
Annual Volatility	0.192826	0.177932	0.190619	0.181411	0.182235	0.184344	0.187141	0.195403	0.236234	0.192557
calmar ratio	0.141179	0.152793	0.129067	0.166047	0.169611	0.154112	0.148658	0.167722	-0.004645	0.204411
omega ratio	1.070751	1.070213	1.068037	1.076235	1.077327	1.073569	1.074515	1.084831	1.003769	1.081964
sharpe ratio	0.367778	0.367786	0.348614	0.399237	0.404926	0.381932	0.389352	0.438950	0.020859	0.431786
sortino ratio	0.670225	0.681685	0.642966	0.722606	0.730229	0.697970	0.700187	0.772924	0.151664	0.765924
tail ratio	0.948165	0.975678	0.945191	0.976945	0.981464	0.961075	0.991248	0.950181	0.940200	1.006407
Mean Absolute Deviation	0.008221	0.007645	0.008003	0.007815	0.007854	0.007861	0.008034	0.008346	0.010396	0.008366
Value at Risk	0.018707	0.017150	0.018213	0.017264	0.017352	0.017820	0.017670	0.019367	0.023519	0.018365
Expected Shortfall	0.028560	0.026460	0.028382	0.027091	0.027211	0.027246	0.028141	0.028866	0.034159	0.028563
Worst Scenario	0.107596	0.097662	0.108392	0.101202	0.101730	0.104070	0.106110	0.113250	0.104077	0.099931
maximum drawdown	0.672542	0.564784	0.695997	0.576889	0.573260	0.609420	0.675269	0.700160	0.865234	0.513988

Table 19: Minimum Variance: Metrics Summary.

Metrics	Equal Weights	Baseline	Naive	ES	Theta	KNN	Random Forest	Pipeline	VAR	ESRNN
Information Ratio	0.023168	0.026125	0.025705	0.028006	0.024294	0.029599	0.027309	0.014353	0.017989	0.017989
Annual Return	0.074986	0.081514	0.080384	0.101902	0.082230	0.097775	0.107889	0.046541	0.060060	0.060060
Annual Volatility	0.192826	0.179725	0.180377	0.235787	0.210485	0.197581	0.281243	0.198211	0.216546	0.216546
calmar ratio	0.141179	0.174382	0.181669	0.142003	0.127284	0.186132	0.130757	0.077660	0.092186	0.092186
omega ratio	1.070751	1.080058	1.077920	1.086187	1.074408	1.090439	1.083570	1.042899	1.054496	1.054496
sharpe ratio	0.367778	0.414721	0.408049	0.444583	0.385658	0.469863	0.433517	0.227845	0.285565	0.285565
sortino ratio	0.670225	0.747138	0.738835	0.763188	0.689197	0.814818	0.732312	0.462865	0.529658	0.529658
tail ratio	0.948165	0.977983	0.981200	0.922178	0.933849	0.962315	0.951546	0.936760	0.954394	0.954394
Mean Absolute Deviation	0.008221	0.007671	0.007777	0.010041	0.008961	0.008493	0.012031	0.008524	0.009242	0.009242
Value at Risk	0.018707	0.016944	0.017127	0.023700	0.020727	0.019475	0.028144	0.019086	0.020684	0.020684
Expected Shortfall	0.028560	0.026778	0.026766	0.034517	0.030864	0.029304	0.041159	0.029554	0.032935	0.032935
Worst Scenario	0.107596	0.103291	0.099334	0.130149	0.115515	0.115447	0.139149	0.100560	0.114453	0.114453
maximum drawdown	0.672542	0.556483	0.517727	1.088830	0.924482	0.648139	1.432378	0.817894	0.951323	0.951323

Table 20: Quadratic Utility Algorithm: Metrics Summary.

Metrics	Equal Weights	Baseline	Naive	ES	Theta	KNN	Random Forest	Pipeline	VAR	ESRNN
Information Ratio	0.023168	0.023377	0.023084	0.023069	0.023092	0.022627	0.025043	0.026155	0.022469	0.023053
Annual Return	0.074986	0.074126	0.074430	0.073822	0.073985	0.071833	0.079457	0.085669	0.072968	0.074336
Annual Volatility	0.192826	0.184914	0.191299	0.188354	0.188816	0.185293	0.185293	0.196210	0.194344	0.191338
calmar ratio	0.141179	0.147333	0.141750	0.144933	0.144531	0.139688	0.153275	0.156451	0.133394	0.141550
omega ratio	1.070751	1.071071	1.070126	1.070039	1.070099	1.068884	1.076364	1.080279	1.067916	1.069910
sharpe ratio	0.367778	0.371097	0.366448	0.366209	0.366577	0.359188	0.397547	0.415190	0.356689	0.365948
sortino ratio	0.670225	0.680889	0.670732	0.672160	0.672472	0.662876	0.714023	0.737709	0.655843	0.670309
tail ratio	0.948165	0.960558	0.957211	0.958655	0.961111	0.954394	0.976560	0.957083	0.953244	0.960043
Mean Absolute Deviation	0.008221	0.007921	0.008196	0.008076	0.008097	0.007918	0.007934	0.008358	0.008361	0.008212
Value at Risk	0.018707	0.017981	0.018609	0.018260	0.018281	0.017961	0.017471	0.019226	0.019085	0.018608
Expected Shortfall	0.028560	0.027418	0.028242	0.027856	0.027915	0.027514	0.027816	0.028974	0.028565	0.028227
Worst Scenario	0.107596	0.102916	0.105309	0.103846	0.104032	0.102907	0.106326	0.112475	0.104706	0.104839
maximum drawdown	0.672542	0.620721	0.675262	0.638161	0.644741	0.641247	0.657881	0.718065	0.722340	0.678393

Table 21: Hierarchical Risk Parity: Metrics Summary(1996/03 - 2020/07).

Metrics	Equal Weights	Baseline	Naive	ES	Theta	KNN	Random Forest	Pipeline	VAR	ESRNN
Information Ratio	0.023168	0.025744	0.022160	0.023923	0.023786	0.020314	0.007353	0.025838	-0.017158	0.002538
Annual Return	0.074986	0.080455	0.070172	0.075175	0.074922	0.064592	-0.105342	0.084078	-0.694146	0.005220
Annual Volatility	0.192826	0.180199	0.184282	0.181778	0.182599	0.185484	0.643285	0.193993	1.299345	0.216689
calmar ratio	0.141179	0.175741	0.131315	0.146570	0.145380	0.120566	-0.105617	0.157048	-0.694146	0.007966
omega ratio	1.070751	1.079364	1.068426	1.073488	1.072995	1.061746	1.021944	1.079304	0.953057	1.007291
sharpe ratio	0.367778	0.408670	0.351780	0.379766	0.377587	0.322478	0.116721	0.410161	-0.272383	0.040287
sortino ratio	0.670225	0.738723	0.650740	0.692756	0.689154	0.608703	0.212502	0.731038	-0.366192	0.184797
tail ratio	0.948165	0.976797	0.956254	0.979538	0.979652	0.969456	0.932695	0.952417	1.025796	0.928977
Mean Absolute Deviation	0.008221	0.007643	0.007762	0.007715	0.007757	0.007914	0.027448	0.008261	0.058417	0.009536
Value at Risk	0.018707	0.016730	0.017618	0.017162	0.017279	0.017608	0.064862	0.018968	0.127311	0.021123
Expected Shortfall	0.028560	0.026785	0.027542	0.027177	0.027281	0.027640	0.095557	0.028720	0.185264	0.032566
Worst Scenario	0.107596	0.102493	0.105070	0.104414	0.104541	0.101037	0.284991	0.111361	0.447854	0.103363
maximum drawdown	0.672542	0.527926	0.676167	0.640005	0.644867	0.683009	3.935323	0.682606	9.469643	0.950402

Table 22: Black-Litterman: Metrics Summary(1996/03 - 2020/07).

Metrics	Equal Weights	Baseline	Naive	ES	Theta	KNN	Random Forest	Pipeline	VAR	ESRNN
Information Ratio	0.023168	0.018958	0.022577	0.016378	0.016272	0.026601	0.026601	0.022346	0.026601	0.026601
Annual Return	0.074986	0.060934	0.073972	0.053084	0.052824	0.087099	0.087099	0.072994	0.087099	0.087099
Annual Volatility	0.192826	0.190247	0.198074	0.195446	0.196774	0.195971	0.195971	0.196897	0.195971	0.195971
calmar ratio	0.141179	0.114168	0.126762	0.094246	0.092405	0.160838	0.160838	0.129933	0.160838	0.160838
omega ratio	1.070751	1.057711	1.069281	1.049062	1.048658	1.083600	1.083600	1.067964	1.083600	1.083600
sharpe ratio	0.367778	0.300946	0.358397	0.259998	0.258304	0.422278	0.422278	0.354733	0.422278	0.422278
sortino ratio	0.670225	0.577706	0.655705	0.517157	0.514086	0.744207	0.744207	0.650736	0.744207	0.744207
tail ratio	0.948165	0.931926	0.952954	0.953387	0.952934	0.950824	0.950824	0.945004	0.950824	0.950824
Mean Absolute Deviation	0.008221	0.008091	0.008400	0.008414	0.008485	0.008168	0.008168	0.008420	0.008168	0.008168
Value at Risk	0.018707	0.018339	0.019140	0.018799	0.019000	0.018763	0.018763	0.019232	0.018763	0.018763
Expected Shortfall	0.028560	0.027968	0.029105	0.028476	0.028642	0.029268	0.029268	0.028950	0.029268	0.029268
Worst Scenario	0.107596	0.101292	0.109650	0.096598	0.096486	0.118530	0.118530	0.107209	0.118530	0.118530
maximum drawdown	0.672542	0.697186	0.792947	0.759480	0.776437	0.680989	0.680989	0.745620	0.680989	0.680989

Table 23: Critical Line Algorithm (Maximum Sharpe Ratio): Metrics Summary.

Metrics	Equal Weights	Baseline	Naive	ES	Theta	KNN	Random Forest	Pipeline	VAR	ESRNN
Information Ratio	0.023168	0.023129	0.023225	0.023581	0.023360	0.021896	0.015714	0.022346	0.026601	0.026601
Annual Return	0.074986	0.076051	0.075496	0.075980	0.075194	0.071442	0.050805	0.072994	0.087099	0.087099
Annual Volatility	0.192826	0.199241	0.194533	0.190964	0.190616	0.196586	0.193559	0.196897	0.195971	0.195971
calmar ratio	0.141179	0.128824	0.136026	0.142920	0.141852	0.123648	0.093073	0.129933	0.160838	0.160838
omega ratio	1.070751	1.070763	1.071833	1.072823	1.072171	1.066398	1.047060	1.067964	1.083600	1.083600
sharpe ratio	0.367778	0.367157	0.368680	0.374329	0.370832	0.347586	0.249452	0.354733	0.422278	0.422278
sortino ratio	0.670225	0.668174	0.671424	0.681818	0.676882	0.642253	0.502999	0.650736	0.744207	0.744207
tail ratio	0.948165	0.940122	0.946365	0.944400	0.946019	0.941158	0.945070	0.945004	0.950824	0.950824
Mean Absolute Deviation	0.008221	0.008481	0.008194	0.008061	0.008040	0.008425	0.008327	0.008420	0.008168	0.008168
Value at Risk	0.018707	0.019479	0.018683	0.018395	0.018348	0.019301	0.018429	0.019232	0.018763	0.018763
Expected Shortfall	0.028560	0.029222	0.028723	0.028197	0.028161	0.028767	0.028219	0.028950	0.029268	0.029268
Worst Scenario	0.107596	0.109920	0.110557	0.108947	0.108691	0.105611	0.095838	0.107209	0.118530	0.118530
maximum drawdown	0.672542	0.805449	0.735077	0.688514	0.684213	0.781399	0.725576	0.745620	0.680989	0.680989

Table 24: Critical Line Algorithm (Minimum Volatility): Metrics Summary.

Charts for Optimization Result (1996-03-19 - 2020-01-02)

Metrics	Equal Weights	Baseline	Naive	ES	Theta	KNN	Random Forest	Pipeline	VAR	ESRNN
Information Ratio	0.024139	0.023003	0.019507	0.024739	0.024749	0.023355	0.022753	0.027036	0.023049	0.016620
Annual Return	0.076541	0.071574	0.061570	0.077811	0.078066	0.072626	0.070203	0.086068	0.075100	0.052436
Annual Volatility	0.184897	0.177821	0.181572	0.182057	0.183052	0.177811	0.174849	0.186164	0.195326	0.177612
calmar ratio	0.144107	0.148149	0.117984	0.150980	0.149293	0.139586	0.130404	0.160120	0.125672	0.113175
omega ratio	1.072461	1.067824	1.057490	1.073096	1.073121	1.069886	1.067608	1.081508	1.067960	1.048460
sharpe ratio	0.383187	0.365153	0.309671	0.392727	0.392872	0.370752	0.361190	0.429184	0.365894	0.263838
sortino ratio	0.701354	0.683522	0.599167	0.720639	0.720255	0.687954	0.672146	0.767592	0.674263	0.538742
tail ratio	0.956660	0.967906	0.960179	0.983172	0.977297	0.966327	0.971043	0.949734	0.955514	0.950028
Mean Absolute Deviation	0.008026	0.007845	0.007976	0.008034	0.008079	0.007734	0.007652	0.008079	0.008617	0.007854
Value at Risk	0.018407	0.017822	0.017653	0.018086	0.018240	0.017490	0.016991	0.018763	0.019736	0.017521
Expected Shortfall	0.027249	0.026036	0.026599	0.026605	0.026731	0.026302	0.026131	0.027407	0.028364	0.025768
Worst Scenario	0.084767	0.080875	0.086548	0.079494	0.079640	0.084420	0.082216	0.081498	0.082760	0.084463
maximum drawdown	0.672542	0.599360	0.655334	0.663812	0.677639	0.653282	0.696618	0.692087	0.826320	0.543678

Table 25: Maximum Diversification: Metrics Summary.

Metrics	Equal Weights	Baseline	Naive	ES	Theta	KNN	Random Forest	Pipeline	VAR	ESRNN
Information Ratio	0.024139	0.024130	0.022964	0.026448	0.026810	0.024803	0.026385	0.028298	0.001887	0.027912
Annual Return	0.076541	0.073401	0.072374	0.080891	0.082180	0.076804	0.081874	0.090442	-0.000273	0.088721
Annual Volatility	0.184897	0.170787	0.182524	0.173537	0.174315	0.176870	0.178047	0.187276	0.234025	0.185623
calmar ratio	0.144107	0.155605	0.132228	0.170232	0.173767	0.155412	0.155604	0.167567	-0.000439	0.205471
omega ratio	1.072461	1.071924	1.069931	1.078757	1.079838	1.074533	1.078573	1.085230	1.005406	1.082898
sharpe ratio	0.383187	0.383057	0.364550	0.419853	0.425599	0.393740	0.418848	0.449214	0.029960	0.443089
sortino ratio	0.701354	0.712192	0.675427	0.761012	0.768646	0.723901	0.752189	0.796191	0.166576	0.787736
tail ratio	0.956660	0.986253	0.963665	0.982604	0.979826	0.966802	0.981253	0.952094	0.945345	1.020805
Mean Absolute Deviation	0.008026	0.007467	0.007803	0.007620	0.007657	0.007679	0.007813	0.008149	0.010322	0.008187
Value at Risk	0.018407	0.016645	0.017641	0.016917	0.017005	0.017565	0.017299	0.019007	0.023280	0.017880
Expected Shortfall	0.027249	0.025274	0.027031	0.025777	0.025887	0.026026	0.026617	0.027564	0.033642	0.027429
Worst Scenario	0.084767	0.080702	0.090932	0.075776	0.075364	0.084095	0.076792	0.079488	0.104077	0.087651
maximum drawdown	0.672542	0.564784	0.695997	0.576889	0.573260	0.609420	0.675269	0.700160	0.865234	0.513988

Table 26: Minimum Variance: Metrics Summary.

Metrics	Equal Weights	Baseline	Naive	ES	Theta	KNN	Random Forest	Pipeline	VAR	ESRNN
Information Ratio	0.024139	0.027077	0.026575	0.028043	0.024615	0.030402	0.026768	0.016072	0.020467	0.020467
Annual Return	0.076541	0.082336	0.081174	0.100169	0.081903	0.097852	0.104102	0.051710	0.067777	0.067777
Annual Volatility	0.184897	0.171979	0.173153	0.227678	0.202934	0.189070	0.274245	0.190053	0.205297	0.205297
calmar ratio	0.144107	0.176139	0.183452	0.139588	0.126779	0.186278	0.126168	0.086285	0.104031	0.104031
omega ratio	1.072461	1.081464	1.079214	1.085035	1.074256	1.091121	1.081095	1.047326	1.060796	1.060796
sharpe ratio	0.383187	0.429833	0.421866	0.445164	0.390743	0.482615	0.424936	0.255136	0.324903	0.324903
sortino ratio	0.701354	0.777670	0.766459	0.770521	0.704284	0.841286	0.723544	0.509587	0.593894	0.593894
tail ratio	0.956660	0.975624	0.992897	0.930130	0.949868	0.955227	0.949141	0.938709	0.968197	0.968197
Mean Absolute Deviation	0.008026	0.007481	0.007597	0.009835	0.008772	0.008288	0.011838	0.008313	0.008962	0.008962
Value at Risk	0.018407	0.016592	0.016704	0.023296	0.020174	0.019140	0.027869	0.018567	0.020053	0.020053
Expected Shortfall	0.027249	0.025499	0.025571	0.033236	0.029647	0.027931	0.040052	0.028197	0.031058	0.031058
Worst Scenario	0.084767	0.078951	0.075479	0.090206	0.087253	0.085294	0.095232	0.093339	0.091524	0.091524
maximum drawdown	0.672542	0.556483	0.517727	1.088830	0.924482	0.648139	1.432378	0.817894	0.951323	0.951323

Table 27: Quadratic Utility Algorithm: Metrics Summary.

Metrics	Equal Weights	Baseline	Naive	ES	Theta	KNN	Random Forest	Pipeline	VAR	ESRNN
Information Ratio	0.024139	0.024327	0.023895	0.023925	0.023934	0.023659	0.026682	0.026875	0.023135	0.023830
Annual Return	0.076541	0.075470	0.075544	0.075019	0.075147	0.073511	0.082364	0.086064	0.073828	0.075369
Annual Volatility	0.184897	0.177346	0.183792	0.180881	0.181352	0.177638	0.176457	0.188125	0.187162	0.183925
calmar ratio	0.144107	0.150005	0.143872	0.147284	0.146800	0.142951	0.158883	0.157172	0.134967	0.143516
omega ratio	1.072461	1.072693	1.071396	1.071431	1.071449	1.070797	1.079764	1.081034	1.068850	1.071092
sharpe ratio	0.383187	0.386183	0.379317	0.379798	0.379934	0.375576	0.423566	0.426629	0.367264	0.378285
sortino ratio	0.701354	0.711491	0.697872	0.700411	0.700360	0.695494	0.761038	0.762928	0.679231	0.696567
tail ratio	0.956660	0.963956	0.965901	0.969503	0.970964	0.960376	0.976535	0.954476	0.960286	0.962288
Mean Absolute Deviation	0.008026	0.007735	0.008011	0.007892	0.007913	0.007730	0.007720	0.008161	0.008182	0.008029
Value at Risk	0.018407	0.017645	0.018206	0.017910	0.017950	0.017605	0.017200	0.018924	0.018732	0.018244
Expected Shortfall	0.027249	0.026169	0.027001	0.026617	0.026676	0.026246	0.026339	0.027662	0.027385	0.027001
Worst Scenario	0.084767	0.082632	0.084022	0.083458	0.083477	0.084038	0.077836	0.081957	0.084569	0.083719
maximum drawdown	0.672542	0.620721	0.675262	0.638161	0.644741	0.641247	0.657881	0.718065	0.722340	0.678393

Table 28: Hierarchical Risk Parity: Metrics Summary(1996/03 - 2020/01).

Metrics	Equal Weights	Baseline	Naive	ES	Theta	KNN	Random Forest	Pipeline	VAR	ESRNN
Information Ratio	0.024139	0.026499	0.023366	0.025187	0.025040	0.021564	0.006241	0.026681	-0.015300	0.005082
Annual Return	0.076541	0.080855	0.072311	0.077159	0.076925	0.067206	-0.115933	0.084843	-0.677988	0.015140
Annual Volatility	0.184897	0.172826	0.176165	0.173601	0.174430	0.177755	0.644316	0.185829	1.290741	0.208688
calmar ratio	0.144107	0.176614	0.135316	0.150439	0.149265	0.125445	-0.116236	0.158477	-0.677988	0.023104
omega ratio	1.072461	1.080280	1.070866	1.075932	1.075425	1.064453	1.018634	1.080423	0.958022	1.014447
sharpe ratio	0.383187	0.420658	0.370927	0.399827	0.397502	0.342317	0.099067	0.423555	-0.242876	0.080677
sortino ratio	0.701354	0.763936	0.687992	0.731235	0.727376	0.646280	0.186903	0.759276	-0.324098	0.246649
tail ratio	0.956660	0.982949	0.968031	0.996301	0.993990	0.959688	0.927718	0.952178	1.028376	0.936675
Mean Absolute Deviation	0.008026	0.007462	0.007564	0.007517	0.007558	0.007722	0.027434	0.008063	0.058012	0.009313
Value at Risk	0.018407	0.016421	0.017150	0.016640	0.016792	0.017225	0.065142	0.018568	0.126860	0.020514
Expected Shortfall	0.027249	0.025551	0.026191	0.025803	0.025914	0.026337	0.095858	0.027386	0.184095	0.031201
Worst Scenario	0.084767	0.084520	0.087880	0.082916	0.082992	0.086864	0.284991	0.081856	0.447854	0.103363
maximum drawdown	0.672542	0.527926	0.676167	0.640005	0.644867	0.683009	3.935323	0.682606	7.928845	0.950402

Table 29: Black-Litterman: Metrics Summary(1996/03 - 2020/01).

Metrics	Equal Weights	Baseline	Naive	ES	Theta	KNN	Random Forest	Pipeline	VAR	ESRNN
Information Ratio	0.024139	0.019678	0.023143	0.016937	0.016796	0.027550	0.027550	0.023058	0.027550	0.027550
Annual Return	0.076541	0.062337	0.074505	0.054437	0.054107	0.087913	0.087913	0.074016	0.087913	0.087913
Annual Volatility	0.184897	0.183315	0.190552	0.189374	0.190787	0.186930	0.186930	0.189448	0.186930	0.186930
calmar ratio	0.144107	0.116797	0.127676	0.096649	0.094650	0.162341	0.162341	0.131753	0.162341	0.162341
omega ratio	1.072461	1.059034	1.069906	1.050124	1.049634	1.084906	1.084906	1.069032	1.084906	1.084906
sharpe ratio	0.383187	0.312381	0.367390	0.268863	0.266628	0.437347	0.437347	0.366032	0.437347	0.437347
sortino ratio	0.701354	0.602651	0.677251	0.537175	0.533168	0.775747	0.775747	0.675411	0.775747	0.775747
tail ratio	0.956660	0.942749	0.939034	0.953227	0.948716	0.943793	0.943793	0.944522	0.943793	0.943793
Mean Absolute Deviation	0.008026	0.007916	0.008213	0.008257	0.008329	0.007951	0.007951	0.008234	0.007951	0.007951
Value at Risk	0.018407	0.017962	0.018940	0.018474	0.018702	0.018486	0.018486	0.019003	0.018486	0.018486
Expected Shortfall	0.027249	0.026814	0.027877	0.027462	0.027643	0.027787	0.027787	0.027726	0.027787	0.027787
Worst Scenario	0.084767	0.093212	0.089883	0.094373	0.094309	0.087918	0.087918	0.086604	0.087918	0.087918
maximum drawdown	0.672542	0.697186	0.792947	0.759480	0.776437	0.680989	0.680989	0.745620	0.680989	0.680989

Table 30: Critical Line Algorithm (Maximum Sharpe Ratio): Metrics Summary.

Metrics	Equal Weights	Baseline	Naive	ES	Theta	KNN	Random Forest	Pipeline	VAR	ESRNN
Information Ratio	0.024139	0.023627	0.023932	0.024319	0.024115	0.022365	0.016332	0.023058	0.027550	0.027550
Annual Return	0.076541	0.076325	0.076259	0.076729	0.076017	0.071816	0.052384	0.074016	0.087913	0.087913
Annual Volatility	0.184897	0.191798	0.186644	0.183127	0.182782	0.189595	0.187540	0.189448	0.186930	0.186930
calmar ratio	0.144107	0.129288	0.137400	0.144328	0.143403	0.124295	0.095966	0.131753	0.162341	0.162341
omega ratio	1.072461	1.071161	1.072784	1.073822	1.073235	1.066818	1.048330	1.069032	1.084906	1.084906
sharpe ratio	0.383187	0.375059	0.379913	0.386053	0.382818	0.355041	0.259263	0.366032	0.437347	0.437347
sortino ratio	0.701354	0.687926	0.696754	0.707940	0.703420	0.661039	0.524394	0.675411	0.775747	0.775747
tail ratio	0.956660	0.941279	0.947008	0.947636	0.948386	0.947439	0.946197	0.944522	0.943793	0.943793
Mean Absolute Deviation	0.008026	0.008295	0.007999	0.007868	0.007848	0.008249	0.008170	0.008234	0.007951	0.007951
Value at Risk	0.018407	0.019172	0.018435	0.018157	0.018063	0.019023	0.018186	0.019003	0.018486	0.018486
Expected Shortfall	0.027249	0.028017	0.027429	0.026914	0.026876	0.027636	0.027204	0.027726	0.027787	0.027787
Worst Scenario	0.084767	0.088306	0.089944	0.088387	0.088813	0.088148	0.095046	0.086604	0.087918	0.087918
maximum drawdown	0.672542	0.805449	0.735077	0.688514	0.684213	0.781399	0.725576	0.745620	0.680989	0.680989

Table 31: Critical Line Algorithm (Minimum Volatility): Metrics Summary.

VIII Reference

[1] Babaitsev, V. A., et al.(2012). *Fast Algorithm for the Markowitz Critical Line Method*, Mathematical Models and Computer Simulations, vol. 4, no. 2, 2012, pp. 241–250. Retrieved from: doi:10.1134/s2070048212020020.

[2] Bianchi, Daniele, et al.(2018) *Bond Risk Premia with Machine Learning*. SSRN Electronic Journal, 2018. Retrieved from: doi:10.2139/ssrn.3232721.

[3] *Black-Litterman Allocation*, Black-Litterman Allocation - PyPortfolioOpt 1.2.3 Documentation. Retrieved from: `http://pyportfolioopt.readthedocs.io/en/latest/BlackLitterman.html.`

[4] Botchkarev, A.(2019) *A new typology design of performance metrics to measure errors in machine learning regression algorithms*. Retrieved from: Interdisciplinary Journal of Information, Knowledge, and Management 14, pp. 045– 076, 2019.

[5] Brogaard, Jonathan, and Abalfazl Zareei.(2018). *Machine Learning and the Stock Market*. SSRN Electronic Journal, 2018. Retrieved from:doi:10.2139/ssrn.3233119.

[6] Bryzgalova, Svetlana, et al.(2019). *Bayesian solutions for the factor zoo: models*. SSRN Electronic Journal, 2019a. Retrieved from: doi:10.2139/ssrn.3481736.

[7] Bryzgalova, Svetlana, et al.(2019). *Forest Through the Trees: Building Cross-Sections of Stock Returns*.SSRN Electronic Journal, 2019b. Retrieved from: doi:10.2139/ssrn.3493458.

[8] Campbell R., Harvey, and Liu, Yan.(2019).*Lucky Factors*.SSRN Electronic Journal, 2019. Retrieved from:doi:10.2139/ssrn.2528780.

[9] Chen, Luyang, et al. (2019). *Deep learning in asset pricing*.SSRN Electronic Journal, 2019. Retrieved from: doi:10.2139/ssrn.3350138.

[10] Chiu, Ching-Wai (Jeremy), et al. *A new approach for detecting shifts in forecast accuracy*. SSRN Electronic Journal, 2019. Retrieved from:doi:10.2139/ssrn.3163414.

[11] Cheng, Mingmian, et al. (2019). *Forecast Evaluation*.

[12] Choi, Darwin, et al. (2019). *Alpha go everywhere: machine learning and international stock returns*. SSRN Electronic Journal, 2019. Retrieved from: doi:10.2139/ssrn.3489679.

[13] Choueifaty, Yves, and Yves Coignard (2008). *Toward Maximum Diversification*. Journal of Portfolio Management - J PORTFOLIO MANAGE, vol.35, pp. 40-51. Retrieved from: doi: 10.3905/JPM.2008.35.1.40.

[14] Fan, Jianqing, et al.(2017). *Sufficient forecasting using factor models*. SSRN Electronic Journal, 2014. Retrieved from: doi:10.2139/ssrn.2607666.

[15] Feng, Guanhao, et al.(2019). *Taming the Factor Zoo: A Test of New Factors.*SSRN Electronic Journal, 2019a. Retrieved from:doi:10.3386/w25481.

[16] Feng, Guanhao, et al.(2018). *Deep Learning for Predicting Asset Returns.* Retrieved from: arXiv Electronic Journal, 2018a.

[17] Feng, Guanhao, et al.(2018) *Deep learning factor alpha.* SSRN Electronic Journal, 2018b. Retrieved from:doi:10.2139/ssrn.3243683.

[18] Feng, Guanhao, et al.(2019). *Deep Learning in Characteristics-Sorted Factor Models.* SSRN Electronic Journal, 2019b. Retrieved from:doi:10.2139/ssrn.3243683.

[19] Franses, Philip Hans.(2016). *A Note on the Mean Absolute Scaled Error.* International Journal of Forecasting, vol. 32, no. 1, 2016, pp. 20–22. Retrieved from: doi:10.1016/j.ijforecast.2015.03.008.

[20] Freyberger, Joachim, et al.(2017). *Dissecting Characteristics Nonparametrically.* The Review of Financial Studies, Volume 33, Issue 5, May 2020, Pages 2326–2377. Retrieved from:doi:10.3386/w23227.

[21] Gasthaus, Jan, et al.(2019). *Probabilistic Forecasting with Spline Quantile Function RNNs.* Proceedings of Machine Learning Research 89, pp. 1901–1910, 2019.

[22] Gibbs, Christopher G.(2017). *Conditionally Optimal Weights and Forward-Looking Approaches to Combining Forecasts.* SSRN Electronic Journal. 2017. Retrieved from: doi:10.2139/ssrn.2919117.

[23] Goodwin, Paul.(2014). *Using naïve forecasts to assess limits to forecast accuracy and the quality of fit of forecasts to time series data.* SSRN Electronic, 2014.Retrieved from: doi:10.13140/2.1.1979.2324.

[24] Granziera, Eleonora, and Tatevik Sekhposyan.(2018). *Predicting Relative Forecasting Performance: An Empirical Investigation.*SSRN Electronic Journal, 2018. Retrieved from:doi:10.2139/ssrn.3280994.

[25] Gu, Shihao, et al.(2019). *Autoencoder asset pricing models.* SSRN Electronic Journal, 2019a. Retrieved from:doi:10.2139/ssrn.3335536.

[26] Gu, Shihao, et al.(2019). *Empirical asset pricing via machine learning.*SSRN Electronic Journal, 2019b. Retrieved from:doi:10.3386/w25398.

[27] Holt, Charles C.(2003).*Forecasting Seasonals and Trends by Exponentially Weighted Moving Averages.* International Journal of Forecasting, vol. 20, no. 1, 2004, pp. 5–10. Retrieved from:doi:10.1016/j.ijforecast.2003.09.015.

[28] Hsiao, Cheng, and Shui Ki Wan.(2014). *Is There an Optimal Forecast Combination?.* Journal of Econometrics, vol. 178, 2014, pp. 294–309.Retrieved from:doi:10.1016/j.jeconom.2013.11.003.

[29] Hyndman, Rob J., and Anne B. Koehler.(2006). *Another look at measures of forecast accuracy*In: International Journal of Forecasting 22(4), pp. 679–688, 2006. Retrieved from:doi:10.1016/j.ijforecast.2006.03.001.

[30] Hyndman, Rob, Billah, Baki.(2003). *Unmasking the Theta method*. International Journal of Forecasting, vol.19, pp. 287-290. Retrieved from: doi:10.1016/S0169-2070(01)00143-1.

[31] Hyndman, Rob, and George Athanasopoulos. *Principles and Practice.*,OTexts, 2018.

[32] Jin, Sainan, et al.(2005). *Robust Forecast Comparison*. SSRN Electronic Journal, 2015. Retrieved from:doi:10.2139/ssrn.2605927.

[33] Li, Yimou, et al.(2020). *Beyond the Black Box: An Intuitive Approach to Investment Prediction with Machine Learning.*The Journal of Financial Data Science, 2020.

[34] Liew, Jim Kyung-Soo, and Boris Mayster.(2017). *Forecasting ETFs with Machine Learning Algorithms*. SSRN Electronic Journal, 2017. Retrieved from:doi:10.2139/ssrn.2899520.

[35] Kim, Hyeongwoo, and Nazif Durmaz.(2012). *Bias correction and out-of-sample forecast accuracy*. International Journal of Forecasting, vol. 28, no. 3, 2012, pp. 575–586. Retrieved from: doi:10.1016/j.ijforecast.2012.02.009.

[36] Kok, S.(2017). *The Quest for a Better Forecast Error Metric: Measuring More than the Average Error*.

[37] Kozak, Serhiy, et al.(2017). *Shrinking the Cross Section*. SSRN Electronic Journal, 2017.Retrieved from:doi:10.2139/ssrn2945663.

[38] Makridakis, et al.(2018). *The M4 Competition: Results, findings, conclusion and way forward*. International Journal of Forecasting. Retrieved From:doi: 10.1016/j.ijforecast.2018.06.001.

[39] Makridakis, Spyros, et al.(2020). *The M4 Competition: 100,000 time series and 61 forecasting methods.*International Journal of Forecasting, vol. 36, no. 1, 2020, pp. 54–74. Retrieved from:doi:10.1016/j.ijforecast.2019.04.014.

[40] Markus Loning, Franz J. Kiraly.(2005).*Forecasting with Sktime: Designing Sktime's New Forecasting API and applying it to Replicate and extend the M4 study*. University College London. Retrieved from : `https://arxiv.org/abs/2005.08067`

[41] Markus Loning, et al.*sktime: A Unified Interface for Machine Learning with Time Series*. The Alan Turing Institute. Retrieved from : `https://github.com/alan-turing-institute/sktime`

[42] Messmer, Marcial, and Francesco Audrino.(2017). *The (Adaptive) Lasso in the Zoo - Firm Characteristic Selection in the Cross-Section of Expected Returns*. SSRN Electronic Journal, 2017. Retrieved from: doi:10.2139/ssrn.2930436.

[43] Messmer, Marcial.(2017). *Deep Learning and the Cross-Section of Expected Returns.* SSRN Electronic Journal, 2017. Retrieved from:doi:10.2139/ssrn.3081555.

[44] Niedermayer, Andras, and Daniel Niedermayer.(2010). *Applying Markowitz's Critical Line Algorithm.*. Handbook of Portfolio Construction, 2010, pp. 383–400. Retrieved from: doi:10.1007/978-0-387-77439-8_12.

[45] *Other Optimisers.*, Other Optimisers - PyPortfolioOpt 1.2.3 documentation. Accessed August 19, 2020. Retrieved from: `https://pyportfolioopt.readthedocs.io/en/latest/OtherOptimisers.html.`

[46] Petropoulos, Fotios, et al.(2019). *The Inventory Performance of Forecasting Methods: Evidence from the M3 Competition Data*. International Journal of Forecasting, vol. 35, no. 1, 2019, pp. 251–265. Retrieved from:doi:10.1016/j.ijforecast.2018.01.004.

[47] Popescu, Andreea Victoria.(2019). *The Macroeconomy and the Cross-Section of International Equity Index Returns: A Machine Learning Approach*. SSRN Electronic Journal, 2019. Retrieved from:doi:10.2139/ssrn.3480042.

[48] Rapach, David E., et al.(2019). *Industry Return Predictability: A Machine Learning Approach*. The Journal of Financial Data Science, 2019. Retrieved from:doi:10.3905/jfds.2019.1.3.009.

[49] Ryll, Lukas, and Sebastian Seidens.(2019).*Evaluating the Performance of Machine Learning Algorithms in Financial Market Forecasting: A Comprehensive Survey.* ArXiv.org, 6 July 2019. Retrieved from:arxiv.org/abs/1906.07786.

[50] Samuels, Jon D., and Rodrigo M. Sekkel.(2017). *Model Confidence Sets and Forecast Combination*. International Journal of Forecasting, vol. 33, no. 1, 2017, pp. 48–60.Retrieved from:doi:10.1016/j.ijforecast.2016.07.004.

[51] Siliverstovs, Boriss.(2017). *Dissecting Models' Forecasting Performance*. Economic Modelling, vol. 67, 2017, pp. 294–299. Retrieved from:doi:10.1016/j.econmod.2017.01.008.

[52] Simonian, Joseph, et al.(2019).*A Machine Learning Approach to Risk Factors: A Case Study Using the Fama–French–Carhart Model*. The Journal of Financial Data Science, 2019. Retrieved from:doi:10.3905/jfds.2019.1.032.

[53] Smyl, Slawek.(2019).*A Hybrid Method of Exponential Smoothing and Recurrent Neural Networks for Time Series Forecasting.* ,International Journal of Forecasting, vol. 36, no. 1, 2020, pp. 75–85. Retrieved from : doi:10.1016/j.ijforecast.2019.03.017.

[54] Sun, Ying, et al.(2019). *Short-Timescale Gravitational Microlensing Events Prediction with ARIMA-LSTM and ARIMA-GRU Hybrid Model.*, Big Scientific Data Management Lecture Notes in Computer Science, 2019, pp. 224–238. Retrieved from: doi:10.1007/978-3-030-28061-1_ 23.

[55] Viet, D Le. *JPM-Big Data and AI Strategies Machine Learning and Alternative Data Approach to Investing*. Medium, Zungybungy, 10 May 2019.

[56] Weigand, A.(2019). *Machine learning in empirical asset pricing.* . Financ Mark Portf Manag 33, 93–104,2019. Retrieved from: https://doi.org/10.1007/s11408-019-00326-3

[57] Xu, Xiaoliang, et al.(2008).*A Fast Algorithm for Solving Large Scale Nonlinear Optimization Problems Using RNN.*, 2008 IEEE Conference on Cybernetics and Intelligent Systems, 2008. Retrieved from : doi:10.1109/iccis.2008.4670798.