SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Portfolio Optimization Using Multi-Fidelity Gaussian Process

Chengye Zhang

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Portfolio Optimization Using Multi-Fidelity Gaussian Process

Portfolio Optimierung mithilfe des Multi-Fidelity Gauß-Prozesses

Author:	Chengye Zhang
Supervisor:	Prof. Hans Bungartz
Advisor:	Kislaya Ravi
Submission Date:	12.01.2024

I confirm that this master’s thesis is my own work and I have documented all sources and material used.

Munich, 12.01.2024 Chengye Zhang

Acknowledgments

Abstract

This work is about data fusion and multi-fidelity data in finance. Data fusion has recently been widely used in finance data analyses. Portfolio optimization, a special multi-objective optimization problem, is rebalancing asset allocation. This work aims to improve portfolio optimization using multi-fidelity data and data fusion methods to generate a new portfolio, which can bring higher returns and a better Sharpe ratio. Our work is the first attempt to apply multi-fidelity for portfolio optimization. We define fidelity in portfolio optimization. We propose two different multi-fidelity models in portfolio optimization. One is the sampling interval based multi-fidelity model. The other one is the prediction based multi-fidelity model. According to the experiment results, the sampling interval based multi-fidelity model is feasible. We can generate a fused long-term portfolio from short-term and long-term portfolios in the sampling interval based multi-fidelity model. This fused long-term portfolio can outperform long-term portfolios. Moreover, this model can refine the Sharpe ratio and the return by a binary prediction of local optimal γ^*_{short}.

Contents

1 Introduction

This work explores using multi-fidelity data fusion and Gaussian processes in portfolio optimization. Data fusion has recently been widely used in finance data analyses[42][15][43] [38]. Mainly, Gaussian processes and Bayesian optimization have been successful in finance analysis [14][28]. This work wants to apply Gaussian processes and multi-fidelity data for new financial areas and portfolio optimization. Portfolio optimization,a special multi-objective optimization problem,[25] rebalances asset allocation based on multi-objectives. Trend-following strategy, a traditional algorithm to rebalance asset allocation, must manually adjust some hyperparameters[8][14]. These suitable hyperparameters can be selected faster using Bayesian optimization with a squared exponential kernel in hyperparameter space. With these optimized hyperparameters, a trend-following strategy can generate some outstanding portfolios.

This work aims to improve the performance of portfolio optimization to generate a better portfolio, which can bring a better return and higher Sharpe ratio. [39] Before our work, Tseng and Allen used multi-fidelity methods in financial engineering. In their research, they try to solve the problem that information about alternative surrogate systems' accuracy or 'fidelity' may be ambiguous and difficult to determine in practical applications. In our work, we need to solve a similar problem: What is the accuracy of portfolio optimization? Our work is the first trial to apply multi-fidelity for portfolio optimization. We need to define the fidelity of portfolio optimization. Traditionally, the fidelity is equivalent to the accuracy. A high-fidelity model can generate a more exact solution than a low-fidelity model. However, in portfolio optimization, the definition of fidelity needs to be clarified because accuracy is hard to measure.

Based on different definitions of fidelity, we propose two different multi-fidelity models in portfolio optimization. One is the sampling interval based multi-fidelity model. The other one is the prediction based multi-fidelity model. According to some tests, we found that the prediction based multi-fidelity model needs to be revised. Fortunately, the sampling interval based multi-fidelity model is feasible according to the experiment results.

Our work goal is to improve the return and Sharpe ratio of the portfolio. The sampling interval-based multi-fidelity model can fuse a short-term portfolio into a long-term portfolio to generate a fused long-term portfolio. The fused long-term portfolio can outperform the long-term portfolio. Moreover, γ_{short}, a hyperparameter in the sampling interval based multi-fidelity model, can affect the Sharpe ratio and the return of the fused long-term portfolio. The local optimal γ_{short} distribution is similar to a two-point distribution.

The main contribution of this work is to propose a new model based on the different sample intervals and fusion algorithms for portfolio optimization. We observe that it improves the Sharpe ratio and increases long-term portfolio returns. Moreover, this model can refine the Sharpe ratio and the return by a binary prediction of local optimal γ_{short}.

1.1 Outline

Chapter 2 covers the theoretical background behind portfolio optimization, multi-fidelity, and Gaussian process. We mention financial data characteristics, assets, portfolios, and how portfolio optimization can work. The reasons for the difficulty of portfolio optimization and the performance benchmark in our work are also introduced. At the same time, we also introduce multi-fidelity models, Gaussian Processes, and Gaussian Processes Regression. In Chapter 3, we will describe two process models. One is process models of portfolio optimization. One is process models of multi-fidelity methods. Chapter 4 introduces two different multi-fidelity portfolio optimization models in this work. One is the prediction based multi-fidelity model. The other one is the sampling interval based multi-fidelity model. We introduce their design principles and trade-offs one by one. Chapter 5 covers the implementation of our work. We post a UML class diagram and a package diagram. Chapter 6 records the experiment results of the sampling interval based multi-fidelity model. Six different experiments are designed and conducted to prove that our model can improve the long-term portfolio's return and Sharpe ratio. Finally, we conclude our work and plan for future work in Chapter 7.

2 Theoretical Background

This section recalls basic concepts about assets, portfolios, portfolio optimization problems, and multi-fidelity.

2.1 Financial Data Characteristics

Portfolio optimization[25][27] refers to the optimal combination of financial assets. The target of portfolio optimization is financial assets and the corresponding financial data. With these financial data come new challenges associated with their analysis. Financial data sets may contain thousands of transactions or posted quotes in a single day, stamped to the nearest second. The sizes of these financial data sets are huge, but the valuable information could be more extensive. The analysis based on these financial data is a challenging task because it is complicated due to the low signal-to-noise ratio and non-stationary time series.

2.1.1 Low Signal-to-Noise Ratio

Financial markets are highly uncertain, and the signal/information hidden behind financial data is weak compared with the amount of noise[22]. The signal-to-noise ratio (SNR) is the ratio between desired information and undesired information[1]. It is a measurement to compare the level of a desired signal to the level of background noise in engineering and science. Different SNRs can reflect different abilities of the data source. One prevalent assumption is that the signal-to-noise ratio can partly indicate the prediction skill, how much predictability exists within a system [2][9]. The other assumption is that we need more data to validate a signal for the signal prediction when the signal-to-noise ratio is very low.[21]. The low signal-to-noise property may lead to severe overfitting issues and inaccurate prediction of future financial data. According to [22], the signal-to-noise ratio of the financial data is very low.

Many researchers expressed skepticism about whether the current models can sufficiently distinguish between signal and noise. Some real-world examples reinforce this sentiment. Many machine learning models about financial data mistook noise for signal and launched strategies that looked good on backtests but failed post-launch[22].

2.1.2 Non-Stationary Time-Series Data

Most financial data is non-stationary time-series data [37]. In most research [20], it is assumed that financial data is stationary time series data. Using non-stationary time series data in financial models can produce unreliable and spurious results. The solution to this problem is to do transformations and adjustments of the time series data so that it becomes stationary[20]. It can be transformed into a stationary process by differencing [20] to compute the differences between consecutive observations when the non-stationary process is a random walk with or without a drift. On the other hand, if the time series data analyzed exhibits a deterministic trend, detrending can avoid spurious results [6].

2.2 Asset

In portfolio optimization, the target is optimal asset allocation. For example, how much asset should we buy/sell at a particular time? Before making an asset allocation, the corresponding financial data should be analyzed. In this work, we focus on the stock in the secondary market. The asset in our work is the stock in the secondary market. Hence, our financial data is the stock price in the secondary market. Before introducing the portfolio, we need to define asset price and return.

2.2.1 Asset Price

In this work, our asset is the stock in the secondary market. Hence, the asset prices are stock prices in the secondary market. In the secondary market, the participator can purchase the stock at a target price using an order. Each order is assigned a price. In real-world situations, the number of orders is million upon billion for each stock daily. It is necessary to choose one of them to represent the asset value.

Without loss of generality, the stock price can be represented by the closing price, the raw price of the last transacted price before the market officially closes for normal trading. In other words, the closing price is from the last stock transaction every day. Closing price is a trendy choice by many researchers to represent the stock price.

2.2.2 Asset Return

The immediate research data in portfolio optimizations is not asset price but asset return. The asset return is [34]:

$$asset\ return = \frac{Change\ in\ value\ of\ the\ asset\ +\ accumulated\ cash\ flows}{Original\ value\ of\ the\ asset} \tag{2.1}$$

The change in the asset's value is the difference between the sales price and the purchase price, while the asset's original value is the purchase price. In our work, all stock prices are the adjusted closing price. Hence, dividend payments, new offerings, and stock splits can be ignored. The accumulated cash flows are zero.

2.2.3 Expected Asset Return

The expected asset return is the expected value of the asset asset return. The expected asset return represents the return brought by the asset. In portfolio optimization, the classical choice of the expected asset return calculation is the historical average returns, which can calculated by a simple moving average.

Simple Moving Average (SMA)

A simple moving average (SMA) in portfolio optimization is the unweighted mean of the previous k data points, where k is the rolling window size. Let us consider data-points $R_1, R_2 ... R_n$. These data points are the historical returns. We denote the mean over the last k data points as SMA_k. It is calculated as

$$SMA_k = \frac{R_{n-k+1} + R_{n-k+2} + ... + R_n}{k} = \frac{1}{k} \sum_{i=n-k+1}^{n} R_i \tag{2.2}$$

2.3 Portfolio

A portfolio is a collection of investments constructed and held by an institution or a private individual for diversification[25].

2.3.1 Definition of Portfolio

A portfolio is a combination of n assets. Let $x \in \mathbb{R}^n$ and $x = (x_1, x_2, ..., x_n)$ be the allocation vector of weights in the portfolio . And we assume that portfolio is fully invested : $x_1 + x_2 + ... + x_n = 1$.

2.3.2 Portfolio Return

Let $R = (R_1, R_2, ..., R_n)$ denote the vector of asset returns. The portfolio return $R(x)$ is then equal to

$$R(x) = \sum_{i=1}^{n} x_i R_i = x^{\mathrm{T}} R \tag{2.3}$$

2.3.3 Expected Portfolio Return

Let $\mu = \mathbb{E}[R]$ be the vector of expected returns of asset returns.

The expected portfolio return [35] can be :

$$\mu(x) = \mathbb{E}[R(x)] = \mathbb{E}[x^{\mathrm{T}}R] = x^{\mathrm{T}}\mathbb{E}[R] = x^{\mathrm{T}}\mu \tag{2.4}$$

2.3.4 Portfolio Variance (Portfolio Volatility)

Portfolio variance or portfolio volatility is a portfolio risk measurement by a measure of uncertainty[18][35]. Portfolio variance can be regarded as the covariance or correlation coefficients for the assets in a portfolio. Generally, a lower correlation between assets in a portfolio results in a lower portfolio variance. The portfolio variance statistic is calculated using the standard deviations of each asset in the portfolio as well as the correlations of each asset pair in the portfolio.

Let us consider an asset i with returns R_{it}, where t indexes time. The variance of this asset's returns σ_i^2 is :

$$\sigma_i^2 = \mathbb{E}[(R_i - \mu_i)^2] = (1/T)(\sum_{T}^{t=1}(R_{it} - \mu_i)^2) \tag{2.5}$$

where μ_i denotes the mean return of asset i and T the number of observations.

The covariance between asset i and asset j , σ_{ij} is given by :

$$\sigma_{ij} = E[(R_i - \mu_i)^2] = (1/T)(\sum_{T}^{t=1}(R_{it} - \mu_i)(R_{jt} - \mu_j)) \tag{2.6}$$

Let $\Sigma = \mathbb{E}[(R-\mu)(R-\mu)^{\mathrm{T}}]$ be the vector of the covariance matrix of asset returns. The portfolio variance $\sigma^2(x)$ is

$$\sigma^2(x) = x^T\Sigma x \tag{2.7}$$

where Σ is a $n \times n$ matrix of covariances ,

$$\Sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \dots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \dots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \dots & \sigma_{nn} \end{pmatrix} \tag{2.8}$$

2.3.5 Goal of Portfolio

The portfolio is an asset allocation. When determining an asset allocation, the corresponding goal is to maximize the expected portfolio return and minimize the portfolio risk. A portfolio is preferred to another if it has a higher expected return and lower variance. However, how can we find a portfolio that can maximize the expected portfolio return and minimize the portfolio risk? The best way is portfolio optimization.

2.4 Portfolio Optimization

Portfolio optimization is the process of rebalancing asset allocation based on multi-objectives[25][27]. It can be treated as a particular multi-objective optimization problem. Traditionally, portfolio optimization is going to select different assets to invest to maximize the overall return and minimize the overall risk simultaneously. The selection of different assets is based on the mean-variance optimization framework(mean-variance analysis).

2.4.1 Goal of Portfolio Optimization

Portfolio optimization is to select the best asset allocation according to some special objective[25][27]. The result of portfolio optimization is an optimal portfolio. As introduced in the section, a better portfolio is a portfolio that can bring a better return and lower risk. The actual return and the actual risk are generally unobservable and unknown, The expected return and the expected risk can be measured. Portfolio optimization is to select an optimal allocation that can maximize the expected return and minimize the expected risk simultaneously.

In a typical portfolio optimization (equation 2.9 and equation 2.10), We want to maximize the expected return of the portfolio $u(x)$ under some constraints or minimize the portfolio volatility $\sigma(x)$ under some constraints. The portfolio volatility $\sigma(x)$ represents the risk.

$$Max\ u(x)\ s.t.\ constraints \tag{2.9}$$

$$Min\ \sigma(x)\ s.t.\ constraints \tag{2.10}$$

2.4.2 Modern Portfolio Theory (Mean-Variance Analysis)

This section introduces the modern portfolio theory, or mean-variance analysis, and discusses its different formulations.

Mean-variance analysis is an approach to the optimal portfolio choice problem introduced by Harry Markowitz in 1950[25][27]. According to Markowitz's suggestion, the portfolio choice should be made concerning two criteria: the expected portfolio return and the variance of the portfolio return. A portfolio is preferred to another if it has a higher expected return and lower variance. Mean-variance analysis can be regarded as an optimization problem and geometric interpretation of the trade-off between the expected return and variance. (a mean-variance optimization framework that can assist in selecting the most efficient portfolio by analyzing various possible portfolios of the given assets.) In this work, we consider long position (long position means that $x_i >= 0$) and full allocation $\sum_{i=1}^{n} x_i = \mathbf{1}^T x = 1$ in this work.

The portfolio choice problem is typically treated as a one-period problem in classical portfolio optimization. The main principle of mean-variance analysis consists of two different ways:

- find the portfolio with the minimum variance from all feasible portfolios with a lower bound on the expected performance, like equation 2.10.
- find the portfolio with maximum expected performance from all feasible portfolios with a given upper bound on the variance of portfolio return, like equation 2.9.

In this work, we focus on the second principle, to maximize the expected performance with a given upper bound on the variance of portfolio return. The corresponding optimization problem of equation 2.9 is:

$$\begin{aligned} \max_{x} \quad & x^T \mu \\ \text{s.t.} \quad & x^T \Sigma x \leq R^\star \\ & x^T e = 1 \\ & x \geq 0 \end{aligned} \tag{2.11}$$

Where :

- $R^\star$ is the upper bound on the portfolio variance.
- $x \geq 0$ means that all components of the vector are non-negative, $x_i \geq 0$, $i = 1, ..., n$.

The corresponding optimization problem of equation 2.10 is:

$$\begin{aligned} \min_{x} \quad & x^T \Sigma x \\ \text{s.t.} \quad & x^T \mu \geq R_\star \\ & x^T e = 1 \\ & x \geq 0 \end{aligned} \tag{2.12}$$

Where :

- $R_\star$ is the lower bound on the expected performance (the expected portfolio return).
- $x \geq 0$ means that all components of the vector are non-negative, $x_i \geq 0$, $i = 1, ..., n$.

2.4.3 Regularized Markowitz Portfolio Optimization

The idea of using penalization comes from the regularization problem of linear regressions. Penalization has mainly been used in forecasting to improve out-of-sample forecasting [44]. Markowitz portfolio optimization is related to linear regression [36], and regularizations may improve the performance of Markowitz portfolio optimization.

L2 Constrained Portfolio

The L2 regularization method is one of the most famous regularization procedures. Using L2 regularization, we could prevent overfitting by penalizing complex models. In portfolio optimization, overfitting can cause an extremely imbalanced portfolio, which contains many negligible weights[13]. In order to reduce the number of negligible weights, we can add the L2 regularization to the objective function in portfolio optimization [13][8].

In our work, the L2 regularization is written as follows:

$$-\gamma x^T x \tag{2.13}$$

Where :

- γ is L2 regularization parameter.

When this L2 regularization parameter is a maximum value, all weights of the portfolio are equally distributed. Furthermore, when this L2 regularization parameter is a minimum value, the entire portfolio is allocated to one asset. Hence, it can reduce the amount of negligible weight.

2.4.4 How to Solve Optimization Problem in Mean-Variance Analysis?

Mathematical optimization is a very difficult problem regarding complex objectives and constraints. Fortunately, mean-variance analysis with standard objectives and constraints, like equation 2.11 and equation 2.12, are convex optimization problems that are a well-understood class of problems[13]. The task of how to solve a convex optimization problem is done directly by CVXPY. [41]

2.5 Portfolio's Performance Benchmark

The goal of portfolio optimization is to generate a good portfolio. A good portfolio can bring a very high actual return and a low actual risk. Therefore, portfolio optimization aims to maximize the return and minimize the variance. This section introduces some relevant benchmarks for comparing a portfolio's performance .[3] Traditionally, there are three different types of benchmarks in portfolio performance evaluation: absolute return measures, risk-adjusted return measures, and performance attribution. In our work, we focus on absolute return measures and risk-adjusted return measures.

2.5.1 Average Return

To evaluate the portfolio, this section introduces our first benchmark, average return [35]. In real situations, the return on a portfolio must be measured when a portfolio needs to be rebalanced over time. If we want to see the portfolio's performance every period, average return is the best choice. Let us consider the portfolio return at time t is $R_t(x)$. The average return of the portfolios x between time 1 to time t is equal to

$$R_{AVG} = (\sum_{i=1}^{t} R_i(x))/t \tag{2.14}$$

Where :

- x is the portfolio.
- $R_i(x)$ is portfolio return at time i
- R_{AVG} is the average return between time 1 to time t

2.5.2 Cumulative Return

This section introduces our second benchmark, cumulative return[35]. If we want the performance of portfolio total periods, cumulative return is more suitable. Let us consider the portfolio return at time t is $R_t(x)$. The cumulative return of the portfolios x between time 1 to time t is equal to

$$R_{CUM} = (\prod_{i=1}^{t}(R_i(x) + 1)) \tag{2.15}$$

Where :

- x is the portfolio.

- $R_i(x)$ is portfolio return at time i
- R_{CUM} is the average return between time 1 to time t

2.5.3 Sharpe Ratio

Sharpe ratio, or reward-to-variability ratio, is one of the risk performance measures that play an essential role in the evaluation of the risk performance of a portfolio. A higher positive Sharpe ratio represents that an asset can bring higher returns or lower volatility. It considers not only risk parameters but also return. Sharpe ratio is stated as follows.

$$S_a = \frac{R(x) - R_{rf}}{\sigma(x)} \tag{2.16}$$

Where :

- $R(x)$ is the return of the portfolio
- R_{rf} is the risk-free rate
- $\sigma(x)$ is the standard deviation of the portfolio

2.6 Time Frame in the Application of Portfolio Optimization

The time frame, or investment time horizon, about how frequently to rebalance the portfolio is significant in applying portfolio optimization. For example, the portfolio must be rebalanced every day when the time frame is one day. The portfolio needs to be rebalanced every month when the time frame is one month. The portfolio optimization applications with different time frames can be regarded as different fidelity models.

Trend-following strategy can generate different portfolios using different time frames. A short-term portfolio, which is generated using a short-term time frame, focuses on the short-term asset return. The short-term asset price always brings more uncertainty because asset prices may be affected by some jamming in a short time. The asset price could come back to a fair value after the jamming. A long-term portfolio, which is generated using a long-term time frame, can overcome the jamming and bring a more stable return. Hence, the long-term portfolio can be more stable than the short-term portfolio. In other words, a long-term portfolio does not need to refresh (rebalance) itself frequently. However, sometimes, the performance of the long-term portfolio could be better than the performance of the short-term portfolio.

2.7 Multi-Fidelity

Measuring real-world data is fundamental for decision-making in many scientific fields[30]. Although data can always be generated with a certain quality and quantity in a computer simulation, it still needs to trade between working with more precise but huge-cost data or inaccurate but low-cost data in some complex competition. In this section, we introduce a formalization of multi-fidelity, which we will use in the following sections.

2.7.1 Multi-Fidelity Models

In this section, we introduce and formalize multi-fidelity models[30].

Let us consider a function $f : Z \to Y$ mapping an input $z \in Z \subseteq \mathbb{R}^d$ to an output $y \in Y \subseteq \mathbb{R}$. The evaluating cost of function f is a certain workload $c \in \mathbb{R}^+$. A high-fidelity model is the function $f_{hi} : Z \to Y$ with a evaluation workload $c_{hi} \in \mathbb{R}^+$. A low-fidelity model is the $f_{lo}^{(i)} : Z \to Y$ with a evaluation workload $c_{lo}^{(i)} \in \mathbb{R}^+$. The evaluating cost of high-fidelity model f_{hi} is more expensive than the evaluating cost of $f_{lo}^{(i)}$. The high-fidelity model f_{hi} and the low-fidelity model $f_{lo}^{(i)}$ are the approximations of the underlying function f_{exact}. The high-fidelity model f_{hi} is a better approximation of the underlying function f_{exact} than all low-fidelity model $f_{lo}^{(i)}$. In theory , we consider a transformation function $h_{trans-lo}^{(i)}$, which can mapping the underlying function f_{exact} into $f_{lo}^{(i)}$, like :

$$f_{lo}^{(i)} = h_{trans}^{(i)} \circ f_{exact} \tag{2.17}$$

The more complex the underlying transformation function $h_{trans}^{(i)}$ leads the worse approximation performance of $f_{lo}^{(i)}$. Similarly, we consider a transformation function $h_{trans-hi}$, which can mapping the underlying function f_{exact} into f_{hi} , like :

$$f_{hi} = h_{trans-hi} \circ f_{exact} \tag{2.18}$$

The transformation function $h_{trans-hi}$ is more complex than the transformation function $h_{trans}^{(i)}$. Figure 2.1 describes the trade-off between low- and high-fidelity models in the classical multi-fidelity model. As we can see, a high-fidelity model requires more computation costs but can bring a more exact solution. A low-fidelity model can just bring an inexact solution, but its computation cost is inexpensive.

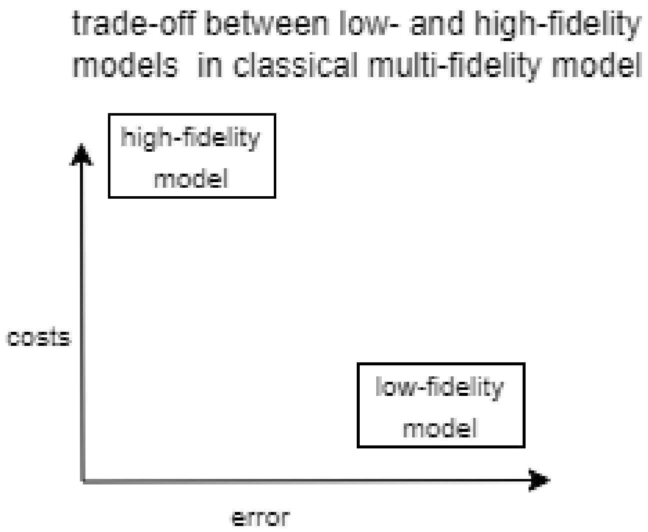

Figure 2.1: Illustrations of the Trade-off between Low- and High-Fidelity Model in Classical Multi-Fidelity Model [30]

2.7.2 Multi-fidelity Model Management Strategies

This section introduces a multi-fidelity model management strategy[30]. A model management strategy typically describes combining and applying different fidelity models. There are three types of multi-fidelity model management strategies: adaptation, fusion, and filtering. Our work is related to adaptation and fusion.

Adaptation The first type of multi-fidelity model management strategy is an adaptation, enhancing the low-fidelity model with information from the high-fidelity model while the computation proceeds [30]. One typical case of adaptation is the correction of low-fidelity model outputs. The correction, which is derived from the high-fidelity model, can be added to the low-fidelity model output to generate a better output. In this situation, the low-fidelity model provides a primary result with a lower evaluation cost. In contrast, the high-fidelity model provides the correction that can improve the accuracy of the basic result. Therefore, the low-fidelity model can turn into a more accurate model after such an adaptation process,

Fusion The second multi-fidelity model management strategy type is information fusion, combining information from different fidelity sources. One typical case of fusion is the control variate framework [7][17][30]. In the control variate framework, the correlation between high- and low-fidelity models can reduce the variance of Monte

Carlo estimators. An estimator with a low variance can be obtained from a large number of low-fidelity model evaluations, A few high-fidelity model evaluations can be leveraged to obtain unbiased estimators of the statistics of interest, which can be derived from the estimator with a low variance.

2.8 Gaussian Process

In this section, we introduce Gaussian progress and Gaussian progress regression.

2.8.1 Multivariate Gaussian Distributions

This section recalls the basic concept of Gaussian distributions and multivariate Gaussian distributions[40].

Gaussian distributions are the fundamentals of Gaussian process regression. The Gaussian distributions $N(\mu, \sigma^2)$ can described by its probability density function :

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \tag{2.19}$$

Where:

- The parameter μ is the mean or expectation of the distribution
- The parameter σ^2 is the variance of the distribution.

The multivariate Gaussian distribution of a d-dimensional random vector $\boldsymbol{X} = (X_1, ..., X_d)^d$ can described in the following notation: $\boldsymbol{X} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ Where:

- The parameter $\boldsymbol{\mu}$ is d-dimensional mean vector with $\boldsymbol{\mu} = E[\boldsymbol{X}] = (E(X_1), E(X_2), ..., E(X_d))^T$
- The parameter $\boldsymbol{\Sigma}$ is $d \times d$ covariance matrix with $\Sigma_{i,j} = E[(X_i - \mu_i)(X_j - \mu_j)] = COV[X_i, X_j]$

The corresponding probability density function is :

$$f_{\boldsymbol{X}}(x_1, ..., x_d) = \frac{exp(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{u})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{u}))}{\sqrt{(2\pi)^d \, |\boldsymbol{\Sigma}|}} \tag{2.20}$$

Where:

- The parameter $\boldsymbol{x}$ is d-dimensional column vector
- The parameter $|\boldsymbol{\Sigma}| \equiv det\boldsymbol{\Sigma}$ is the determinant of $\boldsymbol{\Sigma}$.
- The parameter $\boldsymbol{\Sigma}$ is $d \times d$ covariance matrix with $\Sigma_{i,j} = E[(X_i - \mu_i)(X_j - \mu_j)] = COV[X_i, X_j]$

2.8.2 Marginalization

Assuming a multivariate Gaussian distribution $\boldsymbol{X} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is given and $\boldsymbol{X_s}$ is a any subset vector of $\boldsymbol{X}$. The marginalization property of Gaussian distributions states that the marginal distribution of $\boldsymbol{X_s}$ is also a multivariate normal distribution.

$$\boldsymbol{X_s} \sim N(\boldsymbol{\mu_s}, \boldsymbol{\Sigma_s})$$

Where:

- The parameter $\boldsymbol{\mu_s}$ drops the irrelevant variables from the mean vector $\boldsymbol{\mu}$
- The parameter $\boldsymbol{\Sigma_s}$ drops the corresponding rows and columns from the covariance matrix $\boldsymbol{\Sigma}$

For example, let us consider $\boldsymbol{X}$ is partitioned as follows:

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x_1} \\ \boldsymbol{x_2} \end{bmatrix}$$

$\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are partitioned as follows:

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu_1} \\ \boldsymbol{\mu_2} \end{bmatrix}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma_{11}} & \boldsymbol{\Sigma_{12}} \\ \boldsymbol{\Sigma_{21}} & \boldsymbol{\Sigma_{22}} \end{bmatrix}$$

The covariance matrix Σ is symmetric , Σ_{11} and Σ_{22} are also symmetric. According to the marginalization property , $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ follow the multivariate Gaussian distributions with the following distributions:

$$\boldsymbol{x_1} \sim N(\boldsymbol{\mu_1}, \boldsymbol{\Sigma_{11}})$$
$$\boldsymbol{x_2} \sim N(\boldsymbol{\mu_2}, \boldsymbol{\Sigma_{22}})$$

2.8.3 Conditional Distributions

Assuming a multivariate Gaussian distribution is given. Let us consider d-dimensional $\boldsymbol{x}$ s partitioned as follows [40][33]:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x_1} \\ \boldsymbol{x_2} \end{bmatrix}$$

with size $\begin{bmatrix} q \times 1 \\ (N-q) \times 1 \end{bmatrix}$.

$\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ can be partitioned as follows:

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu_1} \\ \boldsymbol{\mu_2} \end{bmatrix}$$

with size $\begin{bmatrix} q \times 1 \\ (N-q) \times 1 \end{bmatrix}$.

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma_{11}} & \boldsymbol{\Sigma_{12}} \\ \boldsymbol{\Sigma_{21}} & \boldsymbol{\Sigma_{22}} \end{bmatrix}$$

with size $\begin{bmatrix} q \times q & q \times (N-q) \\ (N-q) \times q & (N-q) \times (N-q) \end{bmatrix}$.

According to [40], the conditional distribution $\boldsymbol{x_1}|\boldsymbol{x_2}$ is also a multivariate Gaussian distribution with the following distributions:

$$\boldsymbol{x_1}|\boldsymbol{x_2} \sim N(\boldsymbol{\mu_{1|2}}, \boldsymbol{\Sigma_{1|2}})$$

Where:

- $\boldsymbol{\mu_{1|2}} = \boldsymbol{\mu_1} + \boldsymbol{\Sigma_{12}}\boldsymbol{\Sigma_{22}}^{-1}(x_2 - \boldsymbol{\mu_2})$
- $\boldsymbol{\Sigma_{1|2}} = \boldsymbol{\Sigma_{11}} - \boldsymbol{\Sigma_{12}}\boldsymbol{\Sigma_{22}}^{-1}\boldsymbol{\Sigma_{21}}$

2.8.4 Gaussian Processes

A stochastic Gaussian process can be described as a collection of random variables indexed by time or space [40][33]. Each collection of those random variables has a multivariate Gaussian distribution. In other words, the distribution of a Gaussian process is the joint distribution of those random variables. This distribution of a Gaussian process is over functions $f(x)$ with a continuous domain. These functions $f(x)$ can be evaluated at $x_1, ..., x_n$. The mean of $f(x)$ is the mean function $m(x)$:

$$m(x) = E[f(x)]$$

The covariance between the two random variables $f(x_i)$ and $f(x_j)$ corresponding to x_i and x_j is the kernel function $k(x_i, x_j)$:

$$k(x_i, x_j) = cov(f(x_i), f(x_j)) = E[(f(x_i) - m(x_i))(f(x_j) - m(x_j))]$$

Gaussian process can be written as :

$$f(x) \sim GP(m(x), k(x, x'))$$

.

The shape of the Gaussian processes model is defined by the mean function $m(x)$ and the kernel function $k(x, x')$. Some common covariance functions include:

- Constant : $K_C(x, x') = C$
- Linear : $K_L(x, x') = x^T x'$
- Squared exponential : $K_{SE}(x, x') = exp(-\frac{d(x,x')^2}{2l^2})$, where l is the length scale of the kernel and $d(\cdot, \cdot)$is the Euclidean distance.

2.8.5 Gaussian Processes Regression

In this section, we are going to introduce Gaussian processes regression[40]. According to, the regression function modeled by a multivariate Gaussian is defined as:

$$P(\boldsymbol{f}|\boldsymbol{X}) = N(\boldsymbol{f}|\boldsymbol{\mu}, \boldsymbol{K})$$

, where:

- $\boldsymbol{X}$ are the the observed data points with $\boldsymbol{X} = [x_1, ..., x_n]$.
- $\boldsymbol{f} = [f(x_1), ..., f(x_n)]$
- $\boldsymbol{\mu} = [m(x_1), ..., m(x_n)]$ and $m(\cdot)$ represents the mean function.
- $K_{ij} = k(x_i, x_j)$ and $k(\cdot, \cdot)$ represents a positive definite kernel function.

Based on this model, we can predict the result at new points $\boldsymbol{X_*}$ as $\boldsymbol{f(X_*)}$. Let us consider the joint distribution of $\boldsymbol{f}$ and $\boldsymbol{f_*}$ can be described as :

$$\begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{f_*} \end{bmatrix} \sim N(\begin{bmatrix} m(\boldsymbol{X}) \\ m(\boldsymbol{X_*}) \end{bmatrix}, \begin{bmatrix} \boldsymbol{K} & \boldsymbol{K_*} \\ \boldsymbol{K_*^T} & \boldsymbol{K_{**}} \end{bmatrix})$$

, where:

- $\boldsymbol{K} = K(\boldsymbol{X}, \boldsymbol{X})$,$\boldsymbol{K_*} = K(\boldsymbol{X}, \boldsymbol{X_*})$ and $\boldsymbol{K_{**}} = K(\boldsymbol{X_*}, \boldsymbol{X_*})$
- $(m(\boldsymbol{X}), m(\boldsymbol{X_*})) = 0$, while the data is often normalized to a zero mean.

In fact , this joint distribution of $\boldsymbol{f}$ and $\boldsymbol{f_*}$ is the joint probability distribution equation $P(\boldsymbol{f}, \boldsymbol{f_*}|\boldsymbol{X}, \boldsymbol{X_*})$ over $\boldsymbol{f}$ and $\boldsymbol{f_*}$. In regressions , we want to know the conditional distribution $P(\boldsymbol{f_*}|\boldsymbol{f}, \boldsymbol{X}, \boldsymbol{X_*})$.

Using the theorem, the corresponding conditional distribution is :

$$\boldsymbol{f_*}|\boldsymbol{f}, \boldsymbol{X}, \boldsymbol{X_*} \sim N(\boldsymbol{K_*^T K f}, -\boldsymbol{K_*^T K^{-1} K_*})$$

3 Process Models

In the last chapter, we recall some background knowledge about portfolio optimization and multi-fidelity. In this section, we want to specify the portfolio optimization process model and the multi-fidelity method process model.

3.1 Process Models of Portfolio Optimization

This section describes the process model of portfolio optimization.

Classical Portfolio optimization is based on the single-period model of investment. A single-period investment always takes place over only one period. Before or after this period, there is not any investment. In other words, the portfolio turnover costs can be ignored. Portfolio optimization can be divided into four steps: *Data Sampling*, *Data Preprocessing*, *Optimization Problem Solving*, and *Result Evaluation*, as illustrated in 3.1.

3.1.1 Data Sampling

The first step is *Data Sampling*. In this step, raw data is collected from the financial market. In our work, the raw data is stock prices from the secondary market. Specifically, the raw data is the closing price based on different time frames. Using different time frames represents using different sampling intervals or sampling periods. The data from financial markets is historical time-series data. The amount of these historical time-series data is limited. A large sampling interval can lead to a smaller data set size. In contrast, a smaller sampling interval can bring a more extensive data set.

3.1.2 Data Preprocessing

The second step is *Data Preprocessing*. Raw data must be preprocessed before being used in *optimization problem solving*. *Data Preprocessing* contains four different calculations.

Adjusted Closing Prices

Firstly, the adjusted closing prices need to be calculated from raw data to ensure comparability because dividend payments, new offerings, and stock splits can make a non-linear change in asset prices[29]. In this work, the raw data are the adjusted closing prices[5] from Yahoo Finance. Hence, this calculation can be ignored.

Historical Asset Returns

Secondly, the historical asset returns need to be calculated from these adjusted closing prices using the formula 2.1. At the same time, the research target in portfolio optimization is not asset prices but asset returns.

Expected Asset Returns

Thirdly, the expected asset return needs to be calculated from historical asset returns because building a portfolio is a forward-looking activity [25]. Thus, the asset returns must be forecast rather than observed. Calculating the expected asset returns for all assets is equivalent to calculating the expected portfolio return. This calculation is based on the formula 2.2 in our work.

Covariances of Asset Returns

Finally, the covariances of asset returns are required to be calculated from historical asset returns using the formula 2.5. A calculation of covariances among all assets is equivalent to a calculation of the portfolio variance.

3.1.3 Optimization Problem Solving

The third step is *optimization problem solving*. This step is the core component of portfolio optimizations because the first and second steps are just about data preparation. This step generates an optimal portfolio by solving an optimization problem. Traditionally, this optimization problem is mean-variance analysis. The input of this optimization problem is the expected portfolio return and the expected portfolio variance. Mean-variance analysis with standard objectives and constraints are convex optimization problems that are a well-understood class of problems. Hence, it can be solved by mathematical tools, such as quadratic and linear programming. The detailed setting of the optimization problem is mentioned in the section. In this work, the optimization problem is directly solved by CVXPY.[41]

Optimization Problem Setting

In our work, our target optimization problem in portfolio optimization is as follows:

$$
\begin{aligned}
\max_{x} \quad & x^T \mu - \gamma x^T x \\
\text{s.t.} \quad & x^T \Sigma x \leq R^\star \\
& x^T e = 1 \\
& x \geq 0
\end{aligned}
\tag{3.1}
$$

Where :

- $R^\star$ is the upper bound on the portfolio variance.
- $x \geq 0$ means that all components of the vector are non-negative, $x_i \geq 0, i = 1, ..., n$.
- γ is L2 regularisation parameter, defaults to 1.
- $R^\star = 3$, which can allow a very high portfolio variance.

3.1.4 Result Evaluation

The final step is *result evaluation*. In this step, the portfolio optimization performance must be evaluated according to the actual asset returns. Before portfolio return calculation, the portfolio need to be normalized.

Let us consider that $x \in \mathbb{R}^n$ and $x = (x_1, x_2, ..., x_n)$ be the allocation vector of weights in the portfolio before normalization. The normalized portfolio x_{norm} is specified as follow :

$$x_{norm} = (x_{1,norm}, x_{2,norm}, ..., x_{n,norm}) \tag{3.2}$$

Where :

- $x_{i,norm} = \frac{x_i}{(x_1 + x_2 + ... + x_n)}$, $i \in [1, n]$

In our work, the benchmarks in *result evaluation* are average return, cumulative return, and Sharpe ratio. See section 2.5.

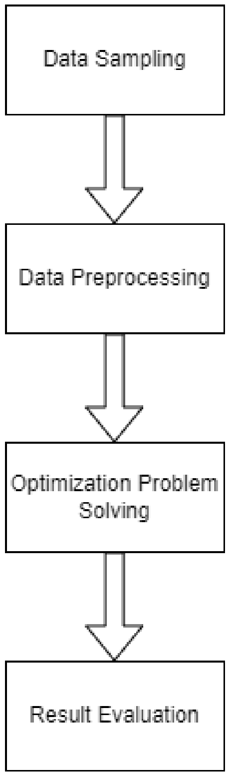

Figure 3.1: Portfolio Optimization Process Model

3.2 Process Models of Multi-Fidelity methods

This section describes the process model of multi-fidelity methods for outer-loop applications. Multi-fidelity methods for outer-loop applications are the notion from [30], see Figure 3.2. Outer-loop applications are defined as computational applications that form outer loops around a model. The multi-fidelity model can receive input from outer-loop applications and generate output. At the termination of the outer loop, an overall outer-loop result can be obtained. Outer-loop applications should target a specific outer-loop result.

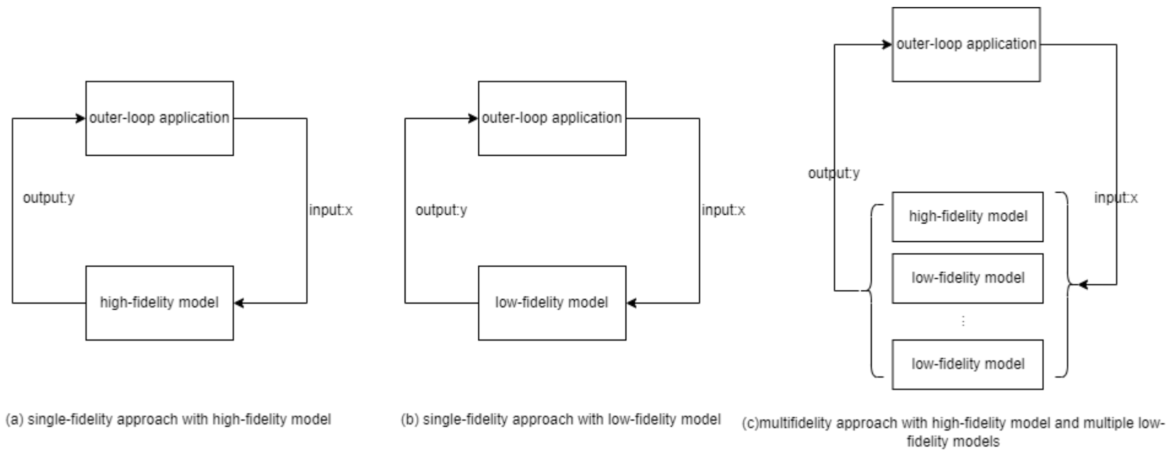

Figure 3.2: Process Models of Multi-Fidelity Methods for Outer-Loop Applications[30]

4 Multi-Fidelity Portfolio Optimizations Models

In this Chapter, we discuss the fidelity of portfolio optimization and propose two different multi-fidelity portfolio optimization models. One is the prediction based multi-fidelity model. The other one is the sampling interval based multi-fidelity model.

4.1 Fidelity in Portfolio Optimizations

Our work is the first trial to link multi-fidelity to portfolio optimizations. Therefore, we need to define the fidelity of portfolio optimizations. We want to find a mapping between the process model of multi-fidelity methods and the process model of portfolio optimization. Without loss of generality, we focus on low-fidelity and high-fidelity in our work. We want to improve the performance of portfolio optimization to generate a new and better portfolio offering a higher return and a better Sharpe ratio. The multi-fidelity model's output should be the new portfolio's evaluation results when the target outer-loop applications detect whether or not the new portfolio can outperform. The overall outer-loop result is whether or not portfolio optimization is improved. The difficulty in linking multi-fidelity methods to portfolio optimization is not in these outer-loop applications but in low-fidelity and high-fidelity models. We must solve a problem: What is a high-fidelity (low-fidelity) model in portfolio optimizations?

4.1.1 What Are High-Fidelity (Low-Fidelity) Models in Portfolio Optimizations?

What are high-fidelity (low-fidelity) models in portfolio optimizations? The intuitive idea is that high-fidelity models can generate a better portfolio, while low-fidelity models can generate a worse one. A better portfolio can bring a higher return and a lower risk, while a worse portfolio can offer a lower return and a higher risk. Can a particular portfolio optimization model generate a better portfolio than others? Although some have suggested that some variants of classical portfolio optimization can outperform for some special situations[25][13][8], there is currently no known documentation to support that they can outperform in most situations[16]. As far as we know, there is no acknowledged high-fidelity portfolio optimization model or an acknowledged low-fidelity portfolio optimization model. Hence, we must define high-fidelity (low-fidelity) models in portfolio optimizations.

As introduced in the section 3.1, portfolio optimization consists of four steps: *data sampling*, *data preprocessing*, *optimization problem solving*, and *result evaluation*. Each step receives the input from its last step or data source and generates the corresponding output. Suppose we find a high-fidelity model to generate a better output for one of these steps. In that case, this better output may lead to a better portfolio optimization performance evaluation result. Hence, our target is to test whether or not these steps can be high-fidelity models(steps) or low-fidelity models(steps). In other words, we want to know whether or not some better approach or parameter adjustments can refine these steps of classical portfolio optimization.

4.1.2 Why High-Fidelity Model Could Not Be a Complex Model in Portfolio Optimizations?

Traditionally, the low-fidelity model is simplified, while the high-fidelity model is more complex but better. The evaluating cost of the high-fidelity model is larger than the evaluating cost of the low-fidelity model[30]. Multifidelity methods leverage a low-fidelity model to obtain computational speedups. If we found a more complex model for portfolio optimization, multi-fidelity methods would leverage the portfolio optimization with a low-fidelity model to obtain computational speedups. These more complex models include the expected return calculation based on a more complex model, the portfolio variance calculation based on a more complex model, and the optimization problem with more constraints. Using a more complex model increases the computation cost of optimization. Unfortunately, portfolio optimization with a more complex model may underperform in most situations compared to portfolio optimization with a simple model[23]. Traditionally, a more complex model in portfolio optimization represents using more constraints in optimization problems. These complex constraints may interact with each other. Using these multiple and interacting constraints can lead to risk-return optimization being more sensitive to input changes, see section 4.1.3.

4.1.3 Why Is It Difficult to Improve Portfolio Optimization ?

There are two reasons why improving portfolio optimizations is so hard. One is the uncertainty from financial data. The other one is the uncertainty of optimization problems.

Uncertainty from Financial Data and Time Series Decomposition

Financial data is uncertain because its signal-to-noise ratio is meager, as introduced in section 2.1.1. It can lead to a phenomenon: garbage in, garbage out. One classical solution to this phenomenon is time series decomposition, which decomposes a time series into trends, seasonality, and cycles. Although it is possible to make a time series decomposition using some filter to increase the signal-to-noise ratio, it is also equivalent to adding more limitations to the portfolio optimization. If the input of the portfolio optimization came from a particular time series decomposition, its portfolio could offer a higher return and lower risk only in the corresponding trend cycle. Using time series decomposition is unsuitable for portfolio optimization because no such approach can guarantee that a trend must be detected in any time interval. The investment-based portfolio optimization must be continuous without any break because the portfolio returns and performance must be calculated periodically. Investments with such a filter can lead to a huge loss when a trend can not be detected. Hence, time series decomposition, or filter, is unsuitable for portfolio optimization when the investment must be rebalanced regularly.

Uncertainty of Optimization Problems

Solving the optimization problem is uncertain because of mean-variance instability. In portfolio optimization, the estimation errors of the covariance matrix Σ_t of asset returns and the expected return u can Introduce some huge estimation errors and instability in the optimal solution. Academics have largely studied this stability issue with estimators based on historical figures. Before going into the details of this subject, we propose to illustrate the stability problem of the optimal portfolio with the following example from [35].

Let us consider four assets,x_1,x_2,x_3,x_4. The expected asset returns are $\hat{u}_1$ = 5%, $\hat{u}_2$ = 6%,$\hat{u}_3$ = 7%,$\hat{u}_4$ = 8% whereas the variance of the asset returns, are equal to $\hat{\sigma}_1$ =10%,$\hat{\sigma}_2$ =12%,$\hat{\sigma}_3$ =14%,$\hat{\sigma}_4$ =15%, Our assumption is that the correlations are the same and we have $\hat{\rho}_{i,j}$ =$\hat{\rho}$=70% We solve the optimization problem without constraints using these parameters. In this case the optimal portfolio is x_1^*=23.49%,x_2^*=19.57%,x_3^*=16.78%, x_4^*=28.44%. Hence, 23.49 % of the total capital should be used to buy asset x_1. 19.57 % of the total capital should be used to buy asset x_2 .

In table 4.1, we indicate how a small perturbation of input parameters changes the optimized solution. For instance, if the variance of the second asset increases by 3%, the weight on this asset becomes -14.04% instead of 19.57%. If the realized return of the first asset is 6% and not 5%, the optimal weight of the first asset is almost three times larger (63.19% versus 23.49%). As a consequence, the optimized solution is very sensitive to

$\hat{\rho}$	70%	80%	70%	80%	70%
$\hat{\sigma}_2$	12%	12%	15%	15%	12%
$\hat{u}_1$	5%	5%	5%	5%	6%
x_1^*	23.49%	19.43%	36.55%	39.56%	63.19%
x_2^*	19.59%	16.19%	-14.04%	-32.11%	8.14%
x_3^*	16.78%	13.88%	26.11%	28.26%	6.98%
x_4^*	28.44%	32.97%	37.17%	45.87%	18.38%

Table 4.1: Sensitivity of the portfolio to input parameters [35]

estimation errors. The stability problem comes from the solution structure. Indeed, the solution involves the inverse of the covariance matrix $I = \hat{\Sigma}^{-1}$called the information matrix. The eigenvectors of the two matrices are the same, but the eigenvalues of I are equal to the inverse of the eigenvalues of $\hat{\Sigma}$.

4.1.4 Data Sampling in High-Fidelity Model / Low-Fidelity Model

This section discusses how we can improve the output of *Data Sampling*. According to signal processing, sampling is the reduction of a continuous-time signal to a discrete-time signal by a sampler, which is a component that extracts samples from a continuous signal[24]. The sampler can sample from the continuous-time signal every time interval T, called the sampling interval or sampling period. The sampling rate is how frequently the sampler extracts samples from a continuous signal. According to the Nyquist–Shannon sampling theorem[26], the fidelity of the sampling depends on the sample rate. The lower sample rate can generate a high-fidelity result with a lower probability than the higher sample rate. Without loss of generality, the high-fidelity result comes from the sampling with a higher sample rate. The low-fidelity result comes from sampling with a lower sample rate. In other words, the high-fidelity result can come from sampling with a smaller sampling interval, while the low-fidelity result comes from a larger sampling interval.

However, there is a problem: a smaller sampling interval can lead to a larger data set size. In comparison, a larger sampling interval can lead to a smaller data set size because the range of dates is limited in the time series data. The number of generated portfolios within a range of dates differs according to the different sizes of the data sets. For example, using daily data with a daily sampling, the investors should rebalance their portfolio by solving optimization problems daily. In comparison, the investors should rebalance their portfolios by solving optimization problems monthly using monthly data with a monthly sampling. There are 365 or 366 portfolios generated by daily rebalance every year, while 12 portfolios are generated by monthly rebalance every year. Comparing the daily portfolio and monthly portfolio is a problem. For more details about how to solve this problem, see section 4.3.2.

4.1.5 Data Preprocessing in High-Fidelity Model / Low-Fidelity Model

This section discusses how we can improve the output of *Data Preprocessing*. As introduced in, *Data Preprocessing* consists of four different calculations, but we focus on the expected asset returns and the covariances of asset returns. The intuitive idea is that high-fidelity *Data Preprocessing* can calculate more precise expected asset returns and covariances of asset returns. The expected asset return calculation and the covariances of asset returns calculation are based on rolling-window analysis of time-series models with a fixed rolling window size w. A larger rolling window size means that the expected asset returns and the covariances of asset returns can learn more things from the historical data.

However, more evidence is needed to prove that a more precise calculation can benefit much from a larger rolling window size. According to our experiment result, see section 6.1, using a larger rolling window size can only improve the accuracy of these calculations when the time range about related data is smaller than 3 or 4 years. For example, the expected monthly asset return calculation with $w = 40$ is better than it's with $w = 70$ or $w = 12$. The data set with $w = 70$ contains too much data that needs to be more timely.

4.1.6 Optimization Problem Solving in High-Fidelity Model / Low-Fidelity Model

This section discusses how we can improve the output of *Optimization Problem Solving*. An optimal portfolio can be generated by solving a classical optimization problem. The optimization problem can be mean-variance analysis. In this optimization problem, the hyperparameter tuning can lead to different results and computational costs. Is it possible to select a suitable hyperparameter to give the optimization problem a better result with higher computational cost? According to our experiments in section 6.3, changing the L2 regularisation parameter γ, can affect the result to generate a better portfolio.

4.1.7 Result Evaluation in High-Fidelity Model / Low-Fidelity Model

This section discusses how we can improve the output of *Result Evaluation*. In *Result Evaluation*, the portfolio optimization performance needs to be evaluated according to the actual asset returns. Normally, there is nothing to do because an optimal portfolio is generated from the last step, and the actual returns are collected directly from the market. The calculations of average return, cumulative return, and Sharpe ratio are based on formula 2.14, formula 2.15, and formula 2.16. The actual asset returns and these calculations are invariant. Nothing can be done. However, one interesting thing is whether or not we can get a higher portfolio return if we can reduce the difference between the actual and expected asset returns. Can an unknown mapping function be found between actual and expected asset returns?

Unknown Mapping Function between Actual Asset Returns and Expected Asset Returns

This section discusses the unknown mapping function between actual and expected asset returns in portfolio optimization; see Figure4.1. Before discussing the unknown mapping function, whether or not the unknown mapping function exists must be discussed.

Does the Unknown Mapping Function Exist? The unknown function can map the expected return into the actual return. Let us assume that the expected return can be calculated from the historical asset return R , where $R = [R_1, R_2..., R_n]$ is a time-series data. The corresponding actual return is R_{n+1} for R_n . This unknown mapping function is equivalent to time series data forecasting. There is no evidence to prove that we can forecast the financial time series data. In other words, the unknown mapping function may not exist.

Unknown Mapping Function There is much research on asset return prediction based on time series data[20][37]. The most famous hypothesis is the random walk hypothesis, which states that stock market prices evolve according to a random walk (so price changes are random), and thus, the stock return cannot be predicted [10]. In our work, we assume that the asset return can be predicted in a short time. The relationship between the historical returns and the actual return is a non-linear relationship according to [20]. Hence, the unknown mapping function may be non-linear. In our work, we have tried using linear and Gaussian processes regression with some multi-fidelity methods to predict the actual asset return. Unfortunately, the actual asset return can not be predicted successfully; more details are in section 4.2.

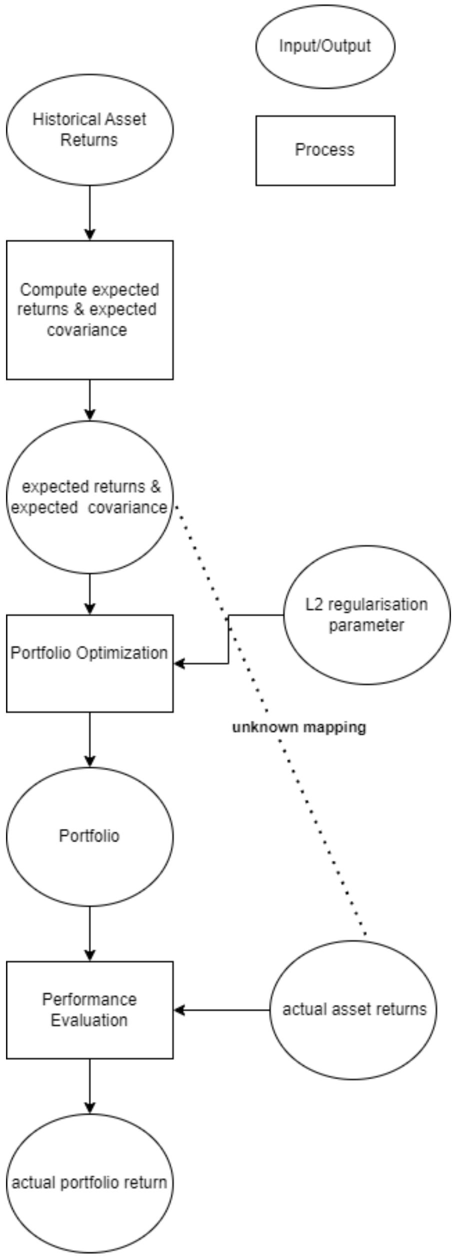

Figure 4.1: Portfolio Optimization with unknown mapping function

4.2 Prediction Based Multi-Fidelity Model — a Unsuccessful Case

This section describes the prediction based multi-fidelity model. As introduced in the section 4.1.7, we want to find an unknown mapping function between actual and expected asset returns. If the difference/error between actual and expected asset returns can be reduced, the portfolio optimization can be improved. If this difference can disappear, the actual asset returns can be predicted perfectly by the historical asset returns. Evaluating this unknown mapping function is equivalent to predicting the actual asset returns. That is the reason why this model is named a prediction-based multi-fidelity model. Although the prediction-based multi-fidelity model is not feasible according to our test, we still propose our idea.

4.2.1 Relationship between Expected Asset Return and Actual Asset Return

This section discusses the relationship between expected asset returns and actual asset returns again. As introduced in the section 2.4, portfolio optimization is based on the expected asset return and the expected covariance. The portfolio is the optimal solution when the expected asset return is the actual asset return. The expected asset return can be regarded as the result of a low-fidelity model when the actual asset return results from a high-fidelity model. However, this high-fidelity model is an unknown function that can predict the actual asset returns from the historical asset returns in *Data Preprocessing*.

Forecasting the actual asset return is a challenging task. Some research has shown that predicting the actual asset returns may be impossible, like the random walk hypothesis[10]. However, there is still much research about how to predict the actual asset return[32][19][31][11]. In this work, classical portfolio optimization, which needs to calculate the expected asset returns, is regarded as a low-fidelity model. In contrast, particular portfolio optimization, which can predict asset returns, is regarded as a high-fidelity model.

We assume that a black box function, a precise but huge-cost approach, can make this prediction. The classical portfolio optimization can be improved if we can build a mapping function between the high-fidelity and low-fidelity models to estimate the distribution of the high-fidelity model's output.

4.2.2 Prediction Based Multi-Fidelity Model Design

This section describes the design of the prediction based multi-fidelity model. A multi-fidelity model should consist of low-fidelity models and high-fidelity models.

The low-fidelity models in the prediction-based multi-fidelity model are classical portfolio optimization; see Figure 4.2. The expected returns and covariance are derived from the historical asset returns. Portfolio optimization can solve an optimization problem based on the expected returns and the expected covariance to generate a portfolio. Ultimately, this portfolio is evaluated with the actual asset returns in *Result Evaluation*.

The high-fidelity models in the prediction-based multi-fidelity model are the portfolio optimization using the black box function, which can calculate the perfect expected asset returns, see Figure 4.3. The expected covariance is calculated from the historical asset returns. The perfect expected returns come from the black box function. Portfolio optimization can solve an optimization problem based on the perfect expected returns and the expected covariance to generate a perfect portfolio. Ultimately, this portfolio is evaluated with the actual asset returns in performance evaluation.

Suppose we can find a function that can map the expected returns into the perfect expected returns; see mapping A in Figure 4.4. In that case, the actual portfolio returns can be improved in the low-fidelity model with mapping A. Similarly, the actual portfolio returns can benefit from finding a function that can map the portfolio into the perfect portfolio; see mapping B in Figure 4.4, too.

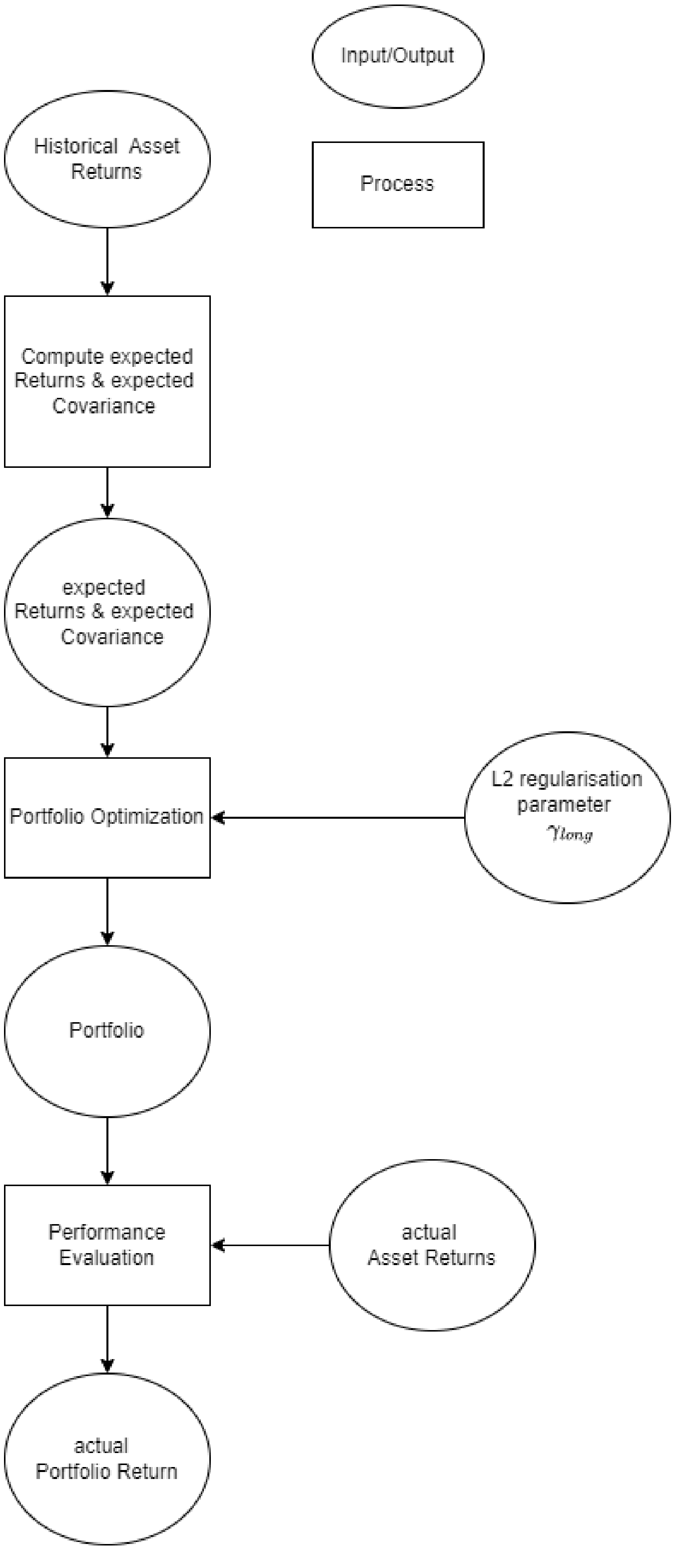

Figure 4.2: Low-Fidelity Model in Prediction-Based Multi-Fidelity Model

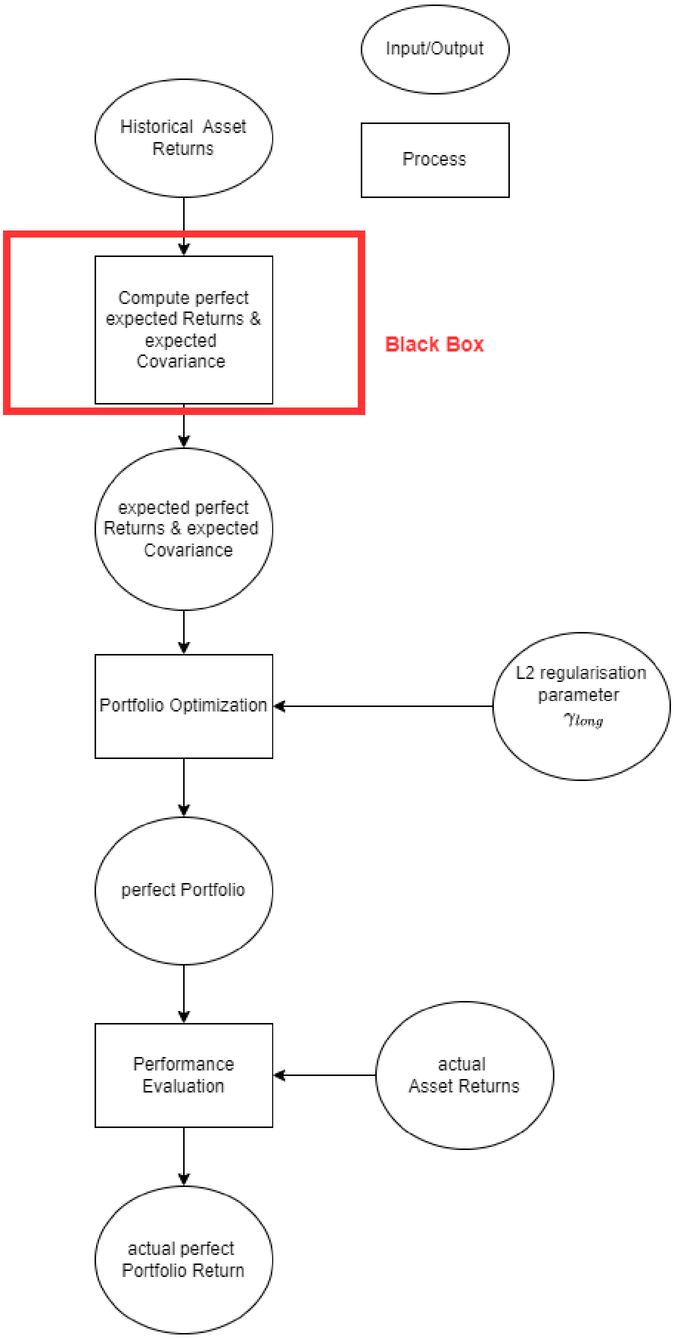

Figure 4.3: High-Fidelity Model in Prediction-Based Multi-Fidelity Model

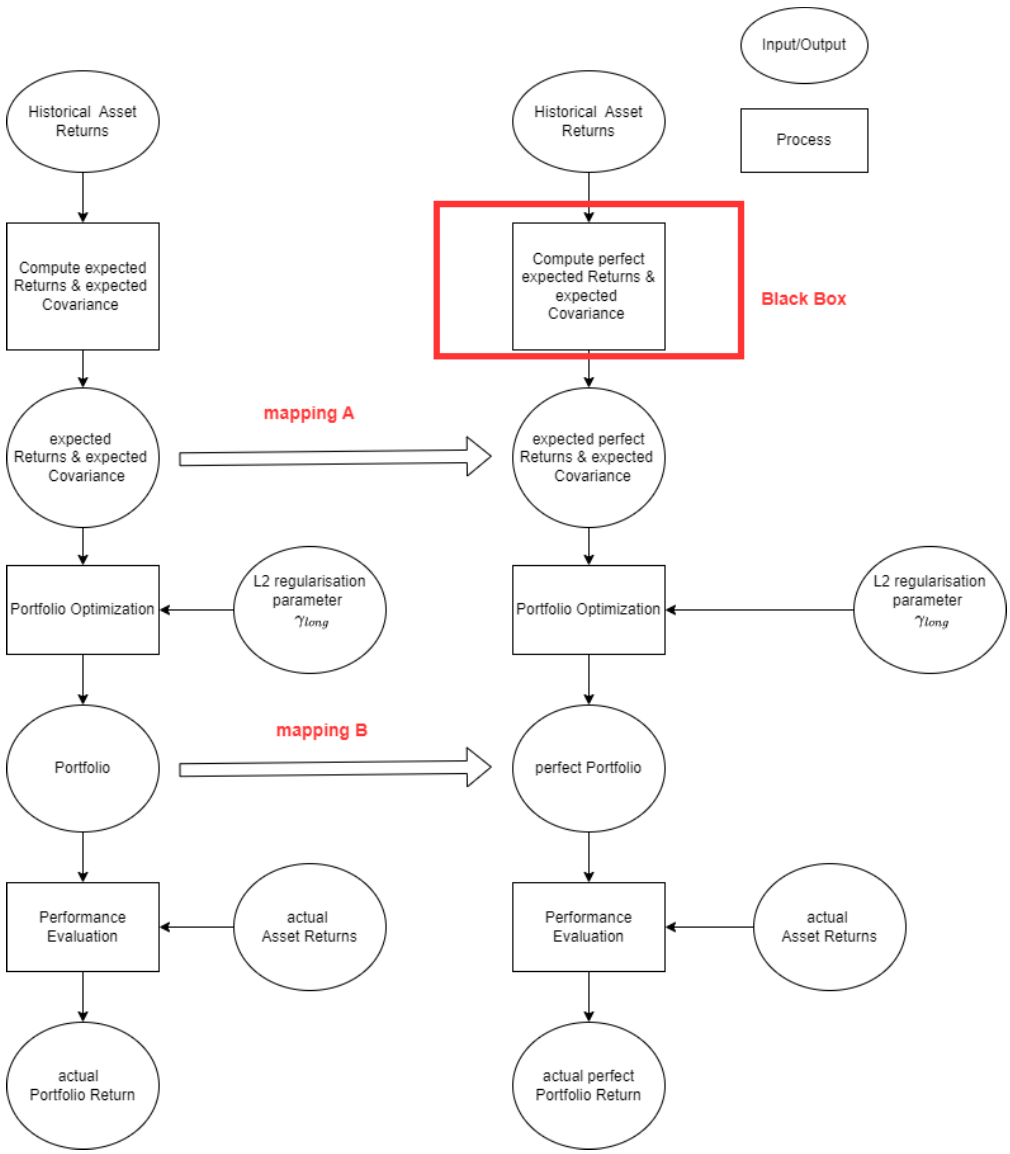

Figure 4.4: Prediction Based Multi-Fidelity Model

4.2.3 Trade-off in Prediction Based Multi-Fidelity Model

This section discusses the trade-off in the prediction-based multi-fidelity model. The trade-off between low- and high-fidelity models in the prediction-based multi-fidelity models is similar to the classical multi-fidelity model; see Figure 4.5. A high-fidelity model can calculate the perfect expected asset returns equal to the actual asset returns.

The expected asset returns can be calculated as usual in the low-fidelity model. Let us consider that the black box calculation of the perfect expected asset return requires infinite computation cost with zero error in a high-fidelity model, while the calculation of the expected asset return requires finite computation cost with a significant error in a low-fidelity model. Hence, high-fidelity models require huge costs and offer a perfect result with negligible error. Low-fidelity models can provide a classically expected asset return with a visible error between the expected and actual asset returns but need a reasonable computation cost.

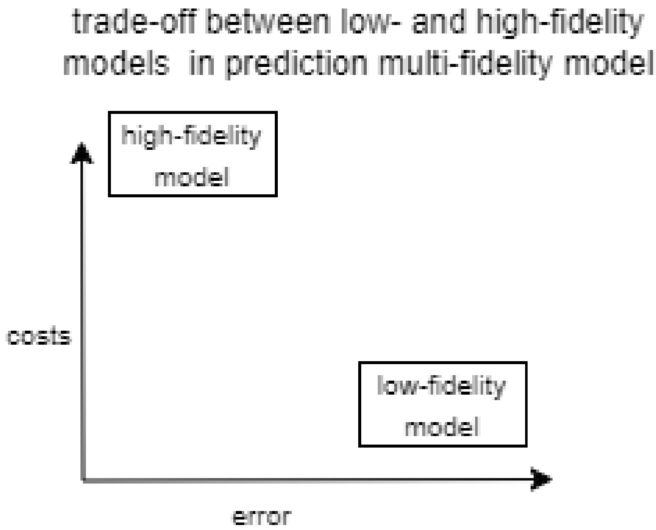

Figure 4.5: Illustrations of the Trade-Off between Low- and High-Fidelity Model in Prediction Based Multi-Fidelity Model

4.2.4 Difference between Multi-Fidelity Methods and Unknown Mapping Function

In the prediction-based multi-fidelity model, we try to estimate the perfect expected returns or the perfect portfolio using classical multi-fidelity methods. However, there is a problem with multi-fidelity methods and unknown mapping functions. The output of multi-fidelity methods and unknown mapping functions are different types. Classical multi-fidelity methods, like autoregressive schemes and nonlinear autoregressive multi-fidelity GPs, are based on Gaussian processes. A distribution mean and a distribution standard deviation describe these methods' output. In contrast, the output of the unknown mapping function is based on the input of *Optimization Problem Solving* and the input of *Result Evaluation* The input of these two steps is some real number. Hence, the output of the unknown mapping function must be a real number. According to our test, we can successfully estimate the distribution of the perfect expected returns and portfolio. We still need to solve the conflict between two different outputs.

4.2.5 Unsuccessful Methods linking Multi-Fidelity Methods to Unknown Mapping Function

In the prediction-based multi-fidelity model, the most important problem is how we can link the output of multi-fidelity methods to the output of an unknown mapping function. We need to find a new unknown mapping,which can project a distribution (a distribution mean and a distribution standard deviation) into a real number. In our work, we have tried the following methods to solve this problem.

Distribution Mean

Design Principle The first method is that we regard the distribution mean as the real number directly. The reason why we do such a thing is that the expected return is the arithmetic mean of historical returns. The distribution mean is a measure of central tendency. In a symmetrical distribution, the mean of distribution can be an arithmetic mean of all data. In other words, we assume that the distribution is symmetrical.

Test Result We have tried to predict the perfect expected returns or the perfect portfolio using Gaussian process regression. The distribution mean is directly regarded as perfect expected returns or portfolios. The error between the distribution mean, and perfect expected return always underperforms the baseline for unknown mapping A when the baseline is the error between the perfect expected return and the average value of the historical return. For unknown mapping B, the distribution mean is regarded as the proportions of portfolios. We predict the different assets ′ proportions independently using Gaussian process regression. Each asset has its distribution mean. We integrate the distribution mean into a predicted portfolio and normalize this predicted portfolio. The normalized and predicted portfolios are evaluated to calculate the return and Sharpe ratio. Compared to the portfolio in the low-fidelity model, the predicted portfolio sometimes can outperform the portfolio when we just focus on a special period. But its return and Sharpe ratio are always lower than the low-fidelity model.

Weighted Arithmetic Mean

Design Principle The second method is using weighted arithmetic mean to calculate the real number. Using Gaussian process regression, we can calculate the 95% confidence interval. We select the upper and lower bounds of this confidence interval as the input of the weighted arithmetic mean, while the distribution mean is also put into the weighted arithmetic mean as input. We want to adjust the weights of the upper bounds, lower bounds, and mean to change the investment style. For example, the weight of lower bounds should be larger for the perfect expected returns calculation if we prefer a more safe investment.

Test Result We tried to find suitable weights using linear regression. The observed values are the perfect expected returns during the perfect expected returns calculation. Similarly, the observed values are the perfect portfolio during the perfect portfolio calculation. According to our test, the weighted arithmetic mean using linear regression also underperforms the baseline for unknown mapping A when the baseline is the average value of the historical return. For unknown mapping B, the weighted arithmetic mean is the proportion of portfolios. The weighted arithmetic mean with linear regression can outperform using the mean of distribution as the proportion of the perfect portfolio. However, the weighted arithmetic mean still underperforms the baseline when focusing on average return, cumulative return, and Sharpe ratio.

4.2.6 Conclusion

The prediction based multi-fidelity model looks like a logical design. Although the black box function is unknown, the output and the input of the black box function are known. If we can link the output of multi-fidelity methods to the output of an unknown mapping function, it is possible to implement the prediction based multi-fidelity model. Unfortunately, according to our test in a small data set, we can not find a suitable approach to help us build a helpful connection between two different outputs. According to our observations, predicting the proposition of portfolio is better than predicting the return. Although the normalized predicted portfolio sometimes can outperform the baseline, the average return, the cumulative return, and the Sharpe ratio of the normalized predicted portfolios are always under the baseline for the whole data set. Because we are unable to solve this problem, we tried to design a new model,the sampling interval based multi-fidelity model.

4.3 Sampling Interval Based Multi-Fidelity Model — a Successful Case

This section introduces the sampling interval based multi-fidelity model. As introduced in the section 4.1.4, the high-fidelity data come from sampling with a smaller sampling interval in the data sampling. In comparison, the low-fidelity data come from sampling with a larger sampling interval. Can the high-fidelity data and the low-fidelity data be utilized by the multi-fidelity method? Before solving this problem, it is necessary to show how these high-fidelity and low-fidelity data work in portfolio optimization.

4.3.1 High-Fidelity Data and Low-Fidelity Data in Portfolio Optimization

This section introduces how portfolio optimization can generate a portfolio from high-fidelity or low-fidelity data.

The first step of portfolio optimization is *Data Sampling*, which collects asset prices from the financial market. *Data Sampling* with different sampling intervals can generate different data sets with different sizes. The historical data from the financial markets is finite. A smaller sampling interval can expand the size of the corresponding data set, while a larger sampling interval can shrink the size of its data set. In financial investment, the long-term investment is always based on some data set collected with a larger sampling interval. This data set may need to be more timely, and its quantities are rare. Hence, the data from these data sets are low-fidelity data. The low-fidelity data set based on a larger sampling interval is named long-term data because it is used for long-term investment. In contrast, the short-term investment is always based on some data set collected with a smaller sampling interval because the data from these data sets is timely enough. Hence, the data from these data sets are high-fidelity data. The high-fidelity data set based on a smaller sampling interval is named short-term data in our work.

The asset prices are sampled regularly from the financial market with different sampling intervals. A larger sampling interval causes the long-term asset prices, while the short-term asset prices come from a smaller sampling interval. These asset prices can easily be converted into historical asset returns according to the formula 2.1. Similarly, historical asset returns based on long-term asset prices are the historical long-term asset returns. The expected long-term asset returns and the expected long-term portfolio variance are calculated with the historical long-term asset returns. The long-term portfolio is generated by portfolio optimization with expected long-term asset returns and expected long-term portfolio variance. Hence, long-term portfolio optimization is the optimization of these long-term data to generate a long-term portfolio. Due to a lack of timeliness, long-term portfolio optimization is low-fidelity portfolio optimization.

And vice versa.

Figure 4.6 describes generating a long-term portfolio from historical long-term asset returns. The expected long-term returns and covariance are computed according to the historical long-term asset returns. The expected long-term returns, the expected long-term covariance, and the L2 regularisation parameter γ_{long} play as the input of the portfolio optimization problem to generate a long-term portfolio. Finally, this long-term portfolio is evaluated by the actual long-term asset returns to calculate the long-term portfolio return.

In order to simplify the process structure, the computation of the expected long-term returns and the expected long-term covariance are integrated into one process, long-term portfolio generation. Long-term portfolio generation produces a long-term portfolio directly from the historical long-term asset returns.

Figure 4.7 describes generating a short-term portfolio from historical short-term asset returns. The computation of the expected short-term returns and the expected short-term covariance, as well as the portfolio optimization, are integrated into one process: short-term portfolio generation. Short-term portfolio generation with the L2 regularisation parameter γ_{short} produces a short-term portfolio directly from the historical short-term asset returns.

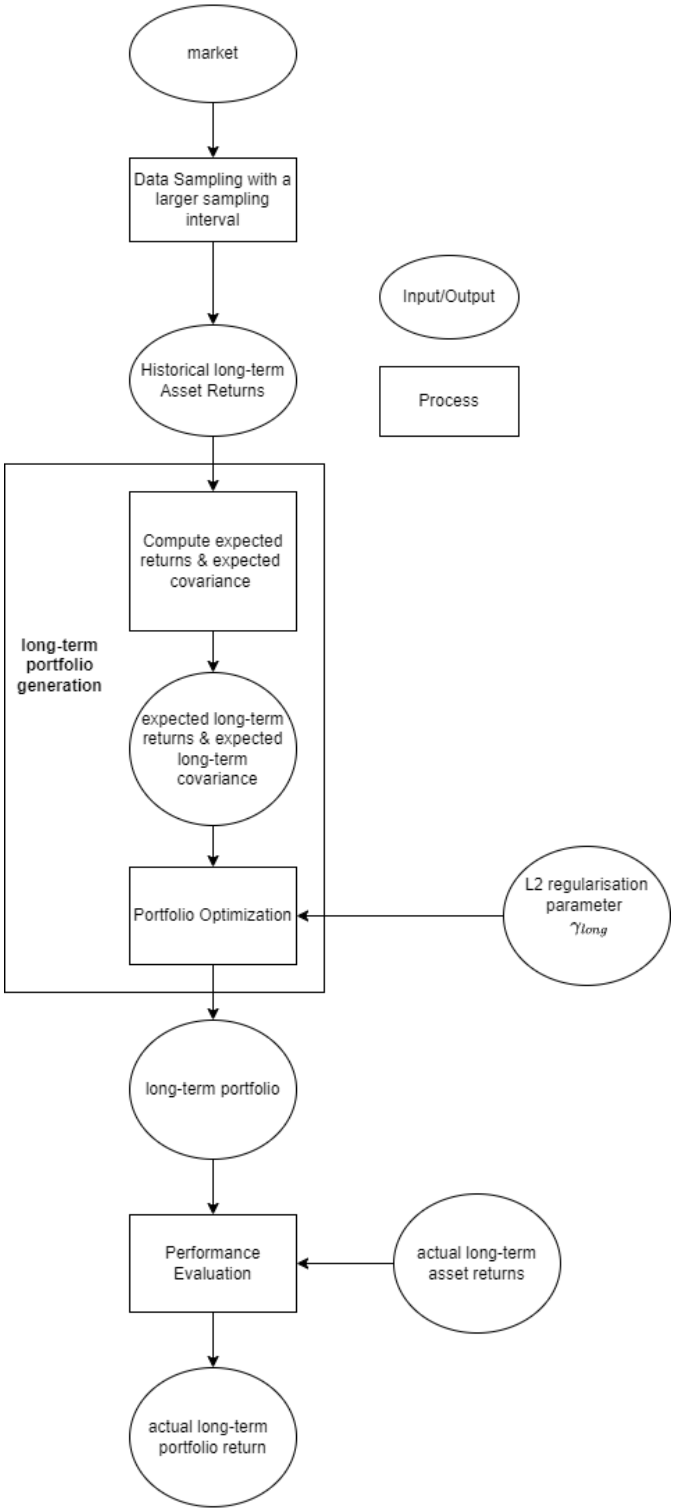

Figure 4.6: Long-term Portfolio Generation in Low-Fidelity Portfolio Optimization

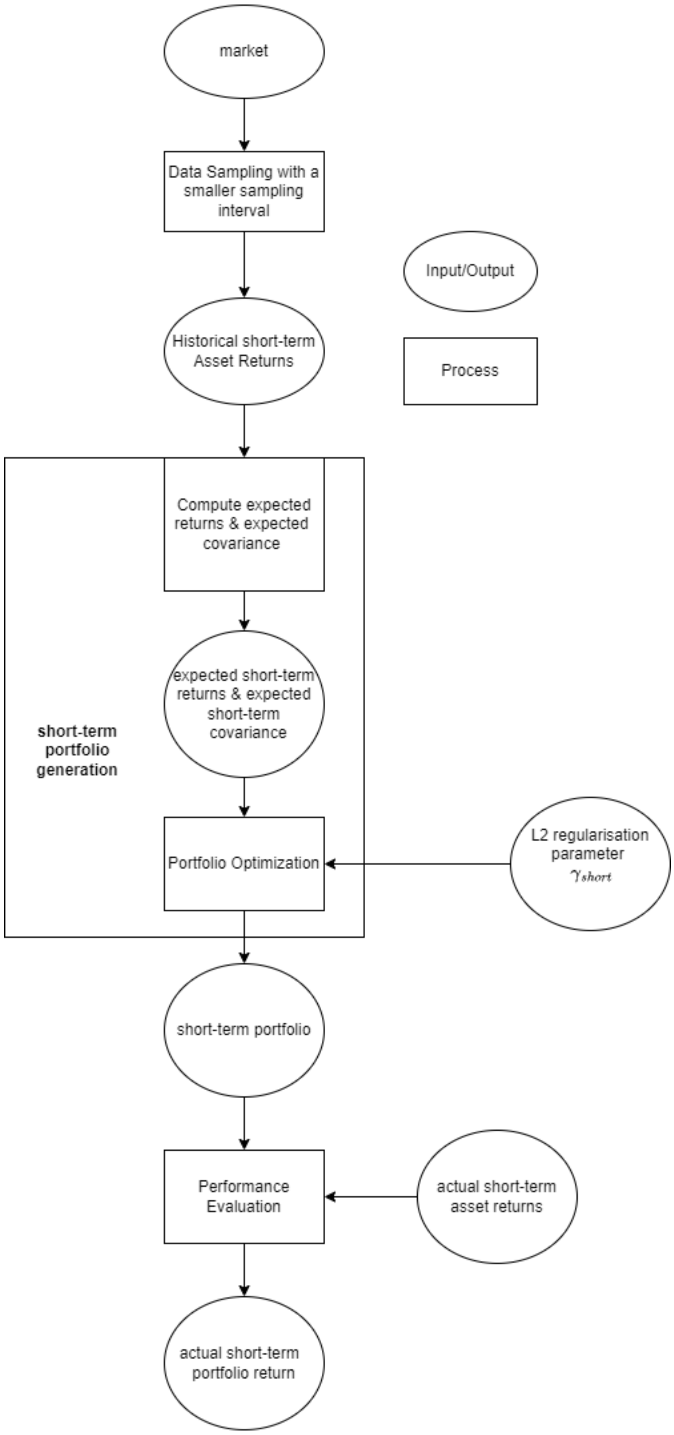

Figure 4.7: Short-Term Portfolio Generation in High-Fidelity Portfolio Optimization

4.3.2 Sampling Interval Based Multi-Fidelity Model Design

This section proposes the sampling interval-based multi-fidelity model design. In portfolio optimization, the generation of a long-term portfolio requires historical long-term asset returns, while the generation of a short-term portfolio requires historical short-term asset returns. The generation of the historical short-term asset returns and the historical short-term asset returns has been discussed in this section. Because of the different sampling intervals, the number of historical short-term asset returns is larger than that of historical long-term asset returns. The historical short-term asset returns can contain more information than the historical long-term asset returns. Therefore, the short-term portfolio may involve more timely information than the long-term one. In other words, the short-term portfolio may bring more return and a higher Sharpe ratio. The short-term portfolio optimization, which can generate the short-term portfolio, is high-fidelity portfolio optimization. Although short-term portfolio optimization may outperform long-term portfolio optimization, most individual investors are passive investors who prefer to hold an investment over a long period. It means that their portfolios are rebalanced infrequently, like a long-term portfolio. Can we improve the long-term portfolio with a short-term portfolio?

Figure 4.8 describes how a short-term portfolio is fused into a long-term portfolio. A short-term portfolio SP can be generated from the historical short-term asset returns from the historical asset prices with a γ_{short}. A long-term portfolio LP can be generated from the historical long-term asset returns from the historical asset prices, with a γ_{long}. Using different γ , different portfolios with different risks can be generated by portfolio optimization; see section 2.4.3. Then, the short-term portfolio is combined with the long-term portfolio via a fusion algorithm to generate a fused long-term portfolio FLP. The fusion algorithms are introduced in the section 4.3.3. The fused long-term portfolio FLP should outperform the long-term portfolio. For example, the corresponding average return, cumulative return, and Sharpe ratio should be better than the long-term portfolio. Moreover, we want to adjust the γ_{short} to improve the fused long-term portfolio.

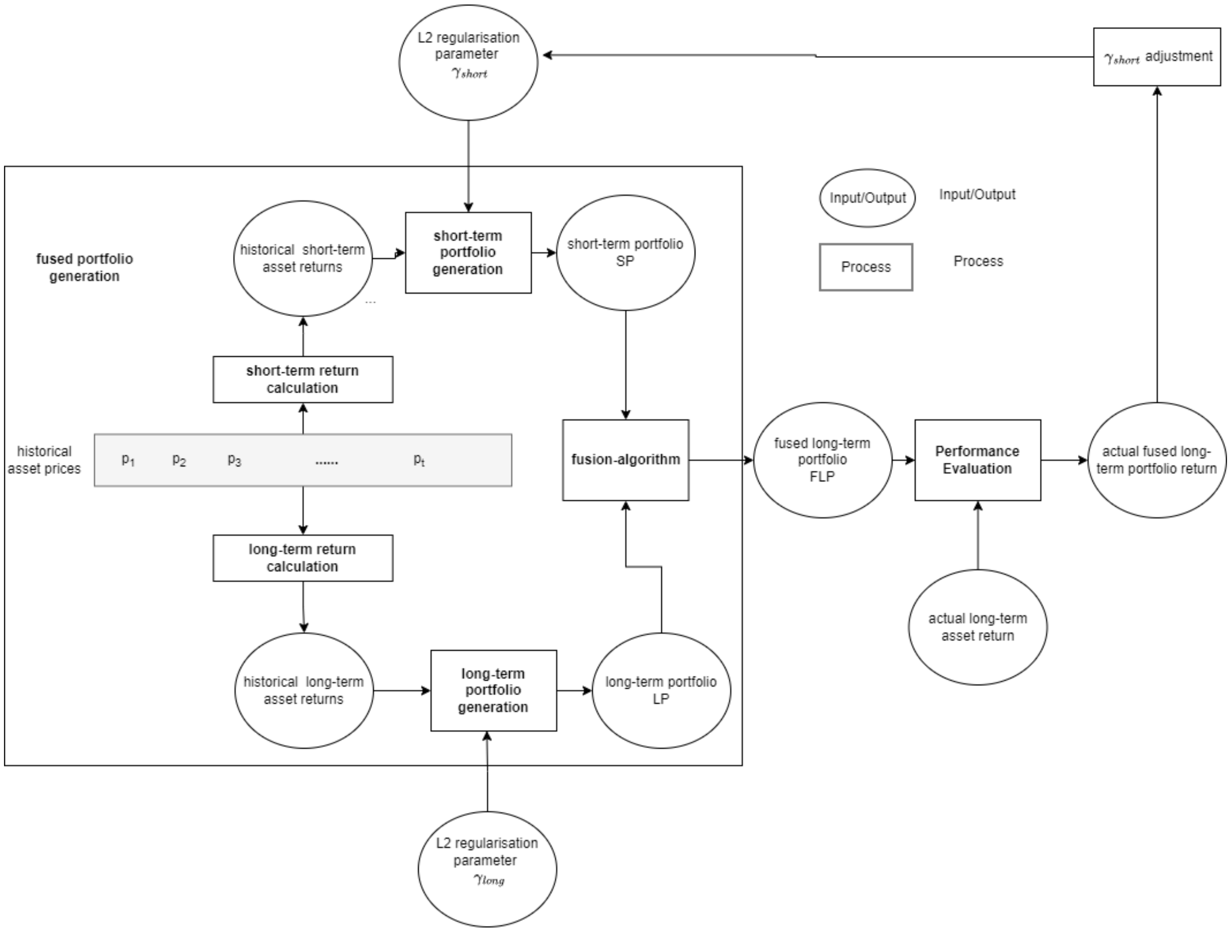

Figure 4.8: Data fusion between High-Fidelity Portfolio Optimization and low-fidelity portfolio optimization

4.3.3 Fusion Algorithm

This section introduces some fusion algorithms used in data fusion of long-term and short-term portfolios. As introduced in the last section, a short-term portfolio SP is fused into a long-term portfolio LP to generate a fused long-term portfolio FLP. However, how can we fuse SP into LP? In our work, our fusion algorithm is a simple average, as follows:

Simple Average

$$FLP = (0.5)LP + (0.5)SP \tag{4.1}$$

Reason Using Simple Average

We want to use simple averages as the fusion algorithm for controlling variables. This is the first trial of the sampling interval-based multi-fidelity model. We do not aim to maximize the performance but to prove that the sampling interval-based multi-fidelity model can improve performance. We need to control variables. Controlling variables can benefit from the simple average because the weights of the simple average are fixed.

4.3.4 Intuitive Design Principles

This section explains the intuitive design principles. However, our experiment result in section 6.3,section 6.4,and section 6.5, violates the second principle partly. There are two intuitive reasons why the fused long-term portfolio should outperform.

The first reason is that the short-term portfolio can offer the newest information and the most recent trends. The fused long-term portfolio can benefit from these trends and this information. As mentioned in the introduction 2.4, a portfolio is generated from the asset returns. The asset returns are based on the historical asset prices. According to [20][14], the momentum of asset return follows some asset return cycles. Using different sampling intervals, different asset return cycles can be detected. The short-term asset returns can display more asset return cycles than the long-term asset returns because the short-term asset returns contain more information. The expected return and covariance, calculated from the short-term asset returns, can reveal the most recent trend that can not be detected in the long-term asset returns. Hence, the short-term portfolio can catch the most recent trends that the long-term cannot.

The second reason is that the fused long-term portfolio comes from two portfolios with different *gamma*. One portfolio with smaller γ can offer a higher return with higher risk. The other portfolio with larger γ can offer a lower return with lower risk. Using a simple average to fuse one portfolio into another is equivalent to building a portfolio of portfolios. The average return of the fused long-term portfolio is just the average of the average returns of the long-term portfolio and short-term portfolio when the fusion algorithm is the simple average. However, the cumulative return and the Sharpe ratio of the fused long-term portfolio are not just the average values. It is possible that the fused long-term portfolio can outperform with a better Sharpe ratio and cumulative return than the short-term portfolio and the long-term portfolio. The worst case is that the fused long-term portfolio can outperform just one of these portfolios.

4.3.5 Trade-Off in Sampling Interval Based Multi-Fidelity Model

This section discusses the trade-off in sampling interval-based multi-fidelity model. In the classical multi-fidelity model and the prediction-based multi-fidelity model, the trade-off of the low- and high-fidelity models is for the error of the result and the computation cost. However, the trade-off in the sampling interval-based multi-fidelity models differs from the classical and prediction-based multi-fidelity models.

As introduced in the section 4.3.2, a sampling interval-based multi-fidelity model is related to high-fidelity and low-fidelity data. High-fidelity data can bring a higher return, portfolio variance, and transaction cost for the short-term portfolio when it must frequently rebalance itself. In contrast, low-fidelity data used for long-term portfolios can offer a lower return, portfolio variance, and transaction cost, while the short-term portfolio needs to be rebalanced at a lower frequency. In this section, the return represents the portfolio performance.

Figure 4.9 illustrates two examples of multi-fidelity setting. In the traditional multi-fidelity setting, the trade-off between low- and high-fidelity models is for computation cost and error. High-fidelity models can bring higher costs and lower errors, while low-fidelity models can bring lower costs and higher errors. In sampling interval-based multi-fidelity model, the trade-off between low- and high-fidelity models is between return/portfolio variance and transaction cost. High-fidelity models can bring a higher return, variance, and cost, while low-fidelity models can bring a lower return, variance, and cost. A different thing is that high-fidelity models can not guarantee that their return can always be higher than those of low-fidelity models at any time. Like figure 4.10, some outliers that low-fidelity models can outperform high-fidelity models sometimes are avoidless. It can guarantee that the variance and cost of high-fidelity models are higher than those of low-fidelity models.

In this work, we apply the fusion algorithm in portfolio optimization to generate a fused-low-fidelity model. This fused-low-fidelity model should outperform the worst one of between low- and high-fidelity models at least; see section 4.3.3. In other words, the fusion algorithm can offer a lower bounds that the fused-low-fidelity model can outperform the low-fidelity model, like Figure 4.11. But how can we find the upper bounds in a sampling interval-based multi-fidelity model? We can find the upper bounds in the sampling interval-based multi-fidelity model by optimizing the L2 regularisation parameters γ_{short}.

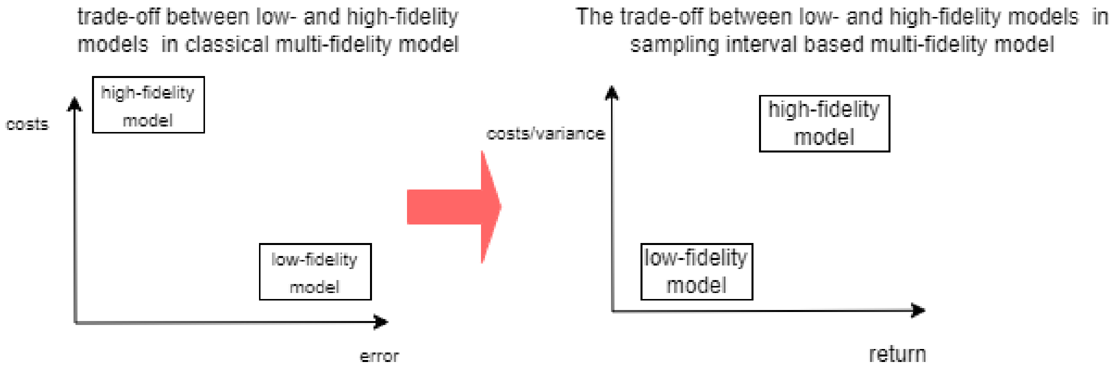

Figure 4.9: Illustrations of trade-off between low- and high-fidelity models

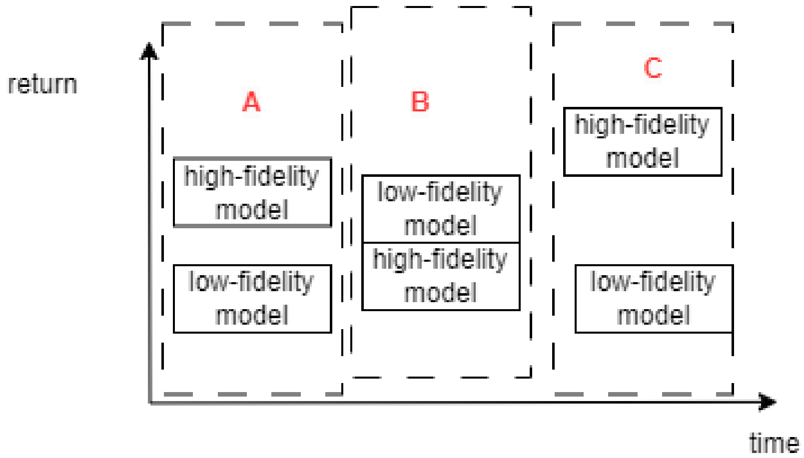

Figure 4.10: Illustrations of the profits in different time periods

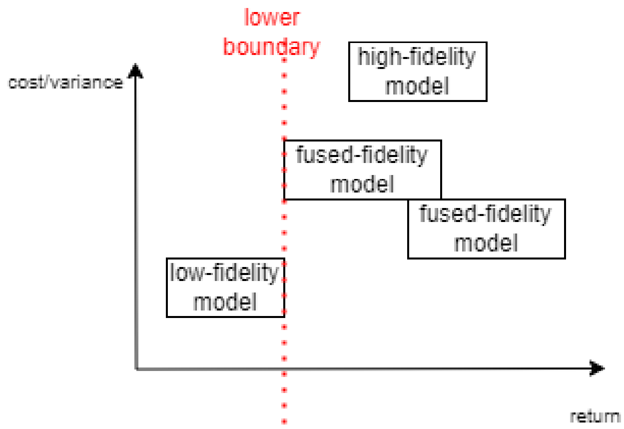

Figure 4.11: Illustrations of the fused-low-fidelity model

4.3.6 L2 γ_{short} in Sampling Interval Based Multi-Fidelity Model

This section introduces how L2 regularisation parameters γ_{short} can affect the fused long-term portfolio. As introduced in section 2.4.3, the L2 regularization parameter γ can reduce the amount of negligible weight in portfolio optimization. In other words, L2 regularization can reduce the variance and the cost in portfolio optimization. The portfolio, generated with a very small γ in portfolio optimization, can bring high risk and return because the allocation is extremely imbalanced. If we find the relationship between L2 γ_{short} and the portfolio optimization, we can improve the portfolio optimization by adjusting L2 γ_{short}. Before discussing this relationship, we must introduce two new concepts, local optimal L2 γ^*_{short} and global optimal L2 γ^*_{short}.

Global Optimal L2 γ^*_{short}

Sampling interval-based multi-fidelity model with a fixed L2 γ_{short}, a fixed L2 γ_{long}, and a fixed historical data can generate a set of fused long-term portfolios. This set can be evaluated by average return, cumulative return, and Sharpe ratio. L2 γ_{short} can affect these average return, cumulative return, and Sharpe ratio. If we change L2 γ_{short} , these average returns, cumulative returns, and Sharpe ratios must be changed. We want to find a global optimal L2 γ^*_{short}, which can make these average return, cumulative return, and Sharpe ratio as large as possible. Intuitively, a imbalance portfolio using very small L2 γ_{short} should bring higher risk and return ,compared to a more balance portfolio using large L2 γ_{short}. But, according to our experiment result, see section 6.3, the relationship between L2 γ_{short} and these returns a high positive correlation. In other words, the best way to improve the fused long-term portfolios is using a huge L2

γ_{short}. When L2 γ_{short} is very large, the allocation of assets in the short-term portfolio is a balanced portfolio. Because the long-term portfolio is also a balanced portfolio, the fused long-term portfolio must be a very balanced portfolio. It seems that equally weighted portfolios are better than mean–variance portfolios [23] [12]. In our work, L2 γ_{short} is selected from a range between 0.05 and 1.5. Hence,the global optimal L2 γ^*_{short} is 1.5.

Local Optimal L2 γ^*_{short}

In the last section, we have introduced global optimal L2 γ^*_{short}. This section describes local optimal L2 γ^*_{short}. Sampling interval-based multi-fidelity model with a fixed L2 γ_{short}, a fixed L2 γ_{long}, and a fixed historical data can generate a set of fused long-term portfolios. Now, we only focus on this whole set, but each fused long-term portfolio in this set. These fused long-term portfolios belong to time series data. Let us consider a set of fused long-term portfolios with $\gamma_{short} = x$ written as follow :

$$FLP_{\gamma_{short}=x} = [FLP_{1,x}, FLP_{2,x}, ..., FLP_{T,x}] = [FLP_{t,x}] \tag{4.2}$$

Where :

- $t \in [1, T]$ is the time point.
- $x \in [0.01, 1.5]$ is L2 γ_{short}

The local optimal L2 γ^*_{short} is x at time t ,which can make :

$$R(FLP_{t,x}) \geq R(FLP_{t,y}), \forall y \in [0.05, 1.5] \tag{4.3}$$

Where :

- $R(FLP_{t,x})$: the actual portfolio return
- $t \in [1, T]$: the time point.
- $x \in [0.05, 1.5]$

If we can predict such local optimal L2 γ^*_{short} at each time t, we can maximize the return of the fused-long-term portfolio. The maximized return of the fused-long-term portfolios is our trade-off 's upper bounds in the sampling interval based multi-fidelity model. Actually, according to our experiment result in section 6.5, the distribution of local optimal L2 γ^*_{short} is similar to the two-point distribution when we ignore some outliers; see figure4.12. To predict a data point in a two-point distribution is equivalent to the binary prediction.

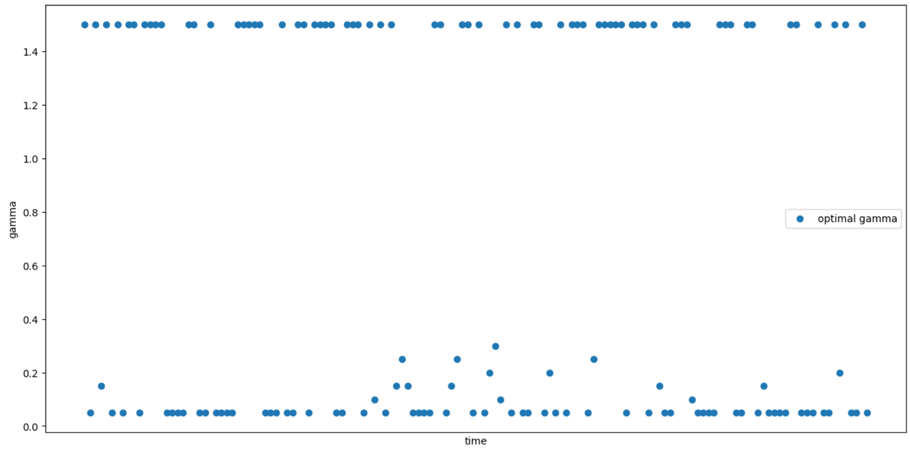

Figure 4.12: Local Optimal γ^* distribution (from section 6.4)

How to predict the local optimal γ^*

As introduced in the last section, the local optimal γ^* distribution can be regarded as a two-point distribution. The best way to predict the data point in a two-point distribution, or binary prediction, is a machine learning or deep learning model. It is out of our work expectations. We try to use Gaussian process classification to predict the local optimal γ^* at time t.

4.3.7 Conclusion

The sampling interval based multi-fidelity model is a trial directly using multi-fidelity method to improve the portfolio optimization. In this model, the short-term portfolio from the high-fidelity model is fused into a long-term portfolio from the low-fidelity model to generate a fused long-term portfolio. This fused long-term portfolio owns a better lower bound than the long-term one. The corresponding upper bounds of the fused long-term portfolio are also mentioned. We also provide a possible idea about how to get closer to the upper bounds. Generally, the sampling interval based multi-fidelity model aims to improve portfolio optimization. The problem of improving portfolio optimization is converted into a binary prediction problem using a multi-fidelity method.

5 Implementation

This section describes the implementation of this work. In our work, we do not implement some classical methods one more time because many famous open-source packages can offer outstanding implementations of these traditional approaches. Our work focuses on designing a feasible multi-fidelity model to improve portfolio optimization. The link for the GitHub repository of the implementation is $https://github.com/ZhangArcher/MT_github.git$.

5.1 UML Package Diagram of the Implementation

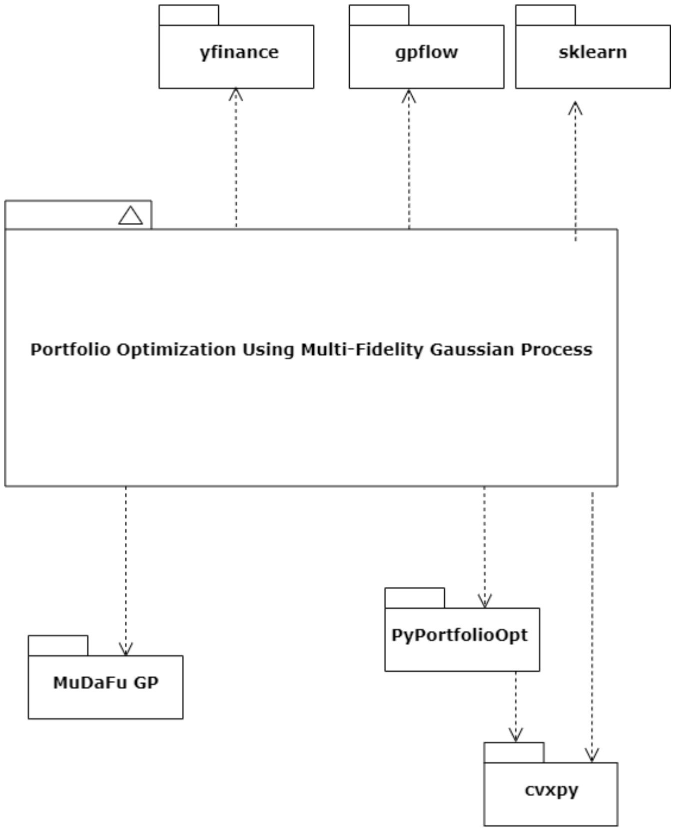

Figure 5.1: UML Package Diagram of the Implementation

Figure 5.1 illustrates the package diagram of the implementation. In our work, optimization problems are solved by PyPortfolioOpt, a very famous library that has implemented many portfolio optimization methods. The portfolio optimization problems with standard objectives and constraints can be solved as a convex optimization by cvxpy, the fantastic python-embedded modelling language for convex optimization, in PyPortfolioOpt. MuDaFu GP is used directly to implement the prediction based multi-fidelity model. MuDaFu GP can offer some outstanding implementations of classical multi-fidelity methods. Gpflow is used to create basic GP regression models for the sampling interval based multi-fidelity model. Yfinance, an open-source tool that uses Yahoo's publicly available APIs to collect financial data, is used to collect the financial data in our work. scikit-learn(sklearn) can offer some outstanding implementations of machine learning. We predict the local optimal γ^*_{short} using Gaussian process classification which is implemented by scikit-learn.

5.2 UML class diagram of the implementation

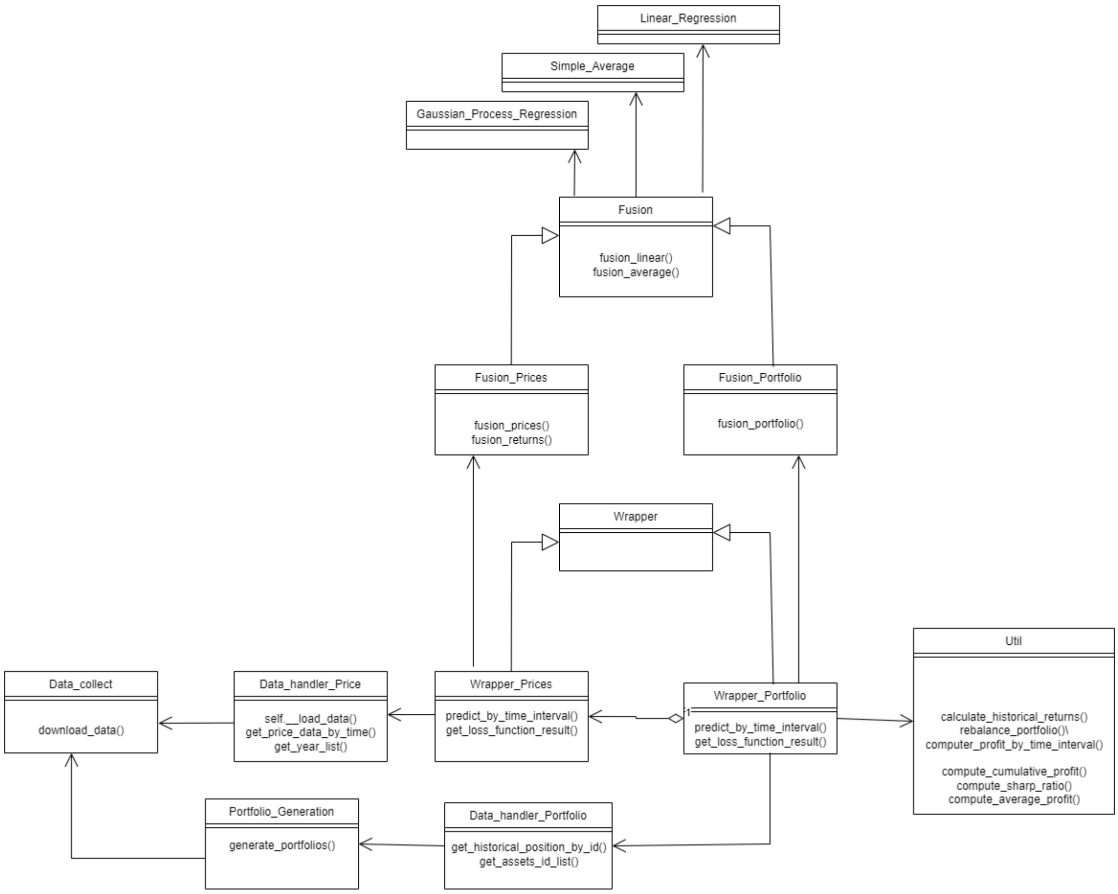

Figure 5.2: UML Class Diagram of the Implementation

Figure 5.2 illustrates the UML class diagram of the implementation. In order to keep things simple, the illustrated classes contain only the most important methods. All classes have been added the necessary code comments. More detailed documentation can be found in the GitHub repository $https://github.com/ZhangArcher/MT_github.git$.

6 Experiment

In this Chapter, we introduce some experiments about the sampling interval based multi-fidelity model , to prove that fused long-term portfolios can outperform the long-term portfolios.

6.1 Experiment 1: Is a Larger Rolling Window Size Better For the Calculation of the Expected Asset Return ?

6.1.1 Experiment Goal

We want to verify whether or not using a larger rolling window size can generate a better-expected asset return in data preprocessing.

6.1.2 Experiment Design

1. calculates the expected asset return from the historical prices of stocks using different methods with different rolling window sizes
2. calculates the error between the expected and actual asset returns using mean absolute error.
3. calculate the Pearson correlation coefficient between the error and the rolling window sizes

6.1.3 Data set

In this experiment, the data set is the historical monthly prices of stocks between $1995 - 01 - 01$ and $2015 - 01 - 01$ and the historical daily prices of stocks between $1995 - 01 - 01$ and $2015 - 01 - 01$.

The relative stocks are:
"AMD","BA","CSCO","DHR","INTC","JPM","NKE","PG","TXN","WMT".

6.1.4 Experiment Result

Error with different rolling window sizes

Figure 6.1 describes the error between the expected and actual asset returns with the different rolling window sizes when the data set is the historical monthly prices of stocks between 1995 − 01 − 01 and 2015 − 01 − 01. As we can see clearly, the errors decrease with increasing the rolling window size until that rolling window size is 40. And then, the errors rebound from the lowest point and never go down in the future. The reasonable rolling window size is about 40. In other words, we should calculate expected monthly asset returns using the data of the most recent 3 or 4 years.

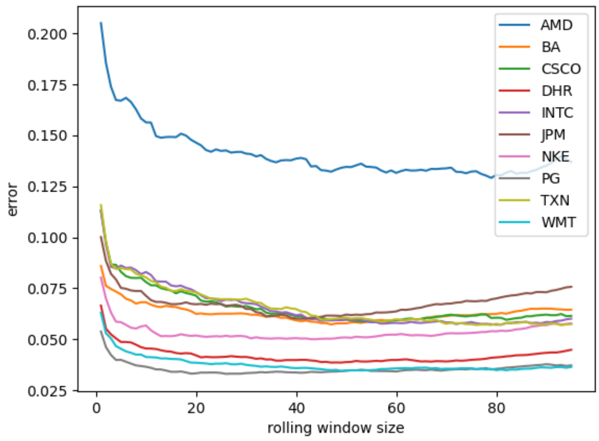

Figure 6.1: Development of the errors between the expected asset return and the actual asset return with the different rolling window sizes (based on the historical monthly prices)

Figure 6.2 describes the error between the expected and actual asset returns with the different rolling window sizes when the data set is the historical daily prices of stocks between 1995 − 01 − 01 and 2015 − 01 − 01. The errors drop rapidly at the beginning with increasing rolling window size. But the errors level off earlier, compared to Figure 6.1. It begins to bound back when the roll window size exceeds 900. Hence, the sensible range of the historical data is the most recent 2 or 3 years for calculating expected daily asset returns.

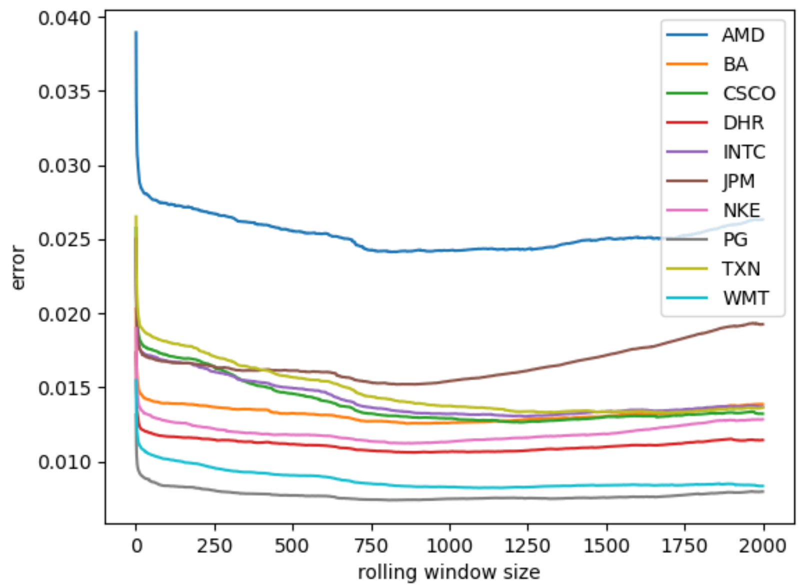

Figure 6.2: Development of the errors between expected asset return and actual asset return with the different rolling window sizes (based on the historical daily prices)

Pearson correlation coefficients between errors and rolling window sizes

Figure 6.3 describes the correlation coefficient between the error and the rolling window sizes when the data set is the historical monthly prices of stocks between $1995-01-01$ and $2015-01-01$. ID represents the stock ID, whose errors between the expected asset return and the actual asset return are calculated. Range means the range of the rolling window size. For example, the Pearson correlation coefficient between the error about AMD and the rolling window sizes is -0.87189272 in the rolling window size range between 1 and 12. When the range is small, like [1,12], all coefficients are between -0.8 and -1. In other words, it represents a very high negative correlation. For this range, increasing the rolling window size can certainly reduce the error. When enlarging the range, these coefficients get closer to 0. The coefficient of PG and the coefficient of NKE are especially under -0.1 when the range is [1,96]. A coefficient, being in a range [-0.19 ,0.19] , implies a very low correlation.

In other words, when the rolling window size is small enough, the relationship between errors and rolling window sizes is a high negative correlation. The high negative correlation disappears with increasing rolling window size.

range \ id	AMD	BA	CSCO	DHR	INTC	JPM	NKE	PG
[1, 12]	-0.871889272	-0.87492	-0.83144	-0.83719	-0.73153	-0.87428	-0.80828	-0.85869
[1, 24]	-0.878160683	-0.87914	-0.84625	-0.83962	-0.85787	-0.80471	-0.76915	-0.82233
[1, 48]	-0.85805147	-0.88046	-0.88691	-0.80721	-0.92623	-0.83021	-0.66813	-0.63051
[1, 96]	-0.741281074	-0.41136	-0.72235	-0.4487	-0.82809	-0.07279	-0.07637	-0.0994

Figure 6.3: Pearson correlation coefficients between errors and rolling window sizes, when using different ranges of rolling window sizes and the monthly data (For example, the Pearson correlation coefficient between the error about AMD and the rolling window sizes, is -0.87189272 in the rolling window size range between 1 and 12)

Similarly, figure 6.4 describes the correlation coefficient based on the historical daily prices between $1995-01-01$ and $2015-01-01$ At the beginning, all coefficients are between -0.8 and -1 for the small range [1,10]. In other words, the error can be reduced significantly by increasing the rolling window size within a range between 1 and 10. After that, these coefficients rise up to around -0.5, when the range is enlarged until the range [1,180]. After [1,180], most coefficients rebound and sink again with increasing rolling window size. This rebound ends at range [1,1000]. When the range is [1,1000], most coefficients are between -0.8 and -1 again. In the rest of these ranges, the coefficients climb to -0.2 again.

Totally, the coefficients between errors and rolling window sizes fluctuate, ranging from $-0.2$to -0.9 . Combined with Figure 6.2, it is clear that a larger rolling window size can offer some limited help to reduce errors between the expected asset return and the actual asset return.

id / range	AMD	BA	CSCO	DHR	INTC	JPM	NKE	PG	TXN	WMT
[1, 10]	−0. 863079105	−0. 842716546	−0. 861116471	−0. 862433842	−0. 861700668	−0. 841580751	−0. 884110877	−0. 847694558	−0. 868374695	−0. 874897302
[1, 30]	−0. 735586523	−0. 704668375	−0. 707269818	−0. 709269942	−0. 696724632	−0. 695938411	−0. 731153751	−0. 713223774	−0. 72105751	−0. 727989535
[1, 90]	−0. 616161219	−0. 602654918	−0. 596322151	−0. 607559633	−0. 563202393	−0. 571440961	−0. 646809728	−0. 709312702	−0. 639788535	−0. 690190906
[1, 180]	−0. 577182871	−0. 53420142	−0. 614247616	−0. 565613287	−0. 564165512	−0. 524009059	−0. 658063417	−0. 678274266	−0. 650447208	−0. 710421187
[1, 360]	−0. 708224502	−0. 623736888	−0. 842576521	−0. 617276756	−0. 803185171	−0. 623016551	−0. 781417715	−0. 770731623	−0. 852035245	−0. 838093428
[1, 720]	−0. 87539907	−0. 819113813	−0. 961788787	−0. 788852438	−0. 939852806	−0. 733724583	−0. 818494542	−0. 829178543	−0. 946839256	−0. 888304722
[1, 1000]	−0. 923798629	−0. 900183415	−0. 964074404	−0. 875238899	−0. 96779424	−0. 857725566	−0. 865485651	−0. 8403476	−0. 969896254	−0. 927439124
[1, 1250]	−0. 890608932	−0. 862111966	−0. 942742274	−0. 85411242	−0. 952946208	−0. 707368786	−0. 780328439	−0. 769594626	−0. 961846534	−0. 913914208
[1, 1500]	−0. 798861416	−0. 746050907	−0. 900234203	−0. 745502228	−0. 912149749	−0. 227378645	−0. 622158526	−0. 708920043	−0. 94105662	−0. 862476685
[1, 1750]	−0. 675148583	−0. 558807475	−0. 840526523	−0. 55950162	−0. 850852406	0. 295088354	−0. 338321993	−0. 613538318	−0. 913039928	−0. 80674983
[1, 2000]	−0. 449100758	−0. 232353023	−0. 773352902	−0. 276918042	−0. 775029728	0. 589171464	0. 040791935	−0. 407486739	−0. 870956663	−0. 754355824

Figure 6.4: Pearson correlation coefficients between errors and rolling window sizes, when using different ranges of rolling window sizes and daily data(For example, the Pearson correlation coefficient between the error about AMD and the rolling window sizes is -0.61616 within a rolling window size range between 1 and 90)

Compare the error with the actual asset returns

In this experiment, performance evaluation is based on mean absolute error. Ideally, the expected asset return is a measure of the center of the distribution of the return. The actual asset return should follow this distribution. However, it is well-known that the difference between the expected asset return and the actual asset return is large. As we can see in Figure 6.1, the minimal absolute error between them is larger than 0.025 when the average monthly return of AMD between $2000-01-01$ and $2015-01-01$ is only 0.00196. The percentage error between them can be 1275%. The error is very much larger than the average true value.

6.1.5 Conclusion

Using a larger rolling window size can generate a better expected asset return when the rolling window size is in a suitable range. When the rolling window sizes are huge, the errors can not be reduced anymore. Although the reduction can decrease the error to 50%, the errors are still very large compared to the actual asset returns.

6.2 Experiment 2: Can a Larger Rolling Window Size Bring a Better Portfolio?

6.2.1 Experiment Goal

We want to verify whether or not using a larger rolling window size can generate a better portfolio in portfolio optimization.

6.2.2 Experiment Design

1. generate portfolios with different rolling window sizes in portfolio optimization
2. calculates cumulative returns, average returns, and Sharpe ratios.
3. calculate the correlation coefficient between the rolling window sizes and the average returns
4. calculate the correlation coefficient between the rolling window sizes and the cumulative returns
5. calculate the correlation coefficient between the rolling window sizes and the Sharpe ratios

6.2.3 Data set

In this experiment, the data set is the historical monthly prices between $1995 - 01 - 01$ and $2015 - 01 - 01$ and the historical daily prices between $1995 - 01 - 01$ and $2015 - 01 - 01$.

The relative stocks are:
"AMD","BA","CSCO","DHR","INTC","JPM","NKE","PG","TXN","WMT".

6.2.4 Experiment Result

Average Portfolio Returns and its Correlation Coefficients

Figure 6.5 describes the development of the average portfolio returns from different portfolios when these portfolios are generated by portfolio optimization with the historical monthly prices and different rolling window sizes. As we can see, the average return goes up and down with an upward trend until the rolling window size is around 43. The peak of the average return is 0.0143 (1.43%). After the peak, the average returns slump into 0.0082(0.82%) with some bounce back. The Pearson correlation coefficient between average portfolio returns and the corresponding rolling window sizes is -0.315828878607737, when the data set is the historical monthly prices. There is only a low negative correlation between them.

Figure 6.6 describes the development of the average portfolio returns from different portfolios when these portfolios are generated by portfolio optimization with the historical daily prices and different rolling window sizes. As we can see, the development of the average portfolio returns in figure 6.6 is similar to the development in figure 6.5. The average return goes up and down with an upward trend until the rolling window size is around 800. The peak of the average return is 0.00053 (0.053%). And then, the average returns sink into 0.00037(0.037%). The Pearson correlation coefficient between average portfolio returns and the corresponding rolling window sizes is -0.483744502406371, when the data set is the historical daily prices. There is only a moderate negative correlation between them.

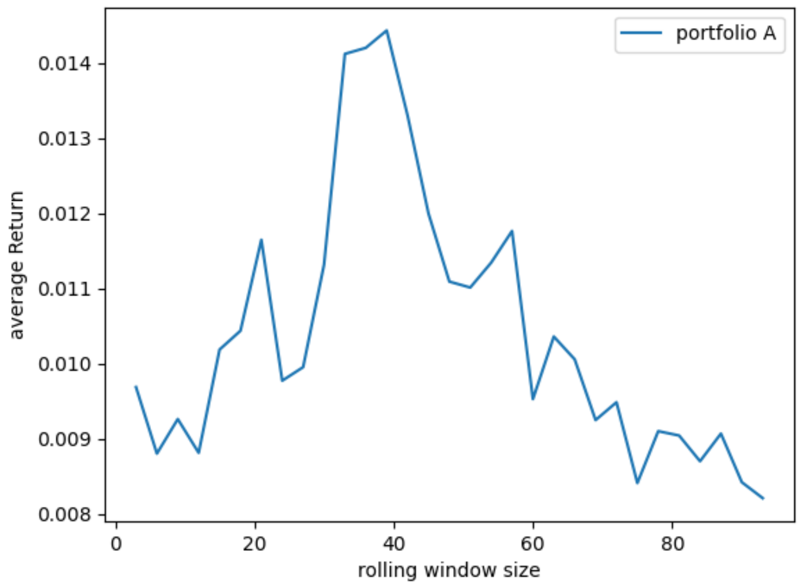

Figure 6.5: Development of the average portfolio returns from different portfolios that are generated by portfolio optimization with different rolling windows sizes (based on the historical monthly prices)

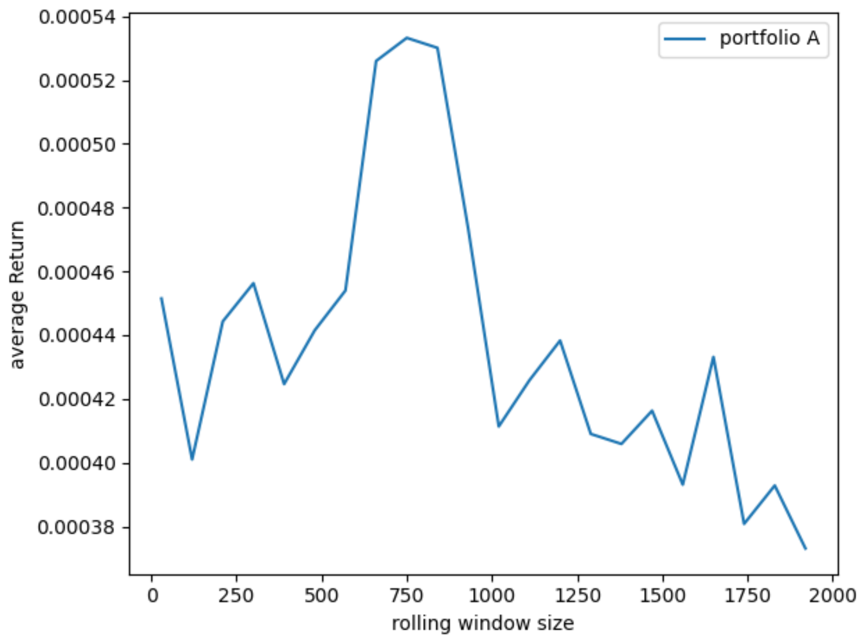

Figure 6.6: Development of the average portfolio returns from different portfolios that are generated by portfolio optimization with different rolling windows sizes (based on the historical daily prices)

Cumulative Returns and its Correlation Coefficients

Figure 6.7 describes the development of the cumulative returns from different portfolios when these portfolios are generated by portfolio optimization with the historical monthly prices and different rolling window sizes. As we can see, the average return goes up and down with an upward trend until the rolling window size is around 43. The peak of the average return is 4.5 (450%). After the peak, the average returns slump into 0.75 (75%) with some bounce backs. The Pearson correlation coefficient between cumulative returns and the corresponding rolling window sizes is -0.483744502406371 , when the data set is the historical monthly prices. There is only a moderate negative correlation between them.

Figure 6.8 describes the development of the cumulative returns from different portfolios when these portfolios are generated by portfolio optimization with the historical daily prices and different rolling window sizes. The average return goes up and down with an upward trend until the rolling window size is around 750. The peak of the average return is 2.5 (250%) . And then, the average returns sink into 0.6 (60%). The

Pearson correlation coefficient between cumulative returns and the corresponding rolling window sizes is -0.821987539560751 , when the data set is the historical daily prices. There is only a very high negative correlation between them.

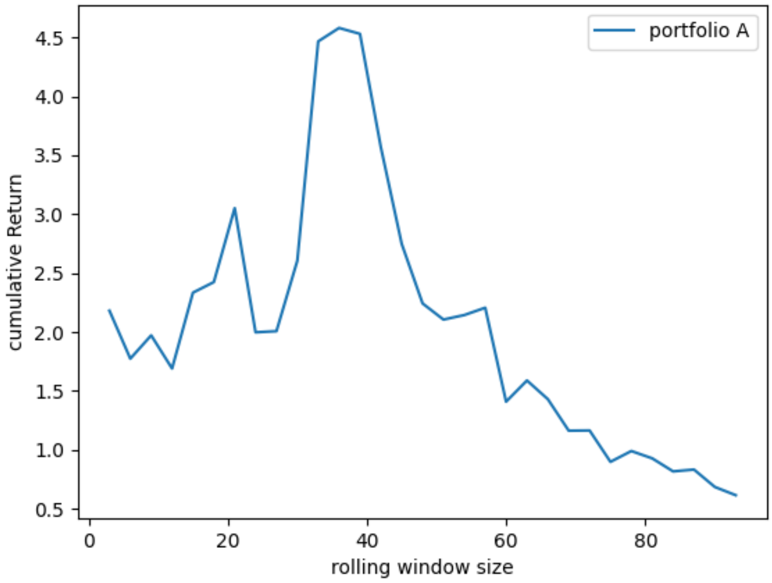

Figure 6.7: Development of cumulative returns from different portfolios that are generated by portfolio optimization with different rolling windows sizes (based on the historical monthly prices)

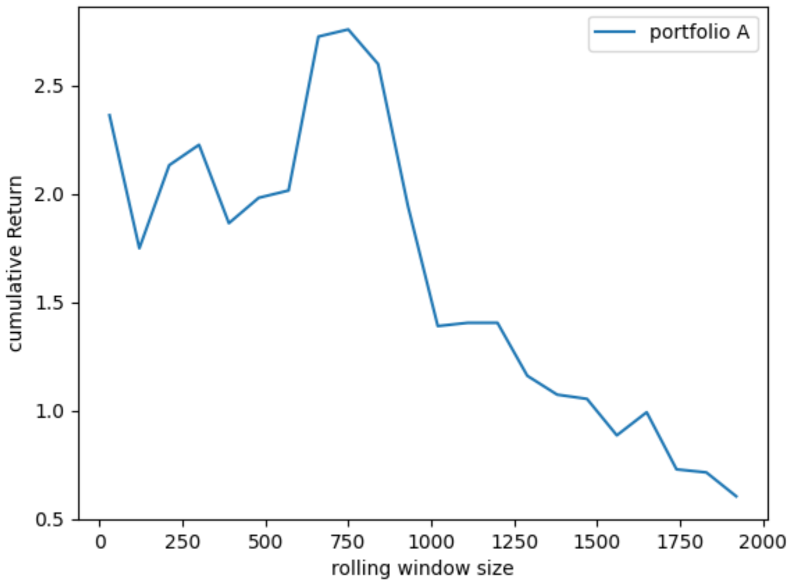

Figure 6.8: Development of cumulative returns from different portfolios that are generated by portfolio optimization with different rolling windows sizes (based on the historical daily prices)

Sharp Ratio and its correlation coefficients

Figure 6.9 describes the development of Sharpe ratios from different portfolios when these portfolios are generated by portfolio optimization with the historical monthly prices and different rolling window sizes. As we can see, the average return goes up and down with an upward trend until the rolling window size is around 40. The peak of the average return is 2.5. After the peak, the average returns slump to 0.9 with some bounce-backs. The Pearson correlation coefficient between Sharpe ratios and the corresponding rolling window sizes is -0.513842048881781 , when the data set is the historical monthly prices. There is only a moderate negative correlation between them.

Figure 6.10 describes the development of Sharpe ratios from different portfolios when these portfolios are generated by portfolio optimization with the historical daily prices and different rolling window sizes. The average return goes up and down with an upward trend until the rolling window size is around 750. The peak of the average return is 2.2. And then, the average returns sink to 0.9. The Pearson correlation coefficient between Sharpe ratios and the corresponding rolling window sizes is -

0.715591403398691, when the data set is the historical daily prices. There is only a high negative correlation between them.

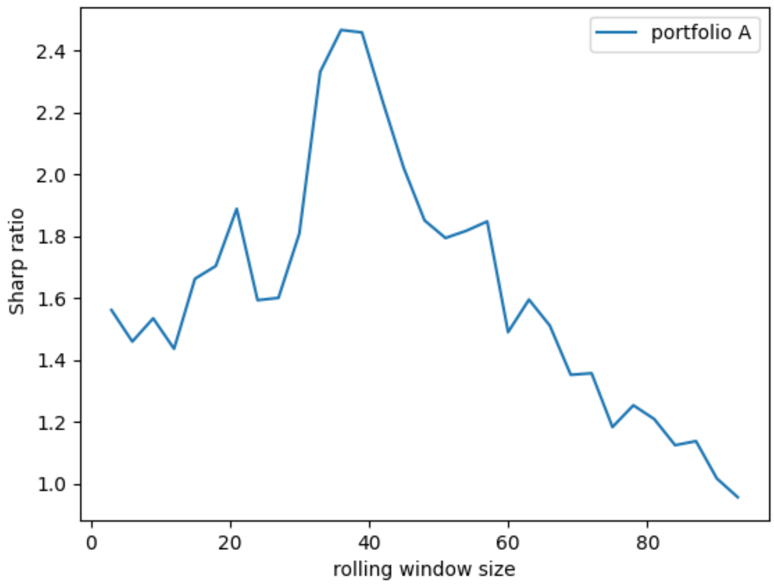

Figure 6.9: Development of Sharpe ratios from different portfolios that generated by portfolio optimization with different rolling windows sizes (based on the historical monthly prices)

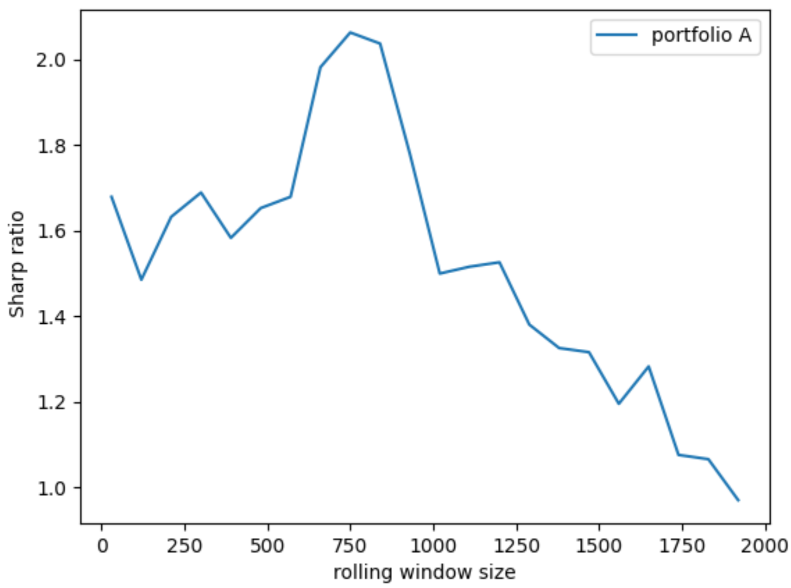

Figure 6.10: Development of Sharpe ratios from different portfolios that generated by portfolio optimization with different rolling windows sizes (based on the historical daily prices)

6.2.5 Conclusion

It is similar to the section; a better portfolio can be generated using a larger rolling window size when the rolling window size is in a suitable range. But when the rolling window size is too large, the portfolio becomes terrible rapidly.

6.3 Experiment 3: Sampling Interval Based Multi-Fidelity Model Using Different γ_{short}

6.3.1 Experiment Goal

In this experiment, we want to test the performance of the fused long-term portfolio, compared to the short-term portfolio and long-term portfolio, when using different γ_{short}. The range of γ_{short} is between 0.05 and 1.5.

6.3.2 Experiment Design

1. generate long-term portfolios with $\gamma_{long} = 1$ in portfolio optimization
2. generate short-term portfolios with different γ_{short} in portfolio optimization
3. fuses the short-term portfolio with γ_{short} into the long-term portfolio to generate a fused long-term portfolio with γ_{short} when the long-term portfolio and the short-term portfolio are at the same time point.
4. calculate cumulative returns, average returns, and Sharpe ratios for the short-term portfolio with γ_{short}, long-term portfolio, and fused long-term portfolio.

6.3.3 Data set

In this experiment, the data set is the historical monthly prices between $1995 - 01 - 01$ and $2015 - 01 - 01$ and the historical daily prices between $1995 - 01 - 01$ and $2015 - 01 - 01$.

The relative stocks are:
"AMD","BA","CSCO","DHR","INTC","JPM","NKE","PG","TXN","WMT".

6.3.4 Experiment Result

Average Portfolio Returns and its Correlation Coefficients

Figure 6.11 describes the development of average portfolio returns when using different γ_{short}. Because a long-term portfolio, using the fixed $\gamma_{long} = 1$, it can be a baseline. Long-term, short-term, and fused long-term portfolios rebalance themselves every month. Hence, the average portfolio returns of the long-term portfolio, short-term portfolio, and fused long-term portfolio are the monthly return The red curve is the average monthly return of the short-term portfolio based on daily rebalance. We assume that there are 30 days every month. Hence, the red curve is the result that the average daily return of the short-term portfolio based on daily rebalance is multiplied by 30. As we can see, the peak of the red one is closer to 0.018 (1.8%) at the beginning, while γ_{short} is closer to 0. After that, the average return dwindles with increasing γ_{short} without any recovery. There is a similar trend for the average of short-term portfolios and the average of fused long-term portfolios. Both fall at first and rebound from $\gamma_{short} = 0.15$. After that, they never go down. The baseline, the average portfolio returns of a long-term portfolio, always be 0.00303990108268414 (0.303990108268414%) , due to the fixed $\gamma_{long} = 1$

Table 6.1 describes the Pearson correlation coefficients between the average return and γ_{short} for the short-term portfolio, the fused long-term portfolio, and the short-term

	fused long-term portfolio	short-term portfolio	short-term portfolio based on daily rebalance
Pearson correlation coefficients	0.91296972	0.91296972	-0.697389618

Table 6.1: Pearson correlation coefficients between the average return and γ_{short}

portfolio based on daily rebalance. Pearson correlation coefficient of fused long-term portfolio and the coefficient of short-term portfolio is in the range between 0.8 and 1. There is a high positive correlation between the average return of the fused long-term portfolio (the short-term portfolio). In other words, increasing γ_{short} can increase the average return of a fused long-term portfolio(the short-term portfolio).

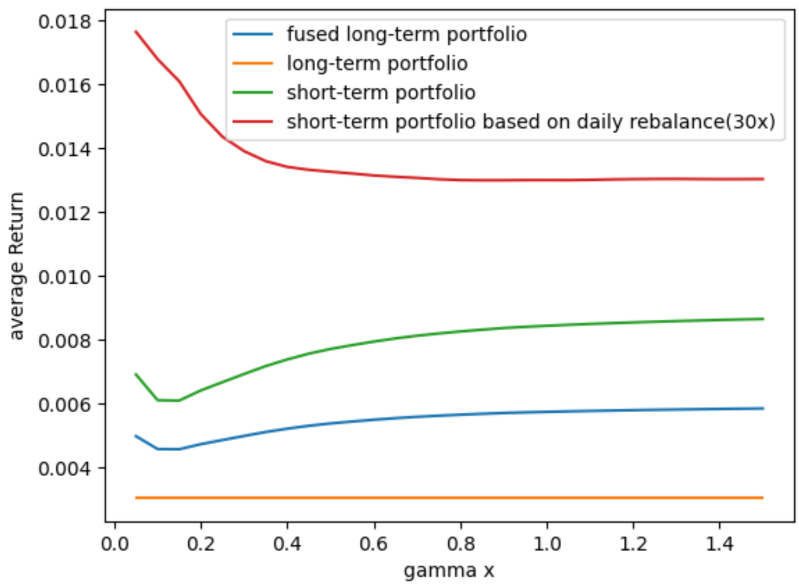

Figure 6.11: Development of average portfolio returns when using different γ_{short}

Cumulative Returns and its Correlation Coefficients

Figure 6.12 describes the development of cumulative returns when using different γ_{short}. The long-term portfolio is the baseline due to the fixed $\gamma_{long} = 1$. The cumulative returns of the long-term portfolio is 1.00085 (100.085%). Compared to figure 6.11, the trend of the short-term portfolio based on daily rebalance is different. The red one rises

	fused long-term portfolio	short-term portfolio	short-term portfolio based on daily rebalance
Pearson correlation coefficients	0.909286988309415	0.916197063052294	-0.602477854280436

Table 6.2: Pearson correlation coefficients between the cumulative return and γ_{short}

up at first until $\gamma_{short} = 0.2$. The corresponding peak is 3.8 (380%). After $\gamma_{short} = 0.2$, its cumulative returns rebound from the peak and fall into the lowest level. The trend of the short-term portfolio and the trend of the fused long-term portfolio are similar to Figure 6.12. Both of them can outperform the long-term portfolio.

Table 6.2 describes the Pearson correlation coefficients between the cumulative return and γ_{short} for the short-term portfolio, the fused long-term portfolio, and the short-term portfolio based on daily rebalance. Similar to table 6.1, there is a high positive correlation between the cumulative return of fused long-term portfolio (the short-term portfolio) and γ_{short}.

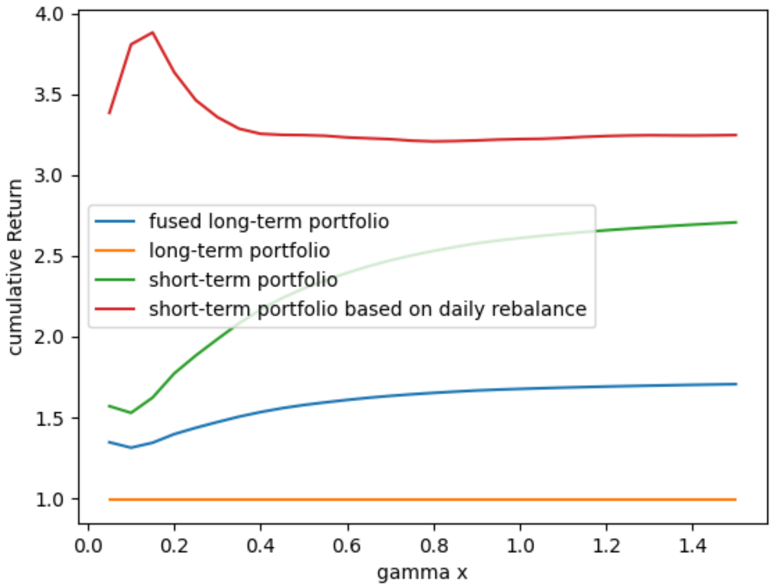

Figure 6.12: Development of cumulative returns when using different γ_{short}

	fused long-term portfolio	short-term portfolio	short-term portfolio based on daily rebalance
Pearson correlation coefficients	0.906947959067862	0.908923985557281	0.272903586203281

Table 6.3: Pearson correlation coefficients between the Sharpe ratios and γ_{short}

Sharpe ratios and its correlation coefficients

Figure 6.13 describes the development of Sharpe ratios when using different γ_{short}. The long-term portfolio is the baseline due to the fixed $\gamma_{long} = 1$. The Sharp ratio of the long-term portfolio is around 0.4. Similar to figure 6.12, the Sharpe ratios of the short-term portfolio, based on daily rebalance, rise up at first until $\gamma_{short} = 0.2$. The corresponding peak is 1.712. After $\gamma_{short} = 0.2$, its Sharpe ratios rebound from the peak and fluctuate. Compared to Figure 6.12 and Figure 6.11, the difference point is that the Sharpe ratios of the short-term portfolio can outperform the Sharpe ratios of the short-term portfolio based on daily rebalance when γ_{short} is larger than 1.1. The rest are similar.

Table 6.3 describes the Pearson correlation coefficients between the Sharpe ratios and γ_{short} for the short-term portfolio, the fused long-term portfolio, and the short-term portfolio based on daily rebalance. Similarly, there is a high positive correlation between the Sharpe ratios of the fused long-term portfolio (the short-term portfolio) and γ_{short}. Compared to table 6.2, the difference thing is that Pearson correlation coefficients of short-term portfolios based on daily rebalance are positive. Hence, there is a low positive correlation between the Sharpe ratios of the short-term portfolio based on daily rebalance and γ_{short}.

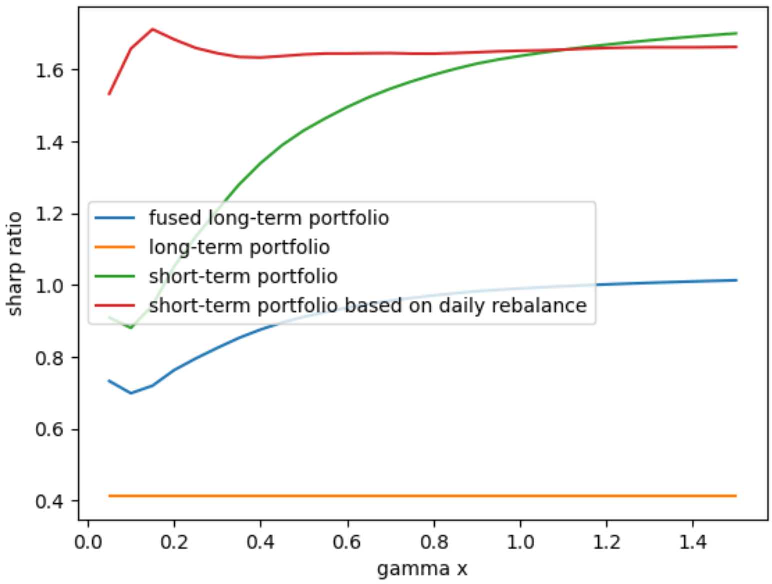

Figure 6.13: Development of Sharpe ratios when using different γ_{short}

6.3.5 Conclusion

As we can see, γ_{short} can affect the returns of these portfolios when focusing on the whole data set over long time periods. We can find a relationship between γ_{short} and the returns of these portfolios easily. The distribution of Sharpe ratios / cumulative returns / average return looks like a continuous distribution. It is possible to find a global optimal γ_{short} to improve Sharpe ratios, cumulative returns, and average returns.

6.4 Experiment 4: Local Optimal γ^*_{short} Distribution in Sampling Interval Based Multi-Fidelity Model

6.4.1 Experiment Goal

In this experiment, we want to describe the local optimal γ^*_{short} distribution at different time points (for each data point) , when the range of γ_{short} is between 0.05 and 1.5.

6.4.2 Experiment Design

1. generate long-term portfolios with $\gamma_{long} = 1$ at time t

2. generate short-term portfolios with different γ_{short} at time t
3. fuse the short-term portfolio with $\gamma_{short} = x$ into long-term portfolio with $\gamma_{long} = 1$, to generate a fused long-term portfolio with $\gamma_{short} = x$ at time t
4. find out the local optimal γ^*_{short} from these fused long-term portfolio with $\gamma_{short} = x$ at time t , when the fused long-term portfolio with $\gamma_{short} = \gamma^*_{short}$ should bring the highest return at time t ,compared to the other portfolio with other γ_{short})
5. repeat for each time point within the range of the historical data set.
6. display these γ^*_{short} at different time points.

6.4.3 Data set

In this experiment, the data set is the historical monthly prices between $1995 - 01 - 01$ and $2015 - 01 - 01$ and the historical daily prices between $1995 - 01 - 01$ and $2015 - 01 - 01$.

The relative stocks are:
"AMD","BA","CSCO","DHR","INTC","JPM","NKE","PG","TXN","WMT".

6.4.4 Experiment Result

Figure 6.14 describe local optimal γ^*_{short} distribution. As we can see, the largest local optimal γ^*_{short} is only 1.5 because our experiment setting about the range of γ_{short} is just between 0 and 1.5. Most of these local optimal γ^*_{short} are closer to the extreme value in the default range of γ_{short}. It looks like a particular step function or a special two-point distribution.

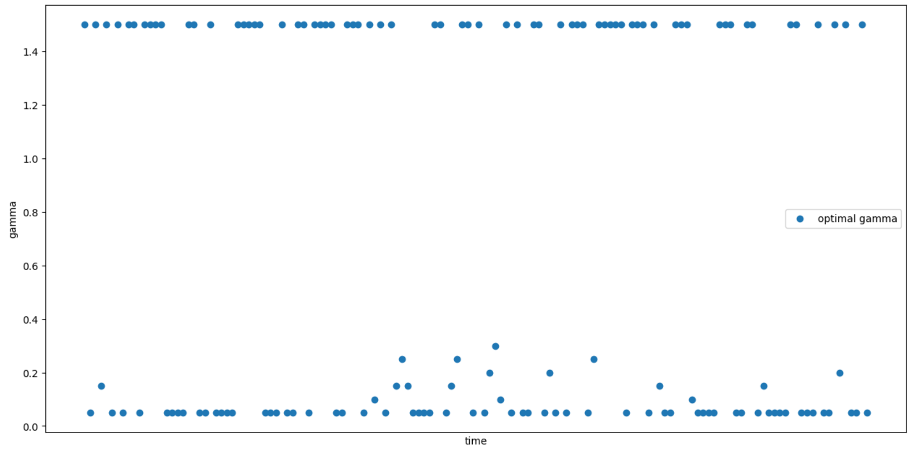

Figure 6.14: optimal γ^*_{short} distribution

6.4.5 Conclusion

The local optimal γ^*_{short} distribution is similar to a two-point distribution. If we can ignore some non-extreme values, it can be regarded as a two-point distribution.

6.5 Experiment 5: Sampling Interval based Multi-Fidelity Model with the local optimal γ^*_{short}

6.5.1 Experiment Goal

In this experiment, we want to test the performance of the fused long-term portfolio using the local optimal γ^*_{short} , compared to the short-term and long-term portfolios.

6.5.2 Experiment Design

1. generate local optimal γ^*_{short} for each time point t like Experiment 4 long-term portfolios with $\gamma_{long} = 1$ at time t
2. generate the fused long-term portfolio with local optimal γ^*_{short}
3. calculate cumulative returns, average returns, and Sharpe ratios for the fused long-term portfolio with local optimal γ^*_{short}

4. compare with Experiment 3

6.5.3 Data set

In this experiment, the data set is the historical monthly prices between $1995-01-01$ and $2015-01-01$ and the historical daily prices between $1995-01-01$ and $2015-01-01$.

The relative stocks are:
"AMD","BA","CSCO","DHR","INTC","JPM","NKE","PG","TXN","WMT".

6.5.4 Experiment Result

Average Portfolio Returns

Figure 6.15, based on figure 6.11 , describes the development of average portfolio returns when using different γ. Because these local optimal γ^*_{short} are fixed for each time point, the corresponding average portfolio return does not change with different $\gamma short$. As we can see, the average portfolio return of the fused long-term portfolio using a local optimal γ^*_{short} is about 0.013 (1.3%). It outperforms the fused long-term portfolio, short-term portfolio, and long-term portfolio.

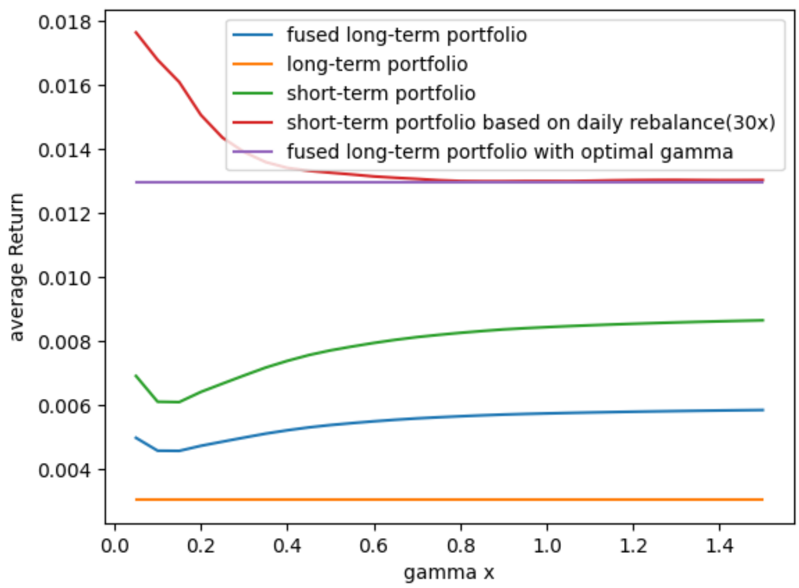

Figure 6.15: Development of average portfolio returns when using the local optimal γ^*_{short}

Cumulative Returns

Figure 6.16, based on figure 6.12 , describes the development of cumulative returns when using different γ. Because these local optimal γ^*_{short} are fixed, the corresponding average portfolio return does not change with different γ_{short}. As we can see, the cumulative returns of the fused long-term portfolio using the local optimal γ^*_{short} is about 4.6(460%). It is the best among all of these portfolios.

Sharpe ratios

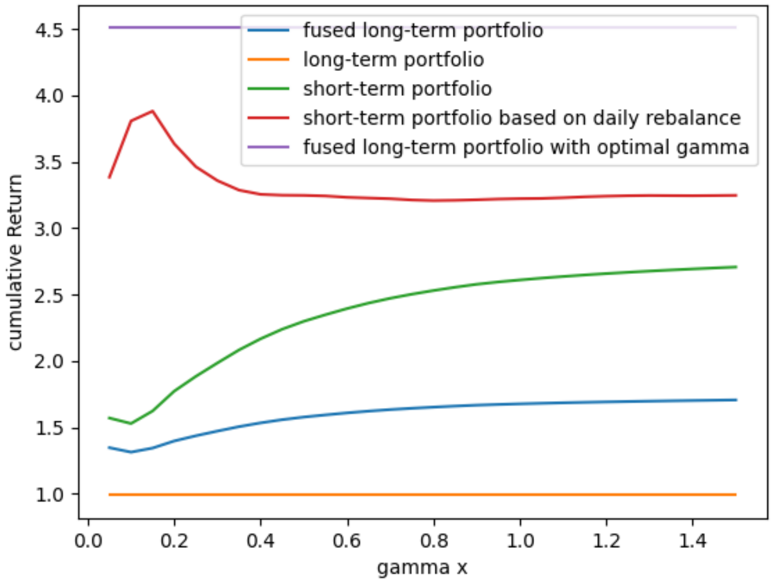

Figure 6.16: Development of cumulative returns when using the local optimal γ^*_{short}

Figure 6.17, based on figure 6.13 , describes the development of Sharpe ratios when using different γ_{short}. Because these local optimal γ^*_{short} are fixed, the corresponding average portfolio return does not change with different γ_{short}. As we can see, the Sharp ratio of the fused long-term portfolio using the local optimal γ^*_{short} is 2.148. It's better than the rest of the portfolio.

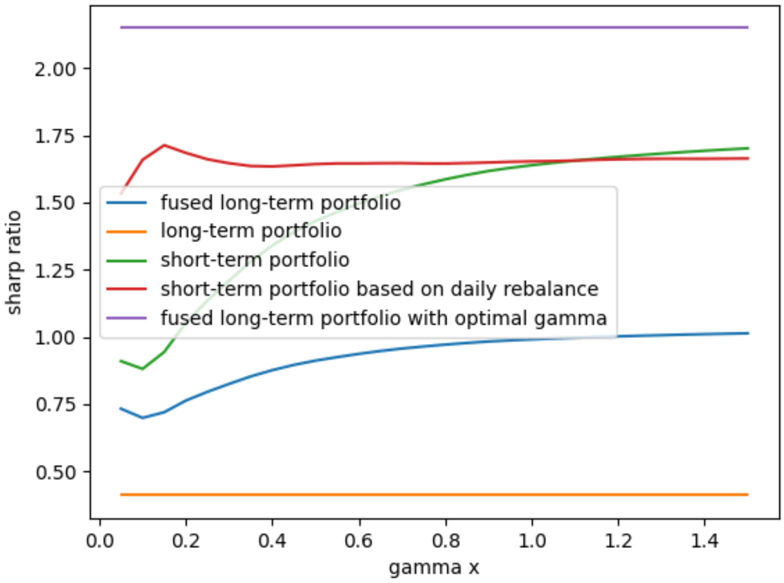

Figure 6.17: Development of Sharpe ratios when using the local optimal γ^*_{short}

6.5.5 Conclusion

As we can see, the performance of the fused long-term portfolio using the local optimal γ^*_{short} is better than the others. If we can predict the local optimal γ^*_{short} , the fused long-term portfolio can be improved.

6.6 Experiment 6: Sampling Interval based multi-fidelity Model with predicted local optimal γ^*_{short} and binary local optimal γ^*_{short}

6.6.1 Experiment Goal

In this experiment, we want to test the performance of the fused long-term portfolio using predicted local optimal γ^*_{short} and binary local optimal γ^*_{short}, compared to the short-term portfolio and long-term portfolio.

6.6.2 Experiment Design

1. generate local optimal γ^*_{short} for each time point t like Experiment 4 and long-term portfolios with $\gamma_{long} = 1$.
2. predict local optimal γ^*_{short} at each time t using a Gaussian process classifier .
3. calculate binary local optimal γ^*_{short} (If a local optimal γ^*_{short} is larger than 0.8, then the corresponding binary local optimal γ^*_{short} is 1.5 . Others binary local optimal γ^*_{short} are 0.05).
4. generate the fused long-term portfolio with predicted local optimal γ^*_{short}.
5. calculates cumulative returns, average returns, and Sharpe ratios for the fused long-term portfolio with predicted local optimal γ^*_{short} and binary local optimal γ^*_{short} .
6. compare with experiment 5

6.6.3 Data set

In this experiment, the data set is the historical monthly prices between $1995-01-01$ and $2015-01-01$ and the historical daily prices between $1995-01-01$ and $2015-01-01$.

The relative stocks are:
"AMD","BA","CSCO","DHR","INTC","JPM","NKE","PG","TXN","WMT".

6.6.4 Experiment Result

In this experiment, we predict the local optimal γ^*_{short} using Gaussian process classifier implemented by scikit-learn [4]. The local optimal γ^*_{short} distribution is similar to a two-point distribution. The best way to predict data points from two-point distributions is a machine learning model or deep learning model. Unfortunately, we can not predict the data using a machine learning or deep learning model due to a lack of time and data. Therefore, Gaussian process classifier is used in our experiment to predict the local optimal γ^*_{short}. Our data set size of local optimal γ^*_{short} is 144. The accuracy of the predictions is 54.8%.

Average Portfolio Returns

Figure 6.18, based on figure 6.15, describes the development of average portfolio returns when using predicted local optimal γ^*_{short} and binary local optimal γ^*_{short}. Because the predicted local optimal γ^*_{short} and the binary local optimal γ^*_{short} are fixed for each time point, the corresponding average portfolio return does not change with different $\gamma short$. As we can see, the average portfolio return of binary local optimal γ^*_{short} is about 0.0123 (1.23%). Compared to the local optimal γ^*_{short} ,the average portfolio return of binary local optimal γ^*_{short} drops down slightly.

The average portfolio return of predicted local optimal γ^*_{short} is just 0.004 (0.4%).

It seems that we should keep γ^*_{short} as large as possible if we can not predict the local optimal γ^*_{short} in a high accuracy. However, it still outperforms the long-term portfolio.

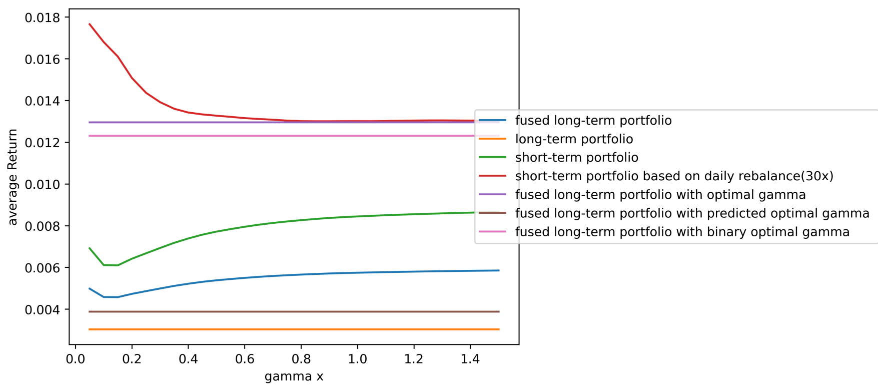

Figure 6.18: Development of average portfolio returns when using predicted local optimal γ^*_{short} and Binary Local Optimal γ^*_{short}

Cumulative Returns

Figure 6.19, based on figure 6.16, describes the development of cumulative returns when using predicted local optimal γ^*_{short} and binary local optimal γ^*_{short}. Because the predicted local optimal γ^*_{short} and the binary local optimal γ^*_{short} are fixed for each time point, the corresponding cumulative portfolio return does not change with different $\gamma short$.

As we can see, the cumulative returns of the fused long-term portfolio using the local optimal γ^*_{short} is about 4.6(460%). It is the best among all of these portfolios.

As we can see, the cumulative return of binary local optimal γ^*_{short} is about 4.1 (410%). Compared to the local optimal γ^*_{short} ,the cumulative return of binary local optimal γ^*_{short} drops down slightly.

The cumulative portfolio return of predicted local optimal γ^*_{short} is just 1.38 (138%).

It seems that we should keep γ^*_{short} as large as possible if we can not predict the local optimal γ^*_{short} correctly in a high probability. However, it still outperforms the long-term portfolio.

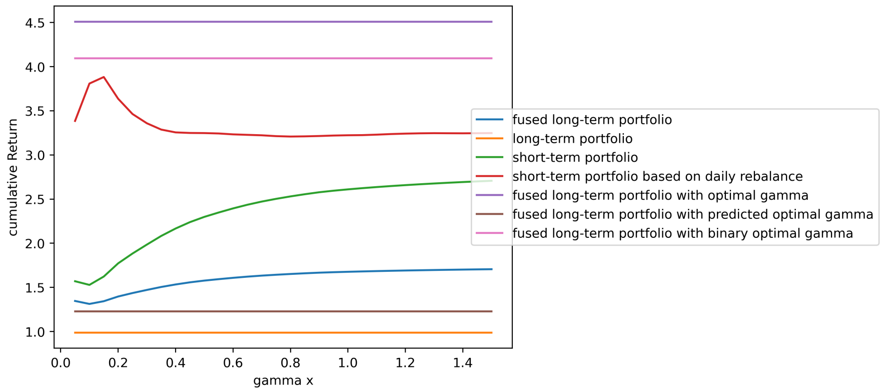

Figure 6.19: Development of Cumulative Returns When Using Predicted Local Optimal γ^*_{short} and Binary Local Optimal γ^*_{short}

Sharpe Ratios

Figure 6.20, based on figure 6.17, describes the development of Sharpe ratios when using predicted local optimal γ^*_{short} and binary local optimal γ^*_{short}.

Because the predicted local optimal γ^*_{short} and the binary local optimal γ^*_{short} are fixed for each time point, the corresponding Sharpe ratios do not change with different $\gamma short$.

As we can see, the Sharpe ratio of binary local optimal γ^*_{short} is 2.02. There is also a little decline compared to the local optimal γ^*_{short}. The Sharpe ratio of predicted local optimal γ^*_{short} is only 0.61. But it is still better than the long-term portfolio.

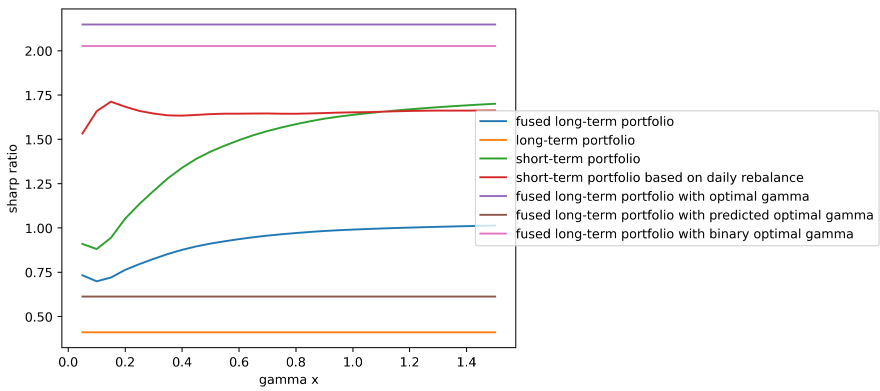

Figure 6.20: Development of Sharpe Ratios When Using Predicted Local Optimal γ^*_{short} and Binary Local Optimal γ^*_{short}

6.6.5 Conclusion

As we can see, there is a slight decline of the performance when using binary local optimal γ^*_{short}, compared to using local optimal γ^*_{short}. However, the performance using binary local optimal γ^*_{short} is better than the baseline. The performance of predicted local optimal γ^*_{short} just outperforms the long-term portfolio can be improved because the accuracy of the predictions is very low.

7 Conclusion

Our work has given an overview of portfolio optimization, multi-fidelity, and Gaussian processes. We described financial data characteristics, assets, portfolios, portfolio optimization, traditional multi-fidelity model, Gaussian process, and some regression algorithms. Process models about how portfolio optimization can work are also mentioned. We discuss the definition of multi-fidelity in portfolio optimizations. According to the process models of portfolio optimization, we discuss whether or not each step can be a high-fidelity / low-fidelity in portfolio optimization. According to the result of the discussion, two different types of multi-fidelity portfolio optimization models are proposed. One is the sampling interval based multi-fidelity model, a feasible idea. The other one is the prediction-based multi-fidelity model, an unreasonable idea. The reason why the prediction-based multi-fidelity model is unreasonable is that we can not transform a distribution into a useful expected return or a useful portfolio. Some experiments about the sampling interval based multi-fidelity model are designed and implemented. These experiment results explain the total workflow of the sampling interval based multi-fidelity model, and how it can be used in the future. Experiments 1 and 2 describe why we can not improve portfolio optimization with a huge data set. Experiment 3 proves that it is possible that a short-term portfolio can be fused into a long-term portfolio to generate a fused long-term portfolio, which can bring higher returns and a higher sharp ratio than the long-term portfolio. Experiment 4 displays the local optimal γ^*_{short} distribution. This distribution can be treated as a two-point distribution if we ignore some non-extreme values. Experiment 5 shows an ideal situation of how good the fused long-term portfolio can be while using local optimal γ^*_{short} at each time t. But how can the local optimal γ^*_{short} for the fused long-term portfolio be predicted at each time point t? The local optimal γ^*_{short} distribution is similar to a two-point distribution. The best way to predict data points from two-point distributions is a machine learning model or deep learning model. Unfortunately, we can not predict the data using a machine learning or deep learning model due to a lack of time and data. In experiment 6, we predict the local optimal γ^*_{short} using a Gaussian process classifier implemented by scikit-learn [4].

7.1 Future Works

Many things about this work still need to be done in the future. The first thing is how to predict the local optimal γ^*_{short}. The second thing is how to refine the fusion algorithm in sampling interval based multi-fidelity model.

7.1.1 Predicting Local Optimal γ^*_{short}

Predicting optimal γ^*_{short} is a challenging thing. However, fortunately, the distribution of local optimal γ^*_{short} can be regarded as a two-point distribution. Normally , binary prediction is easier than multiple outcome prediction because the machine learning and deep learning model can outperform in such tasks. We can use machine learning or deep learning to calculate the probability of local optimalγ^*_{short} at time t. Because it follows the two-point distribution, two probabilities,P_0 and P_1, can be calculated. P_0 is the probability of the minimum γ. P_1 is the probability of the maximum γ. These probabilities can be used in the fusion algorithm. In order to use machine learning or deep learning, it is necessary to collect more data from different data source, not just stock prices.

7.1.2 Refine Fusion Algorithm with the probability of the maximum of local optimal γ^*_{short}

In this work, our fusion algorithm is just a simple average method. In fact, due to the prediction of local optimal γ^*_{short}, it is possible that we can apply the probability of local optimal γ^*_{short} at time t into the fusion algorithm. For example, we can allocate the portfolio according to these probabilities, such as:

$$FLP = ((0.5)LP + (0.5)SP)P_0 + ((0.5)LP + (0.5)SP)P_1 \tag{7.1}$$

It can reduce the risk by pooling risk.

Abbreviations

List of Figures

List of Tables

Bibliography

[1] Signal-to-noiseratio(2023)Wikipedia.Availableat:https://en.wikipedia.org/wiki/Signal-to-noise_ratio(Accessed:07January2024).. [Accessed 07-01-2024].

[2] https://tradingenigma.wordpress.com/2021/07/19/what-is-the-signal-to-noise-ratio-and-why-is-it-important-in-algorithmic-trading/. [Accessed 07-01-2024].

[3] https://www.financestrategists.com/wealth-management/investment-management/portfolio-performance-evaluation/practical-considerations-in-portfolio-performance-evaluation)/. [Accessed 07-01-2024].

[4] https://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.GaussianProcessClassifier.html)/. [Accessed 07-01-2024].

[5] *adjustOHLC: Adjust Open,High,Low,Close Prices For Splits and Dividends in quantmod: Quantitative Financial Modelling Framework — rdrr.io*. https://rdrr.io/cran/quantmod/man/adjustOHLC.html. [Accessed 08-01-2024].

[6] *An Introduction to Non-Stationary Processes — investopedia.com*. [Accessed 07-01-2024].

[7] P. Bratley. *A guide to simulation*. Springer Verlag, 1996.

[8] B. Bruder, N. Gaussel, J.-C. Richard, and T. Roncalli. "Regularization of portfolio allocation." In: *Available at SSRN 2767358* (2013).

[9] *Can signal-to-noise ratio indicate prediction skill? Based on skill assessment of 1-month lead prediction of monthly temperature anomaly over Japan — frontiersin.org*. https://www.frontiersin.org/articles/10.3389/fclim.2022.887782/full. [Accessed 07-01-2024].

[10] P. Cootner. *The Random Character of Stock Market Prices*. Risk classics library. M.I.T. Press, 1964. ISBN: 9781899332847.

[11] A. Cowles. "Stock market forecasting." In: *Econometrica, Journal of the Econometric Society* (1944), pp. 206–214.

[12] V. DeMiguel, L. Garlappi, and R. Uppal. "Optimal Versus Naive Diversification: How Inefficient is the 1/N Portfolio Strategy?" In: *The Review of Financial Studies* 22.5 (Dec. 2007), pp. 1915–1953. ISSN: 0893-9454. DOI: 10.1093/rfs/hhm075. eprint: https://academic.oup.com/rfs/article-pdf/22/5/1915/24429471/hhm075.pdf.

[13] *General Efficient Frontier; PyPortfolioOpt 1.5.4 documentation — pyportfolioopt.readthedocs.io*. https://pyportfolioopt.readthedocs.io/en/latest/GeneralEfficientFrontier.html. [Accessed 07-01-2024].

[14] J. Gonzalvez, E. Lezmi, T. Roncalli, and J. Xu. *Financial Applications of Gaussian Processes and Bayesian Optimization*. 2019. arXiv: 1903.04841 [q-fin.PM].

[15] C. Gu. "Application of data mining technology in financial intervention based on data fusion information entropy." In: *Journal of Sensors* 2022 (2022), pp. 1–10.

[16] A. Gunjan and S. Bhattacharyya. "A brief review of portfolio optimization techniques." In: *Artificial Intelligence Review* 56.5 (2023), pp. 3847–3886.

[17] R. T. Haftka. "Combining global and local approximations." In: *AIAA Journal* 29.9 (1991), pp. 1523–1525. DOI: 10.2514/3.10768. eprint: https://doi.org/10.2514/3.10768.

[18] *How Can I Measure Portfolio Variance? — investopedia.com*. https://www.investopedia.com/ask/answers/071515/how-can-i-measure-portfolio-variance.asp. [Accessed 07-01-2024].

[19] W. Huang, Y. Nakamori, and S.-Y. Wang. "Forecasting stock market movement direction with support vector machine." In: *Computers & operations research* 32.10 (2005), pp. 2513–2522.

[20] J. HYNDMAN ROB. *Forecasting: Principles and practice*. OTEXTS, 2021.

[21] *Is Low Signal to Noise Ratio Really A Problem For Financial Machine Learning? - ENJINE — enjine.com*. https://www.enjine.com/blog/low-signal-noise-ratio-really-problem-financial-machine-learning/. [Accessed 07-01-2024].

[22] R. Israel, B. T. Kelly, and T. J. Moskowitz. *Can machines "learn" finance?* July 2020.

[23] P. N. Kolm, R. Tütüncü, and F. J. Fabozzi. "60 years of portfolio optimization: Practical challenges and current trends." In: *European Journal of Operational Research* 234.2 (2014), pp. 356–371.

[24] R. I. Kumar. *Signals and Systems*. Delhi, India: PHI Learning, Dec. 2009.

[25] H. Markowitz. "PORTFOLIO SELECTION*." In: *The Journal of Finance* 7.1 (1952), pp. 77–91. DOI: https://doi.org/10.1111/j.1540-6261.1952.tb01525.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1540-6261.1952.tb01525.x.

[26] R. Marks. *Introduction to Shannon Sampling and Interpolation Theory*. Springer Texts in Electrical Engineering. Springer New York, 2012. ISBN: 9781461397083.

[27] M. H. Miller. *Portfolio Selection: Efficient Diversification of Investments.* 1960.

[28] T. Niu, J. Wang, H. Lu, and P. Du. "Uncertainty modeling for chaotic time series based on optimal multi-input multi-output architecture: Application to offshore wind speed." In: *Energy Conversion and Management* 156 (Jan. 2018), pp. 597–617. DOI: 10.1016/j.enconman.2017.11.071.

[29] V. Norton. "Adjusted closing prices." In: *arXiv preprint arXiv:1105.2956* (2011).

[30] B. Peherstorfer, K. Willcox, and M. Gunzburger. *Survey of multifidelity methods in uncertainty propagation, inference, and optimization*. 2018. arXiv: 1806.10761 [math.NA].

[31] D. H. B. Phan, S. S. Sharma, and P. K. Narayan. "Stock return forecasting: Some new evidence." In: *International Review of Financial Analysis* 40 (2015), pp. 38–51.

[32] D. Rapach and G. Zhou. "Forecasting stock returns." In: *Handbook of economic forecasting*. Vol. 2. Elsevier, 2013, pp. 328–383.

[33] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, Nov. 2005. ISBN: 9780262256834. DOI: 10.7551/mitpress/3206.001.0001.

[34] *Return on assets - Wikipedia — en.wikipedia.org*. https://en.wikipedia.org/wiki/Return_on_assetscite_note-isbn0-618-73661-1-1. [Accessed 07-01-2024].

[35] T. Roncalli. *Introduction to Risk Parity and Budgeting*. 2014. arXiv: 1403.1889 [q-fin.PM].

[36] B. Scherer and R. D. Martin. *Modern Portfolio Optimization with NuOPTTM, S-PLUS®, and S+ BayesTM*. Springer Science & Business Media, 2007.

[37] R. S. Tsay. *Analysis of Financial Time Series*. Wiley India, 2016.

[38] S.-H. Tseng and T. Allen. "A Simple Approach for Multi-Fidelity Experimentation Applied to Financial Engineering." In: *Appl. Stoch. Model. Bus. Ind.* 31.5 (Sept. 2015), pp. 690–705. ISSN: 1524-1904. DOI: 10.1002/asmb.2075.

[39] S.-H. Tseng and T. Allen. "A Simple Approach for Multi-fidelity Experimentation Applied to Financial Engineering." In: *Applied Stochastic Models in Business and Industry* 31.5 (2015), pp. 690–705. DOI: https://doi.org/10.1002/asmb.2075. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/asmb.2075.

[40] J. Wang. *An Intuitive Tutorial to Gaussian Processes Regression*. 2022. arXiv: 2009.10862 [stat.ML].

[41] *Welcome to CVXPY 1.4 ; CVXPY 1.4 documentation — cvxpy.org*. https://www.cvxpy.org/. [Accessed 07-01-2024].

[42] F. Yuan and H. Zhan. "Stock market investment behavior based on behavioral finance based on data fusion algorithm." In: *IETE Journal of Research* (2022), pp. 1–7.

[43] L. Zhang, L. Zhang, W. Teng, and Y. Chen. "Based on Information Fusion Technique with Data Mining in the Application of Finance Early-Warning." In: *Procedia Computer Science* 17 (2013). First International Conference on Information Technology and Quantitative Management, pp. 695–703. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2013.05.090.

[44] H. Zou and T. Hastie. "Regularization and variable selection via the elastic net." In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 67.2 (2005), pp. 301–320.