Research on Interpretable Machine Learning Portfolio Based on Multi-factor Clustering

JiHui Shi[1], WenZheng Zhang[2]

***Abstract*—The 'black box' phenomenon and limited interpretability present significant obstacles in machine learning and deep learning for portfolio management. Additionally, standard metrics for interpretability in machine learning often struggle to effectively elucidate model features in portfolio decision contexts. This research aims to address these challenges by introducing a methodology for generating easily interpretable portfolios. The approach involves using Random Forest feature importance analysis within multi-factor models, followed by clustering based on stock factors. Portfolios are generated using the Mean-CVaR model, and the effectiveness of the proposed explainable portfolios is evaluated through comparative analysis with two machine learning interpretability tools: SHAP and Permutation methods.**

***Index Terms*—Interpretability, Multi-factor Mode, Stock Clustering, Random Forest, Portfolio**.

I. INTRODUCTION

The application of machine learning techniques in investment decision-making has seen a significant rise, highlighting the need to improve interpretability to address the "black box" problem, making research in this area crucial. The origins of conventional portfolio design strategies trace back to the Capital Asset Pricing Model (CAPM) introduced by Sharpe [1], the first theoretical framework to statistically elucidate the correlation between stock returns and market risk, including idiosyncratic risks of individual stocks. CAPM established the foundation for developing portfolio design methodologies based on market risk assessment. Eugene Fama [2] enhanced the explanatory capability of stock returns by incorporating additional components beyond the scope of CAPM.

In the 21st century, big data has given researchers unparalleled access to extensive data resources, yet presenting obstacles in identifying non-linear interactions within large market data volumes. Machine learning technology, renowned for handling non-linear and non-stationary data, has become a popular tool in portfolio design research. Huang [3] devised a technique using Support Vector Regression (SVR) and Genetic Algorithms (GAs) to construct effective portfolios, optimizing model parameters to select the best subset of input variables for the SVR model. In portfolio creation, J. Xiujuan [4] introduced the Random Forest Support Vector Machine (RFSVM), using random forests to process and reduce the dimensions of initial data variables, thereby enhancing decision-making reliability and efficacy. Shen [5] proposed the VIKOR-DANP model for improved stock selection, mitigating limitations of conventional regression models. This methodology assesses stocks using criteria like profitability and cash flow earning ability, determining modified scoring weights for identifying high-quality growth stocks.

Nevertheless, the interpretation of investment portfolios generated through machine learning faces challenges: the inherent "black box" nature of these algorithms complicates understanding [6], and some methodologies focus on selecting high-return stocks based on predicted return series, failing to capture the relationship between selected stocks and cross-sectional return explanatory factors, potentially overlooking stocks correlated with return characteristic factors.

Within the domain of machine learning, interpretability is defined as "the capacity to elucidate or convey information in a comprehensible manner to human beings" [7]. This is crucial in stock market applications for investors and model researchers to enhance investing methods. The primary objective is to identify the elements impacting prediction outcomes, with the ultimate goal of making them understandable. Taha Buğra Çelik [8] used Explainable Artificial Intelligence (XAI) techniques to evaluate forecast reliability, focusing on comprehensive predictive efficacy using XAI indicators, rather than solely on accuracy. This approach aims to reduce suboptimal decision-making risks by investors. Ribeiro[9] introduced Local Interpretable Model-Agnostic Explanations (LIME) to address transparency and interpretability challenges in machine learning algorithms. LIME achieves local interpretability by training an interpretable model near the prediction. Similarly, Lundberg S M [10] employed Shapley values, derived from cooperative game theory, to quantify each feature's impact on machine learning model predictions, ensuring equitable reward distribution based on marginal contributions. However, Kyung Keun Yun [11] noted that LIME and SHAP do not address temporal and collective feature dependencies in stock sequences. Yun proposed integrating evolutionary algorithms with machine learning regression and optimal feature selection for stock price prediction, enhancing local interpretability by capturing significant feature behaviors over certain periods, thereby providing dynamic interpretations for short-term data. Current literature primarily focuses on feature selection and interpretability

JiHui Shi and WenZheng Zhang are eqaual contribution to the paper. JiHui Shi is corresponding author (e-mail: jacky-shi@zjhu.edu.cn). [1] works with School of Economic Management, Huzhou University, Huzhou Zhejiang, China, [2]works with School of Information Engineering, Huzhou University, Huzhou Zhejiang China ,

indicator development to explain machine learning model outputs from various perspectives. In machine learning portfolio creation, however, the exclusive emphasis on model prediction interpretability still challenges the effective application of analytical results in portfolio construction decision-making. There is often a tendency to reinforce confidence in a model's predictive capabilities, while overlooking the connection between interpretability metrics and portfolio formation methods, as well as the subsequent portfolio building process.

Hence, this article presents an exceptionally interpretable method for constructing portfolios. It does so by aggregating specific stock factors within a multi-factor model framework, establishing a clear link between factor-driven portfolio construction strategies and machine learning interpretive indicators. Including high-importance factors in the model not only enhances its interpretability for investment decision-making but also mitigates the 'black box' issue associated with machine learning, addressing the challenges faced by traditional interpretive index methods in intuitive feature interpretation within portfolio decision-making contexts. This approach employs machine learning feature importance analysis to pinpoint the most predictively significant influencing factors, leading to a strategy for constructing clustered portfolios using these identified factors. Grounded in empirical evidence and bolstered by clear economic reasoning, the selected equities demonstrate superior performance across key predictors. By forging this link between machine learning interpretability and portfolio construction strategies, we offer a new framework for financial practice, enabling more effective application of machine learning model predictions in portfolio strategy formulation. Moreover, it assists investors in better understanding these stock selection strategies when constructing their actual portfolios.

II. DATA AND METHODOLOGY

This study begins by employing the Random Forest technique to identify key factors influencing stock returns during the specified timeframe. These factors form the foundation of the analysis. Bayesian optimization is then used to fine-tune the Random Forest model's parameters for each training period. With these optimized parameters, the model assesses the significance of each factor in predicting returns. The model's interpretability is enhanced through the distribution of factor importance, focusing particularly on the two most important factors as the base.

The dataset of stock factors undergoes MiniBatchKMeans clustering to account for dynamic market changes. We employ a dynamic clustering method with a sliding window to capture variable trends monthly. The time-series data of stock factors are divided into overlapping subsets, each analyzed within a sliding window at a fixed step length. Clustering is performed in these windows to group factor features. The resulting clusters, which represent various stock factor characteristics, undergo statistical analysis. Concurrently, these clusters are assessed and prioritized based on fundamental variables identified by the Random Forest algorithm. Stocks most frequently appearing in top-rated clusters are selected for investment.To balance portfolio returns and risk management, the Mean-CVaR model is applied for portfolio creation. A genetic algorithm determines the optimal investment weights for the selected stocks. The research flowchart in Fig. 1 details the sequence of these steps.

The following section provides a concise overview of the primary machine learning methodologies and models used in this study.

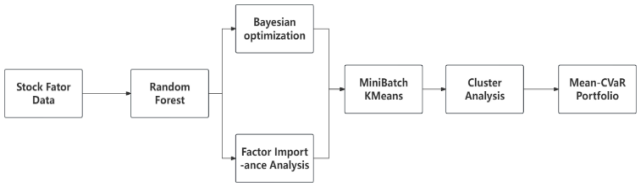

Fig. 1. Research flow chart

A. Bayesian Optimization Random Forest

The prediction of stock returns over time is frequently influenced by various factors, making the identification of the most significant explanatory elements a considerable challenge. Among a range of machine learning algorithms, Random Forests [12] are particularly adept at elucidating factor importance due to their tree structure. Therefore, this study employs the Random Forest algorithm to predict stock returns, leveraging its inherent feature importance distribution tool. A key aspect of this approach is determining the ideal hyperparameters of the Random Forest model to achieve high performance and ensure generalizability [13].

Traditionally, cross-validation techniques are used to identify suitable hyperparameters by iteratively training and validating the dataset, evaluating the model's performance across different configurations, and selecting the optimal values. However, this method can be computationally intensive, especially when handling large volumes of stock factor data inputs. To address this, the study incorporates Bayesian optimization [14], a method for iterative optimization aimed at finding a global optimum within a limited number of iterations. This technique enhances the efficiency of hyperparameter identification.

In this study, the Bayesian optimization parameters for Random Forest include the number of decision trees, the maximum depth of each tree, the minimum sample size for splitting an internal node, and the minimum sample size for a leaf node. These parameters directly affect the model's performance and generalizability. Bayesian optimization uses a Gaussian process model to emulate the relationship between the target function and model parameters. Through iterative parameter selection and Gaussian process model updates, the ideal parameter combination is determined, potentially addressing overfitting and underfitting issues, and improving feature element importance.

B. Muti-Factor Models

Barr Rosenberg [15] was the pioneer in introducing the multi-factor model, a methodology designed to explain fluctuations in stock returns by incorporating multiple factors. In this model, the returns of stock securities are represented as a linear combination of various factors. The fundamental structure of the multi-factor model is characterized as follows:

$$r_{i,t} = \alpha_i + \sum_{j=1}^{k} \beta_{i,j} f_{j,t} + \epsilon_{i,t} \quad (1)$$

The rate of return of asset i at time t is denoted by $r_{i,t}$, while the rate of return of the j_{th} factor at time t is represented by $f_{j,t}$. The excess return of asset i is denoted by $\propto_i$, the return of asset i to the sensitivity coefficient of j is represented by $\beta_{i,j}$, and the error term of asset i at time t is denoted by $\epsilon_{i,t}$.

Multifactor models provide robust decision support for investors by assessing asset risk and return attributes. However, in practice, different factors may have varying cyclical impacts on asset returns. To address the challenge of identifying these cyclical influences in stock data, this article employs machine learning techniques to dynamically model stock feature data for each period. This approach allows for the selection of an optimal combination of factors that best suits the specific characteristics of that period, thereby enhancing the model's relevance and accuracy in different market conditions.

C. MiniBatchKMeans Cluster Mean-CvaR Portfolio

To efficiently capture dynamic factor changes and select robust explanatory factors during portfolio optimization, this section employs a sliding window methodology combined with MiniBatchKMeans clustering [16]. This clustering method, suitable for extensive datasets, groups data points into K clusters based on similarity and shared attributes, minimizing similarities between different clusters.

The approach involves partitioning stock factor time-series data into overlapping subsets. Within each window, clustering operations are performed on the factor features. Initially, factor data is divided into fixed-length windows at each time frequency. After aggregating data within a window, it is systematically slid by a predetermined step length. The clustering process is repeated for new factor data in each window to accommodate cyclical variations. Bayesian optimization is used to determine the optimal sliding window length, step duration, and cluster quantity for each period.

MiniBatchKMeans clustering results in clusters reflecting various stock factor characteristics, which undergo statistical analysis. Robust feature factors that significantly explain stock returns, as identified by the Random Forest algorithm, are combined. Scores are assigned to each cluster based on these factors, with the overall score of each cluster being calculated. The highest-scoring clusters are used to select frequently appearing stocks in each period, indicative of a strong association with specific factors.

Tail risks in the portfolio, due to deviations from normality in real-world stock return distributions, are evidenced by skewness and kurtosis. To address this, the Mean-CVaR model [17] is chosen for portfolio construction using selected equities. Unlike traditional mean-variance and mean-semivariance models, Mean-CVaR more effectively accounts for asymmetric distribution of stock returns and tail risks, thereby mitigating potential portfolio losses. A detailed description of the utilized model is provided below:

Assume investors allocate their initial capital into a portfolio of high-risk assets within the financial market. R_i denotes the random rate of return for each high-risk asset i, while $r_i=E(R_i)$ represents the expected rate of return. The proportion of the investor's portfolio at risk in asset i is denoted by xi, and the weight vector of the portfolio is x=(x1, x2,..., xn). Investors aim to optimize the negatively weighted sum of the portfolio's expected return and risk, while ensuring that no constraints are violated. One of the ways in which risk is quantified is through the conditional value at risk (CVaR).

Suppose an investor allocates their initial wealth to n different risky assets in the financial market. For each risky asset i, we define Ri as its random rate of return and $r_i=E(R_i)$ as its expected rate of return. The investor's investment proportion in risky asset i is denoted as xi, and the portfolio's weight vector is x=(x1, x2, ..., xn). Under the premise of not violating any constraints, the investor aims to maximize the negative weighted sum of the portfolio's expected rate of return and risk. Here, risk is represented in the form of Conditional Value at Risk.

The expected return on the investment portfolio is:

$$r_p = \sum_{i=1}^{n} x_i * r_i \quad (2)$$

The return on the investment portfolio is:

$$R_p = \sum_{i=1}^{n} x_i * R_i \quad (3)$$

A portfolio's CVaR is defined as the expected loss in excess of value at risk (VaR). At the α confidence level, VaR is defined as the worst α% case of the portfolio return, that is :

$$\mathrm{VaR} = \inf\{r \mid P(R_p \le r) \ge \alpha\} \quad (4)$$

CVaR is defined as the expected loss when the rate of return is lower than VaR, that is:

$$\mathrm{CVaR} = \mathbb{E}[R_p \mid R_p \le \mathrm{VaR}] \quad (5)$$

Therefore, our objective function is:

$$\mathrm{Maximize}(r_p - \lambda \times \mathrm{CVaR}) \quad (6)$$

Among them, λ is the risk aversion coefficient, which indicates the investor's aversion to risk.

The constraints are: The investment weight of each stock is greater than 0, that is, xi >= 0, i=1,2,...,n. The sum of investment weights is 1, that is, $\sum$xi= 1, i=1,2,...,n.

The above is the basic form of the mean-CVaR model, which provides a framework for quantifying the trade-off between expected return and risk for investors. By solving this optimization problem, investors can determine the portfolio allocation that achieves the maximum expected return while satisfying their risk aversion.

D. Data Selection

This study's sample includes every constituent stock of the CSI All-Share Index during the periods of January to June 2018 and January to June 2019, with data collected daily. Data preprocessing was essential to ensure accurate and reliable processing by the machine learning models. This involved filling missing values, addressing outliers, and normalizing data dimensions.

Stock returns were predicted using the Random Forest algorithm, optimized via Bayesian optimization. For each prediction period, stock factor data was split into training and test sets in an 8:2 ratio. Given the importance of input variable selection in time series prediction, sixteen factors representing various market characteristics of each stock were selected. These include liquidity, valuation, momentum, size, volatility, leverage, technical indicators, among others.

The specific categories of these factors are detailed in Table I, located in the appendix, which provides comprehensive information about the sixteen stock feature factors.

The model's accuracy was evaluated using three metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).

TABLE I STOCK FACTOR TABLE

Share Turnover	Absolute Return to volume	Earnings to Price Ratio	Operating Cash Flow to Price Ration
Long Term Reverse	Momentum Change	Float Capitalizatio n	Firm Size
Book to Market Equity	Idiosyncratic Volatility	Market Leverage	Down to Up Volatility
ASI	MACD	ATR	RI

III. RESULTS AND DISCUSSION

The efficacy of the Random Forest model in fitting stock return data is significantly influenced by the choice of hyperparameters. In this study, the Bayesian optimization method is utilized to fine-tune the model's parameters, enhancing its prediction accuracy for stock returns. The optimal parameters determined for each period, which contribute to the model's enhanced performance, are presented in Table II.

A. Comparative Analysis of Bayesian-Optimized Random Forest Ablation Experiments

To highlight the advantages of Bayesian optimization, this study compares the improvements in prediction performance and reduction in model error of Random Forest models before and after optimization.

1) Feature Significance

Following Bayesian optimization, the Random Forest model shows a more distinct distribution in feature importance, particularly in highlighting the key stock factors, compared to its pre-optimization state. This optimized model more effectively emphasizes the explanatory capability of the two fundamental components, which are crucial in predicting the rate of return. This enhancement in focusing on the most influential factors for return prediction is illustrated through the data feature distribution charts for 2018 and 2019, as depicted in Figs. 2, 3, 4, and 5. These figures clearly demonstrate the changes in feature importance distribution, underscoring the improved focus and clarity brought about by Bayesian optimization in understanding stock return drivers.

TABLE II RANDOM FOREST BEST PARAMETERS TABLE

	Maximum depth of each tree	Minimum leaf node samples	Fewest sample splits	number of decision trees
2018.1	6	8	16	442
2018.2	7	9	13	445
2018.3	7	9	18	221
2018.4	7	9	7	460
2018.5	6	8	3	109
2018.6	7	4	20	492
2019.1	8	2	17	478
2019.2	5	1	3	100
2019.3	6	4	18	161
2019.4	8	3	4	66
2019.5	7	9	7	460
2019.6	5	1	20	429

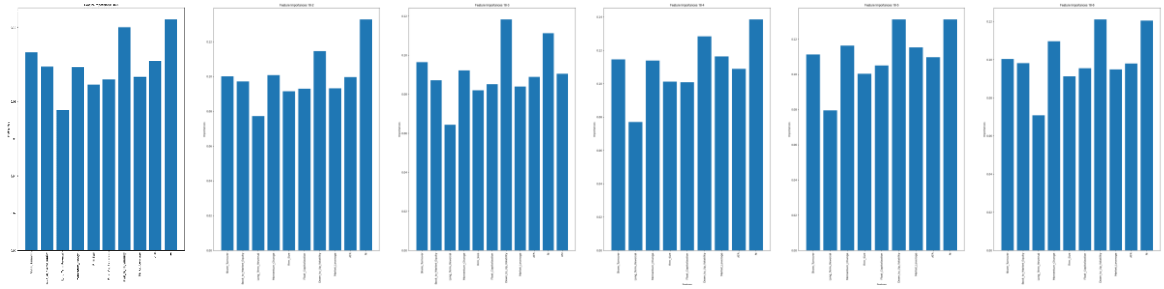

Fig. 2. 2018 Unoptimized Random Forest Feature Distribution Chart

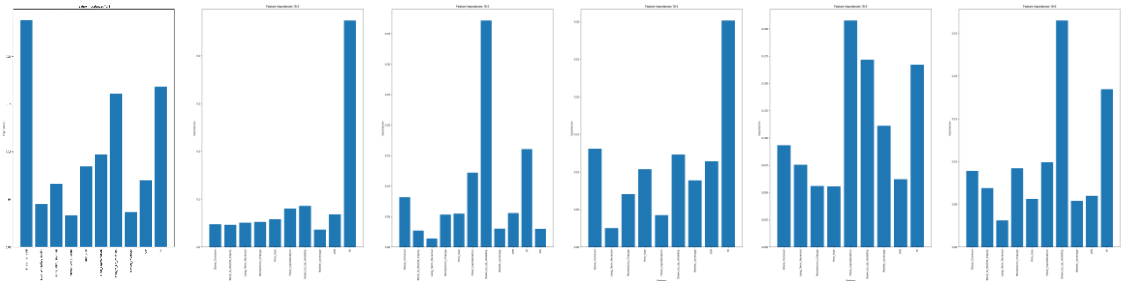

Fig. 3. 2018 Bayesian Optimized Random Forest Feature Distribution Chart

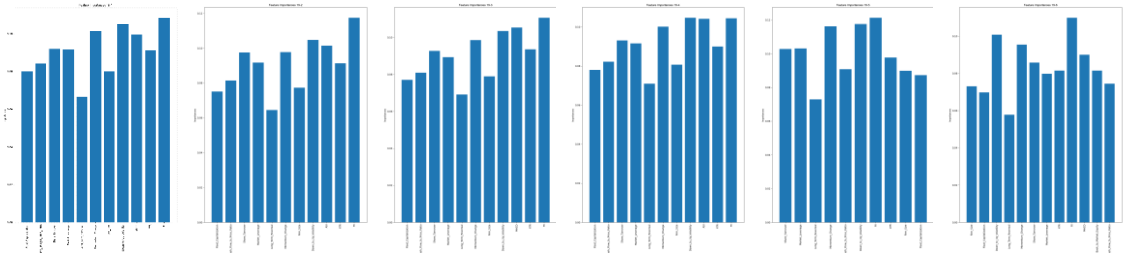

Fig. 4. 2019 Unoptimized Random Forest Feature Distribution Chart

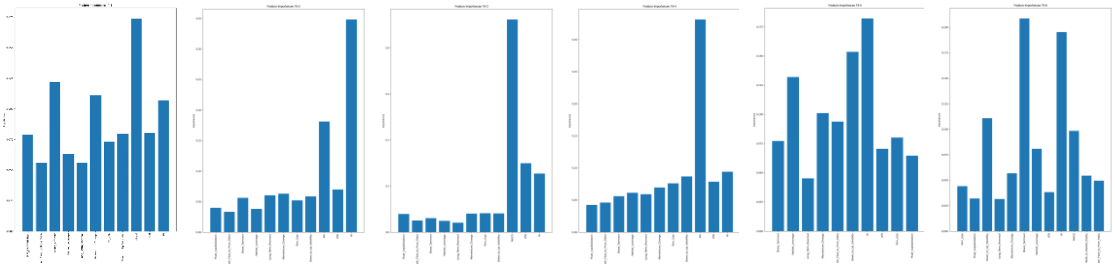

Fig. 5. 2019 Bayesian Optimized Random Forest Feature Distribution Chart

2) Pre-optimization and post-optimization error analysis

To evaluate the effectiveness of Bayesian optimization, this study assesses the error magnitude in the Random Forest model before and after its application. The comparative analysis, focusing on changes in error metrics, is presented in Tables III and IV. These tables provide detailed insights into the model's performance, highlighting the reduction in prediction error achieved through optimization.

Table III 2018 Pre-optimization and post-optimization error analysis

	Pre-optimization	MAE	MSE	RMSE	Post-optimization	MAE	MSE	RMSE
January		0.139	0.035	0.186		0.132	0.032	0.180
February		0.131	0.032	0.180		0.126	0.031	0.175
March		0.134	0.032	0.180		0.129	0.030	0.174
April		0.142	0.035	0.188		0.135	0.032	0.180
May		0.138	0.034	0.184		0.132	0.032	0.179
June		0.135	0.033	0.180		0.129	0.030	0.174

Table IV 2019 Pre-optimization and post-optimization error analysis

	Pre-optimization	MAE	MSE	RMSE	Post-optimization	MAE	MSE	RMSE
January		0.140	0.033	0.181		0.135	0.031	0.176
February		0.135	0.030	0.173		0.130	0.028	0.166
March		0.143	0.034	0.185		0.135	0.031	0.177
April		0.142	0.033	0.182		0.136	0.031	0.177
May		0.139	0.033	0.182		0.133	0.031	0.176
June		0.145	0.034	0.185		0.136	0.031	0.177

3) Baseline Model Comparison

The study compares the prediction error of the Random Forest model optimized via Bayesian methods with that of various baseline models to assess relative performance. Specifically, the predictive efficacy of stock return rate predictions using the KNN algorithm [18], Decision Tree method [19], and XGboost algorithm [20] is examined. The comparative results of each model are depicted in Figs. 6 and 7, highlighting the differences in prediction accuracy.

Upon reviewing these figures, it becomes apparent that the Bayesian-optimized Random Forest (BORF) model consistently exhibits lower error metrics (MAE, MSE, RMSE) compared to the standard Random Forest, KNN, and Decision Tree models. This observation indicates that the BORF model provides more accurate predictions of stock return rates, demonstrating the effectiveness of Bayesian optimization in enhancing model performance.

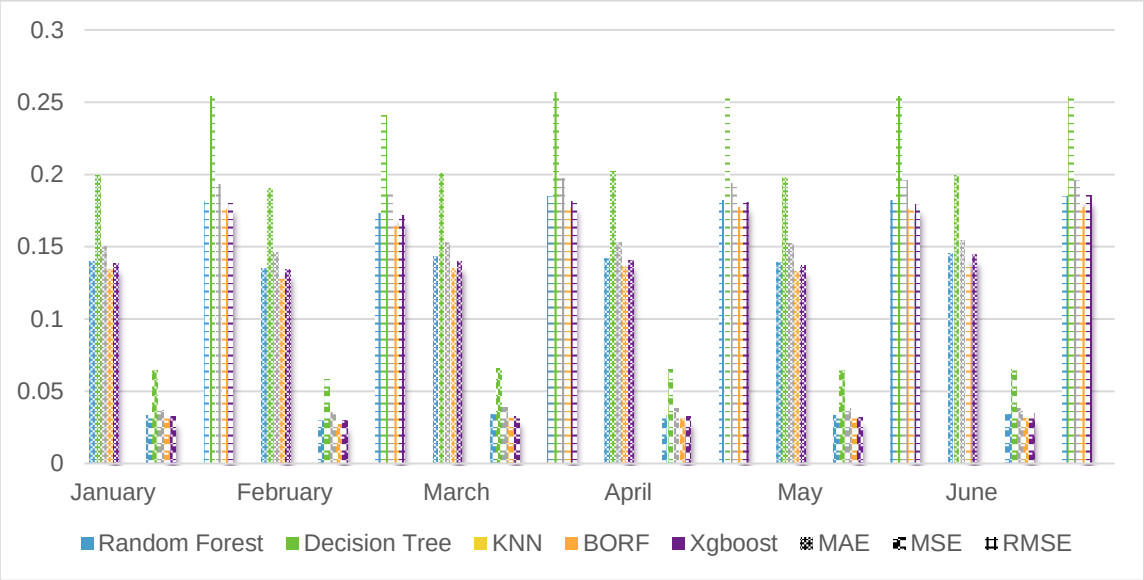

Fig. 6. Model Error Metrics Graph for 2018

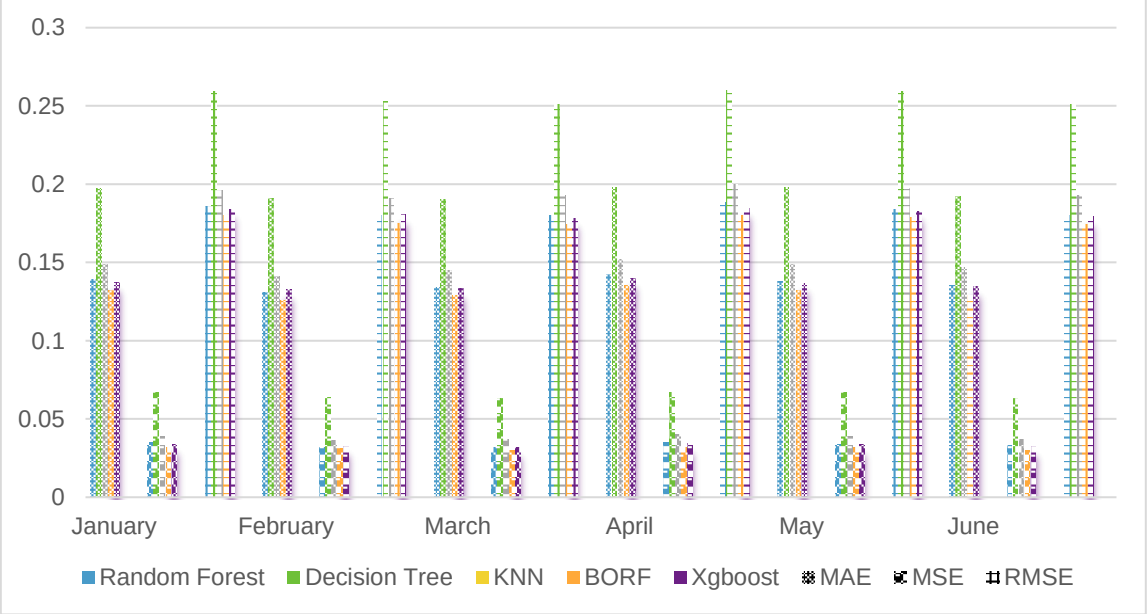

Fig. 7. Model Error Metrics Graph for 2019

B. Analysis of clustered portfolio construction results

MiniBatchKMeans clustering is employed to create distinct clusters representing stock factor characteristics. Based on the Random Forest algorithm's analysis for each period, the two most influential features in explaining returns are identified as the base factors. Clustering portfolio construction around these base factors establishes a direct link between the construction process and the machine learning model's interpretability indicators. The weighted scores of the base factors in each cluster highlight the clusters most strongly associated with key return drivers. Specifically, the cluster with the highest score is prioritized for investment.

For portfolio construction, the top five stocks appearing most frequently at the summit of the highest-scoring cluster are selected. This approach ensures that the chosen stocks are closely aligned with the identified base features influencing returns. Table V details the number of clusters and the optimal monthly sliding window parameters obtained through Bayesian optimization, providing insights into the clustering methodology and parameter selection process.

TABLE V NUMBER OF CLUSTERS AND SLIDING WINDOW PARAMETERS

	Number of clusters	Steps	Window length
2018.01	2	1	20
2018.02	3	4	15
2018.03	3	2	20
2018.04	3	3	18
2018.05	4	6	20
2018.06	3	3	20
2019.01	6	5	20
2019.02	2	3	15
2019.03	4	3	20
2019.04	4	3	20
2019.05	2	5	20
2019.06	2	1	10

C. Model interpretive analysis

In 2018 and 2019, the Random Forest algorithm, optimized through Bayesian optimization, was used to forecast monthly stock returns from January to June. Each month, the two features with the highest explanatory power for returns were identified as the foundational factors for the training cycle. This study leverages the Random Forest model's inherent feature importance distribution to select features for predicting model outcomes.

In 2018 and 2019, the Random Forest algorithm, optimized through Bayesian optimization, was used to forecast monthly stock returns from January to June. Each month, the two features with the highest explanatory power for returns were identified as the foundational factors for the training cycle. This study leverages the Random Forest model's inherent feature importance distribution to select features for predicting model outcomes.

To evaluate the interpretability of this feature importance distribution, two machine learning interpretability tools, SHAP (SHapley Additive Explanations) [21] and Permutation Importance [22], were employed. SHAP, inspired by the Shapley value from game theory, assigns a "fair" contribution degree to each feature by calculating the difference in model predictions with and without each feature. On the other hand, Permutation Importance measures a feature's significance by observing the decrease in model performance when each feature subset is scrambled and evaluated for its impact on performance.

Tables VI and VII present a comparison of these three methods regarding feature selection and feature weights in model prediction results. As indicated in these tables, the Random Forest model's feature importance distribution consistently selects the two most significant features across most test data periods, demonstrating its stability and reliability across all three interpretability methods.

D. Mean-CVaR portfolio numerical analysis

This segment evaluates the monthly performance of stock portfolios across different investment weights. To further assess the effectiveness of the proposed model, comparisons are made with major indices such as the CSI 300, Shanghai Composite Index, and Shenzhen Component Index, alongside equal-weighted portfolio models.

The study employs genetic algorithms for optimizing the Mean-CVaR model's weights, considering the need for a multivariate solution under multiple constraints. Genetic algorithms, simulating genetic and natural selection processes, are adept at finding global optimal solutions, avoiding the limitations of local optima that conventional algorithms might encounter.

The CVaR for each stock is calculated using the Monte Carlo method, followed by the application of a target function to determine optimal weights for each period, as shown in Tables VIII and IX. The portfolio's returns are derived from these weights, and the results are compared with other models and key market indices, as detailed in Tables VIII and V. Furthermore, Figures 4 and 5 contrast the returns of portfolios constructed using SHAP and Permutation Importance methods.

Tables VIII and V present the average monthly returns across various months and years, including comparisons with the Mean-CVaR model, Shanghai Composite Index, CSI 300, and Shenzhen Component Index. The distribution of returns from portfolios built using the method described in this paper is compared with those constructed using other interpretability methods in Figs. 8 and 9. Through this comparative analysis, we can draw the following conclusions:

1. Performance Comparison: The Mean-CVaR model demonstrated superior performance relative to other reference models and indices, achieving an average monthly return of 2.92% during the study period. This was notably higher compared to the Shanghai Composite Index (1.14%), CSI 300 (1.05%), and Shenzhen Component Index (1.02%). Additionally, the equal-weight model also showed strong profitability with an average monthly return of 2.55%.

2. Risk Analysis: The models and indices exhibited comparable performance in terms of return volatility, with standard deviations ranging from 0.053 to 0.070. Specifically, the Mean-CVaR model had a volatility of 0.068, slightly higher than the other indices but still within an acceptable range when considering all models and indices.

3. Interpretability Portfolio Returns: Comparing the investment portfolio construction model with SHAP and Permutation Importance methods revealed that the Random Forest investment portfolio incurred fewer losses in most months of the 2018 bear market. With an average monthly return of 0.32%, it significantly outperformed the other two interpretability methods. During the bull market phase of 2019, the Random Forest portfolio showed superior and more consistent performance, averaging 7.95% per month, compared to 4.95% for SHAP and 6.71% for Permutation portfolios.

In summary, the Mean-CVaR model, while not surpassing the Shanghai Composite Index in terms of the highest monthly return, demonstrated a notable advantage in overall returns, stability, and risk resistance during the analysis period. This strength stemmed from the Mean-CVaR investment portfolio, constructed based on the interpretative base factors identified by the Random Forest model. The comparative analysis with portfolios generated using different interpretability methods highlighted the efficacy of the portfolio construction approach outlined in this paper, especially in terms of asset allocation and utility. This underscores the potential of the proposed method in enhancing portfolio performance and managing investment risks more effective

TABLE VI 2018 FEATURE SELECTION AND NORMALIZATION OF WEIGHTS

	Random Forest	Weights	SHAP	Weights	Permutation Importance	Weights
2018.01	Share_Turnover	0.59	Share_Turnover	0.57	Share_Turnover	0.55
	RI	0.41	RI	0.43	RI	0.45
2018.02	RI	0.85	RI	0.86	RI	0.93
	Down_to_Up_ Volatility	0.15	Down_to_Up_ Volatility	0.14	Float_Capitalization	0.07
2018.03	Down_to_Up_ Volatility	0.7	Down_to_Up_ Volatility	0.69	Down_to_Up_ Volatility	0.71
	RI	0.3	RI	0.31	RI	0.29
2018.04	RI	0.7	RI	0.62	ATR	0.68
	Share_Turnover	0.3	ATR	0.38	Share_Turnover	0.32
2018.05	Float_Capitalization	0.55	Float_Capitalization	0.54	Float_Capitalization	0.7
	Down_to_Up_Volatility	0.45	RI	0.46	RI	0.3
2018.06	Down_to_Up_Volatility	0.59	Down_to_Up_Volatility	0.57	Down_to_Up_Volatility	0.67
	RI	0.41	RI	0.43	RI	0.33

TABLE VII 2019 FEATURE SELECTION AND NORMALIZATION OF WEIGHTS

	Random Forest	Weights	SHAP	Weights	Permutation Importance	Weights
2019.01	MACD	0.59	MACD	0.51	MACD	0.63
	Share_Turnover	0.41	Share_Turnover	0.49	Share_Turnover	0.37
2019.02	RI	0.61	RI	0.66	RI	0.66
	Down_to_Up_ Volatility	0.39	ASI	0.34	Float_Capitalization	0.34
2019.03	MACD	0.76	MACD	0.76	MACD	0.76
	ATR	0.24	ATR	0.24	ATR	0.24
2019.04	ASI	0.78	ASI	0.79	ASI	0.88
	RI	0.22	Down_to_Up_ Volatility	0.21	ATR	0.12
2019.05	RI	0.54	RI	0.52	RI	0.51
	Down_to_Up_ Volatility	0.46	Down_to_Up_ Volatility	0.48	Down_to_Up_ Volatility	0.49
2019.06	Share_Turnover	0.52	Share_Turnover	0.57	Down_to_Up_ Volatility	0.5
	RI	0.48	RI	0.43	RI	0.5

TABLE VIII 2018 OPTIMAL STOCK WEIGHTS

January	**000581.SZ**	**600856.SH**	**600230.SH**	**000935.SZ**	**600782.SH**
	0.187	0.178	0.177	0.238	0.22
February	**603658.SH**	**600054.SH**	**002400.SZ**	**002128.SZ**	**000990.SZ**
	0.211	0.194	0.202	0.185	0.208
March	**603268.SH**	**300426.SZ**	**603098.SH**	**300429.SZ**	**603566.SH**
	0.196	0.201	0.205	0.179	0.22
April	**603019.SH**	**002432.SZ**	**300386.SZ**	**002569.SZ**	**002769.SZ**
	0.195	0.195	0.212	0.18	0.219
May	**600578.SH**	**601099.SH**	**601928.SH**	**600926.SH**	**603858.SH**
	0.177	0.208	0.197	0.189	0.229
June	**000089.SZ**	**002503.SZ**	**300073.SZ**	**002419.SZ**	**600718.SH**
	0.191	0.21	0.198	0.201	0.2

TABLE VII 2019 OPTIMAL STOCK WEIGHTS

January	**300164.SZ**	**000996.SZ**	**002110.SZ**	**300171.SZ**	**300623.SZ**
	0.182	0.185	0.275	0.134	0.223
February	**000420.SZ**	**000429.SZ**	**000570.SZ**	**000338.SZ**	**000683.SZ**
	0.277	0.2	0.198	0.175	0.15
March	**000650.SZ**	**300294.SZ**	**300357.SZ**	**300031.SZ**	**600801.SH**
	0.249	0202	0.188	0.149	0.212
April	**000929.SZ**	**000888.SZ**	**002144.SZ**	**600509.SH**	**600259.SH**
	0.234	0.163	0.203	0.214	0.186
May	**603318.SH**	**000969.SZ**	**002511.SZ**	**600337.SH**	**000792.SZ**
	0.147	0.214	0.201	0.245	0.193
June	**600330.SH**	**603727.SH**	**600604.SH**	**601298.SH**	**002417.SZ**
	0.19	0.213	0.194	0.197	0.207

TABLE VIII 2018 PORTFOLIO EARNING RATE

	Equal-Weight Model	Shanghai Composite Index	CSI 300	Shenzhen Component Index	Mean-CVaR
January	10.90%	3.96%	4.61%	-0.16%	10.97%
February	-2.60%	-5.44%	-5.23%	-0.33%	-2.49%
March	-0.74%	-3.20%	-3.72%	-0.68%	-0.57%
April	-5.73%	-2.56%	-3.35%	-4.78%	-5.81%

May	1.34%	0.46%	1.03%	-0.46%	1.45%
June	-1.55%	-7.41%	-6.88%	-7.77%	-1.66%

TABLE IX 2019 PORTFOLIO EARNING RATE

	Equal-Weight Model	Shanghai Composite Index	CSI 300	Shenzhen Component Index	Mean-CVaR
January	3.59%	4.62%	4.84%	7.82%	5.02%
February	10.57%	17.54%	12.33%	12.99%	11.68%
March	12.11%	8.06%	3.23%	3.27%	12.45%
April	-7.04%	-5.78%	2.90%	-1.53%	-6.60%
May	8.26%	-0.23%	-0.27%	-1.49%	9.09%
June	1.44%	3.63%	3.07%	5.33%	1.50%

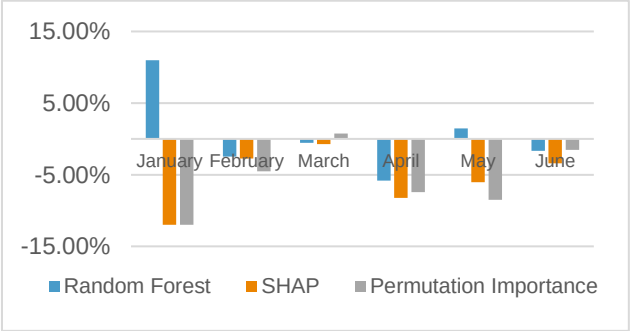

Fig. 8. 2018 Interpretative Portfolio Return Distribution Plot

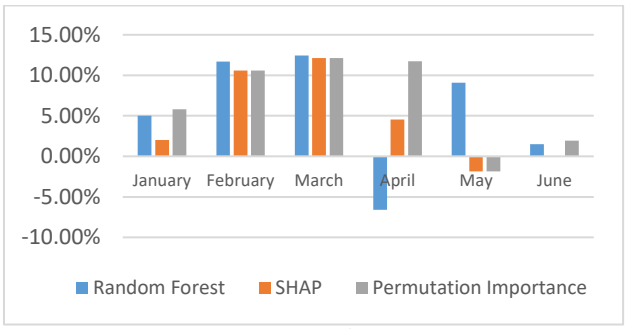

Fig. 9. 2019 Interpretative Portfolio Return Distribution Plot

IV. CONCLUSION

This article explores the integration of machine learning techniques with portfolio model optimization, introducing a method for constructing factor investment portfolios. This method combines sliding window MiniBatchKmeans clustering and Bayesian Random Forest, further optimized by the Mean-CVaR model. It creates a direct link between machine learning interpretability indicators and portfolio construction strategy formulation.

The portfolio model construction process is as follows: Initially, superior equities are identified using MiniBatchKmeans clustering and Bayesian-optimized Random Forest. Then, the Mean-CVaR portfolio model is constructed with stocks that demonstrate a strong correlation to return explanatory factors, as determined through clustering. This approach allows for the calculation of specific investment proportions for each stock.

The study conducts a numerical analysis using constituents of the CSI All-Share Index. The aim is to compare the Random Forest interpretability portfolio with other prominent indices and interpretability portfolios. Analysis results indicate that the Random Forest interpretability portfolio, created by clustering and selecting interpretative features from prediction results, outperformed the SHAP and Permutation interpretability portfolios and major indices in terms of returns. This highlights the Random Forest interpretability portfolio's effectiveness.

Ultimately, the objective of this paper is to establish a link between the interpretability of machine learning and portfolio construction using an indicator-driven strategy. This approach emphasizes the potential importance of interpretability portfolios in asset allocation, offering valuable insights for investors and researchers.

REFERENCES

1. Sharpe W F. Capital asset prices: A theory of market equilibrium under conditions of risk[J]. The journal of finance, 1964, 19(3): 425-442.
2. Fama E F, French K R. A five-factor asset pricing model[J]. Journal of financial economics, 2015, 116(1): 1-22.
3. Huang C F. A hybrid stock selection model using genetic algorithms and support vector regression[J]. Applied Soft Computing, 2012, 12(2): 807-818.
4. Xiujuan J. Quantitative Stock Selection Based on Support Vector Machine of Random Forest [J]. Journal of Regional Financial Research, 2019, 1: 27-30.
5. Shen K Y, Yan M R, Tzeng G H. Combining VIKOR-DANP model for glamor stock selection and stock performance improvement[J]. Knowledge-Based Systems, 2014, 58: 86-97.
6. Carvalho D V, Pereira E M, Cardoso J S. Machine learning interpretability: A survey on methods and metrics[J]. Electronics, 2019, 8(8): 832.
7. Doshi-Velez F, Kim B. Towards a rigorous science of interpretable machine learning[J]. arXiv preprint arXiv:1702.08608, 2017.
8. Çelik T B, İcan Ö, Bulut E. Extending machine learning prediction capabilities by explainable AI in financial time series prediction[J]. Applied Soft Computing, 2023, 132: 109876.
9. Ribeiro M T, Singh S, Guestrin C. " Why should i trust you?" Explaining the predictions of any classifier[C]//Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. 2016: 1135-1144.
10. Lundberg S M, Lee S I. A unified approach to interpreting model predictions[J]. Advances in neural information processing systems, 2017, 30.
11. Yun K K, Yoon S W, Won D. Interpretable stock price forecasting model using genetic algorithm-machine learning regressions and best feature subset selection[J]. Expert Systems with Applications, 2023, 213: 118803.
12. Breiman L. Random forests[J]. Machine learning, 2001, 45: 5-32.
13. Probst P, Wright M N, Boulesteix A L. Hyperparameters and tuning strategies for random forest[J]. Wiley Interdisciplinary Reviews: data mining and knowledge discovery, 2019, 9(3): e1301.
14. Jones D R, Schonlau M, Welch W J. Efficient global optimization of expensive black-box functions[J]. Journal of Global optimization, 1998, 13(4): 455.
15. Rosenberg B, Marathe V. Common factors in security returns: Microeconomic determinants and macroeconomic correlates[R]. University of California at Berkeley, 1976.
16. Sculley D. Web-scale k-means clustering[C]//Proceedings of the 19th international conference on World wide web. 2010: 1177-1178.
17. Rockafellar R T, Uryasev S. Optimization of conditional value-at-risk[J]. Journal of risk, 2000, 2: 21-42.
18. Cover T, Hart P. Nearest neighbor pattern classification[J]. IEEE transactions on information theory, 1967, 13(1): 21-27.
19. Quinlan J R. Induction of decision trees[J]. Machine learning, 1986, 1: 81-106.
20. Chen T, Guestrin C. Xgboost: A scalable tree boosting system[C]//Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016: 785-794.
21. Lundberg S M, Erion G G, Lee S I. Consistent individualized feature attribution for tree ensembles[J]. arXiv preprint arXiv:1802.03888, 2018.
22. Altmann A, Toloşi L, Sander O, et al. Permutation importance: a corrected feature importance measure[J]. Bioinformatics, 2010, 26(10): 1340-1347.

APPENDIX

TABLE I STOCK FACTORS

Sequence	Factor Name	Description	Additional description
1	Share Turnover	Mean of the daily turnover rate sequence over the last K months.	Daily Turnover Rate = Trading Volume / Free-Float Capital
2	Absolute Return to volume	ILLI$Q_{i,\ t}$=Avg($\lvert R_{i,d}\rvert / VOLD_{i,d}$)	$R_{i,d}$is the daily return rate of stock i during month t; $VOLD_{i,d}$is the daily trading volume of stock i during month t
3	Earnings to Price Ratio	Net profit attributable to the parent company in the last 12 months (TTM) / Total Market Value	The inverse of the Price-to-Earnings Ratio
4	Operating Cash Flow to Price Ration	Net cash flow from operating activities in the last 12 months (TTM) / Total Market Value	The inverse of the Price-to-Cash Flow Ratio
5	Long Term Reverse	At the end of month t, calculate the cumulative daily return rate from month (t-59) to month (t-12)	
6	Momentum Change	Cumulative daily return rate from month (t-6) to month (t-1) – Cumulative daily return rate from month (t-12) to month (t-7)	
7	Float Capitalization	Closing price of the day / Free-float capital of the day	
8	Firm Size	Closing price of the day / Total capital of the day	
9	Book to Market Equity	Total equity attributable to shareholders of the parent company in the latest reporting period / Total Market Value	The inverse of the Price-to-Book Ratio
10	Idiosyncratic Volatility	Standard deviation of residuals from CAPM or Fama-French 3-factor model regression over the recent K months $std(\varepsilon_{i,t})$	CAPM: $r_{i,t} - r_{f,t} = \alpha_i + \beta_i(r_{m,t} - r_{f,t}) + \varepsilon_{i,t}$ FF3-factor model: $r_{\{i,t\}} - r_{\{f,t\}} = \alpha_i + \beta_{\{mkt,i\}}(r_{\{mkt,t\}} - r_{\{f,t\}}) + \beta_{\{SMB,i\}}SMB_t + \beta_{\{HML,i\}}HML_t + \epsilon_{\{i,t\}}$
11	Market Leverage	Total assets of the latest reporting period / Total shareholder equity of the same period.	
12	Down to Up Volatility	$DUVOL_i = \log\left(\frac{(n_u - 1)\sum_d (r_{i,t} - \bar{r}_i)^2}{(n_d - 1)\sum_u (r_{i,t} - \bar{r}_i)^2}\right)$	$r_{i,t}$is the return of stock i at time t; $n_u$is the number of days with returns above the average compound return; $n_d$is the number of days with returns below the average compound return
13	ASI	Accumulative Swing Index	
14	MACD	Moving Average Convergence Divergence	
15	ATR	Average True Range	
16	RI	Regional Index	

Jihui Shi, PhD is an associate professor in the School of Economics and Management at Huzhou University with an interest in electronic commerce and social media. His research has appeared in Cluster Compute and other academic journals in China Mainland. Contact: Huzhou University, Erhuan East Road No.759, Huzhou City, China. Email: jacky-shi@zjhu.edu.cn.

Wenzheng Zhang, is a Master's candidate in Computer Science and Engineering at Huzhou Teachers College, primarily focuses on research areas including quantitative trading, investment portfolio optimization, and artificial intelligence and machine learning. Contact: Huzhou University, Erhuan East Road No.759, Huzhou City, China. Email: appharos@gmail.com