A Study of Stock Price Prediction Models Based on Kernel Clustering Localized Sliced Inverse Regression and Bayesian Model Averaging

Zhenpei Yang *

Management College, Ocean University of China Qingdao, China, 266110

yzp626929@outlook.com

Abstract. Multifactor prediction models are pivotal in quantitative finance research, yet face issues such as the curse of dimensionality, complex interconnections, and model overfitting. To address these challenges, this study introduces a machine learning predictive model grounded in sufficient dimension reduction and model averaging principles, tailored for stock price forecasting. This method innovatively employs kernel function-based clustering and weighting to refine classic Sliced Inverse Regression, thereby mitigating the curse of dimensionality while maximally preserving the efficacy of predictive factors on stock prices. Furthermore, this approach utilizes Bayesian Model Averaging to navigate the intricate relationships between factors and stock prices, alleviating the risks of overfitting and underfitting. Empirical analysis demonstrates that, compared to traditional quantitative prediction models, this approach produces lower mean squared error, absolute error, and relative error in stock price forecasting, thereby confirming its accuracy and robustness.

Keywords: Sufficient Dimension Reduction, Bayesian Model Averaging, Machine Learning, Stock Price Prediction.

1. Introduction

In recent years, quantitative trading has become increasingly popular in the stock market, utilizing computer algorithms and mathematical models for trading decisions, integrating techniques from finance, mathematics, computer science, and statistics. In quantitative finance, stock price prediction remains a focal point for investors, as its accuracy is crucial for effective investment decisions and maximizing returns. Traditional analysis relies on fundamental and technical analysis; however, with the rapid development of machine learning, its application in predicting stock prices has become a new research trend. Machine learning can handle vast datasets, identify complex patterns, and enhance predictive accuracy, showcasing unique advantages in processing high-dimensional, non-linear data with significant market noise influence. Initially, researchers assumed a linear relationship between the response variable Y and the predictor X, attempting to construct predictive models using multiple linear regression. An early multifactor model was the three-factor model proposed by Fama and French (1993) [1], suggesting that a stock's excess returns can be explained by market risk, size, and value factors. Later, Fama and French (2015) [2] expanded on this model by adding profitability and investment factors, highlighting the role of financial indicators in stock pricing and providing investors with a more comprehensive framework for evaluating and selecting stocks.

However, researchers found that as multifactor models expanded, the number of predictors X increased, leading to the "curse of dimensionality" - a rapid rise in model complexity due to the growing number of features, significantly reducing model stability and interpretability. To address these issues, penalized linear regression methods were introduced. For instance, Xie and Hu (2017) [3] compared the LASSO and Elastic Net methods in multifactor quantitative investment models, finding that Elastic Net performed better in selecting effective factors and constructing portfolios, aiding investors in achieving higher excess returns. Zeng and Zhou (2017) [4] reviewed variable selection methods for high-dimensional data, highlighting penalized linear regression techniques like Lasso, SCAD, and the Dantzig selector. Their effectiveness in variable selection and model construction was analyzed through examples. While these methods mitigate overfitting to some extent, they fall short when the true relationship between the response variable Y and the predictor X is a

complex, unknown nonlinear form. Nonlinear regression methods like Support Vector Machines (SVM), KNN regression, Decision Trees, BP Neural Networks, Random Forest, and XGBoost have gained research interest. These methods don't require assumptions about data distribution or model form, offering great flexibility. Wu and Lai (2018) [5] investigated the method of combining Support Vector Machine (SVM) algorithm with stock price trend analysis for stock price prediction and verified the effectiveness of this method in improving the accuracy of trend prediction and the total profit rate through actual data. Xian (2020) [6] applied the KNN algorithm in a quantitative trading strategy for A-shares, showing its efficacy in enhancing annual returns and reducing volatility with 11 selected factors. Lin et al. (2020) [7] conducted an applied research study, demonstrating that a composite model combining the Grey GM (1,1) model and the BP (Backpropagation) Neural Network can offer higher prediction accuracy in stock price forecasting, which holds significant reference value for the dynamic analysis of stock prices. Li (2024) [8] exhibited the application of a decision-tree-based multifactor stock selection model in the Chinese stock market, confirming its efficacy in predicting stock returns and achieving excess returns through empirical analysis. Deng and Li (2020) [9] developed a Random Forest stock prediction model based on technical indicators, showing superiority in accuracy and AUC values after grid search optimization. Chen et al. (2018) [10] combined Pearson correlation analysis with the XGBoost algorithm for stock price prediction, achieving promising training and prediction results. However, these models face challenges with high-dimensional data, such as slower convergence and sensitivity to initial weight settings in methods like BP neural networks, potentially leading to long training times, high computational costs, and issues like gradient vanishing or exploding.

Addressing the aforementioned challenges, this research proposes a cutting-edge stock price forecasting model based on Kernel Clustering Localized Sliced Inverse Regression [12] and Bayesian Model Averaging [11] (KCLSIR-BMA). It refines traditional slice inverse regression by kernel clustering, leverages kernel functions for efficient high-dimensional data management, and enhances feature extraction for stock price prediction. Subsequently, it employs Bayesian Model Averaging (BMA) to aggregate various predictive models, with BMA adaptively tuning model weights to detect possibly complex relationships between factors and stock prices while also alleviating overfitting and underfitting. In validation, the KCLSIR-BMA approach outperforms six classical forecast models, predicting stock Gap (current opening price - previous closing price), with lower mean squared error, absolute error, and relative error across nearly all stocks and periods, evidencing the model's robustness.

2. Stock Price Prediction Model Based on Kernel Clustering Localized Sliced Inverse Regression and Bayesian Model Averaging

This study combines KCLSIR with BMA techniques, effectively processes high-dimensional financial data using Gaussian kernel functions, and subsequently employs sliced inverse regression for key feature extraction to achieve sufficient dimensionality reduction. The Bayesian Model Averaging then consolidates forecasts from multiple models, optimizes weight allocation, and captures the intricate relationship between stock prices and predictors, enhancing model robustness. Empirical analysis confirms that the model performs superiorly.

Kernel Clustering Localized Sliced Inverse Regression (KCLSIR) represents a functional data analysis technique that merges kernel methods with the concept of localization. KCLSIR is designed to address the high dimensionality and non-linearity of financial time series data, encompassing opening prices, highest prices, lowest prices, closing prices, previous closing prices, price changes, percentage changes, trading volumes, and turnover. It initially smooths the data using Gaussian kernel functions to mitigate random fluctuations or noise and to reveal inherent data structures or patterns. Through localization, KCLSIR identifies and retains local features that most significantly impact stock price variations, thus achieving effective dimensionality reduction. The implementation process of the proposed approach is detailed as follows:

Data preprocessing is performed to remove missing and outlier values. The nine datasets—opening price, highest price, lowest price, closing price, previous closing price, price change, percentage change, trading volume, and turnover—are standardized using the Z-score method. This involves calculating the mean and variance as follows:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i, \tag{1}$$

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n} (x_i - \mu)^2}. \tag{2}$$

The data from the single-factor dataset, denoted by x_i and the number of samples in it, denoted by n, with the sample size in this article corresponding to the number of trading days from January 1, 2023, to December 31, 2023. The mean μ and standard deviation σ are computed for standardization. Each observation x_i is transformed using the standardization formula:

$$x' = \frac{x_i - \mu}{\sigma}, \tag{3}$$

where x' represents the standardized data with a mean of 0 and standard deviation of 1. This standardization removes discrepancies in scale between features, ensuring equal importance in the model regardless of their original units of measurement.

Subsequently, the global covariance operator $\hat{\Gamma}$ for the dataset X is computed, which describes the interrelatedness among all features within the data. The global covariance operator $\hat{\Gamma}$ is determined using the following formula:

$$\hat{\Gamma} = \frac{1}{n}X^T X - outer(\bar{x}, \bar{x}). \tag{4}$$

Here, X is the matrix of standardized features, X^T denotes the transpose of X, n is the number of elements in the matrix, and $\bar{x}$ is the mean vector of each column (i.e., each feature) in X, The term $outer()$ denotes the computation of the outer product. This method computes the covariance operator as the weighted average of inner products minus the outer product of the mean vector $\bar{x}$ with itself, resulting in a centralized (around $\bar{x}$) covariance matrix.

Let the response variable be represented by Y. Localization is achieved by binning Y, with each bin representing a localized region, and the number of bins being H. For each local region h, a corresponding subset X_{slices} is selected from X, and then for each data point, we calculate its local mean $\hat{\mu}_{i,local}$ and local covariance operator $\hat{\Gamma}_{local}$.

Kernel smoothing uses kernel functions to reduce random fluctuations or noise in the data, thus more clearly revealing the intrinsic structure or patterns. Common kernel functions include the linear, polynomial, Gaussian (RBF), sigmoid, and cosine similarity kernels, among others. After performing 4-fold cross-validation to assess various kernel functions for their effectiveness in kernel smoothing, this study selected the Gaussian kernel function, with the formula as follows:

$$W(x_i, x_j) = exp\left(-\frac{\| x_j - x_i \|^2}{2{\sigma_b}^2}\right). \tag{5}$$

In this context:

- x_i and x_j are input vectors representing financial factor vectors selected from X_{slices}.
- $\| x_j - x_i \|$ is the squared Euclidean distance between x_j and x_i,with n denoting the dimension of the vectors:

$$\| x_j - x_i \| = \sqrt{(x_{j1} - x_{i1})^2 + (x_{j2} - x_{i2})^2 + \cdots + (x_{jn} - x_{in})^2}. \quad (6)$$

- σ_b is the bandwidth parameter of the Gaussian kernel, which determines the smoothing extent. The larger the σ_b, the wider the "reach" of the kernel, incorporating more points and flattening the function's shape; conversely, a smaller σ_b results in a sharper, more localized function.
- $W(x_i, x_j)$ signifies the similarity between x_i and x_j; the value of $W(x_i, x_j)$ increases as they grow closer and approaches zero when they are distant.

Using kernel averaging, the average kernel function within the slice for x_j is computed, where the local mean is calculated by weighting the top t nearest neighbors with the Gaussian kernel function:

$$\hat{\mu}_{i,local} = \frac{\sum_{j \in s} x_j W(x_i, x_j)}{\sum_{j \in s} W(x_i, x_j)}. \quad (7)$$

Here, s denotes the nearest neighbor index set for data point x_i and the weights $W(x_i, x_j)$ are computed by the Gaussian kernel function. The local covariance operator $\hat{\Gamma}_{local}$ is calculated by summing the outer products of the differences between each local mean $\hat{\mu}_{i,local}$ and the overall mean $\bar{\mu}_{\text{local}}$, as follows:

$$\hat{\Gamma}_{local} = \frac{1}{n} \sum_{i=1}^{n} (\hat{\mu}_{i,local} - \bar{\mu}_{local}) (\hat{\mu}_{i,local} - \bar{\mu}_{local})^T. \quad (8)$$

To prevent matrix singularity, we introduce a regularization term ρI, where ρ is the regularization parameter, and I is the identity matrix. The updated covariance operator $\hat{\Gamma}_{new}$ is computed as follows:

$$\hat{\Gamma}_{new} = (\hat{\Gamma} + \rho I)^{-1} \hat{\Gamma}_{local}. \quad (9)$$

Subsequently, eigenvalue decomposition is performed on $\hat{\Gamma}_{new}$, obtaining eigenvalues λ and eigenvectors ω :

$$\hat{\Gamma}_{ncw} \omega = \lambda \omega. \quad (10)$$

The primary eigenvectors corresponding to the top k eigenvalues ω_i are selected, defining the principal directions of the data, i.e., the reduced feature space Ω:

$$\Omega = [\omega_1, \omega_2, \ldots, \omega_k]. \quad (11)$$

Using Ω , the original feature matrix X is transformed into the reduced feature matrix Z:

$$Z = \Omega^T X. \quad (12)$$

Having obtained the dimensionally reduced feature matrix Z, the dataset now effectively isolates crucial information for stock price prediction through the kernel clustering local slicing inverse regression technique, significantly reducing the dimensionality of the original data. This not only enhances data processing efficiency but also improves the model's accuracy in predicting stock price movements.

Based on this, a Bayesian Model Averaging (BMA) model can be constructed using the feature matrix Z and the corresponding response variable Y, optimizing weight distribution through model integration techniques. Define $P(Y|Z, m)$ as the likelihood function of Y given model m and the reduced feature matrix Z. For each model m, the parameter set θ_m is estimated using Z. The likelihood function $P(Y|Z, \theta_m, m)$ is defined as the probability of observing the response variable Y, given Z, m, and θ_m. $P(Y|Z, m)$ computed as follows:

$$P(Y|Z, m) = \int P(Y|Z, \theta_m, m) P(\theta_m|m) d\theta_m, \quad (13)$$

where, $P(\theta_m|m)$ is the prior distribution of parameter θ_m. It is evident that the marginal likelihood function $P(Y|Z,m)$ does not depend on specific parameter values θ_m. For each model m, the objective is to compute its posterior probability $P(m|Z,Y)$:

$$P(m|Z,Y) \propto P(Y|Z,m)\,P(m), \tag{14}$$

where $P(m)$ is the prior probability assigned to mode m. The obtained posterior probabilities $P(m|Z,Y)$ can then be viewed as weights for each model in the model averaging.

For a new observation (i.e., the test set) Z_{new}, we calculate the weighted average prediction using the BMA model trained on data from the previous three months. The BMA prediction is the weighted average of all model predictions, with weights given by each model's posterior probability $P(m|Z,Y)$, yielding the prediction result $\hat{y}(Z_{new})$:

$$\hat{y}(Z_{new}) = \sum_m P\,(m|Z,Y)\hat{y}_m(Z_{new}), \tag{15}$$

where $\hat{y}_m(Z_{new})$ is the prediction of model m for the test set Z_{new}.

3. Data Analysis

3.1. Data Source and Experimental Setup

Data for this study were sourced from the Tushare big data open community, containing nearly 20,000 daily transaction data points for seven stocks listed on the Shanghai and Shenzhen stock markets, spanning from January 1, 2023, to December 31, 2023. The chosen stocks represent leading companies in various major industries: Vanke A (000002.SZ) for real estate, ZTE Corporation (000063.SZ) for telecommunications technology, Hikvision (002415.SZ) for electronic devices, BYD Company (002594.SZ) for new energy vehicles, Wanhua Chemical (600309.SH) for chemical materials, Kweichow Moutai (600519.SH) for food and beverages, and SMIC (688981.SH) for chip manufacturing. These stocks effectively reflect sector fluctuations, offering valuable insights for investors.

The model uses nine indicators (opening prices, highest prices, lowest prices, closing prices, previous closing prices, price changes, percentage changes, trading volumes, and turnover) from the daily transaction data as independent variables to forecast the dependent variable Gap, which is the difference between the opening price of the current day and the closing price of the previous day, to devise relevant strategies.

This study compares the KCLSIR-BMA method with six traditional stock price prediction models: Elastic-Net[3], LASSO[4], Decision Tree[8], Random Forest[9], Neural Networks[7], and XGBoost[10], to evaluate the quality of the new model.

Rolling prediction, a common forecasting method in time series analysis that continuously updates to adapt to the series' non-stationarity for more accurate forecasts, was employed. Li and Tang (2022) [13] constructed a novel stock price prediction model by integrating technical analysis indicators, fundamental analysis indicators with a hybrid recurrent neural network, and validated its high precision in long-term stock price forecasting using the rolling sample prediction evaluation method. A rolling prediction approach was adopted for the experiment, using data from the three months preceding the target month as the training set and the target month as the test set, with eight rolling forecasts conducted for the period from May 1, 2023, to December 31, 2023. For example, data from February 1, 2023, to April 30, 2023, were used as the training set, with May 1, 2023, to May 31, 2023, serving as the prediction set for evaluation. The assessment metrics included Mean Squared Error (MSE), Mean Absolute Error (MAE), and Mean Relative Error (MRE), calculated as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{16}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|, \tag{17}$$

$$MRE = \frac{1}{n}\sum_{i=1}^{n} \left|\frac{y_i - \hat{y}_i}{y_i}\right|, \tag{18}$$

where y_i is the actual value of Gap, and $\hat{y}_i$ is the corresponding predicted value. Here, n represents the number of $\hat{y}_i$ in a single rolling period. The results consistently show that the KCLSIR-BMA method produced lower MSE, MAE, and MRE across almost all stocks and periods, maintaining relative stability, thereby confirming its accuracy and robustness from an empirical perspective.

The model primarily utilizes R for programming, with Python employed for initial data cleaning, removing outliers and missing values, and standardizing the data using Z-score normalization. The hyperparameters of the KCLSIR-BMA model were determined through cross-validation: $k = 2, H = 4, \rho = 0.5, \sigma_b = 1, t = 2$.

3.2. Comparative Results

Table 1 Summary of metrics data for 000002.SZ

Share	Metrics	Models						
		KCLSIR-BMA	Elastic-Net	LASSO	Decision Tree	Random Forest	Neural Networks	XGBoost
000002.SZ	MSE-Mean	0.0139	0.0148	0.0150	0.0224	0.0160	0.0331	0.0219
	MSE-Var	0.0003	0.0002	0.0002	0.0003	0.0003	0.0004	0.0004
	MAE-Mean	0.0756	0.0792	0.0800	0.1028	0.0801	0.1217	0.0960
	MAE-Var	0.0008	0.0008	0.0008	0.0008	0.0006	0.0014	0.0011
	MRE-Mean	1.3010	1.4375	1.4626	2.0699	1.4966	2.9933	2.0616
	MRE-Var	0.0722	0.2457	0.2873	0.1483	0.0450	1.8728	0.7127

Table 2 Summary of metrics data for 000063.SZ

Share	Metrics	Models						
		KCLSIR-BMA	Elastic-Net	LASSO	Decision Tree	Random Forest	Neural Networks	XGBoost
000063.SZ	MSE-Mean	0.1186	0.1700	0.1743	0.4748	0.2334	1.7122	0.5579
	MSE-Var	0.0101	0.0148	0.0153	0.2167	0.0322	11.5698	0.6097
	MAE-Mean	0.2255	0.2798	0.2834	0.5100	0.3561	0.7784	0.5268
	MAE-Var	0.0103	0.0145	0.0147	0.0911	0.0180	0.3929	0.1464
	MRE-Mean	1.2374	2.5339	2.6251	6.0007	3.6734	11.8349	6.5676
	MRE-Var	0.0695	0.7650	0.9718	20.2361	4.8502	136.7128	41.1616

Table 3 Summary of metrics data for 002415.SZ

Share	Metrics	Models						
		KCLSIR-BMA	Elastic-Net	LASSO	Decision Tree	Random Forest	Neural Networks	XGBoost
002415.SZ	MSE-Mean	0.0511	0.0552	0.0558	0.1087	0.0632	0.1418	0.0931
	MSE-Var	0.0029	0.0026	0.0026	0.0113	0.0039	0.0055	0.0123
	MAE-Mean	0.1295	0.1456	0.1474	0.1989	0.1502	0.2748	0.1827
	MAE-Var	0.0019	0.0018	0.0019	0.0055	0.0032	0.0069	0.0070
	MRE-Mean	1.5501	2.3148	2.3976	3.5476	2.0343	5.7732	3.0264
	MRE-Var	0.2030	0.5666	0.6438	3.1565	0.3328	7.6888	1.2016

Table 4 Summary of metrics data for 002594.SZ

Share	Metrics	Models						
		KCLSIR-BMA	Elastic-Net	LASSO	Decision Tree	Random Forest	Neural Networks	XGBoost
002594.SZ	MSE-Mean	4.4569	6.3143	6.4203	8.9560	6.7815	13.7635	8.6839
	MSE-Var	13.4818	19.2643	18.8862	31.6545	35.7554	99.7523	72.8060
	MAE-Mean	1.3577	1.7349	1.7496	2.0647	1.6106	2.7014	1.8858
	MAE-Var	0.1596	0.3244	0.3441	0.2298	0.2036	1.1884	0.4439
	MRE-Mean	5.1435	11.4493	11.9288	26.1530	17.6194	44.2820	23.1660
	MRE-Var	80.2489	140.1747	166.2398	1112.3597	661.2158	1161.4945	1077.5173

Based on the KCLSIR-BMA model trained, and six comparison models, eight rolling forecasts were conducted for the Gap variable of the seven stocks. The MSE, MAE, and MRE metrics for each forecast (i.e., each month) were compared, yielding the mean (MSE-Mean, MAE-Mean, MRE-Mean) and variance (MSE-Var, MAE-Var, MRE-Var) of these metrics across different months and different predictive variables, as shown in Tables 1 to 7 below:

Table 5 Summary of metrics data for 600309.SH

Share	Metrics	Models						
		KCLSIR-BMA	Elastic-Net	LASSO	Decision Tree	Random Forest	Neural Networks	XGBoost
600309.SH	MSE-Mean	0.3190	0.4129	0.4376	0.5598	0.3485	0.9409	0.4114
	MSE-Var	0.0947	0.0930	0.1037	0.1431	0.0921	0.4833	0.0614
	MAE-Mean	0.3771	0.4345	0.4439	0.5470	0.4127	0.7260	0.4658
	MAE-Var	0.0247	0.0205	0.0211	0.0291	0.0215	0.0808	0.0120
	MRE-Mean	1.5354	2.5949	2.7507	5.8988	2.8876	8.1208	3.9732
	MRE-Var	0.1569	2.7235	3.4338	8.1154	1.4806	14.9892	4.1395

Table 6 Summary of metrics data for 600519.SH

Share	Metrics	Models						
		KCLSIR-BMA	Elastic-Net	LASSO	Decision Tree	Random Forest	Neural Networks	XGBoost
600519.SH	MSE-Mean	272.8312	300.7157	308.5741	483.6427	315.9048	574.3202	375.6666
	MSE-Var	185384.0982	178253.3006	171642.6779	456985.8146	176258.0831	327382.0605	256654.2128
	MAE-Mean	8.6316	9.6563	9.9269	13.8273	10.3819	16.2874	11.7851
	MAE-Var	11.2170	10.8241	11.0683	35.8240	13.3684	44.2045	30.3309
	MRE-Mean	2.9202	2.8664	3.0781	8.4624	3.5127	12.9698	2.9293
	MRE-Var	7.7235	3.4118	4.3960	37.0044	3.6558	217.4740	2.7218

Table 7 Summary of metrics data for 688981.SH

Share	Metrics	Models						
		KCLSIR-BMA	Elastic-Net	LASSO	Decision Tree	Random Forest	Neural Networks	XGBoost
688981.SH	MSE-Mean	0.2220	0.2563	0.2590	0.3686	0.2666	0.7012	0.3962
	MSE-Var	0.0489	0.0415	0.0417	0.0539	0.0454	0.2734	0.1465
	MAE-Mean	0.2870	0.3293	0.3320	0.4090	0.3316	0.5731	0.3702
	MAE-Var	0.0137	0.0158	0.0162	0.0116	0.0128	0.0557	0.0316
	MRE-Mean	1.3251	2.4537	2.4957	3.5709	2.5073	5.3151	3.4810
	MRE-Var	0.2850	3.2473	3.3387	6.2762	8.5485	12.6718	13.5017

Overall, it is evident from the stock price prediction experiments conducted on the selected seven stocks that the KCLSIR-BMA model exhibited lower MSE, MAE, and MRE across all periods for most of the stocks, while also maintaining relative stability, indicative of higher accuracy and robustness. When analyzing the prediction results for "600519.SH Kweichow Moutai", it was noted that the mean and variance data of MSE were particularly distinctive. As a leading company in China's high-end liquor market, Kweichow Moutai has a unique market position and high brand value, which may subject its stock price to different influencing factors compared to other stocks. Moreover, due to the brand effect of Moutai, investor sentiment towards its stock might be more extreme, possibly leading to irrational price fluctuations and thus affecting model predictions.

Taking ZTE Corporation (000063.SZ) as a specific case for analysis, Figures 1 to 3 below display the comparison of MSE, MAE, and MRE for the eight rolling forecasts across different models:

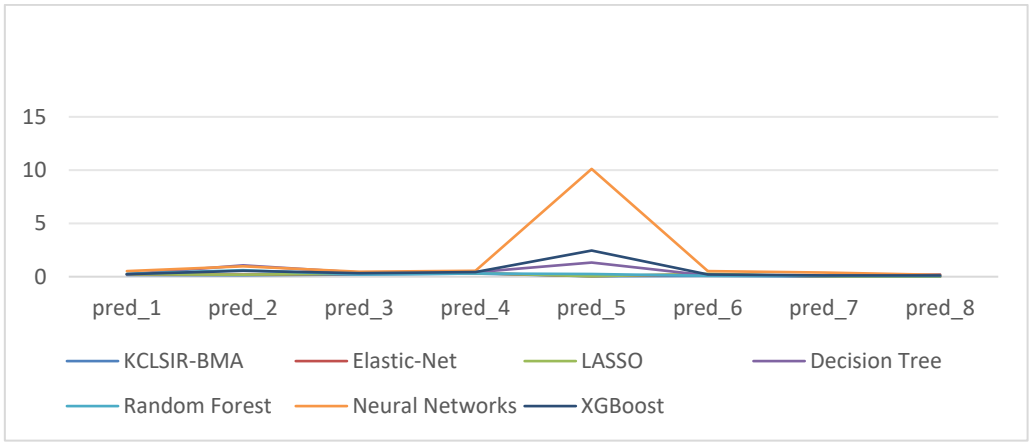

Figure 1 Mean Squared Error

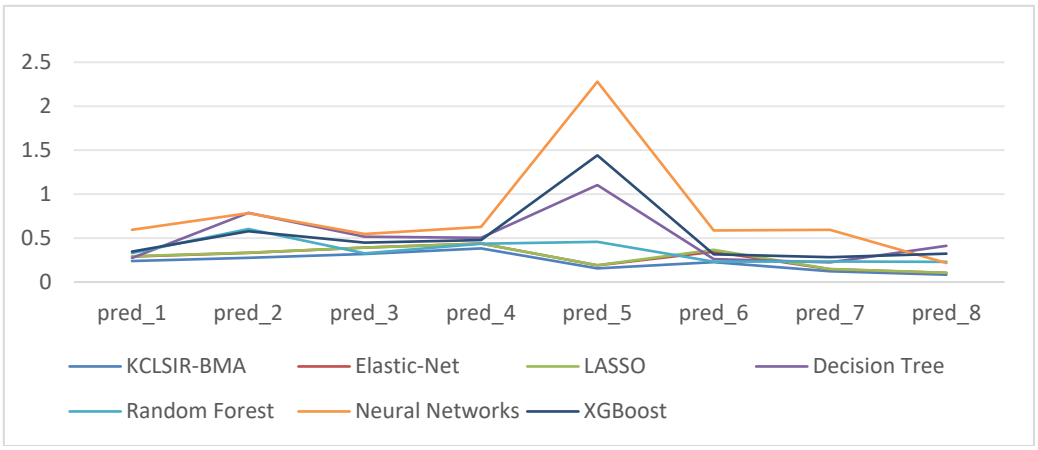

Figure 2 Mean Absolute Error

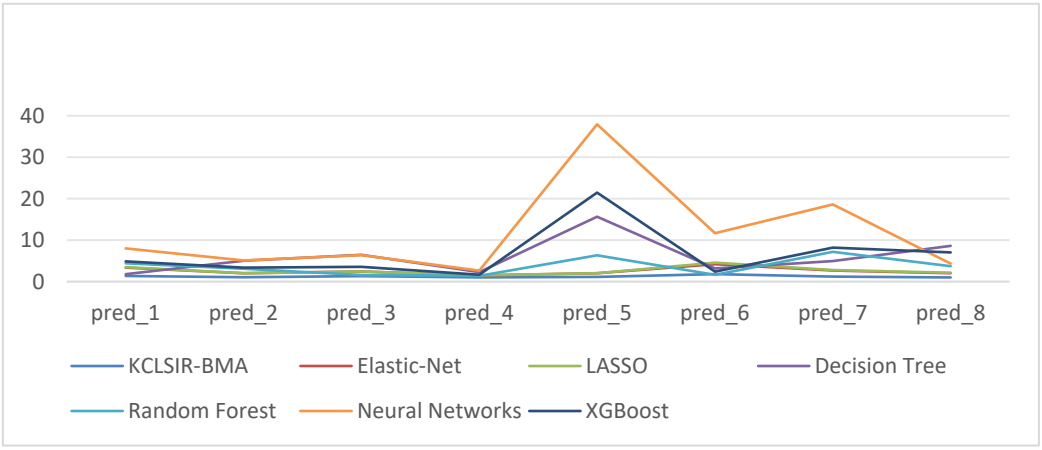

Figure 3 Mean Relative Error

It can be intuitively seen that the KCLSIR-BMA model maintains lower values for all three indices and demonstrates relative stability, significantly outperforming algorithms such as Neural Networks, XGBoost, Decision Trees, and Random Forests.

Taking MSE as an example, we calculate the percentage improvement r_{MSE} of the KCLSIR-BMA model relative to other models using the following formula:

$$r_{MSE} = \left(1 - \frac{MSE_{K-B}}{MSE_i}\right) \times 100\%, \tag{19}$$

where MSE_{K-B} denotes the metrics about MSE (including mean, variance, etc.) of the KCLSIR-BMA model, and MSE_irepresents the MSE and its related metrics of the other comparative models.

Similar formulas can be applied for MAE and MRE to calculate the percentage of improvement. The percentage improvement of MSE-Mean, MSE-Var, MAE-Mean, MAE-Var, MRE-Mean, and MRE-Var of the KCLSIR-BMA model for the six comparison models was calculated and summarized in Table 8 below:

Table 8 Summary of Improvement percentages

	Improvement percentages (r)					
Model	MSE-Mean	MSE-Var	MAE-Mean	MAE-Var	MRE-Mean	MRE-Var
Elastic-Net	30.23%	31.53%	19.41%	29.48%	51.17%	90.92%
LASSO	31.98%	33.94%	20.44%	30.23%	52.86%	92.85%
Decision Tree	75.02%	95.33%	55.79%	88.73%	79.38%	99.66%
Random Forest	49.20%	68.52%	36.69%	42.98%	66.32%	98.57%

The table and chart data indicate that all improvement percentages are positive, confirming that the KCLSIR-BMA model achieved the smallest values among the seven models for these six indicators. It is also observed that the KCLSIR-BMA model improved upon the indicators of the other models by at least 19.41%, with most improvement percentages exceeding 50% and some surpassing 90%. These figures demonstrate the model's significant superiority over traditional machine learning prediction models in terms of higher accuracy and robustness.

4. Conclusion and Extension

This paper presents a stock price prediction model based on Kernel Clustering Local Sliced Inverse Regression and Bayesian Model Averaging (KCLSIR-BMA), developed through in-depth research and empirical analysis. This model demonstrates significant advantages in processing high-dimensional financial time series data, exhibiting higher accuracy and robustness in its predictions. Analysis of data from seven representative stocks listed on the Shanghai and Shenzhen stock markets reveals that the KCLSIR-BMA model can provide smaller Mean Squared Error (MSE), Mean Absolute Error (MAE) and Mean Relative Error (MRE) in most cases, affirming its efficacy and practicality in the domain of stock price forecasting. The implementation of the KCLSIR-BMA model in stock price prediction can potentially lead to more informed investment decisions, thereby enhancing the likelihood of achieving greater financial returns for investors.

Firstly, the KCLSIR-BMA model effectively addresses the curse of dimensionality associated with high-dimensional data through kernel clustering techniques, while also retaining local features that are crucial for stock price variations using the method of sliced inverse regression. Moreover, the application of Bayesian model averaging enables the model to adaptively adjust the weights of different prediction models, capturing the complex nonlinear relationships between stock prices and various factors, thereby mitigating issues related to overfitting and underfitting to a certain degree. In an empirical analysis, the KCLSIR-BMA model displayed lower error metrics compared to six classical quantitative stock price prediction models, underscoring the superior predictive capabilities of this novel model in forecasting dependent variables.

Future research could explore the use of Sliced Average Variance Estimation (SAVE) as an alternative to traditional Sliced Inverse Regression, combined with Kernel Clustering to explore its comparative advantages in dimension reduction efficacy over the KCLSIR approach. Investigating which model averaging techniques (including Bayesian Model Averaging, and Mallows' criterion, among others) yield superior forecasting performance across different market scenarios—such as high market volatility or clear upward or downward trends—is a valuable research direction. In particular, for stocks with high prices and high volatility (such as Kweichow Moutai), it is of great significance to study how to optimize the model structure to achieve lower MSE means and variances for enhancing the practicality and accuracy of the model.

References

[1] Fama, E. F., & French, K. R. (1993). Common risk factors in the returns on stocks and bonds. Journal of Financial Economics, 33(1), 3-56.

[2] Fama, E. F., & French, K. R. (2015). A five-factor asset pricing model. Journal of Financial Economics, 116(1), 1-22.

[3] Xie, H. L., & Hu, D. (2017). The application of multifactor quantitative models in portfolio cfonstruction: A comparative study based on LASSO and Elastic Net. Statistics and Information Forum, 10, 36-42.

[4] Zeng, J., & Zhou, J. J. (2017). A review of variable selection methods for high-dimensional data. Journal of Mathematical Statistics and Management, 04, 678-692.

[5] Wu, C. X., & Lai, J. W. (2018). Research on stock prediction method based on SVM and stock price trend. Software Guide, (4), 42-44.

[6] Xian, J. (2020). A quantitative trading strategy for A-shares based on the KNN algorithm. Huan Bo Hai Economic Outlook, 01, 153.

[7] Lin, G. C., Du, Y. J., & Liu, J. (2020). Application of Two Types of Composite BP Neural Network Models in Stock Price Forecasting. Modern Commercial Trade Industry, 26, 141-143.

[8] Li, M. Y. (2024). Research on multifactor stock selection model based on decision trees. Productivity Research, 02, 145-149.

[9] Deng, J., & Li, L. (2020). The application of parameter-optimized random forest in stock prediction. Software, 01, 178-182.

[10] Chen, Y. S., Tang, Z. J., Luo, Y., & Yang, J. (2018). Stock price prediction research combining Pearson optimization with Xgboost algorithm. Information Technology, 09, 84-89.

[11] Wang, G., & Song, X. (2018). Functional sufficient dimension reduction for functional data classification. Journal of Classification, 35, 250-272.

[12] Raftery, A., Hoeting, J., Volinsky, C., Painter, I., Yeung, K. Y., Sevcikova, M. H., & Suggests, M. A. S. S. (2022). Package 'BMA'.

[13] Li, X. J., & Tang, P. (2022). Stock Price Forecasting Based on Technical Analysis, Fundamental Analysis, and Deep Learning. Statistics and Decision Making, (02), 146-150.