A multi-factor quantitative model based on weighted SIR-SAVE and model averaging

Jialuo Wang [1, *, #], Weihan Wang [2, #], Fan Yang [2, #]

[1] School of Mathematics-East, China University of Science and Technology, Shanghai, China, 201424

[2] Sauder school of business, University of British Columbia, BC, Canada, V6T 1Z4

* Corresponding Author Email: 21012409@mail.ecust.edu.cn

#These authors contributed equally.

Abstract. In financial quantitative research, there are many kinds of predictors that are closely related to stock prices, so how to design effective machine learning algorithms for such high-dimensional prediction problems is a hot issue. Based on this, this paper proposes a stock factor quantization and prediction model based on the idea of sufficient dimensionality reduction and model averaging. The proposed method is based on the principle of conditional independence, which can greatly retain the validity of the predictors on the stock price while solving the dimensionality catastrophe problem. In addition, the method adaptively learns the link function between the quantized factors and stock prices by using the model averaging method to effectively weigh the variance and bias of the prediction model. In the actual data analysis, this paper chooses the weighted SIR-SAVE as the adequate dimensionality reduction method, the mutual information criterion as the assignment rule of model averaging, and some classical high-dimensional regression techniques as the comparison method, and the empirical results show that the proposed method has a higher accuracy under the evaluation indexes of mean squared error and absolute error, and possesses a certain degree of robustness. Finally, since the sub-models in the model averaging are free, the proposed method is also applicable to the problem of classifying the prediction of stock ups and downs.

Keywords: Stock price prediction, high dimensional problems, sufficient dimensionality reduction, model averaging.

1. Introduction

Accurate stock price prediction models can effectively capture market trends and help investors' better grasp investment opportunities and adjust investment strategies to cope with market risks. In financial quantitative research, there are a variety of factors related to stock prices, and how to accurately predict the future rise and fall of stock prices while quantifying the effective factors has been a key research direction in both academia and the industry.

In recent years, a number of statistical learning models have been applied by researchers to the task of stock price prediction, which can be categorized into the following types of methods. The first one is time series-based forecasting methods, such as ARIMA [1] and GARCH [2], both of which are based on features such as serial autocorrelation and volatility to model and forecast stock prices. These methods are good at predicting short-term trends and volatile series, but when faced with long-term trends, structural changes, or complex nonlinear serial relationships, the assumption of linearity and a limited window of historical data constrain the accuracy of these methods for long-term forecasting. For example, Guisheng Zhang and Xindong Zhang [3] pointed out that relying only on the information of the time series itself is not able to perform the task of forecasting complex time series data, and therefore proposed an ARMAD-GARCH model based on the differential information, which improves the ability of the forecasting model to discriminate the direction of the price evolution by integrating the information of the stock price change trend. With the development of deep learning technology, time series prediction models such as Long Short-Term Memory (LSTM) neural networks have made great breakthroughs in the performance of long term series prediction.

Specifically, Zhangyuan Zhou and Xiaoling He [4] learn the internal law of change of the input features based on the LSTM, and utilize the attention mechanism to compute the different weights of the hidden layer states of the LSTM, and then finally, combine the attention weights and LSTM neural networks to make exponential predictions.

However, deep learning itself does not have the ability to mine valid predictors and therefore does not have model interpretability. The second type is the predictor-based regression method. For example, linear regression model can estimate the linear relationship between predictors and stock price and quantify the importance of predictors to stock price. However, the problem of dimensionality catastrophe occurs when the number of predictors is too high. Some improved methods such as LASSO and ridge regression penalization techniques [5-7] can alleviate the model ill-conditioning problem caused by dimensionality catastrophe by restricting the space of the parameters to be evaluated, and also serve to quantify the importance of the factors. However, both ridge regression and LASSO are single generalized linear models, and the tuning parameters are very critical to the prediction performance, and improper settings may cause the prediction model to fall into overfitting or unfitting situations. The third method is based on the combination of feature selection and predictive modeling, for example, using feature selection methods such as Pearson's correlation coefficient, chi-square test, and mutual information criterion to select important features with significant correlation with the stock price [8-9], and then using the regression model to predict the stock price. This type of method has excellent performance when there is multicollinearity among features and the prediction ability of each factor on stock price has a big difference. For example, Yushao Chen [10], for the problem of unsatisfactory effect of a single selection method in feature selection, extracted highly relevant feature data sets and deleted minor features by constructing an integrated feature scorer, comprehensively evaluating the Pearson's correlation coefficient, rank correlation coefficient and other feature selection methods and inverting the validation of weighting factor, and then selecting important features with significant correlations with stock price. The high correlation features are extracted from the data set, and the secondary features are deleted. The improved Stacking algorithm, which integrates ridge regression, random forest and Xgboost, is then applied to predict the closing prices of the three stocks. However, when the predictive power of each factor on stock price is comparable, this type of method cannot completely retain all the effective information on stock price prediction. The fourth method is based on the combination of dimensionality reduction techniques and predictive modeling. These methods map high-dimensional data into a low-dimensional space by linearly or nonlinearly transforming the data and assigning interpretability to the downgraded features. Specifically, Xinrui Xie and others [11] proposed a dual dimensionality reduction method with joint mutual information and improved PCA considering the effectiveness of individual features on labeling and the information redundancy among multiple features, and showed the effectiveness of the proposed dimensionality reduction algorithm by predicting real stock data through neural networks. Xiaodan Zhao and Zhi Qi [12] proposed a nonlinear PCA method based on RBF neural network to address the shortcomings of the traditional PCA (Principal Component Analysis) stock price prediction method in the application of nonlinear processes, which enhanced the accuracy of stock price prediction. However, the current widely used dimensionality reduction methods, such as principal component analysis, are unsupervised dimensionality reduction methods and therefore cannot determine a significant relationship between the reduced factors and the stock price. Some other supervised dimensionality reduction methods, such as ICA, LDA and PLS, also cannot guarantee that the valid information between the original predictors and the stock price is completely retained.

Based on this, this paper proposes a stock factor quantification and prediction model based on the idea of sufficient dimensionality reduction and model averaging. The innovation of this method mainly lies in the following: on the one hand, based on the principle of conditional independence, while solving the problem of dimensional catastrophe, it theoretically retains the effective information of the original predictors on stock price to the greatest extent. On the other hand, the method adaptively learns the connection function between quantized factors and stock prices by applying the

idea of model averaging to weigh the variance and bias of the prediction model. In the empirical study, this paper chooses the classical weighted SIR-SAVE method as the adequate dimensionality reduction technique and adopts the mutual information criterion as the assignment rule for model averaging. The results show that the proposed method has higher prediction accuracy and robustness compared to the comparison methods.

2. Theory and Methods

2.1. Weighted SIR-SAVE method

The specific algorithm of Sliced Inverse Regression (SIR) [13] is shown as follows: assu me that there is the original dataset$(X_1, Y_1), (X_2,\ Y_2), \ldots, (X_n,\ Y_n)$, which is one-dimensional and is a p-dimensional vector. Calculate the sample mean and variance:$\hat{\mu} = E(X), \hat{\Sigma} = \text{var}\ (X)$ then standardize X:$Z_i = \hat{\Sigma}^{-1/2}(X_i - \hat{\mu}), i = 1,2, \cdots, n.$ Then, cut Y uniformly into h intervals and estimate $E(Z \mid Y \in J_l)$

$$E(\ Z \mid Y \in J_l\ ) = \frac{E\big(ZI(Y \in J_l)\big)}{E\big(I(Y \in J_l)\big)}, l = 1,2, \cdots, h \tag{1}$$

Estimate $var(E(Z \mid Y))$: $\hat{\Lambda} = \sum_{l=1}^{h} E(I(Y \in J_l))E(Z \mid Y \in J_l)E(Z^T \mid Y \in J_l)$ Taking the eigenvector $\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_d$ corresponding to the largest d eigenvalues of $\hat{\Lambda}$ as the eigenvector of the kernel matrixΛ, let$\beta_k = \hat{\Sigma}^{-1/2}\hat{v}_k, k = 1,2, \ldots, d$. Then the final dimensionality reduction result is:

$$\hat{\beta}_k^T(X_1 - \hat{\mu}), \hat{\beta}_k^T(X_n - \hat{\mu}) \cdots, k = 1,2, \cdots, d \tag{2}$$

The specific algorithm for SAVE (Sliced Average Variance Estimates) [14] is shown below: assume an original dataset$(X_1, Y_1), (X_2,\ Y_2), \ldots, (X_n,\ Y_n)$, where Y_i is a one-dimensional and X_i is a p-dimensional vector. Calculate the sample mean and variance:$\hat{\mu} = E(X), \hat{\Sigma} = \text{var}\ (X)$ Then standardize on X $Z_i = \hat{\Sigma}^{-1/2}(X_i - \hat{\mu}), i = 1,2, \cdots, n$, then cutting the uniform into intervals and enote by $J_l, l = 1,2,\cdots, h$ the h intervals. Afterwards discretize y:

$$\tilde{Y} = \sum_{l=1}^{h} lI(Y \in J_l) \tag{3}$$

For each sliceJ_l, calculate the conditional variance:

$$\text{var}\left(\ Z \,\middle|\, \tilde{Y} = l\ \right) = \frac{E\left(ZZ^T I\left(\tilde{Y} = l\right)\right)}{E\left(I\left(\tilde{Y} = l\right)\right)} \tag{4}$$

Compute a sample estimate of the kernel matrixΛ_{SAVE}:

$$\hat{\Lambda}_{SAVE} = h^{-1}\sum_{l=1}^{h} EI(\tilde{Y} = l)(I_p - \text{var}\ (Z \mid \tilde{Y} = l))^2 \tag{5}$$

Calculating the eigenvector$v_1, v_2, \ldots, v_d$, corresponding to the largest d eigenvalues of $\hat{\Lambda}_{SAVE}$ yields$\beta_i = \hat{\Sigma}^{-1/2}\hat{v}_i, i = 1,2, \ldots, d$. The final result of dimensionality reduction is then:

$$\hat{\beta}_k^T(X_1 - \hat{\mu}), \hat{\beta}_k^T(X_n - \hat{\mu}) \cdots, k = 1,2, \cdots, d \tag{6}$$

The weighted SIR-SAVE regression model is an improved version based on the weighting method and the SIR-SAVE regression model [15]. Weights are added in front of the covariance matrices of both SIR and SAVE to obtain a SIR-SAVE covariance matrix. The specific algorithm is shown below: Assume that there is an original data set $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$, where Y_i is a one-dimensional and X_i is a p-dimensional vector. Compute the sample mean and variance: $\hat{\mu} = E(X), \hat{\Sigma} = \text{var}(X)$ Then standardize on X: $Z_i = \hat{\Sigma}^{-1/2}(X_i - \hat{\mu}), i = 1,2, \cdots, n$ Construct the covariance matrices SIR and SAVE with weights: the weighted covariance matrix of SIR is:

$$Cov(SIR) = \sum_{i=1}^{n} W_i Z_i Z_i^T \tag{7}$$

Where W_i is the weight of the corresponding sample? The weighted covariance matrix of SAVE is:

$$Cov(SAVE) = \sum_{i=1}^{n} (1 - W_i)(Z_i - \hat{Z}_i)(Z_i - \hat{Z}_i)^T \tag{8}$$

Summing the weighted covariance matrices of SIR and SAVE gives the weighted covariance matrix of:

$$SIR_{SAVE} = \alpha SIR + (1 - \alpha) SAVE \tag{9}$$

Where α is the weight coefficient? Then, the eigenvalue decomposition of SIR_SAVE is performed to obtain the eigenvalues and the corresponding eigenvectors$SIR_SAVE = V \Lambda V^T$, where Λ is the eigenvalue diagonal matrix and V is the eigenvector matrix. Compute the eigenvector $v_1, v_2, \ldots, v_d$ corresponding to the largest d eigenvalues, afterwards we can get $\beta_i = \hat{\Sigma}^{-1/2} \hat{v}_i, i = 1,2, \ldots, d$.Then the final result of dimensionality reduction is:

$$\hat{\beta}_k^T(X_1 - \hat{\mu}), \hat{\beta}_k^T(X_n - \hat{\mu}) \cdots, k = 1,2, \cdots, d \tag{10}$$

2.2. Bagging Algorithm Based on Mutual Information Weighting

Mutual Information is a metric used to measure the correlation between two random variables [16]. It quantifies how much the uncertainty of one random variable is reduced by knowing the information from another random variable.

Mutual Information can be calculated using entropy, and its formula is as follows:

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \tag{11}$$

Where$H(X) = -\Sigma\, p(x) log(p(x))andH(X,Y) = -\Sigma\, \Sigma\, p(x,y) log(p(x,y))$represent the entropy of variables X and Y respectively. $H(X,Y)$ Represents the joint distribution entropy of X and Y. $p(x)$ represents the probability of variable X taking value x, and $p(x,y)$ represents the joint probability of variable X taking value x and variable Y taking value y. By calculating the individual entropies of X and Y and the joint entropy, we can obtain the mutual information. The principle of mutual information is to compare the sizes of the entropies of X and Y and the joint entropy to measure their correlation. If X and Y are independent, the joint entropy is equal to the sum of the individual entropies, and the mutual information is 0. If X and Y are correlated, the joint entropy is smaller than the sum of the individual entropies, and the mutual information is a positive value. Bagging (Bootstrap aggregating) is an ensemble learning method [17] that improves model performance by training multiple base learners in parallel and combining their prediction results. The principle is as follows: First, for the sample set:

$$D = \{(X1, Y1), (X2, Y2), \ldots, (Xm, Ym)\} \quad (12)$$

Choose a weak learning algorithm and the number of iterations T for the weak classifiers. For t = 1, 2, ..., T: Randomly sample the training set for the t-th time, repeated m times, to obtain a sampling set D_tconsisting of m samples. Train the t-th weak learner $G_t(x)$.using the sampling setD_t. If it is a classification algorithm, the final class is determined by the class with the highest number of votes among the T weak learners or one of the classes. If it is a regression algorithm, the final model output is the arithmetic average of the regression results obtained by the T weak learners. The advantage of the Bagging algorithm is that by combining multiple different base learners, it can reduce the variance of the model and improve its generalization ability. Moreover, since the base learners are trained in parallel, the algorithm efficiency can be improved through parallel computation. In the process of integrating prediction results, the weighting of the prediction results of the base learners can be further optimized based on different weights. Similar to the traditional Bagging algorithm, new training sets are created by randomly selecting several samples from the original training set using a sampling with replacement method. At the same time, a weight is assigned to each base model, which can be manually set based on the importance of the samples or automatically calculated based on the distribution of the samples.

2.3. Weighted SIR-SAVE and Weighted Bagging-Based Stock Price Prediction Model

Next, we will provide a detailed description of the proposed method, which utilizes sufficient dimensionality reduction techniques to alleviate the curse of dimensionality while retaining the effective information for the prediction task. Additionally, through the weighted bagging algorithm, the model can adaptively fit the linking function between the prediction factors and the stock price. The specific algorithm is as follows:

1. Divide the data using a rolling window method to create training and testing sets, and further partition the training set into a validation set.
2. Apply the weighted SIR-SAVE method to reduce the dimensionality of the training set,

Generating projection matrices and the reduced-dimensional data. The coefficient parameters can be optimized using cross-validation on the validation set.

3. Utilize the reduced-dimensional training set data in the weighted bagging algorithm to obtain the parameters of each model. The weights of the base models are determined based on the validation set within the training set.
4. Multiply the testing set data by the projection matrices and input it into the trained weighted bagging algorithm to obtain the predicted results on the testing set.
5. Evaluate the effectiveness of the model by comparing the predicted results with the ground truth values using metrics such as mean squared error, absolute error, and posterior error.

After the training of the prediction model, if each base model is an interpretable machine learning method, the relative importance of each quantification factor for the prediction task will be outputted.

3. The data analysis

3.1. Data Source and Experimental Setup

The data used in this study is sourced from the Tushare website (https://tushare.pro/). We collected twelve groups of stock data from January 1, 2023, to May 26, 2023. Specifically, Baiyun Airport, Shanghai Airport, and Dongfeng Motor belong to the service industry; Huaxia Bank and Minsheng Bank are from the financial industry; Wuhu Expressway, Zhongyuan Expressway, and Shanghai Port Group are in the transportation industry; Rizhao Port, Baosteel Group, and Baoshan Iron & Steel are in the manufacturing industry; and Shouchuang Environmental Protection is in the water production and supply industry. As stocks in different industries exhibit different volatility trends (e.g., stocks in the industrial and manufacturing sectors tend to have relatively stable volatility, while stocks in the financial and insurance sectors have more complex volatility patterns), we aim to explore whether the

proposed method maintains its superiority over other comparative methods under different volatility trends. Table 1 presents the prediction factors used in this experiment, which are categorized into two trading volume factors, adjustment factors, technical indicator factors, and moving average factors.

Table 1. Stock Price Quantification Factor Names.

Factor Number	factor name	Factor Type	Factor Description	Discrete or continuous
1	open	Trading Volume Factor	Opening price	Discrete
2	close	Trading Volume Factor	Closing price	Discrete
3	high	Trading Volume Factor	Highest price	Discrete
4	low	Trading Volume Factor	Lowest price	Discrete
5	pre_close	Trading Volume Factor	Yesterday's closing prise	Discrete
6	change	Trading Volume Factor	Rise or fall in price	Discrete
7	pct_change	Trading Volume Factor	Rise or fall in price	Discrete
8	vol	Trading Volume Factor	Average exchange rate	Discrete
9	amount	Trading Volume Factor	turnover	Discrete
10	adj_factor	Reweighting factor	Reweighting factor	Discrete
11	open_hfq	Reweighting factor	Opening price after reweighting	Discrete
12	open_qfq	Reweighting factor	Opening price before reweighting	Discrete
13	close_hfq	Reweighting factor	closing price after reweighting	Discrete
14	close_qfq	Reweighting factor	closing price before reweighting	Discrete
15	high_hfq	Reweighting factor	Adjusted High Price	Discrete
16	high_qfq	Reweighting factor	Adjusted Pre-split High Price	Discrete
17	low_hfq	Reweighting factor	Adjusted Low Price	Discrete
18	low_qfq	Reweighting factor	Adjusted Pre-split Low Price	Discrete
19	pre_close_hfq	Reweighting factor	Adjusted Previous Close Price	Discrete
20	pre_close_qfq	Reweighting factor	Adjusted Pre-split Previous Close Price	Discrete
21	macd_dif	Moving average factor	Moving Average (DIF)	Continuous
22	macd_dea	Moving average factor	Moving Average (DEA)	Continuous
23	macd	Moving average factor	Moving Average	Discrete
24	kdj_k	Technical indicator factor	Fast Stochastic K Line	Discrete
25	kdj_d	Technical indicator factor	Fast Stochastic D Line	Discrete
26	kdj_j	Technical indicator factor	Slow Stochastic J Line	Discrete
27	rsi_6	Technical indicator factor	6-day Relative Strength Index (RSI)	Continuous
28	rsi_12	Technical indicator factor	12-day Relative Strength Index (RSI)	Continuous
29	rsi_24	Technical indicator factor	24-day Relative Strength Index (RSI)	Continuous
30	boll_upper	Technical indicator factor	Bollinger Bands upper	Continuous
31	boll_mid	Technical indicator factor	Bollinger Bands mid	Continuous
32	boll_lower	Technical indicator factor	Bollinger Bands lower	Continuous
33	cci	Technical indicator factor	Commodity Channel Index	Continuous

3.2. Experiment Setup

Regarding the specific experimental setup, we defined the size and categories of the dataset. Considering that an excessive amount of raw data can lead to an abundance of noisy data in the model, resulting in poor prediction performance, we set the data period to around 150 trading days to achieve better prediction results. The original data includes basic data, adjustment factor data, MACD curve data, KDJ indicator data, risk factor data, and Bollinger Bands indicator data. All the data items were not preprocessed.

The experiment first utilized the weighted SIR-SAVE model to process the dataset, compressing the 32 different factors into six major categories of data with different characteristics. Then, the weighted Bagging algorithm was used to adaptively learn the linking function between the quantized factors and stock prices, resulting in predicted data.

For the grouping method, we employed a rolling window approach to divide the dataset into training and testing sets. The regression prediction models used as comparisons in the experiment include multiple linear regression, LASSO regression, ridge regression, decision tree regression, random forest regression, support vector machine regression, and gradient boosting decision tree (GBDT) regression. The evaluation metrics used were mean squared error (MSE) and absolute error (SRE), calculated as follows:

$$SME = \sum_{i=1}^{n} \frac{1}{n} (\hat{y}_i - y_i)^2 \tag{13}$$

$$SRE = \sum_{i=1}^{n} \frac{1}{n} |\hat{y}_i - y_i| \tag{14}$$

The software used for this experiment was Python, and the editor used was Jupyter Notebook. The versions used were Jupyter Notebook 6.4.12 and Python 3.10.3.

3.3. Comparative Results

In the experimental process, all hyperparameters were determined using generalized cross-validation methods. In this section, due to space limitations, we only present the comparative results of four stocks. These stocks come from different industries and exhibit varying degrees of volatility, which can demonstrate the superiority of the proposed method over other comparative methods. Specifically, Table 1 to Table 4 show the mean and standard deviation of the experimental results, as well as the mean squared error, absolute error, and relative error for multiple linear regression, LASSO regression, ridge regression, decision tree regression, random forest regression, support vector machine regression, GBDT regression, and the proposed method. A smaller mean value of MSE, absolute error, and relative error indicates higher prediction accuracy of the model, while a smaller standard deviation indicates a certain level of robustness in the prediction model. From the tables, it is evident that the proposed method outperforms the other comparative methods in terms of evaluation metrics, and it also has the smallest standard deviation, indicating higher robustness compared to other comparative methods.

Figure 1 displays the distribution of different experimental results for the four stocks under each experimental method. From the box plots, the robustness of the proposed method can be clearly observed.

Figure1. The comparison of the predictions of stock 600006.

Methods	SRE_mean	SME_mean	SRE_std	SME_std
PCA_LR	0.8397	0.8244	0.0242	0.0608
PCA_LASSO	0.8397	0.8244	0.0242	0.0608
PCA_Tree	0.7354	0.6588	0.0334	0.0997
PCA_RF	0.7035	0.6162	0.0388	0.1095
PCA_svm	0.9192	0.9415	0.0346	0.0652
PCA_GBDT	0.6922	0.5813	0.0300	0.0802
FSAVE-IMB	0.5982	0.4205	0.0335	0.0635

There may be several reasons why the proposed method is significantly better than other comparative methods: Firstly, by utilizing supervised dimensionality reduction techniques, the method effectively addresses the curse of dimensionality while extracting relevant information from predictive factors for stock prices. Moreover, in the process of dimensionality reduction, the connection function between the predictive factors and corresponding variables is flexible. This allows for the second-stage predictive model to be arbitrary and does not conflict with the dimensionality reduction process. Secondly, during the prediction stage, the method incorporates ensemble learning techniques based on the idea of model averaging. This approach combines multiple sub-models to balance the variance and bias of the predictive model. The importance of each model is adaptively determined on the validation set, significantly enhancing the generalization ability of the predictive model. On the other hand, multiple linear regression, LASSO regression, ridge regression, support vector machine regression, and decision tree regression are all single models. As a result, they are susceptible to issues such as overfitting or underfitting. Although random forest shares similar advantages with the proposed method, it assumes that each sub-model is a random forest with equal weights, limiting its adaptability.

Figure2. The comparison of the predictions of stock 600008.

Methods	SRE_mean	SME_mean	SRE_std	SME_std
PCA_LR	0.8397	0.8244	0.0242	0.0608
PCA_LASSO	0.8397	0.8244	0.0242	0.0608
PCA_Tree	0.7354	0.6588	0.0334	0.0997
PCA_RF	0.7035	0.6162	0.0388	0.1095
PCA_svm	0.9192	0.9415	0.0346	0.0652
PCA_GBDT	0.6922	0.5813	0.0300	0.0802
FSAVE-IMB	**0.5982**	**0.4205**	**0.0335**	**0.0635**

Figure3. The comparison of the predictions of stock 600009.

	SRE_mean	SME_mean	SRE_std	SME_std
PCA_LR	0.5642	0.3553	0.0120	0.0145
PCA_LASSO	0.5642	0.3553	0.0120	0.0145
PCA_Tree	0.6015	0.4093	0.0124	0.0157
PCA_RF	0.5813	0.3783	0.0130	0.0164
PCA_svm	0.5582	0.3431	0.0125	0.0164
PCA_GBDT	0.5794	0.3764	0.0127	0.0167
FSAVE-IMB	**0.4495**	**0.2336**	**0.0159**	**0.0166**

Figure4. The comparison of the predictions of stock 600010.

	SRE_mean	SME_mean	SRE_std	SME_std
PCA_LR	0.5642	0.3553	0.0120	0.0145
PCA_LASSO	0.5642	0.3553	0.0120	0.0145
PCA_Tree	0.6015	0.4093	0.0124	0.0157
PCA_RF	0.5813	0.3783	0.0130	0.0164
PCA_svm	0.5582	0.3431	0.0125	0.0164
PCA_GBDT	0.5794	0.3764	0.0127	0.0167
FSAVE-IMB	**0.4495**	**0.2336**	**0.0159**	**0.0166**

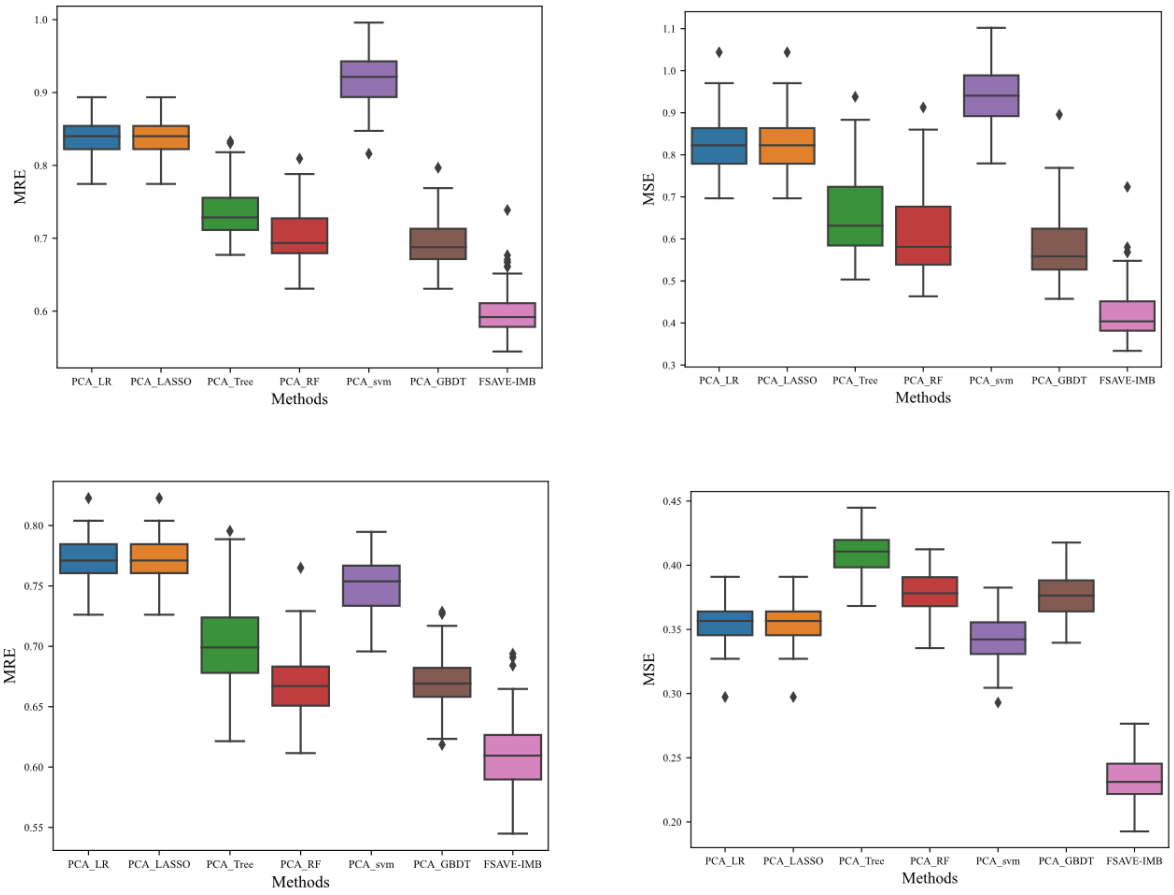

Figure 5. Forecast results of four stocks.

4. Conclusions and Outlook

In this paper, we proposed a stock factor quantification and prediction model based on sufficient dimensionality reduction and the idea of model averaging. The proposed method addresses the high-dimensional regression problem and achieves the quantification of important predictive factors such as technical factors. Based on the principle of conditional independence, the method maximally preserves the effective information of the original predictive factors for stock prices. Additionally, by employing the idea of model averaging, it adaptively balances the variance and bias of the predictive model. Empirical results demonstrate that the proposed method exhibits higher accuracy in terms of evaluation metrics such as mean square error and absolute error, and it possesses a certain degree of robustness. This indicates that the method has great potential for applications and can provide an efficient and accurate solution for practical problems like stock prediction.

Regarding future research directions, on the one hand, further investigation into the idea of model averaging can be conducted to develop more effective weighting rules, thereby enhancing the robustness and accuracy of the model. On the other hand, the proposed method can be extended to a broader range of financial prediction problems, such as exchange rate prediction and futures prediction, to further validate the practicality and applicability of the method. Furthermore, considering that this paper only compared the proposed method with classical high-dimensional regression techniques, future research will explore more diverse high-dimensional regression methods and sufficient dimensionality reduction techniques to further improve the effectiveness of the method.

In summary, the new model and method proposed in this study for stock prediction provide new ideas and directions for future research in financial quantification.

References

[1] Liu, S., Zhang, S. Empirical Research on Stock Price Forecasting Using ARIMA Model. Economic Research Guide, 2021, No. 483(25), 76-78.

[2] Yu, Z., Yang, S. GARCH Stock Price Forecasting Model Based on Error Correction. Chinese Journal of Management Science, 2013, 21(S1), 341-345.

[3] Zhang, G., Zhang, X. ARMAD-GARCH Stock Price Forecasting Model Based on Differential Information. Systems Engineering-Theory & Practice, 2016, 36(05), 1136-1145.

[4] Zhou, Z., He, X. Stock Price Forecasting Method Based on Optimized LSTM Model. Statistics and Decision, 2023, 39(06), 143-148.

[5] Su, D. Stock Price Forecasting Based on Lasso Method and LSTM. (Master's thesis). North China Electric Power University, 2022.

[6] Wang, P. Stock Price Analysis and Forecasting Based on Multivariate Linear Regression. Science & Technology Markets, 2020(01), 84-85.

[7] Wang, W., Cai, W. Research on Probability Prediction of Stock Price Increase Based on Logistic Regression. China Market, 2020, No. 1033(06), 7-8.

[8] Li, Z., Du, J., Nie, B., et al. Overview of Feature Selection Methods. Computer Engineering and Applications, 2019, 55(24), 10-19.

[9] Yao, X., Wang, X., Zhang, Y., et al. Overview of Feature Selection Methods. Control and Decision, 2012, 27(02), 161-166+192.

[10] Chen, Y. Stock Price Forecasting Research Based on Feature Selection and Improved Stacking Algorithm. (Master's thesis). University of South China, 2018.

[11] Xie, X., Lei, X., Zhao, Y. Application of MI and Improved PCA Dimensionality Reduction Algorithms in Stock Price Forecasting. Computer Engineering and Applications, 2020, 56(21), 139-144.

[12] Zhao, X., Qi, Z. Application Research of Nonlinear PCA Method in Stock Price Forecasting. Journal of Jilin Normal University (Natural Science Edition), 2008, 29(04), 70-73.

[13] VanderWeele, T.J. (2009). Marginal structural models for the estimation of direct and indirect effects. Epidemiology, 20(1), 18-26.

[14] Feng, X., Liu, X., & Zhang, J. (2018). A new method for sensitivity analysis in observational studies. Journal of the American Statistical Association, 113(522), 1469-1480.

[15] Wang, Y., & Wang, S. (2019). Weighted regression models for analyzing clustered data: An application to clustered binary data. Biometrical Journal, 61(3), 541-558.

[16] Cover, T. M., & Thomas, J. A. (2006). Elements of information theory. John Wiley & Sons.

[17] Zhou, Z. H. (2012). Ensemble methods: Foundations and algorithms. CRC Press.