Proceedings of ICFTBA 2025 Symposium: Strategic Human Capital Management in the Era of AI
DOI: 10.54254/2754-1169/2025.LD26613

Prediction and Analysis of the Performance of CSI 300 Index Constituent Stocks Based on Integrated Algorithms

Rui Chang

Southwest University of Finance and Economics, Chengdu, China
42201003@smail.swufe.edu.cn

Abstract. This paper aims to explore the effectiveness of stock price prediction models based on ensemble algorithms by predicting the trends of the CSI 300 Index components. The study utilizes 20 feature variables from 242 trading days of the CSI 300 components in 2024, totaling 72,504 data points. Regression interpolation, DBSCAN and wavelet thresholding methods are employed to enhance data stability and quality. During the model construction phase, this study transforms the stock price prediction problem into a binary classification problem, using the direction of price changes in the denoised closing prices as classification labels. Two ensemble learning algorithms—Random Forest and XGBoost—are employed for modeling. Random Forest achieves predictions by constructing multiple decision tree models, with parameters set to n_estimators=500 and max_features=4. XGBoost further optimizes the boosting tree model and determines the optimal parameters through grid search. Both models use a 2:1 ratio to divide the training set and validation set, respectively, for model training and evaluation. XGBoost achieves a prediction accuracy of 55.76%, slightly higher than Random Forest's 55.46%.

Keywords: Random Forest Algorithm, XGBoost Algorithm, Stock Prediction, Spearman Correlation Coefficient, Clustering Algorithm

1. Introduction

Stock price prediction remains a significant challenge in financial research due to market volatility and complex influencing factors. As China's A-share market continues to evolve, the CSI 300 Index—comprising the most representative large-cap stocks—serves as a critical barometer of economic health. Traditional prediction models often struggle with high-dimensional data and noise inherent in emerging markets. To address these limitations, this study leverages ensemble learning algorithms (Random Forest and XGBoost) for directional forecasting of CSI 300 constituent stocks. By transforming numerical price prediction into a binary classification problem and incorporating rigorous data preprocessing—including wavelet denoising and DBSCAN-based outlier handling—we aim to enhance prediction robustness. Our work contributes to the application of integrated machine learning frameworks in quantitative finance, offering methodological insights for stabilizing investment strategies in dynamic markets.

DOI: 10.54254/2754-1169/2025.LD26613

2. Data preprocessing

2.1. Variable selection

The CSI 300 Index is an important indicator reflecting the overall performance of China's A-share market. The selection of CSI 300 component stocks as the primary focus for stock price forecasting is based on the following reasons. First, CSI 300 component stocks encompass the most representative large and medium-sized enterprises in China's A-share market, with a broad industry distribution spanning sectors such as finance, consumer goods, technology, and energy, thereby demonstrating strong representativeness and reflecting the overall operational status of China's economy. Second, CSI 300 component stocks account for approximately 60% of the market capitalization of the A-share market, with good liquidity and large enterprise scale. Third, compared to small- and medium-cap stocks, CSI 300 component stocks exhibit more stable price fluctuations, better reflecting the intrinsic patterns of the market. Fourth, as blue-chip stocks with high market attention, CSI 300 component stocks have higher transparency in information disclosure, with more reliable financial data and market transaction data.

Current research often selects relevant quantitative indicators and basic financial data as variables. Therefore, this paper considers data from the following four aspects: individual stock transaction data, individual stock trend variable data, individual stock transaction derivative indicators, and individual stock financial derivative indicators. Among them, individual stock transaction data includes daily closing price, daily individual stock transaction volume, daily individual stock total market capitalization, and daily individual stock return rate; individual stock trend variable data includes price changes, consecutive days of increases, consecutive days of decreases, weekly high price, and weekly low price; individual stock transaction-derived indicators include price-to-earnings ratio, price-to-book ratio, turnover rate, and liquidity indicators; individual stock financial-derived indicators include earnings per share, net asset value per share, stock value score, net asset growth rate, main business revenue growth rate, operating cash flow growth rate, and stock growth score. This paper obtained 20 variables for the 300 component stocks of the CSI 300 Index over 242 trading days in 2024 from the CSMAR database, totaling 72,504 data points.

2.2. Data cleansing

This paper uses regression interpolation to fill in missing values and clustering algorithms to handle outliers. Since DBSCAN clusters data points based on their density, it can capture clusters with complex shapes and identify outliers well. Furthermore, it does not require the number of classes to be specified before clustering. Therefore, this paper selects the DBSCAN clustering algorithm here.

The steps for handling outliers are as follows:

Step 1. Loop through the constituent stocks.

Step 2. Use the DBSCAN clustering algorithm to identify outliers, with parameters set to default values.

Step 3. Replace the outliers for each stock with its mean value.

The following are descriptive statistics before and after data cleaning.

DOI: 10.54254/2754-1169/2025.LD26613

Table 1. Descriptive statistics (1)

	price-earnings ratio		price-to-book ratio		turnover rate	
clean	No	Yes	No	Yes	No	Yes
count	69368	72504	72504	72504	72504	72504
mean	30.502170	38.403492	4.2357439	3.5495612	0.0115733	0.0115733
std	40.405770	3.441	25.636882	14.652745	0.0157590	0.0157590
50%	19.169367	30.981890	1.8735534	1.8735534	0.00683	0.00683

Table 2. Descriptive statistics (2)

	liquidity indicators		price fluctuation		continued rise days	
clean	No	Yes	No	Yes	No	Yes
count	72504	72504	72504	72504	72504	72504
mean	3.488E-05	3.49E-05	0.0719825	-0.015374	0.8916887	0.7865703
std	3.544E-05	3.54E-05	2.5762796	1.9721910	1.3083355	1.0525327
50%	2.5E-05	2.50E-05	-0.055	-0.07	0	0

Table 3. Descriptive statistics (3)

	continued fall days		life high week		life low week	
clean	No	Yes	No	Yes	No	Yes
count	72504	72504	72504	72504	72504	72504
mean	0.9856697	0.8949230	44.878570	44.882487	42.716356	42.311664
std	1.3577292	1.1443739	106.02099	105.37564	102.24862	100.06583
50%	1	0.75	20.23	20.23	19.280001	19.280001

Table 4. Descriptive statistics (4)

	daily closing price		dnshrtrd		dsmvtll	
clean	No	Yes	No	Yes	No	Yes
count	72504	72504	72504	72504	72504	72504
mean	43.74118	42.39060	55417127	55417127	138303695	138303695
std	104.0225	99.11905	95279482	95279482	216344573	216344573
50%	19.70999	19.70000	26197198	26197198	77880616	77880616

DOI: 10.54254/2754-1169/2025.LD26613

Table 5. Descriptive statistics (5)

	dretwd		earnings per share		net asset value	
clean	No	Yes	No	Yes	No	Yes
count	72504	72504	72504	72504	72504	72504
mean	0.0008828	0.0008864	2.1488628	2.1488628	18.507983	18.507983
std	0.0253700	0.0253439	4.5109717	4.5109717	56.931989	56.931989
50%	-0.000429	-0.000427	1.14256	1.14256	11.858628	11.858628

Table 6. Descriptive statistics (6)

	stock value score		net asset growth rate		revenue growth	
clean	No	Yes	No	Yes	No	Yes
count	72504	72504	72504	72504	72504	72504
mean	0.3033580	0.3033689	0.2209209	0.2209209	0.2071781	0.2071781
std	0.4067671	0.4068692	0.2321439	0.2321439	0.2619683	0.2619683
50%	0.1861915	0.1861915	0.149358	0.149358	0.146616	0.146616

Table 7. Descriptive statistics (7)

	stock value score		net asset growth rate	
clean	No	Yes	No	Yes
count	72504	72504	72504	72504
mean	0.3033580	0.3033689	0.2209209	0.2209209
std	0.4067671	0.4068692	0.2321439	0.2321439
50%	0.1861915	0.1861915	0.149358	0.149358

2.3. Spearman correlation analysis

Pearson correlation coefficients, covariance coefficients, and Spearman correlation coefficients can be selected to determine the correlation between 20 variables. Since the data is not normally distributed, it does not meet the conditions for using Pearson coefficients. At the same time, due to the influence of excessive dimensions, the explanatory power of covariance coefficients is poor. The Spearman correlation coefficient is based on the relative position values obtained after ranking each variable, so this paper selects the Spearman correlation coefficient for correlation analysis. This paper plots a heat map based on the correlation coefficients, and the complete correlation matrix is shown in the appendix.

DOI: 10.54254/2754-1169/2025.LD26613

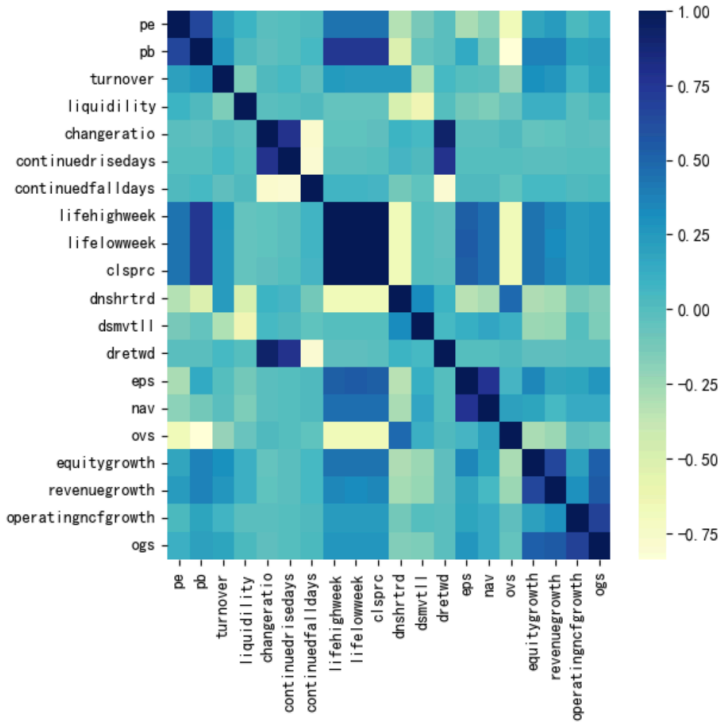

Figure 1. Spearman correlation coefficient heat map

We found that the correlation coefficients between clsprc, lifehighweek, and lifelowweek were extremely high, at 0.9986683961192708, 0.9988503046056649, and 0.998834984048056, respectively. As a result, we excluded lifehighweek and lifelowweek from the analysis.

2.4. Noise reduction of closing prices based on wavelet thresholds

Considering that China's A-share market is still in its developmental stage and the market is not fully efficient, stock price data may contain noise. Wavelet transforms have the ability to process non-stationary financial time series [1]. Therefore, this paper performs wavelet threshold denoising on stock price forecasts, i.e., “tomorrow's closing prices,” based on different stock groupings. Wavelets are waveforms with a mean of zero, characterized by limited energy and the properties of being ‘small’ and “volatile.”

The steps for wavelet threshold denoising are as follows:

Step 1: Wavelet decomposition breaks down the signal into multiple wavelet coefficients with different frequencies and time resolutions. Wavelet coefficients contain important time-frequency information; the wavelet coefficients of the true signal are larger, while those of the noise are smaller.

Step 2: Threshold processing retains the wavelet coefficients of the true signal.

Step 3: Reconstruction is performed to obtain the denoised signal.

Here, the Danbychey4 wavelet, which is commonly used in financial data processing, is selected to perform a two-level scale decomposition on the stock prices of the constituent stocks. The price trends of the constituent stocks of the CSI 300 Index before and after noise filtering (with outliers already processed) are shown in Figures 2 and 3, respectively. Taking Ping An Bank as an example, the price trends of individual stocks before and after noise filtering are shown in Figure 4.

DOI: 10.54254/2754-1169/2025.LD26613

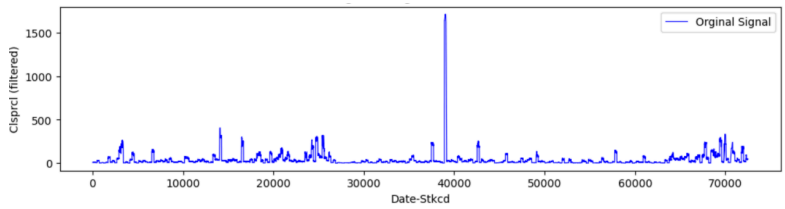

Figure 2. CSI 300 stocks price trends before noise filtering

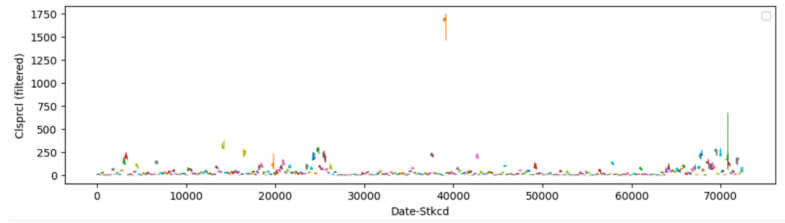

Figure 3. CSI 300 stocks price trends after noise filtering

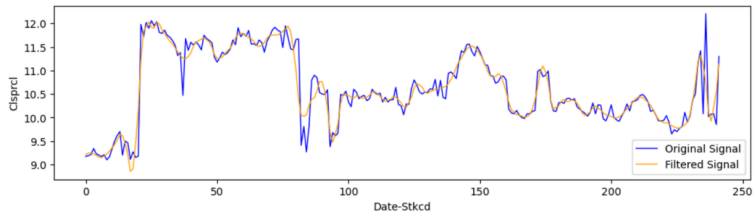

Figure 4. Ping An Bank share price trends before and after noise filtering

3. Establish a model for predicting the performance of constituent stocks

Predicting the exact numerical value of stock prices is nearly an impossible task. However, if one can determine the direction of stock price changes with a high degree of probability and trade based on such signals, it is sufficient to generate profits. Therefore, this paper treats stock closing price prediction as a classification problem. The response factor is the direction of the closing price of the component stocks after denoising over the next trading day. When the denoised closing price rises over the next trading day, it is recorded as 1; and -1 when they fall. The 18 feature variables are used as input factors for predicting the dependent variable of price movement. By defining the dependent variable, identifying the direction of stock closing price movement is transformed into a classic pattern recognition problem in machine learning. The price movement problem is studied as a binary classification problem, estimating the probability of the dependent variable belonging to different categories through input variables and a series of algorithms.

In classification problems, ensemble algorithms build n different decision tree models and obtain the final result based on the voting results of the n decision tree models. Therefore, the probability of the daily price movement of each component stock over the next 1 day, as calculated using ensemble algorithms, can be expressed as:

$$R_{ij} = \arg\max_{y\in\{1,0,-1\}} \sum_{b=1}^{B} \mathbb{I}(y = R_{ij}^{*b}) \tag{1}$$

Among them, Rij is the price change direction of component stock i on trading day j, I(·) is demonstration purposes, R*b is prediction results for the direction of price movement of component stock i on trading day j for decision tree b.

DOI: 10.54254/2754-1169/2025.LD26613

Machine learning can more effectively explore the correlations between variables, thereby making more accurate predictions about events [2,3]. This paper selects the random forest algorithm and XGBoost algorithm to predict the trends of the CSI 300 index components.

4. Prediction of constituent stock trends based on the random forest algorithm

4.1. Random forest algorithm

Random Forest is a classic Bagging model. It randomly samples n different sample datasets from the original dataset and builds n different decision tree models based on their data characteristics [4,5].The first three layers of a single decision tree used in the paper are shown in the figure below.

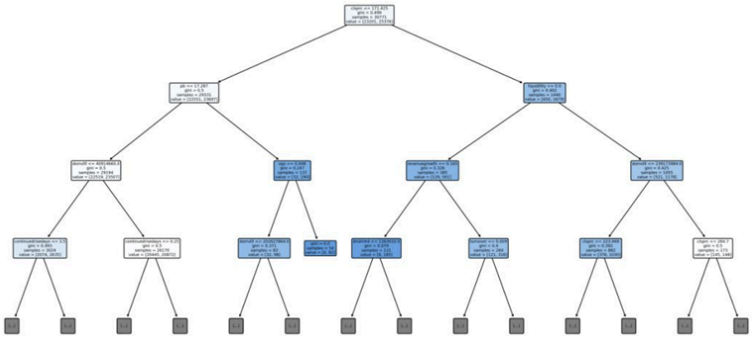

Figure 5. The first decision tree (first three layers) of random forest

In the figure, each node reports four pieces of information: the first row shows the split variable and conditions; the second row shows gini, which is an impurity index used to measure the degree of sample category mixing in the node. The lower the value, the purer the samples in the node, i.e., the more samples of the same category there are; the third row shows samples, which is the sample size; and the fourth row shows value, which is the number of classified samples.

4.2. Algorithm solution

This paper randomly divides the data into training and validation sets at a ratio of 2:1. The actual number of samples in the training set is 48,577, and the number of samples in the validation set is 23,927. This paper selects 123 random seeds.

Additionally, regarding parameter optimization, according to relevant theory, n_estimators is generally greater than or equal to 500, and max_features yields the best results when set to the square root of the number of features. Therefore, this paper sets n_estimator=500 and max_features=4.

After running the code, the prediction results were obtained. The prediction accuracy rate was calculated based on whether the prediction results were correct. The prediction accuracy rate was 0.5546035859071342. The following table further categorizes the prediction results by price movement.

DOI: 10.54254/2754-1169/2025.LD26613

Table 8. Forecasts for different price movements

	predict increase	predict decline
total number	12639	11288
correct number	7388	5882

4.3. Importance ranking

The more frequently a feature is used for splitting in a tree and the more significantly it reduces impurity, the higher its importance. By calculating the average reduction in Gini impurity contributed by each feature across all trees, we can determine the contribution of each feature to the algorithm. Sorting the contribution levels yields an importance ranking for the feature variables, as shown in the table below:

Table 9. Random forest algorithm feature importance ranking

Feature	Importance	Feature	Importance
turnover	0.089936642297	liquidility	0.07450485885059
dnshrtrd	0.089104350127	eps	0.01879284227935
dretwd	0.089009833790	stkcd	0.01859169122410
clsprc	0.088289552350	nav	0.01836762363115
changeratio	0.086624926558	equitygrowth	0.01807878226587
pb	0.080978869492	operatingncfgrowth	0.01805792273989
dsmvtll	0.080880186527	continuedfalldays	0.01793961279699
ovs	0.079665675036	revenuegrowth	0.01761760605874
pe	0.079601743058	continuedrisedays	0.01729097490277

5. Prediction of constituent stock trends based on the XGBoost algorithm

5.1. XGBoost algorithm

EXtreme Gradient Boosting (XGBoost) is an ensemble algorithm based on gradient boosting. To understand the XGBoost algorithm, this paper first introduces the GBDT algorithm, which uses the negative gradient of the loss function (first-order Taylor expansion) as an approximation of the residuals, and iteratively fits regression trees using these residuals. In other words, each new decision tree is constructed to fit the residuals of the previous trees. Building on this foundation, XGBoost introduces optimizations. In the loss function component, XGBoost performs a second-order Taylor expansion of the residuals and incorporates regularization to prevent overfitting [6,7]. Additionally, to enhance computational efficiency, XGBoost employs parallel selection.

5.2. Algorithm solution and parameter optimization

In this section, the same processing as the random forest algorithm is used. The data is randomly divided into a training set and a validation set at a ratio of 2:1. The actual number of samples in the training set is 48,577, and the number of samples in the validation set is 23,927. A random seed of 123 is selected in this paper.

DOI: 10.54254/2754-1169/2025.LD26613

Since XGBoost has better runtime efficiency, parameter tuning was performed in this paper. The optimal parameters were determined through grid search with three-fold cross-validation, with the final selection being eta=0.2 and max-depth=6.

After running the code, the prediction results were obtained, and the prediction accuracy was calculated based on whether the predictions were correct. The prediction accuracy was 0.5575709449575793. The following table further categorizes the prediction results by price movement:

Table 10. Forecasts for different price movements

	predict increase	predict decline
total number	12580	11347
correct number	7649	5692

6. Conclusion

Table 11. Accuracy comparison

	Random forest	XGBoost
accuracy	0.5546035859071342	0.5575709449575793

From the results, the accuracy of the XGBoost algorithm (0.5576) is slightly higher than that of the random forest algorithm (0.5546). This indicates that XGBoost has a slight advantage when processing this dataset. The XGBoost algorithm uses a gradient boosting mechanism to more effectively reduce overfitting and performs well in feature selection and optimization, thereby achieving slightly higher prediction accuracy in such complex data. In contrast, while the Random Forest algorithm is also a powerful ensemble learning algorithm, its decision trees are highly independent, which may prevent it from achieving better performance through global optimization in certain scenarios, unlike XGBoost. Additionally, the relatively small difference in accuracy improvement suggests that the advantages of the XGBoost algorithm may not have been fully realized.

In this study, the model's prediction results validated the application potential of ensemble learning algorithms in stock market data analysis. Although there is still room for improvement in prediction accuracy, further optimization of the algorithm and the addition of more feature data are expected to further enhance prediction accuracy.

In summary, XGBoost and Random Forest algorithms provide effective solutions for predicting the trends of CSI 300 component stocks and offer valuable references for machine learning-based stock market prediction models.

References

[1] He, Y., & Li, H. (2023). Improved NSGA-III-XGBoost algorithm for stock prediction .Computer Engineering and Applications, 59(18), 293–300. (in Chinese)

[2] Liu, J. J., Zheng, C. R., & Hong, Y. M. (2023). How machine learning empowers management research? — Review of domestic and international frontiers and future prospects .Management World, 39(9), 191–216. (in Chinese)

[3] Zhou, W. H., Zhai, X. F., & Tan, H. W. (2022). Research on financial fraud prediction model of listed companies based on XGBoost .The Journal of Quantitative & Technical Economics, 39(7), 176–196. (in Chinese)

DOI: 10.54254/2754-1169/2025.LD26613

[4] Wang, Y. S., & Xia, S. T. (2018). Review of random forest algorithm in ensemble learning .Information and Communications Technologies, 12(1), 49–55. (in Chinese)
[5] Liu, G., & Yi, H. (2024). Deep learning prediction for stock market integrating media information and signal decomposition .Computer Science, 51(S1), 1104–1115. (in Chinese)
[6] Imani, M., Beikmohammadi, A., & Arabnia, H. R. (2025). Comprehensive analysis of random forest and XGBoost performance with SMOTE, ADASYN, and GNUS under varying imbalance levels.Technologies, 13(3), 88.
[7] Chen, T., & Guestrin, C. (2016, August). XGBoost: A scalable tree boosting system. InProceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(pp. 785–794).