Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Combinatorial Press

# Optimization of Random Forest Algorithm and Research on the Effectiveness of its Application in Stock Index Forecasting

Jingdan Luo[1], Yang Shen[1,✉]

[1] *Guilin Institute of Information Technology, Guilin, Guangxi, 541000, China*

ABSTRACT

Random forest algorithm is a kind of integrated learning algorithm with strong universality, high prediction accuracy and not easy to overfitting, and strong stability in stock index prediction application. This study constructs a stock index prediction model based on the random forest algorithm, and predicts the stock index futures price state according to the iteration of the decision tree in the random forest algorithm. Then we propose to use the regular term and ARMA-GARCH time series forecasting model to optimize the overfitting and large forecasting errors in the Random Forest model to achieve the construction of stock index forecasting optimization model. It is verified that the average absolute error of the random forest optimization model proposed in this paper is only 0.0316 in stock index forecasting, and the robustness in stock index forecasting is excellent. The empirical application results of stock index forecasting show that the accuracy of this paper's model for CSI 300 and CSI 500 indexes is above 90%, and the total return of the strategy during the backtesting period is relatively high. The practical application of the stock index forecasting model proposed in this study has the value of further research, which can provide reference and guidance for investors.

*Keywords:* random forest; regular term; ARMA model; GARCH model; stock index forecasting

## 1. Introduction

Random forest algorithm is an integrated learning algorithm, the essence of which is to form a random forest model from multiple decision trees [1]. It constructs multiple decision trees and synthesizes their prediction results to obtain more accurate prediction results. Each decision tree is independent, based on random sampling of samples and random selection of features, different tree structures are constructed, and finally the average or voting results of each decision tree are obtained [2-3]. The advantages of the random forest algorithm include high accuracy and the ability to handle high-dimensional data [4-6]. The decision tree of random forest is a sequence of segmentation by a set of

✉ Corresponding author.
*E-mail address:* arthur_ys@163.com (Y. Shen).
Received 03 March 2024; Revised 20 April 2024; Accepted 10 November 2024; Published Online 16 April 2025.
DOI: 10.61091/jcmcc127b-523

features, where one optimal feature is selected for segmentation at a time, and after the segmentation, the data is divided into two subsets [7-8]. The subsets are partitioned by recursion until the final leaf node satisfies certain conditions, such as the number of data reaches a certain threshold or the information entropy is low to a certain degree. Random forests can be used in a variety of situations such as classification and regression problems, the most commonly used of which is the binary classification problem. It has been widely used in the fields of food quality testing, medical diagnosis, marketing analysis, and credit assessment [9-12].

However, with the increase of data volume and problem complexity, the defects of the random forest algorithm have gradually appeared, such as large computational volume and easy to be affected by noisy data. In order to reduce the computational complexity of the random forest algorithm, researchers have proposed a variety of optimization methods. Among them, feature selection-based optimization method is an effective means, which reduces the computation by selecting important feature subsets to construct decision trees [13-14]. In addition, the computational efficiency of Random Forest algorithm can be significantly improved by using parallel computing technique [15]. For the overfitting problem, researchers have proposed various methods. Among them, Out-of-Bag (OOB) is a commonly used technique. By setting aside a portion of data as a validation set when constructing each decision tree, the generalization ability of the model can be evaluated, thus preventing overfitting [16-18]. In addition, pruning techniques can be used to simplify the structure of decision trees and reduce the risk of overfitting [19-20]. In addition, data noisy data has a large impact on the performance of random forest algorithm. In order to deal with noisy data, researchers have proposed some anti-interference measures, such as through the integrated learning method, the prediction results of multiple models are synthesized, which can reduce the impact of noisy data on the model performance [21-22]. However, the current random forest algorithm still has the disadvantages of frequent training time, high computational overhead, black-box model, and not good at dealing with sparse data, which need to be further optimized.

In this paper, the state of stock index futures price is classified as rising or falling, on the basis of which the stock index forecasting problem is transformed into a classification problem, and the combined iterative improvement of decision tree in the random forest algorithm is utilized to realize the stock index forecasting, and the regular term is used to alleviate the overfitting problem of the model and to smooth the stock index forecasting results of the random forest model. Then use the time series forecasting ARMA model to predict the lag term of the random forest stock index forecasting error, and use the GARCH model to portray the residual term of the ARMA model to construct the ARMA-GARCH model. The model is used to optimize the random forest model to reduce the error of stock index forecasting results. The stock index forecasting performance and robustness of the random forest optimization model are tested and analyzed by the stock index forecasting model robustness test index. Then, CSI 300 index and CSI 500 index are selected for empirical application of stock index forecasting to analyze the stock index return prediction of the random forest optimization model and the strategy return under the application of the model to explore the application effect of the random forest optimization model in stock index forecasting.

## 2. Method

### *2.1. Random Forest-based Stock Index Forecasting Models*

Random forest is an integrated learning algorithm based on decision trees, which can be applied in similar occasions such as predicting the price rise and fall of stock index futures [23]. The basic prediction unit of the model is the decision tree, and the combination of the decision tree model is iteratively improved to achieve the purpose of improving the prediction effect. In the specific process of improving the decision tree, the model is iteratively improved based on the direction of the gradient of the error that exists in the prediction of stock index futures, and according to the formula (1), the

combination of the error situation to form a weight vector, and the combination of the decision tree model produced in the training, to form the overall prediction results:

$$y = \sum_{m=1}^{M} g_k(x) \tag{1}$$

where $g_k(x) = \sum_{j=1}^{J} r_j I\left(x \in R_j\right)$, $r_j$ denote the weights occupied by the $j$ rd decision tree in the final prediction and according to (2), the weights are updated during each iteration:

$$h_k(x) = h_{k-1}(x) + \eta\left(\sum_{j=1}^{J_k} \gamma_{jk} I\left(x \in R_{jk}\right)\right) \tag{2}$$

Where $\eta$ represents the progress of updating the weights during the iterative process, the value, if taken smaller, the random forest uses a smaller magnitude to update the weights, which helps to improve the accuracy of the prediction of stock index futures rise and fall, but also brings the risk of overfitting.

When predicting the price rise and fall of stock index futures, the random forest model has the following characteristics.

1) The random forest model has good predictability. Since the model is based on the second-order Taylor expansion of its loss function, it can be more carefully used to improve the fitting degree of the loss function in the iteration, which helps to improve the predictive effect of the model. At the same time, the combination of regular terms in the model also avoids the overfitting of the Random Forest model in the prediction of time series models such as stock index futures prices, and improves the robustness of the model prediction.

2) The random forest model has good robustness. The random forest model uses random sampling and other strategies to form a collection of random samples from the stock index futures price data collection, and forms a training model based on the randomly selected sample collection, reducing the interference of the noise that may exist in the sample on the prediction results, and reducing the impact of various types of abnormal data in the sample, enhancing the robustness of the model.

3) The random forest model can adapt to non-linear prediction scenarios. The stock market represented by stock index futures has a certain degree of validity, and the influence of various factors on the price of stock index futures is not linear, thus causing the common ARMA time series, logistic regression and other linear models are not well adapted to this kind of prediction scenario. However, there are certain nonlinear mechanisms in the random forest model for fitting the possible nonlinear effects, thus improving the predictive ability of the model.

Therefore, based on the applicability of the random forest model to the prediction of stock index futures price rise and fall analyzed above, this paper chooses this model to construct the final prediction model.

Moreover, in order to facilitate the analysis of the stock index futures rise and fall problem and to provide reference for the subsequent construction of trading strategies, this paper classifies the state of daily stock index futures prices as rising or falling, and transforms the prediction problem into a classification problem, which can therefore be represented using the random forest model of (3):

$$L\left(y_i, \hat{y}_i\right) = 2\sum_{i=1}^{q} \log\left(1 + \exp\left(-2 y_i p_i\right)\right) \tag{3}$$

where $\left\{\left(x_i, y_i\right)\right\}$ represents the historical sample data of stock index futures, where $x_i = \left(x_{1i}, x_{2i}, \ldots x_{qi}\right)$ denotes the indicators entered into the random forest model. $y_i$ represents the rise and fall of stock index futures at the $i$ th trading day. Then, based on the theory of the random forest model, (4) can be used to represent the prediction model, where $K$ represents the number of decision trees formed in the modeling process:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in F \tag{4}$$

Where, $f_k$ denotes that the function when the $k$nd decision tree is used to predict the price rise and fall of the stock index, and the prediction objective can be expressed by Equation (5):

$$Obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{5}$$

In Eq. (5), the final prediction results of the rise and fall of stock index futures are formed by synthesizing the decision tree models generated in the learning process and based on different weights as well as the joint mechanism, and then iterative learning is carried out based on Eq. (6), which is used to update the weights corresponding to the different decision trees in each round of the learning process:

$$Obj^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_i(x_i)\right) + \Omega(f_t) + cons\tan t \tag{6}$$

where $\hat{y}_i^t$ represents the prediction result of the Random Forest model for the stock index futures on day $i$ in $t$ rounds of iterations, and further expansion of Eq. (6) leads to Eq. (7):

$$Obj^{(t)} = \sum_{i=1}^{n} \left[ g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \tag{7}$$

Where, $f_t(x) = \omega_{q(x)}$, $\omega \in R^t$, $q: R^d \to \{1, 2, 3 \ldots T\}$, $\omega$ represent the ratio of leaf nodes in the overall decision tree nodes during the construction of the model, and $q$ represents the specific structural information of the decision tree formed during the construction of the model.

$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \omega_j^2$ is a regularity term that is used to smooth the prediction results of the random forest model for the stock index to mitigate the overfitting problem. Where $T$ denotes the leaf node size of the decision tree generated in the model. Since the decision tree is the basic prediction unit in the random forest and branching is required for its construction, it is represented by $I_j = \{i \mid q(x_i) = j\}$ and can be further varied from (7) to the form of (8):

$$\begin{aligned} Obj^{(t)} &= \sum_{i=1}^{n} \left[ g_i \omega_{q(x_i)} + \frac{1}{2} h_i \omega_{q(x_i)}^2 \right] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \omega_j^2 \\ &= \sum_{i=1}^{n} \left[ \left( \sum_{i \in I_j} g_i \right) \omega_j + \frac{1}{2} \left( \sum_{i \in I_j} h_i + \lambda \right) \omega_j^2 \right] + \gamma T \end{aligned} \tag{8}$$

To further solve the objective function, we can assume $G_j = \sum_{i \in I_j} g_i$, $H_j = \sum_{i \in I_j} h_i$ and find a derivative of $\omega_j$ on both sides of the equation, which is further transformed into:

$$\omega_j^* = -\frac{G_i}{H_j + \lambda} \tag{9}$$

The final objective function of the random forest to predict the price movement of stock index futures is obtained and denoted as:

$$Obj = -\frac{1}{2} \sum_{j=1}^{T} \frac{G_i}{H_j + \lambda} + \gamma T \tag{10}$$

### *2.2. ARMA-GARCH based random forest optimization method*

2.2.1. Model Optimization Principles:

1) $ARMA(i,j)$ Model. ARMA model [24] is a time series forecasting method. Currently, the model is widely used in linear time series forecasting. Among them, the general expression of the $i$ nd order autoregressive model $AR(i)$ is:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_i y_{t-i} + e_t \tag{11}$$

In the above equation, $c$ is the constant term, $\phi_1, \phi_2, \cdots, \phi_i$ is the autocorrelation coefficients of each order, and $e_t$ includes all the new information that the series could not be explained by past values in period $t$. Therefore, $e_t$ and $y_{t-i}(i=1,2,3\ldots)$ are independent of each other.

And the $j$ th order moving average model $MA(j)$ uses the lag term of prediction error $e_t$ to further improve the predicted value of the model with the general expression:

$$y_t = e_t - \Theta_1 e_{t-1} + \Theta_2 e_{t-2} + \cdots + \Theta_j e_{t-j} \tag{12}$$

In the above equation, $\Theta_1, \Theta_2, \ldots, \Theta_j$ is the moving average coefficient of each order.

If the time series contains both an autoregressive and a moving average component, the two models can be combined to obtain the classical $ARMA(i,j)$ -model, which has the following general expression:

$$\begin{aligned} y_t &= c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi y_{t-i} \\ &\quad + e_t - \Theta_1 e_{t-1} - \Theta_2 e_{t-2} - \cdots - \Theta_j e_{t-j} \end{aligned} \tag{13}$$

2) Model $GARCH(p,q)$. The general expression for the $GARCH(p,q)$ model is given below:

$$\begin{cases} y_t = \mu + e_t \\ e_t = \sigma_{t|t-1}\varepsilon_t \\ \sigma^2_{t|t-1} = \omega + \sum_{i=1}^{p} \alpha_i e^2_{t-i} + \sum_{j=1}^{q} \beta_j \sigma^2_{t-j|t-j-1} \end{cases} \tag{14}$$

In the above equation, $\mu$ is the conditional mean term of the model and $e_t$ is the residual term whose conditional variance varies over time. $p$ is the order of ARCH, $q$ is the order of GARCH, $\alpha_i$ and $\beta_j$ are the coefficients of each term, these coefficients are required to satisfy the non-negativity. The GARCH model [25] can be generalized in several ways, for example, the model's conditional mean can not be a fixed value, using the ARMA model to be engraved, while the residual term of the ARMA model and then use the GARCH model to be engraved, then the generalization of the model is $ARMA(u,v)-GARCH(p,q)$ model and its expression is:

$$\begin{cases} y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_u y_{t-u} \\ \quad + e_t - \Theta_1 e_{t-1} - \Theta_2 e_{t-2} - \cdots - \Theta_j e_{t-v} \\ e_t = \sigma_{t|t-1}\varepsilon_t \\ \sigma^2_{t|t-1} = \omega + \sum_{i=1}^{p} \alpha_i e^2_{t-i} + \sum_{j=1}^{q} \beta_j \sigma^2_{t-j|t-j-1} \end{cases} \tag{15}$$

2.2.2. ARMA-GARCH modeling. Based on the above analysis, this paper proposes ARMA-GARCH model for the improvement and optimization of Random Forest stock index forecasting model, there are six main steps in the modeling of ARMA-GARCH model, the specific process is as follows.

1) White noise test on the sequence, if the sequence is white noise then the modeling is over, otherwise go to step 2.
2) Smoothness test of the series, if the series is not smooth, then the difference operation to make it smooth, to be smooth data into the third step.
3) Determine the alternative orders of the model by autocorrelation function and partial autocorrelation function, and set the order of the ARMA model according to the AIC and BIC criteria.
4) Estimate the parameters of the model and test the residuals of the model for ARCH effect, if there is no ARCH effect, the ARMA model establishment is completed, and residuals diagnosis and prediction can be carried out, etc. If there is ARCH effect, then go to the fifth step.
5) Fix the order of the GARCH model and determine the distribution of its new interest, which is usually determined using the AIC and BIC criteria;
6) Estimate the coefficients of the ARMA-GARCH model, modeling is completed, and residuals are diagnosed and predictions are made, etc.

### *2.3. Optimization model construction for stock index forecasting*

The expression of the proposed Random forest-ARMA-GARCH combination model in this paper is as follows:

$$\begin{cases} close_t = f\left(x_t\right) + u_t \\ u_t = \lambda + \sum_{i=1}^{p} \phi_i u_{t-i} + \sum_{j=1}^{q} \theta_j e_{t-j} + e_t \\ e_t = \sigma_{t|t-1} \varepsilon_t \\ \sigma_{t|t-1}^2 = \omega + \sum_{i=1}^{r} \alpha_i e^2{}_{t-i} + \sum_{j=1}^{s} \beta_j \sigma_{t-j}^2 \end{cases} \tag{16}$$

Where, $f\left(x_t\right)$ is the value fitted by the random forest model, do white noise and smoothness test on the fitted residuals $u_t$, if the residuals sequence is not white noise and smooth, then construct ARMA model to fit the residuals, and then ARCH test on the residuals fitted by the ARMA model, if there is a conditional heteroskedasticity property, then construct GARCH model, otherwise do not construct GARCH model.

The flowchart of the Random forest-ARMA-GARCH model obtained after the optimization of the random forest algorithm is shown in Fig. 1. The input part is the training set $\left\{\left(x_t,\ close_t\right)\right\}_{t=1}^{T}$, where $x_t = \left(x_{t,1}, \cdots, x_{t,M}\right)$, and $T+1$ period input variables $x_{T+1}$. The output part is the prediction result $close_{T+1}$.

1) Calculate the correlation coefficients between each feature and the target variable separately, and rank the importance of the features according to the calculation results; establish a random forest model, and rank the features according to the importance of each feature to the model; synthesize the results of feature importance obtained from these two feature screening methods, and select $top\ k$ feature to form a new training set $\left\{\left(x_t', r_t\right)\right\}_{t=1}^{T}$, where $x_t' = \left(x_{t,1}', \cdots, x_{t,k}'\right)$ is the input variable of the random forest.
2) According to the feature selection results in step 1, a new input variable $x_{T+1}' = \left(x_{T+1,1}', \cdots, x_{T+1,k}'\right)$ can be obtained from $x_{T+1} = \left(x_{T+1,1}, \cdots, x_{T+1,M}\right)$.
3) Train the random forest model in the training set and use the Bayesian optimization algorithm to select the optimal parameter set.
4) Fit the training set on the trained random forest to get the fitting result $f\left(x_i'\right)$, and according to Eq. $e_i = y_i^- f\left(x_i'\right)$, get the residual time series $\left\{u_t\right\}_{t=1}^{T}$ of the training set.

5) Do white noise test on $\{u_t\}_{t=1}^{T}$. If the sequence is not white noise, perform the following steps sequentially, otherwise, go to step 11.

6) Construct an ARMA model for the residual time series $\{u_t\}_{t=1}^{T}$, whose order is determined using the autocorrelation function, partial autocorrelation function, and the AIC and BIC criteria, so that the residual time series $\{e_t\}_{t=1}^{T}$ of the ARMA model can be obtained.

7) Do the conditional heteroskedasticity test on $\{e_t\}_{t=1}^{T}$. If there is an ARCH effect on the residual series, perform the following steps sequentially, otherwise go to step 10.

8) Fit to obtain parameter $\lambda, \phi_i, \theta_j, \alpha_i, \beta_j$, thus building a combined forecasting model.

9) Use the $t+1$-period input variables obtained in step 2 and the residual time series $\{u_t\}_{t=1}^{T}$ obtained in step 4 as inputs to the combined model to obtain $close_{t+1}$ and $\sigma_{t+1}$, respectively, and transfer to step 12.

10) At this point, the random forest-ARMA model is constructed, using the $t+1$-period input variables obtained in step 2 and the residual time series $\{u_t\}_{t=1}^{T}$ obtained in step 4 as inputs to the random forest-ARMA model to obtain $close_{t+1}$.

11) Use the $t+1$-period input variables obtained in step 2 as inputs to the random forest model to obtain $t+1$-period predictions $f(x'_{t+1})$, let $close_{t+1} = f(x'_{t+1})$.

12) Output the prediction $close_{t+1}$.

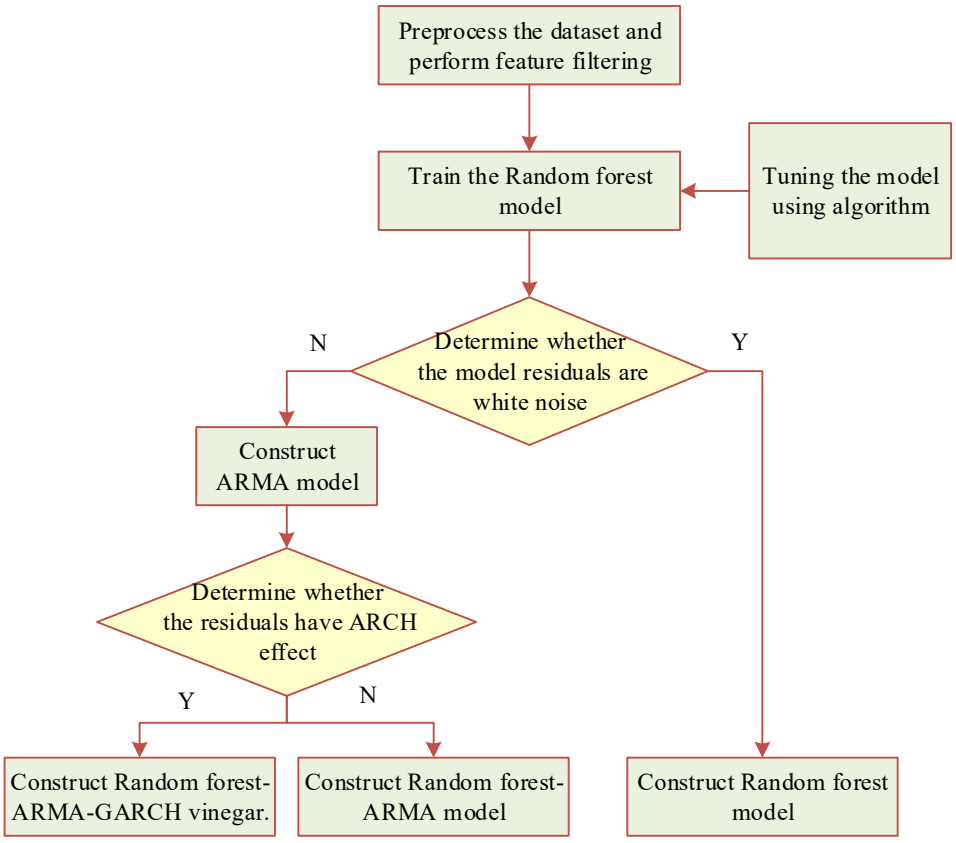



**Fig. 1.** Random forest-ARMA-GARCH model flow chart

## 3. Results and discussion

### *3.1. Stock Index Forecasting Model Robustness Tests*

3.1.1. Robustness assessment indicators. In order to analyze the robustness and predictive performance of the model in stock index forecasting, this paper will refer to the following six evaluation metrics commonly used in machine learning.

1) Mean Absolute Error (MAE), which is the average of absolute errors:

$$MAE = \frac{1}{N}\sum_{t=1}^{N}\left|r_t - \hat{r}_t\right| \tag{17}$$

Where $r_t$ is the actual rate of return at moment $t$; $\hat{r}_t$ is the expected rate of return at moment $t$.

2) Mean Square Error (MSE), is the average of the sum of the squares of the differences between the predicted and true values:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(r_t - \hat{r}_t\right)^2 \tag{18}$$

3) Root Mean Square Error (RMSE), the square root of the mean of the squared differences between the predicted and true values:

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}\left(r_t - \hat{r}_t\right)^2} \tag{19}$$

4) Deviation calculates the expected value of the difference between the predicted and true values:

$$Bias = E\left[r_t - \hat{r}_t\right] \tag{20}$$

5) The variance represents how well the model obtained by training with the samples performs in the test set:

$$Variance = E\left[\hat{r}_t^2\right] - E\left[\hat{r}_t\right]^2 \tag{21}$$

6) HitRatio, which measures the percentage of predictions that are correct in the predicted direction:

$$\begin{aligned} Hit\ ratio &= \frac{Number\ of\ times\ direction\ prediction\ was\ correct}{\left(\begin{array}{l} Number\ of\ correct\ direction\ predictions \\ +\ number\ of\ incorrect\ direction\ predictions \end{array}\right)} \\ &= \frac{1}{N}\sum_{t=1}^{N} 1_{\{r_t * \hat{r}\rangle > 0\}} \end{aligned} \tag{22}$$

Eq. $1_{\{r_t * \hat{r}_t > 0\}} = \begin{cases} 1, & r_t * \hat{r}_t > 0 \\ 0, & r_t * \hat{r}_t \le 0 \end{cases}$.

3.1.2. Results of comparative robustness analysis. The experimental data for the comparative evaluation of stock index forecasting models in this paper come from the Wind Financial Database. In order to comprehensively and systematically compare the robustness of various machine learning forecasting models in stock index forecasting, the input features of all the models are the 1-minute price-volume information and technical indicators, and the logarithmic returns of the output stock index futures are forecasted, and the experiment adopts the cross-validation method by selecting 2 months as the training set and 1 month as the test set. First, at the end of each month, this paper tunes the parameters of each stock index forecasting model with the known most recent 2-month data, and

then uses the tuned model to forecast the log return of stock index futures for the latter 1 month. Calculations are performed based on the six assessment metric formulas, and the overall forecast robustness performance and accuracy of each model are shown in Table 1. In terms of mean absolute error (MAE), mean square error (MSE) and root mean square error (RMSE), these metrics (0.0316, 0.00256, 0.0493) of the improved and optimized Random forest-ARMA-GARCH model proposed in this paper are significantly smaller than those of the other models, which indicates that the difference between its predicted value and the true value is minimal. The reason for the larger error in the decision tree model in the comparison model is that most of the predicted values are smaller than the true values, while the random forest model and the neural network model have a larger error because many of the predicted values are much larger than the true values.

**Table 1.** Analysis of performance indicators of different models

| Model | MAE | MSE | RMSE | Bias | Variance | Hit ratio |
|---|---|---|---|---|---|---|
| Decision tree | 0.0492 | 0.00318 | 0.059 | 3.12E-04 | 3.73E-03 | 0.5031 |
| Random forest | 0.0491 | 0.00305 | 0.0585 | 3.09E-04 | 3.37E-03 | 0.5141 |
| XGBoost | 0.0388 | 0.00305 | 0.0573 | 3.88E-04 | 3.07E-03 | 0.5154 |
| Neural network | 0.0385 | 0.00293 | 0.0548 | 3.30E-04 | 1.73E-03 | 0.5213 |
| Support vector machine | 0.0352 | 0.00285 | 0.0507 | 3.72E-04 | 1.32E-03 | 0.5359 |
| Random forest-ARMA-GARCH | 0.0316 | 0.00256 | 0.0493 | 3.65E-04 | 1.05E-03 | 0.5256 |

Comparing the analysis results of Hit Ratio indicators in the above results, it can be found that the accuracy of the Random forest-ARMA-GARCH model proposed in this paper in predicting the direction is relatively high (0.5256), and in order to make a better comparison, this paper carries out a further analysis based on the Hit Ratio of each prediction window. The results of the analysis of the predicted Hit Ratio metrics for each window of each model are shown in Table 2, although the average Hit Ratio of the neural network model is the highest (0.5282), there is a window (June) of only 0.4927, and the excessive fluctuation of the Hit Ratio indicates that the prediction accuracy is not stable. On the contrary, in the Random forest-ARMA-GARCH model, the Hit Ratio is kept above 0.5156 (0.5156-0.5374), so the prediction accuracy is relatively high and stable compared with other models. Based on the above analysis, the improved and optimized Random Forest model proposed in this paper has the best overall performance in terms of the robustness of the prediction of stock indices, including the smallest model error indexes, the bias-variance trade-off that reduces both of them to the order of magnitude of $10^{-5}$ and below, as well as the prediction accuracy and the stability of the accuracy are both excellent.

**Table 2.** Each window predicts the Hit Ratio

| Model | Decision tree | Random forest | XGBoost | Neural network | Support vector machine | Random forest-ARMA-GARCH |
|---|---|---|---|---|---|---|
| March | 0.4932 | 0.5019 | 0.4977 | 0.5149 | 0.5129 | 0.5156 |
| April | 0.4784 | 0.5004 | 0.4936 | 0.522 | 0.5092 | 0.5252 |
| May | 0.4981 | 0.5072 | 0.5147 | 0.5195 | 0.5109 | 0.5197 |
| June | 0.4645 | 0.506 | 0.4776 | 0.4927 | 0.5044 | 0.5374 |
| July | 0.4616 | 0.502 | 0.4863 | 0.5434 | 0.5017 | 0.5344 |
| August | 0.4561 | 0.5031 | 0.465 | 0.5461 | 0.5083 | 0.5237 |
| September | 0.4576 | 0.5077 | 0.4953 | 0.5342 | 0.5057 | 0.5244 |
| October | 0.4852 | 0.5084 | 0.4989 | 0.5274 | 0.5012 | 0.5238 |
| November | 0.4992 | 0.5049 | 0.4603 | 0.5336 | 0.5111 | 0.5364 |

| | | | | | | |
|---|---|---|---|---|---|---|
| December | 0.4640 | 0.5026 | 0.5172 | 0.5482 | 0.5193 | 0.5332 |
| Average | 0.4758 | 0.5044 | 0.4907 | 0.5282 | 0.5085 | 0.5274 |

### *3.2. Analysis of empirical stock index forecasting applications*

#### 3.2.1. Data Selection and Description:

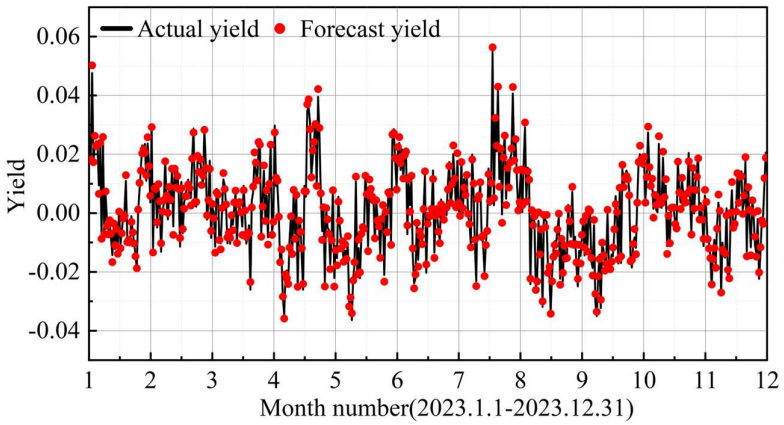
1) Selection of Technical Indicators. Technical indicators are calculated based on stock index time series data, and can be used to measure the profitability, solvency and operational capacity of the sample stocks included in the stock exchange. The technical indicators selected in this paper are relatively commonly used by investors and investment institutions, covering both fundamental analysis indicators and dynamic indicators, which can measure the trend of stock index data from multiple perspectives. The selected technical indicators include market capitalization (CMV), return on equity (ROE), return on assets (ROA), current ratio (CR), price-earnings ratio (PE), price-to-book ratio (PB), earnings per share (EPS), dividend yield (DY), relative strength index (RSI), exponentially smoothed moving average of differences and similarities (MACD), and momentum indicator (MOM). (b) Experimental data.

2) Experimental Data. In this paper, two stocks, CSI 300 index and CSI 500 index, are selected as the object of empirical application analysis. The CSI 300 is the core index of China's A-share market, reflecting the comprehensive movements of the top 300 representative stocks in the Shanghai and Shenzhen markets in terms of liquidity and size. The sample stocks included in CSI 300 are similar to the industry distribution of the market and cover most of the current market capitalization. After excluding the constituents of CSI 300 and the top 300 stocks in terms of total market capitalization in the A-share market, the top 500 stocks in terms of total market capitalization are the sample stocks of the CSI 500 Index, which reflects the consolidated changes in the value of small and mid-cap companies. The CSI 500 Index is calculated in the same way as the CSI 300 Index, but the difference lies in the composition of its constituents. The sample stocks of the CSI 300 Index are mainly large-cap stocks, which are large in size, competitive, sound in operation and have good performance, while the sample stocks of the CSI 500 Index are mainly small and mid-cap stocks, which have good growth and high development potential.

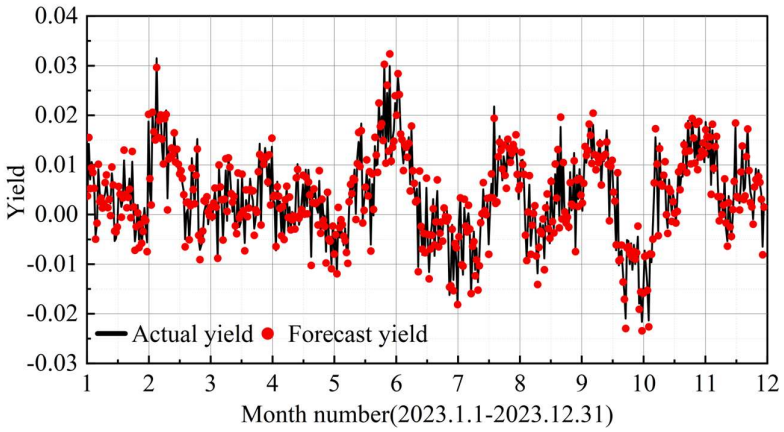
Since the data set is financial time series data, there are some noise values and outliers, for this part of the data to take data smoothing method. Because the numerical magnitude of each technical indicator is different, it is necessary to standardize the sequence of each technical indicator to eliminate the effect of magnitude, so as to map all the data into the range of [-1, 1]. The time span of the experimental data is 2021.1.1-2023.12.31, and the data of each technical indicator within 3 years are obtained to form the experimental sample space, and each trading day data is taken as a sample for calculation. The data from 2021.1.1-2022.12.31 is used as the training sample set for model training and parameter tuning, and the data from 2023.1.1-2023.12.31 is used as the test data set for evaluating the model's stock index prediction effect.

3.2.2. Equity index yield forecasts. The preprocessed technical indicators are used as feature inputs, and the returns are used as labeled values to predict the CSI 300 and CSI 500 index data sets within the interval of 2023.1.1-2023.12.31 on the modified random forest model, and compare them with the actual returns, and the outputs of the comparison results are shown in Fig. 2, with (a) and (b) representing the results of analyzing the returns of the stocks CSI 300 and CSI 500 index yield analysis results. The return volatility of CSI 300 is slightly larger than that of CSI 500, but the overall volatility trend of the returns of large-cap and small- and mid-cap stocks is similar from the time level. And the improved random forest model has better yield prediction effect on CSI 300 index data set and CSI 500

index data set, although there are still some deviations, the overall yield trend prediction is more accurate, and the prediction accuracy is 95.62% and 91.42% on average, respectively.




(a) Shanghai and Shenzhen 300




(b) Certificate 500

**Fig. 2.** Earnings forecast analysis results

### *3.3. Backtest analysis of model application strategies*

3.3.1. Strategy evaluation indicators. This paper uses cumulative return, maximum retracement, and the traditional Sharpe ratio as evaluation metrics for trading strategies.

1) The maximum retracement, which is the maximum value of the magnitude of the return retracement when the price reaches the lowest point looking backward at any point in the selected time period, can represent the worst case scenario that will happen with probability in the trading process, and is a very important risk indicator. The formula is:

$$\max Drawdown = \max\left(\frac{P_i - P_j}{P_i}\right) \tag{23}$$

Where $P_i$ is the price on day $i$; $P_j$ is the price on day $j$, $i < j$.

2) The traditional Sharpe ratio studies the size of the excess return of the portfolio while bearing a unit of total risk, the index can be the classic indicator of risk and return of the two integrated consideration, the formula is:

$$SharpeRatio = \frac{E(R_P) - R_f}{\sigma_P} \tag{24}$$

where $E(R_P)$ is the expected rate of return, $R_f$ is the risk-free rate of return, and $\sigma_P$ is the standard deviation of excess returns.

3.3.2. Strategy backtesting results. This paper takes stock trading strategy as the starting point, imports the prediction results of Random forest-ARMA-GARCH model into the trading platform, and constructs trend timing strategy based on it. The results of the return analysis after the strategy runs backtest are shown in Figure 3. First, from the perspective of total return, the average total return of the strategy (91.57%) far exceeds the benchmark return (16.40%). The strategy return curve in red in the figure is significantly higher than the benchmark return curve in black for most of the time. This suggests that over the long term, the strategy delivers higher returns and more lucrative profits for investors. Second, the volatility of the strategy's returns did not become greater as returns increased. On the contrary, the volatility of the strategy's return curve is relatively small, showing better stability. This reflects the strategy's pursuit of returns while taking into account risk control, reflecting a high investment value. Finally, even during periods of low benchmark returns, the strategy's returns have remained at a relatively high level, which indicates that the strategy has a certain degree of flexibility and can still maintain its excellent return performance under different market conditions. All in all, the strategy's return has the advantages of higher total return, lower volatility and higher flexibility than the benchmark's return.

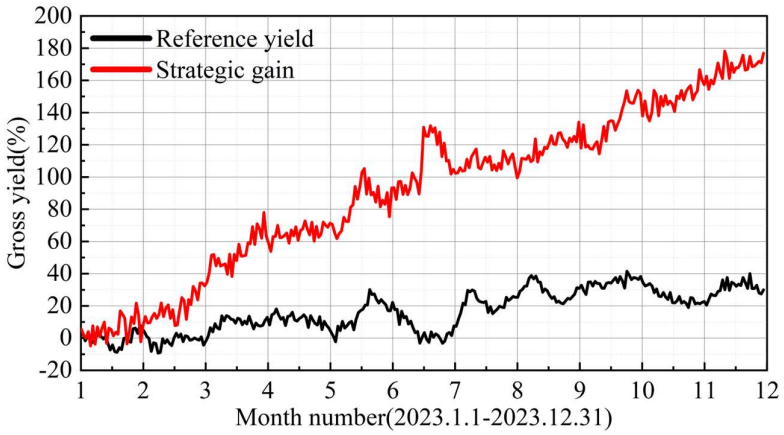



**Fig. 3.** Comparison of strategy returns and benchmark returns

The strategy can be further analyzed in a comprehensive manner by calculating the common evaluation metrics for strategy backtesting, which are shown in Table 3, along with the results. The total return of this strategy during the backtesting period is 176.88%, which means that the investment amount has increased by 176.88%, i.e., based on the initial investment amount, the adoption of this strategy is able to obtain nearly twice the return. The Sharpe ratio is 4.521, which indicates that the excess return obtained by this strategy per unit of total risk taken is much higher than the average, indicating that its risk-adjusted return is excellent. The maximum retracement was 11.62%, which is very small relative to the total return, indicating that the strategy suffered a low capital loss in the event of a market decline. The Beta coefficient is 0.559, which is much lower than 1, indicating that the strategy has low market risk sensitivity, does not incur substantial losses due to market volatility, and has good market independence. The profit-loss ratio is 5.213, which means that the average gain of profitable trades of this strategy far exceeds the loss of average losing trades, and even if the losing trades occur, their impact on the overall return is small. Summarizing the performance of the strategy's indicators, it can be concluded that the trend timing strategy constructed based on the optimized random forest model performs well in terms of return, risk measure and trading evaluation. It not only delivers high returns, but also does a good job in risk control and maintains stable performance even under unfavorable market conditions. In addition, the high win

rate and high profit/loss ratio are further evidence of the effectiveness and profitability of its trade execution.

**Table 3.** Evaluation metrics for trend timing strategy

| Type | Index | Numerical value |
|---|---|---|
| Yield type | Gross yield | 176.88% |
| | Annual yield | 91.57% |
| | Excess yield | 124.62% |
| Risk measurement | Sharpe ratio | 4.521 |
| | Sotanol ratio | 7.694 |
| | Peak back | 11.62% |
| | Beta coefficient | 0.559 |
| Transaction evaluation type | Odds | 0.784 |
| | Break-even ratio | 5.213 |

## 4. Conclusion

In this paper, ARMA-GARCH model is proposed to predict the error generated by Random Forest model in the process of stock index forecasting, in order to improve the performance of Random Forest algorithm in stock index forecasting, and construct Random Forest-ARMA-GARCH stock index forecasting model. The prediction performance and robustness of the optimization model are examined, and the application effect of the model in stock index prediction is analyzed empirically, and the results show that:

1) The average absolute error (0.0316), mean square error (0.00256) and root mean square error (0.0493) of the improved optimized stock index forecasting model proposed in this paper are significantly smaller than those of other comparative models in terms of robustness test, and the values of Hit Ratio indicator are kept above 0.5156 (0.5156-0.5374), which is relatively high and more accurate in forecasting than other models. The prediction accuracy is relatively high and stable compared with other models.

2) In the empirical prediction of CSI 300 and CSI 500 indices, the model of this paper is more accurate in predicting the overall return trend of stocks, with an average prediction accuracy of 95.62% and 91.42%, respectively. It is also found that the total return and profit/loss ratio of the stock strategy under the model application for the backtesting period reaches 176.88% and 5.213, respectively, which means that the average gain of profitable trades of this strategy far exceeds the loss of average losing trades.

In summary, according to the experiments designed and results obtained in this paper, the optimized Random Forest stock index prediction model using ARMA-GARCH model is suitable for the prediction of the rise and fall of the stock price index as well as the construction of quantitative trading strategies, and the findings of the study can provide a certain reference basis for both institutional investors and individual investors.

## References

[1] Boateng, E. Y., Otoo, J., & Abaye, D. A. (2020). Basic Tenets of Classification Algorithms K-Nearest-Neighbor, Support Vector Machine, Random Forest and Neural Network: A Review. forest (RF), 10, 11.

[2] Chutia, D., Borah, N., Baruah, D., Bhattacharyya, D. K., Raju, P. L. N., & Sarma, K. K. (2020). An effective approach for improving the accuracy of a random forest classifier in the classification of Hyperion data. Applied Geomatics, 12(1), 95-105.

[3] Xia, J., Zhang, S., Cai, G., Li, L., Pan, Q., Yan, J., & Ning, G. (2017). Adjusted weight voting algorithm for random forests in handling missing values. Pattern Recognition, 100(69), 52-60.

[4] Shaik, A. B., & Srinivasan, S. (2018, November). A Brief Survey on Random Forest Ensembles in Classification Model. In International Conference on Innovative Computing and Communications: Proceedings of ICICC 2018, Volume 2 (Vol. 56, p. 253). Springer.

[5] Zhu, L., Qiu, D., Ergu, D., Ying, C., & Liu, K. (2019). A study on predicting loan default based on the random forest algorithm. Procedia Computer Science, 162, 503-513.

[6] Darst, B. F., Malecki, K. C., & Engelman, C. D. (2018). Using recursive feature elimination in random forest to account for correlated variables in high dimensional data. BMC Genomic Data, 19.

[7] Jaiswal, J. K., & Samikannu, R. (2017, February). Application of random forest algorithm on feature subset selection and classification and regression. In 2017 world congress on computing and communication technologies (WCCCT) (pp. 65-68). Ieee.

[8] Probst, P., & Boulesteix, A. L. (2018). To Tune or Not to Tune the Number of Trees in Random Forest. Journal of Machine Learning Research, 18, 1-18.

[9] de Santana, F. B., Neto, W. B., & Poppi, R. J. (2019). Random forest as one-class classifier and infrared spectroscopy for food adulteration detection. Food chemistry.

[10] Zhu, M., Xia, J., Jin, X., Yan, M., Cai, G., Yan, J., & Ning, G. (2018). Class Weights Random Forest Algorithm for Processing Class Imbalanced Medical Data. IEEE Access, 6, 4641-4652.

[11] Ładyżyński, P., Żbikowski, K. P., & Gawrysiak, P. (2019). Direct marketing campaigns in retail banking with the use of deep learning and random forests. Expert Systems with Applications, 134.

[12] Arora, N., & Kaur, P. D. (2020). A Bolasso based consistent feature selection enabled random forest classification algorithm: An application to credit risk assessment. Applied Soft Computing, 86, 105936.

[13] Ma, L., & Fan, S. (2017). CURE-SMOTE algorithm and hybrid algorithm for feature selection and parameter optimization based on random forests. BMC Bioinformatics, 18(1), 1-18.

[14] Paul, A., Mukherjee, D. P., Das, P., Gangopadhyay, A., Chintha, A. R., & Kundu, S. (2018). Improved Random Forest for Classification. IEEE transactions on image processing: a publication of the IEEE Signal Processing Society, 27(8), 4012-4024.

[15] Naue, J., Hoefsloot, H. C. J., Mook, O. R. F., Rijlaarsdam-Hoekstra, L., van der Zwalm, M. C. H., Henneman, P., ... & Verschure, P. J. (2017). Chronological age prediction based on DNA methylation: Massive parallel sequencing and random forest regression. Forensic Science International. Genetics, 31.

[16] Barreñada, L., Dhiman, P., Timmerman, D., Boulesteix, A. L., & Van Calster, B. (2024). Understanding overfitting in random forest for probability estimation: a visualization and simulation study. Diagnostic and Prognostic Research, 8(1), 14.

[17] Mohapatra, N., Shreya, K., & Chinmay, A. (2020). Optimization of the Random Forest Algorithm. Advances in Data Science and Management, 201.

[18] Ramosaj, B., & Pauly, M. (2019). Consistent estimation of residual variance with random forest Out-Of-Bag errors. Statistics & Probability Letters, 151(C), 49-57.

[19] García, R., Fernández Anta, A., Mancuso, V., & Casari, P. (2019). A Novel Hyperparameter-Free Approach to Decision Tree Construction That Avoids Overfitting by Design. IEEE Access, 7, 99978-99987.

[20] Ahmed, A. M., Rizaner, A., & Ulusoy, A. H. (2018). A novel decision tree classification based on post-pruning with Bayes minimum risk. PLOS ONE, 13(4), 1-12.

[21] Reis, I., Baron, D., & Shahaf, S. (2019). Probabilistic Random Forest: A Machine Learning Algorithm for Noisy Data Sets. The Astronomical Journal, 157(1), 16.

[22] Lin, W., Wu, Z., Lin, L., Wen, A., & Li, J. (2017). An Ensemble Random Forest Algorithm for Insurance Big

Data Analysis. IEEE Access, 5, 16568-16575.

[23] Liu Donghai,Dai Qianxin,Tang Xinlin,Zhang Rui,Lu Tingjie & Chen Junjie. (2025). An Improved Random Forest–Based Operation Duration Prediction of Long-Distance Tunnel Construction Considering Geological Uncertainty. Journal of Computing in Civil Engineering(2).

[24] Gholamreza Hesamian,Arne Johannssen & Nataliya Chukhrova. (2024). A neural network-based ARMA model for fuzzy time series data. Computational and Applied Mathematics(8),445-445.

[25] Jiawen Wang. (2024). Research on Return Volatility of Shenzhen Stock Exchange Component Index Based on GARCH Model. Journal of Economics and Public Finance(4).