IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 22, 2019, accepted January 16, 2020, date of publication January 24, 2020, date of current version February 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2969293

Integrated Long-Term Stock Selection Models Based on Feature Selection and Machine Learning Algorithms for China Stock Market

XIANGHUI YUAN[1], JIN YUAN[1], TIANZHAO JIANG[2], AND QURAT UL AIN[1]
[1]School of Economics and Finance, Xi'an Jiaotong University, Xi'an 710049, China
[2]Shanghai Foresee Investment Ltd., Liability Company, Shanghai 200120, China

Corresponding author: Jin Yuan (jiny1994@stu.xjtu.edu.cn)

This work was supported by the Chinese Natural Science Foundation under Grant 11631013, Grant 11971372, and Grant 11801433.

ABSTRACT The classical linear multi-factor stock selection model is widely used for long-term stock price trend prediction. However, the stock market is chaotic, complex, and dynamic, for which reasons the linear model assumption may be unreasonable, and it is more meaningful to construct a better-integrated stock selection model based on different feature selection and nonlinear stock price trend prediction methods. In this paper, the features are selected by various feature selection algorithms, and the parameters of the machine learning-based stock price trend prediction models are set through time-sliding window cross-validation based on 8-year data of Chinese A-share market. Through the analysis of different integrated models, the model performs best when the random forest algorithm is used for both feature selection and stock price trend prediction. Based on the random forest algorithm, a long-short portfolio is constructed to validate the effectiveness of the best model.

INDEX TERMS Stock, trend prediction, machine learning, feature selection, long-term investment.

I. INTRODUCTION

The ability of investors to make profit depends mainly on their prediction ability, while most investors in the Chinese A-share market are facing investment loss. One of the main reason is that most of the investors have limited information and ability to predict the stock price trend well. Therefore, how to construct an effective stock selection model to improve investors' predictability is a meaningful topic.

In recent years, the algorithms for stock trend prediction have been continuously proposed. From the perspective of forecasting, they can be divided into two major categories. One is the stock price trend prediction, which is called classification [1]–[5]. The other is the stock price forecast, which is called regression [6]–[10]. In addition, from the perspective of forecasting time, it is roughly grouped into two types: the short-term and the long-term trend forecast of stock price [11]. In general, the length of time for stock price trend prediction is highly correlated with the selected features. For example, indicators such as yesterday's closing price and 5-day moving average closing price are usually used to predict the stock's short-term trend. In contrast, financial factors such as operating income and return on total assets are more useful in predicting the long-term trend of stocks.

In this paper, we focus on the long-term stock price trend prediction in order to construct a better long-term stock selection model. The widely used classic models for long-term stock price trend prediction are the well-known Capital Asset Pricing Model (CAPM) and the Arbitrage Pricing Theory (APT). These two models have been studied and improved by many scholars. For example, Fama and French establish the three-factor model to explain stock returns [12]. These are linear models with the historical features data as the inputs, and with the stock returns as the outputs. However, the stock market is chaotic, complex, and dynamic, for which reasons the linear model assumption may be unreasonable, and it is especially important to consider a nonlinear model to achieve a mapping between features and stock returns. Fortunately, how to establish nonlinear models can be found in many pieces of literature in recent years. Huang, Nakamoria, and Wang state that they have predicted the direction of NIKKEI 225 index, and the results show that the support vector machine (SVM) algorithm can achieve higher accuracy [1]. Patel et. al analyze applications of the four models in

The associate editor coordinating the review of this manuscript and approving it for publication was Sunith Bandaru.

the Indian market including artificial neural network (ANN), SVM, random forest (RF), and Naive Bayes (NB) [13], [14].

At present, there are more than 3,000 listed companies in the Chinese A-share market, and the number of listed companies is still increasing. The traditional strategy is mainly through research on listed companies to determine whether the company is worth buying or not. However, with the increasing number of listed companies, this traditional strategy will also require more manpower and material resources, and thus the sustainability of the strategy is not strong. Numerous experts and scholars verify that the China stock market is still in the developing stage and a large number of individual investors trade in this market. Moreover, due to information asymmetry and other phenomena, the price of some stocks tends to deviate from their own intrinsic value. Therefore, through analyzing historical stock data by computer, the construction of quantitative stock selection strategy model has great potential in such market. This kind of model can generate a set of logical and strict trading instructions to avoid investors from being affected by market sentiment and resulting in incorrect judgments.

This paper focuses on the multi-features stock selection model based on different feature selection algorithms and machine learning based stock price trend prediction algorithms for the China stock market and establishes the nonlinear relationship between factors and stock returns. It expands the development direction of the classical multi-factor model and provides a new investment strategy for investors in the China stock market. In our work, 60 features are obtained to be used as the input of the model, through the financial report, daily opening prices, closing prices, volumes and other data of the A-share market. The main contributions of this research are reflected in the following points. First, the feature selection algorithm is used to filter the features, which can reduce the complexity of the model, and avoid the dimensional disaster caused by too many features. Second, considering the problem of analyzing time series data using the original cross-validation method, the method of time sliding window cross-validation is adopted to make the model more suitable for the actual situation. Third, the stock price trend forecasting algorithm is applied to predict the excess returns of stock of the subsequent month.

The remainder of this paper is organized into the following sections. In Section 2, several common stock price trend prediction algorithms and feature selection algorithms are introduced and also describes the principle and application of each algorithm in detail. Section 3 explains the experiment scheme. Sections 4 discusses the results of the experiment and proposes a effective trading strategy. Section 5 concludes this paper.

II. METHODOLOGY

A. THE METHODS OF STOCK PRICE TREND PREDICTION

1) SVM

The support vector machine was first proposed by Vapnik and then applied in different fields [15]. There are two categories for support vector machines: classification (SVC) [3], [16], and regression (SVR) [17], [18]. The core idea of SVM is the maximum margin hyper plane, in which way it can classify the sample data into two different categories, including positive and negative examples [19].

The steps to build a decision tree are as follows.

$(x_i, label_i), (i = 1, 2, \ldots, n)$ is a series of linearly separable sample data in an N-dimensional plane, where x_i is the data set in the N-dimensional space, $label_i$ is the label corresponding to the sample data and the value is -1 or 1. Then the function obtained by SVM is $w^T \cdot x + b = 0$. The equation between the points is calculated as:

$$D = \frac{|w^T \cdot x + b|}{||w||} = \frac{label(w^T \cdot x + b)}{||w||}$$

The idea of SVM mentioned above is to maximize the minimum interval, so the algorithm can be transformed into an optimization problem as follow.

$$\min \frac{1}{2}||w||^2 \; s.t. \; label_i(w^T \cdot x_i + b) \geq 1 (i = 1, 2, \ldots, n)$$

The above optimization problem is a quadratic programming problem, which can be solved by a corresponding method. Then this paper introduces the second method to solve the classification hyper plane equation: Lagrangian multiplier method. The above optimization problem is transformed into the following questions by the Lagrangian multiplier method and the KKT condition (Lagrange duality):

$$\max_{\alpha \geq 0} \min_{w,b} L(w, b, \alpha)$$

$$L(w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} \alpha_i (label(w^T \cdot x_i + b) - 1)$$

where $label_i$ is the classification label and α_i is the Lagrangian multiplier. The above formula is calculated, and then the following equation can be achieved:

$$\max L(w, b, \alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j \; label_i \; label_j x_i^T x_j$$

$$s.t. \; \alpha_i \geq 0; \quad \sum_{i=1}^{n} \alpha_i \; label_i = 0$$

The optimization problem can be solved by the sequential minimal optimization (SMO) algorithm and then the specific value of the parameter α_i can be known. The algorithm can be described as an optimization function written as:

$$\begin{aligned} f(x) &= sign(w^T \cdot x + b) \\ &= sign((\sum_{i=1}^{n} \alpha_i x_i \; label_i)^T \cdot x + b) \\ &= sign(\sum_{i=1}^{n} \alpha_i \; label_i < x_i, x > + b) \end{aligned}$$

However, the data distribution currently discussed is an ideal situation that is completely linear and can be divided

TABLE 1. Kernel functions.

Kernel Functions	Formula
Linear Function	$K(x_i, x_j) = < x_i, x_j >$
Polynomial Function	$K(x_i, x_j) = (< x_i, x_j > +1)^d$
Gaussian Function	$K(x_i, x_j) = \exp(-\gamma \parallel x_i - x_j \parallel^2)$

without abnormal points. It is not possible to get the hyper plane directly using the maximum-minimum interval. Therefore, the model allows the data to deviate from the hyper plane and get a new optimization function:

$$\max \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j \, label_i \, label_j x_i^T x_j$$

$$s.t. \; 0 \le \alpha_i \le C; \sum_{i=1}^{n} \alpha_i \, label_i = 0$$

where C refers to the slack variable, which changes the tolerance of the model to the abnormal point by changing the size of C.

An advantage of the SVM over the LR model is that the SVM solves the linear indivisibility problem of data by applying the kernel function. In general, data in a low-dimensional space is nonlinear, but after mapping it to a high-dimensional space, the data can become linearly separable. The function of kernel function is to calculate the inner product of two vectors in low-dimensional space and map to the function of high-dimensional space, which not only realizes the mapping from low-dimensional space to high-dimensional space but also reduces the complexity. The function obtained by SVM is:

$$f(x) = sign(\sum_{i=1}^{N} \alpha_i y_i \cdot K(x_i, x_j) + b)$$

K is the kernel function of SVM, which is the biggest feature of SVM. Several common kernel functions are shown in Table1.

2) RF

The random forest algorithm is continuously evolved and based on the decision tree algorithm. The traditional decision tree model is constructed by dividing the sample data according to the features. Each partition determines the optimal partitioning property. As the number of features increases, and the number of branch nodes increases, the model constructed in this way is the decision tree [20], [21].

The steps to build a decision tree are as follows.

(a) the creation of root nodes: all data samples are placed in the root node, and then all features are traversed, and the optimal features are filtered out from there to divide the data samples; thus the original data samples are divided into multiple subsets. The methods for evaluating features are information gain, information gain rate, Gini index, etc.

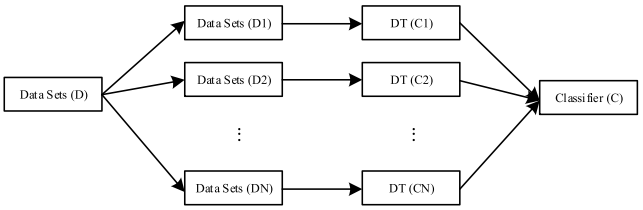


FIGURE 1. The structure of random forest.

For example, ID3 algorithm applies in information gain, and CART tree uses Gini index.

(b) the creation of leaf nodes: the data sets divided by the optimal feature are placed in the leaf node.

(c) the segmentation of leaf node: for each subset of data, the feature sets at this time are the remaining feature after the optimal feature is removed. Then continue to traverse all features, and the best feature is selected to divide the subset of data to form a new subset.

(d) the construction of the decision tree model: continue with step (2) and step (3) until the conditions for stopping the split are met. In general, there are some conditions of the split stop, such as the number of leaf nodes satisfying the condition, all features have been used to divide the data, and so on.

Consequently, the decision tree is constructed in a relatively simple way, and the calculation is not very complicated. However, since each time the data is divided into a local optimization, which is similar to the principle of the greedy algorithm for data sample partitioning, so it is easy to cause over-fitting. In order to further solve the problem, the random forest adopts the random sampling method of putting back, and obtains several sub-datasets from the original data sets, which are respectively used to train a decision tree (weak classifier) model, and the final model is built in this way by voting, taking the mean value and so on.

Fig.1 shows the structure of the random forest [22]. The steps for building the model are as follows. Firstly, the sample data sets are selected based on the Bagging method from the original training data sets and generate N training data sets by the above step. Secondly, the main job is to train N decision tree models separately based on these N training data sets. Thirdly, the random forest will consist of these N decision trees. For the classification problem, the final classification result is decided by the N decision tree classifiers, and for the regression problem, the average of the predicted values of the N decision trees determines the final prediction result [23], [24].

3) ANN

Artificial neural networks have been widely used for stock price trend prediction because it can better complete the construction of nonlinear models [25], [26]. This paper mainly constructs a three-layer, fully connected neural network model, and Fig. 2 describes the specific structure. The function of the model is to use the features data sets as the input to the model, and the final prediction result is obtained

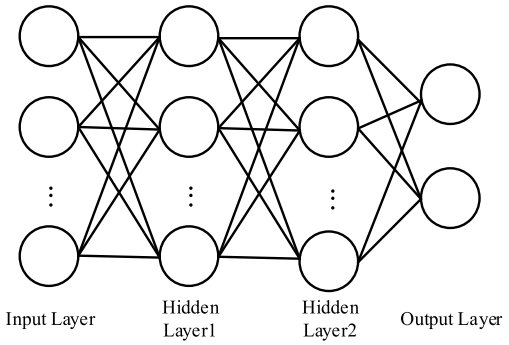


FIGURE 2. The structure of three-layer fully connected neural network.

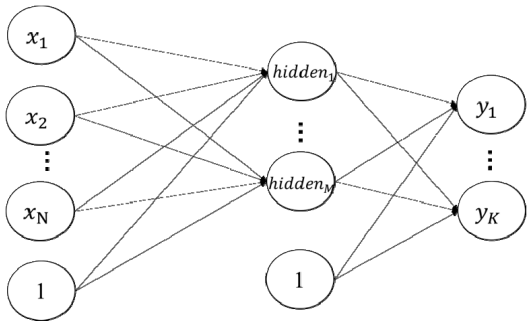


FIGURE 3. The structure of the two-layer fully connected neural network.

through the output layer after the calculation of two hidden layers. The weights on the nodes are calculated by the error back propagation algorithm.

For example, this paper assumes the existence of a two-layer fully connected neural network, presented in Fig. 3. The detail of the figure is as follows: the number of the network input layer nodes is N, which is defined as $\{x_1, x_2, \ldots, x_N\}$, and the number of hidden layer nodes is M, which is defined as $\{hidden_1, hidden_2, \ldots, hidden_M\}$. The number of output layer nodes is K, which is defined as $\{y_1, y_2, \ldots, y_K\}$. The weight from the input layer nodes to the hidden layer nodes is $W_{ij}^1 (i = 1, 2, \ldots, N; j = 1, 2, \ldots, K)$ and the weight from the input layer nodes to the hidden layer nodes is $W_{ij}^2 (i = 1, 2, \ldots, M; j = 1, 2, \ldots, K)$. The bias term of input layer is b_1 and the weight from bias item to hidden layer node is bk_1. The bias term of the hidden layer is b_2 and the weight from bias item of the hidden layer to the output layer node is bk_2.

After the data of the input layer and the output layer, it needs to determine the weight between each node to achieve the construction of the entire network. The method for determining the weight between nodes is the error back propagation algorithm, and the specific flow chart is displayed as in Fig. 4.

(a) Parameter initialization

The initial weight and the offset weight are set and then the initial weight is updated according to the forward propagation and the error back propagation until the error or the number of iterations meets the condition.

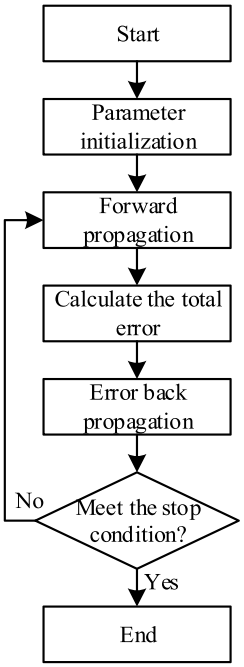


FIGURE 4. The flow chart of the error back propagation algorithm.

(b) Forward propagation

From the input layer to the hidden layer:

$$\begin{aligned} hidden_j &= f(net_j^1) \\ &= f(x_1 W_{1j}^1 + x_2 W_{2j}^1 + \ldots + x_n W_{nj}^1 + b_1 bk_1), \\ &\quad (j = 1, 2, \ldots, M) \end{aligned}$$

where $f(\cdot)$ is the activation function, and it is set as the Sigmoid function, which is written as $f(x) = \frac{1}{1+e^{-x}}$.

From a hidden layer to output layer:

$$\begin{aligned} y_j &= f(net_j^2) \\ &= f(hidden_1 W_{1j}^2 + hidden_2 W_{2j}^2 + \ldots + hidden_m W_{mj}^2 + b_2 bk_2) \end{aligned}$$

(c) Calculate the total error

The algorithm defines a loss function to measure the fit of the model. The smaller the value of the loss function, the better the performance of the fitting. For each data sample, the loss function is:

$$E = \frac{1}{2} \sum_{i=1}^{n} (y_i - t_i)^2$$

where t_i is the target value.

(d) Error back propagation

The weight is updated by error back propagation so that the error can be reduced. There are some common optimization methods such as gradient descent method. The gradient descent method is used as an example to illustrate the process of error back propagation.

(1) Update the weight between the hidden layer and the output layer

Firstly, the effect of each weight on the overall error is calculated, which is the partial error to the bias of the weight:

$$\frac{\partial E}{\partial W_{ij}^2} = \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial net_j^2} \cdot \frac{\partial net_j^2}{\partial W_{ij}^2} = (y_j - t_j) \cdot y_j \cdot (1 - y_j) \cdot hidden_i, \quad (i = 1, 2, \ldots, M; j = 1, 2, \ldots, K)$$

For the bias item weights:

$$\frac{\partial E}{\partial bk_2} = \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial net_j^2} \cdot \frac{\partial net_j^2}{\partial bk_2} = (y_j - t_j) \cdot y_j \cdot (1 - y_j) \cdot b_2$$

Secondly, it needs to set learning efficiency to update weights:

$$W_{ij}^{2+} = W_{ij}^2 + \eta \cdot \frac{\partial E}{\partial W_{ij}^2}$$

$$bk_2^+ = bk_2 + \eta \cdot \frac{\partial E}{\partial bk_2}$$

where W_{ij}^{2+} and bk_2^+ are the updated weights.

(2) Update the weight between the input layer and the hidden layer

Firstly, the partial derivative of the total error versus weight is calculated:

$$\frac{\partial E}{\partial W_{ij}^1} = \frac{\partial E}{\partial hidden_j} \cdot \frac{\partial hidden_j}{\partial W_{ij}^1} = \sum_{k=1}^{K} \left(\frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial net_k^2} \cdot \frac{\partial net_k^2}{\partial hidden_j}\right) \cdot hidden_i \cdot (1 - hidden_i) \cdot x_i$$

Secondly, it needs to set learning efficiency to update weights:

$$W_{ij}^{1+} = W_{ij}^1 + \eta \cdot \frac{\partial E}{\partial W_{ij}^1}$$

$$bk_1^+ = bk_1 + \eta \cdot \frac{\partial E}{\partial bk_1}$$

where W_{ij}^{1+} and bk_1^+ are the updated weights. So far, the weight of the ownership is updated so that the error is reduced, and the loop operation steps (2), (3), and (4) are continued until the error is less than the set threshold or the number of iterations satisfies the condition.

The above part introduces the construction method of a two-layer fully-connected neural network and the weight update method. For multi-layer neural networks, the same algorithm can be used to construct the multi-layer neural network.

B. THE METHODS OF FEATURE SELECTION

When all the features are directly used as input to the model in the process of actually creating the model, the following situations may occur because of unrelated features and correlations between features: (1) higher complexity of the model and (2) after the feature dimension exceeds a certain limit, the performance of the classifier decreases as the feature dimension increases [27].

1) SVM-RFE

The recursive feature elimination (RFE) applies in the machine learning model to perform multiple rounds of training [28]. After each round of training, the features with the lowest importance are eliminated, and then the model is trained based on the new feature set.

SVM-RFE is the recursive feature elimination algorithm based on SVM [29]. The operation steps are as follows. The original sample data set is

$$D = \{x_i^1, x_i^2, \ldots, x_i^n, y_i\}, (i = 1, 2, \ldots, m)$$

where n represents the number of features and m represents the sample data.

(a) The original sample data set is used as input to the linear SVM model to train the SVM model. The classification decision function of the linear SVM model is $f(x) = sign(w^T \cdot x + b)$, where $w_i (i = 1, 2, \ldots, n)$ indicates the weight corresponding to the ith feature.

(b) Calculate the importance score for each feature as:$score_i = w_i^2 (i = 1, 2, \ldots, n)$, where $score_i$ represents the importance score of the ith feature.

(c) Sort the importance of all features in descending order and remove the last ranked feature. The feature-length is changed from n to n-1, and then the data set D is updated.

(d) Cycle through steps (1), (2), (3) until the number of remaining features meets the set conditions.

2) FEATURE SELECTION BASED ON RF

Considering that the decision tree model is constructed in the way that the optimal features are selected from all the features at a time to use to divide the sample data but how to select the optimal feature from a large number of features has become the key issue of the decision tree model. The decision tree model measures the importance of features by the value of information gain, information gain rate, Gini index, and so on. In this way, the importance of all features can be measured well. Therefore, the feature selection algorithm based on the tree model can draw some features with higher importance in this way [30], [31].

With the continuous development of the model, the integrated forests such as random forest and GBDT have begun to emerge, which has solved the over-fitting problem of decision tree to some extent. Therefore, this paper adopts the feature selection algorithm based on random forest. The specific calculation of the model is as follows;

(a) The random forest uses the sampled method of returning to generate the sub-data set, which is used to construct the corresponding decision tree model. Therefore, some data will not be selected during the sampling process. At this time, this part of the data is called the bag (OOB). It is used to analyze the out-of-bag data through the newly constructed decision

tree model to calculate the corresponding error, which is recorded as OOB_error1.

(b) In the data obtained just outside the bag, select all the data corresponding to one of the features, and then randomly change the value of some of the data, that is, add a certain noise to interfere.

(c) Assume that the number of decision trees of the random forest is n and calculate OOB_error1 before adding noise and OOB_error2 after adding noise interference for each decision tree model.

(d) Calculate the importance of features according to the formula $\frac{1}{n}\sum_{i=1}^{n}(OOB\_error1_i - OOB\_error2_i)$. It is reasonable that if the feature is more important, the worse the prediction effect after adding noise. So it can be found by the formula that the larger the value, the greater the importance of the feature.

(e) For each feature, cycle through steps (1), (2), (3) to achieve a calculation of the importance of each feature.

(f) Feature filtering is performed according to the calculated feature importance. All features are sorted, and then the most important features can be selected according to the method of selecting a specified number of features or setting a feature importance threshold.

III. EXPERIMENTS DESIGN

The second chapter introduces the principle of the factor selection algorithm and prediction algorithm, which provides a solid theoretical basis for the construction of this model. This chapter mainly introduces how to realize the multi-factor model based on a machine-learning algorithm to proceed the real-time stock selection. The specific flow chart is shown in Fig. 5.

A. DATA COLLECTION

The data of this paper are from all stocks of the Chinese A-share market, and the work is to predict individual stock trend. Considering that special treatment (ST) stocks and sub-new stocks with shorter time to market are riskier to operate, ST stocks and stocks with less than 3 months are excluded. The data set is from January 1, 2010 to January 1, 2018. On the last trading day of each month, it collects the data set including features and stock excess returns. Further, the sliding window method is employed to divide the data into training sets and testing sets. These experimental data are all obtained from the Wind database.

It is widely known that the predictive effects of machine learning algorithms are closely related to the features, which can contribute to stock returns. To be consistent with multiple literatures, this paper selects 60 features for forecasting. The features which are used as input data for the model to predict the return of the stocks are listed in Table 2. As shown in the table, all features are divided into 10 categories, such as valuation, growth, and so on.

Since the ultimate goal of creating the model is to predict stock price trend, how to deal with stock price trends is particularly important. For example, almost all stocks are on a falling trend in the bear market and on a rising trend in the bull market.. Therefore, it is not very reasonable for the prediction of stock price trends directly. In literature, to prevent stock returns from being affected by market trends, it takes the stock excess returns, which calculated by subtracting the return of the Shanghai Stock Index from stock returns as the forecast target. Moreover, because of the impact of data noise, the models often fail to achieve good results. Thus, for the data in the sample, the stock returns are sorted in descending order first every month and then classify the top 30% of the stock as 1 and the last 30% as −1.

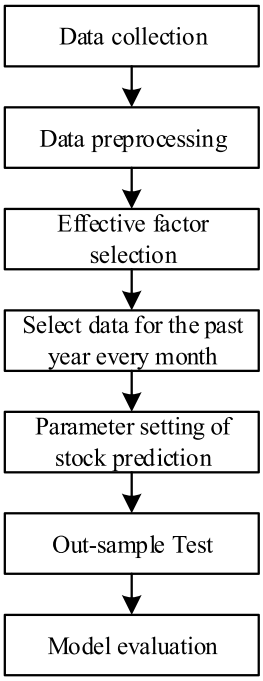


FIGURE 5. The flow chart of building the stock selection model.

B. DATA PREPROCESSING

Considering that the feature data may have extreme values, it will affect the model and lead to abnormal results. This article uses the following methods to process extreme values:

$$x_{i,new} = \begin{cases} x_m + n \times D_{MAD} & x_i \geq x_m + n \times D_{MAD} \\ x_m - n \times D_{MAD} & x_i \leq x_m + n \times D_{MAD} \\ x_i & else \end{cases}$$

where $x_{i,new}$ is the processed value, and x_i is the value of the ith variable. x_m is the median of the sequence. D_{MAD} is the median of a sequence of $|x_i - x_m|$ and n is used to control the amplitude of the upper and lower limits.

Due to a lack of financial reports, calculation errors, etc., it is likely to cause data loss, affecting the accuracy of data analysis. So, the preprocessing of missing values is quite important. In this paper, the missing values are processed using the fill method to make full use of data. Because the factors of stocks in the same industry are roughly similar, the place where the factor exposure is missing is set as the average value of the same stocks in the same industry.

TABLE 2. Input features for the stock market data set.

Category	Description of features
Valuation factors	Net profit(TTM)/Total market value
	The net profit after extraordinary gains and losses(TTM)/Total market value
	Net asset(TTM)/Total market value
	Operating income(TTM)/Total market value
	Net cash flow(TTM)/Total market value
	Operating cash flow(TTM)/Total market value
	Dividend payable(TTM)/Total market value
	Net profit(TTM) compared with the same period of last year/PE_TTM
Growth factors	Operating income growth rate compared with the same period of last year
	The net profit after extraordinary gains and losses growth rate compared with the same period of last year
	Operating cash flow growth rate compared with the same period of last year
	ROE compared with the same period of last year
Financial quality factors	ROE(QTD)
	ROE(TTM)
	ROA(QTD)
	ROA(TTM)
	Gross profit margin(QTD)
	Gross profit margin(TTM)
	The net profit margin after extraordinary gains and losses(QTD)
	The net profit margin after extraordinary gains and losses(TTM)
	Operating cash flow/Net profit(QTD)
	Operating cash flow/Net profit(TTM)
	Inventory turnover(YTD)
	Inventory turnover(TTM)
	Total assets turnover(YTD)
	Total assets turnover(TTM)
Leverage factors	Fixed assets ratio
	Total assets/Net assets
	Non-current liabilities/Net assets
Size factors	*ln*(circulation market value)
	Circulation market value/Total market value
	ln(total market value)
Momentum factors	Return rate 1-month
	Return rate 3-month
	Return rate 6-month
	Return rate 12-month
	Daily turnover rate × daily return rate 1-month
	Daily turnover rate × daily return rate 3-month
	Daily turnover rate × daily return rate 6-month
	Daily turnover rate × daily return rate 12-month
Volatility factors	Highest price/Lowest price 1-month
	Highest price/Lowest price 3-month
	Highest price/Lowest price 6-month
	Highest price/Lowest price 12-month
	Std of daily return rate 1-month
	Std of daily return rate 3-month
	Std of daily return rate 6-month
	Std of daily return rate 12-month
Turnover factors	Daily turnover rate 1 month
	Daily turnover rate 3 month
	Daily turnover rate 6 month
	Daily turnover rate 12 month
Liquidity factors	Current ratio
	Quick ratio
Technical factors	MACD(10,30,15)
	RSI(20)
	PSY(20)
	BIAS(20)

Note: QTD means the latest quarter. TTM indicates the latest four quarters.

For the stocks of the A-Share market, stock returns will also be affected by market capitalization and industry in addition to the 60 factors listed in Table 2. For example, The EP factor of the banking industry's stock is larger than that of the internet industry. The purpose of neutralization is to eliminate the influence of other factors and make the selected stocks more dispersed. The main idea is to obtain a new feature that is independent of the original feature through multiple linear regression models. That is to say, through establishing a linear regression model, the residual value obtained by the regression is used as the new factor data.

Since features have different units and quantity sizes, such as market value and EP, they are obviously different in size. Substituting the features into the model directly will result in different proportions of different features, affecting the prediction results. This problem is solved by the following standardized methods:

$$x_{i,new} = \frac{x_i - u}{\sigma}$$

where $x_{i,new}$ is the value after data normalization and x_i is the value of the ith variable. u is the mean of the sequence and σ is the standard deviation of the sequence.

C. PARAMETER SETTING OF FEATURE SELECTION

1) SVM-RFE

When the SVM-RFE algorithm is applied for feature selection, it needs to select a certain number of features from the features listed in Table 1. Firstly, the importance of all features is sorted in descending order. Secondly, the top 80% of all features are selected, which means that there are 48 features selected.

2) RF

The RF algorithm is very similar to SVM-RFE, and both need to select a certain number of features. For consistency, the top 80% of the features are also selected.

D. PARAMETER SETTING OF STOCK PREDICTON

The stock price trend forecasting algorithms used in this paper include SVM, RF, and ANN. For each model, different parameters of the model will result in different results. In the field of machine learning, the common method of parameter setting is cross-validation. However, it cannot be applied in finance directly. For example, the operation steps of the 10-fold cross-validation method which is one of the most common cross-validation methods exhibit in Fig. 6. If the 10-fold cross-validation method is used in financial data, it is inevitable that future data are used to create model and predict previous data. Thus, in this paper, the time window slicing cross-validation strategy is applied [32]. Considering the application of 12 months of data for cross-validation, the training set data are divided into 12 equal groups, firstly as illustrates in Fig. 7, and then the first 4 groups of data are used as the training set and the next group of data as the validation set each time.

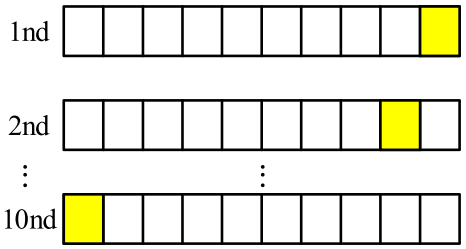


FIGURE 6. 10-fold cross-validation.

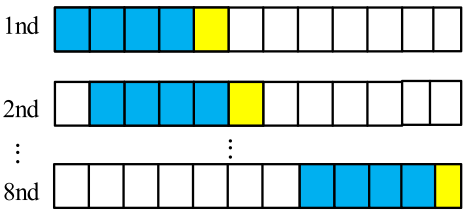


FIGURE 7. The time window slicing cross-validation.

For the SVM algorithm, the function of the Gaussian kernel function is to map data to infinite-dimensional space. Therefore, regardless of the data distribution, it can be mapped to infinite-dimensional space by Gaussian kernel function, which realizes the establishment of the high-dimensional linear model. In this paper, the Gaussian kernel function is used to build the SVM model. The optimization function is:

$$\max \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j \, label_i label_j K(x_i, x_j)$$

$$s.t.\ 0 \le \alpha_i \le C; \quad \sum_{i=1}^{n} \alpha_i \, label_i = 0$$

$$K(x_i, x_j) = \exp(-\gamma ||x_i - x_j||^2)$$

According to the formula, the parameters that need to be adjusted in the SVM model are mainly C and gamma (γ), which are shown in Table 3. For example, when C = 0.001 and gamma = 0.0001, the results obtained by the time window slicing cross-validation are used as evaluation results of the model. Through testing each set of C and gamma values, it can get the result for each model and finally select the best performing model.

In addition, when choosing the RF algorithm, there are some parameters that need to be set, such as the number of decision trees, the maximum number of features that each decision tree needs to consider when classifying, the minimum number of samples required for internal node subdivision, the minimum sample of leaf nodes and so on. In order to reduce the complexity of the model, the paper artificially defines some of these parameters. For example, the number of decision trees is 100. The maximum number of features that each decision tree needs to consider when classifying is the square of the total number of features. It marks the minimum number of samples required for internal node subdivision as "s" and the minimum sample of leaf nodes as "l". They are described in detail in Table 4.

TABLE 3. The parameters of SVM.

	0.001	0.01	0.1	1	10
0.0001	(0.001, 0.0001)	(0.01, 0.0001)	(0.1, 0.0001)	(1, 0.0001)	(10, 0.0001)
0.001	(0.001, 0.001)	(0.01, 0.001)	(0.1, 0.001)	(1, 0.001)	(10, 0.001)
0.01	(0.001, 0.01)	(0.01, 0.01)	(0.1, 0.01)	(1, 0.01)	(10, 0.01)
0.1	(0.001, 0.1)	(0.01, 0.1)	(0.1, 0.1)	(1, 0.1)	(10, 0.1)
1	(0.001, 1)	(0.01, 1)	(0.1, 1)	(1, 1)	(10, 1)

Note: the row data indicates the value of *C,* and the column data indicates the value of gamma.

TABLE 4. The parameters of RF.

	2	5	10	50	100
1	(2, 1)	(5, 1)	(10, 1)	(50, 1)	(100, 1)
5	(2, 5)	(5, 5)	(10, 5)	(50, 5)	(100, 5)
10	(2, 10)	(5, 10)	(10 10)	(50, 10)	(100, 10)
50	(2, 50)	(5, 50)	(10, 50)	(50, 50)	(100, 50)
100	(2, 100)	(5, 100)	(10, 100)	(50, 100)	(100, 100)

Note: the row data indicates the value of "s" and the column data indicates the value of "l".

TABLE 5. The parameters of ANN.

	0.00001	0.0001	0.001	0.01	0.1
100	(0.00001, 100)	(0.0001, 100)	(0.001, 100)	(0.01, 100)	(0.1, 100)
200	(0.00001, 200)	(0.0001, 200)	(0.001, 200)	(0.01, 200)	(0.1, 200)
500	(0.00001, 500)	(0.0001, 500)	(0.001, 500)	(0.01, 500)	(0.1, 500)
1000	(0.00001, 1000)	(0.0001, 1000)	(0.001, 1000)	(0.01, 1000)	(0.1, 1000)

Note: the row data indicates the value of "lbfgs" and the column data indicates the value of "max_iter".

TABLE 6. Confusion matrix.

	Actual = Positive	Actual = Negative	
Predicted =Positive	a	b	Precision = a/(a+b)
Predicted = Negative	c	d	
	Recall = a/(a+c)	FP= b/(b+d)	Accuracy = (a+d)/(a+b+c+d)

In this paper, the three-layer fully connected neural network is applied to predict the stock price trend. Compared with the other two algorithms, the algorithm has more parameters, such as the number of hidden layer neurons, the optimization function, L2 penalty (regularization term) parameter, the maximum number of iterations, and so on. The number of the first hidden layer neurons is set to 20, and then another is 10. The optimization function is ''lbfgs'' which is an optimizer in the family of quasi-Newton methods. It marks L2 penalty (regularization term) parameter as ''alpha'' and the maximum number of iterations as ''max_iter''. They are described in detail in Table 5.

E. OUT-SAMPLE TEST

The sliding window method, which has been widely used in stock price trend prediction, is applied to divide the sample into different groups of training and testing sets. The main reason for applying this method is that investors always pay attention to the recent trend of stocks, but not interested in data long ago. Therefore, the model needs to be continuously updated during the process of the application. In order to obtain the latest stock information as much as possible, the model is regenerated every month. For example, the training set of the first group is over the period from January 2010 to January 2011, while the testing set is over the period from January 2011 to February 2011. The specific division method is shown in Fig. 8.

F. MODEL EVALUATION

For the classification of the algorithm, there are several common evaluation indicators, such as accuracy, precision, and so on. They are calculated from a confusion matrix, as defined in Table 6.

Most people always like to measure the success of classifier tasks based on the correct rate, while accuracy is the most important indicator for evaluating the model. However, the method of calculating accuracy needs to manually set a threshold to achieve classification. The accuracy is greatly affected by this threshold, and so the paper further uses the AUC indicator to evaluate the model in this paper. AUC is calculated from the area covered by the ROC curve, which the x-axis is the FP, and the y-axis is the Recall.

In order to further evaluate the model, the paper constructs a strategy model and implements the back-testing of historical data. There are some indicators, as exhibits in Table 7, which are evaluated the strategy model [33]. The annualized return

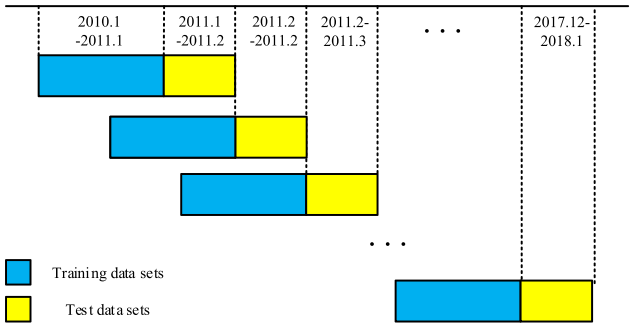


FIGURE 8. Sliding window by one month.

TABLE 7. Evaluation indicators.

	Formula
Annualized Return	$R_p = ((1+P)^{\frac{250}{n}} - 1) \times 100\%$
Sharpe Ratio	$Sharpe - Ratio = \frac{R_p - R_f}{\sigma_p}$
Win Rate	$Win - Rate = \frac{profitable\_transactions}{total\_transactions}$
Profit Loss Ratio	$profit - loss\ ratio = \frac{total\_profit}{total\_loss}$
Max Drawdown	$Max\_Drawdown = Max(P_x - P_y) / P_x$

refers to the return that can be obtained after one year of investment. The Sharpe ratio is an indicator used to measure both returns and risks of the model. The max drawdown is the maximum degree of loss that the model may have in a certain period of time in the past. In the case of investment, if there is a large max drawdown, it will often lead investors to lose confidence in the model. Therefore, the max drawdown of the reasonable control model is particularly important. It is especially important to control the max drawdown of the model reasonably.

Where R_p is the annualized return and P is the total return. n is the number of days the strategy is conducted and R_f is the risk-free rate. σ_p is the volatility of the return. P_x and P_y are the total value of stocks and cash on a certain day, and the requirement is $y > x$.

TABLE 8. Prediction accuracy of different models.

	None	SVM-RFE	RF
SVM	51.7287%	51.7710%	51.8295%
RF	52.9331%	52.7187%	52.7804%
ANN	52.3216%	52.3497%	52.3204%

Note: "None" means no feature selection.

TABLE 9. Prediction AUC of different models.

	None	SVM-RFE	RF
SVM	53.4145%	53.4035%	53.3552%
RF	53.7561%	53.5921%	53.5580%
ANN	52.8368%	52.6750%	52.5320%

IV. RESULTS

A. EMPIRICAL RESULTS AND DISCUSSION

In this paper, the procedure described in Section 3 is used for the experiment. The above steps are repeated every three months according to the method of the time window slicing, and some results were obtained and shown as follows.

First, the prediction accuracy and AUC of different integrated models are analyzed, which has been shown in Table 8 and Table 9. As can be seen from the two tables, the stock price trend forecast result is the best when adopting the RF model. Consider that the prediction performance of the three models is relatively close, the paper conducts a more in-depth analysis of these three models.

Second, the main job is to analyze the profitability of different integrated models. In order to further explain the

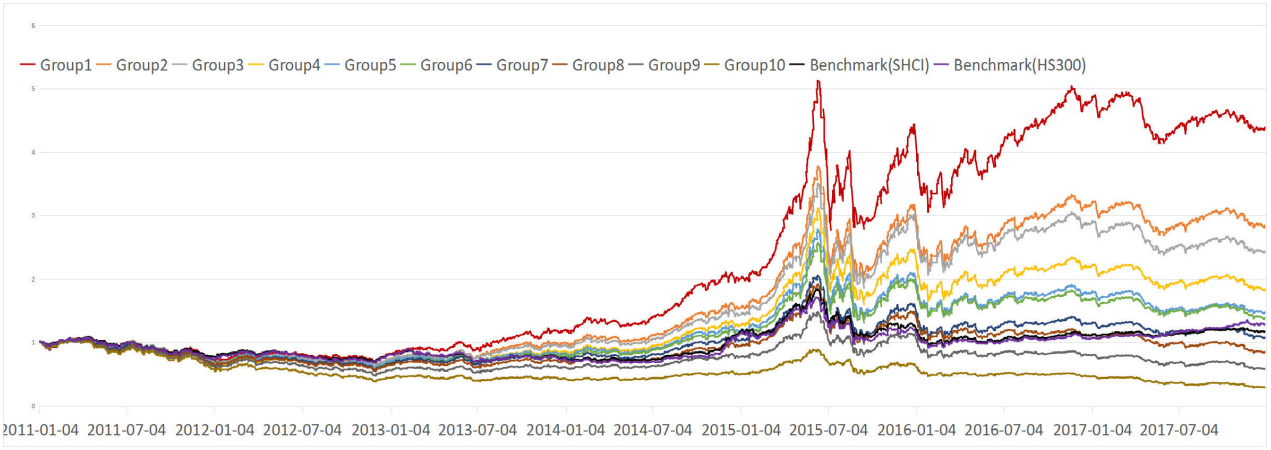


FIGURE 9. Hierarchical combined back-testing net value.

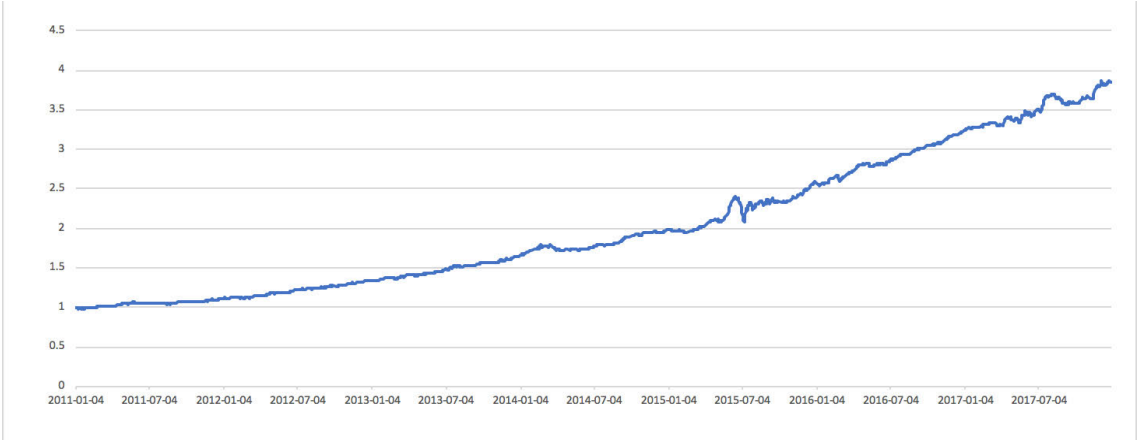


FIGURE 10. The new long-short portfolio net value.

TABLE 10. Annualized return of different models.

	1%			3%			5%		
	None	SVM-RFE	RF	None	SVM-RFE	RF	None	SVM-RFE	RF
APT	21.23%	19.36%	21.14%	22.51%	22.36%	22.95%	23.67%	23.38%	24.09%
SVM	25.21%	28.89%	28.61%	25.70%	26.57%	24.06%	23.90%	23.73%	22.52%
RF	26.34%	25.24%	**29.51%**	27.09%	27.33%	**27.54%**	24.62%	25.93%	**26.91%**
ANN	21.22%	23.25%	21.02%	21.33%	21.64%	18.51%	20.35%	22.15%	18.19%

TABLE 11. Sharp ratio of different models.

	1%			3%			5%		
	None	SVM-RFE	RF	None	SVM-RFE	RF	None	SVM-RFE	RF
APT	0.681	0.599	0.698	0.731	0.731	0.752	0.776	0.762	0.791
SVM	0.774	0.891	0.851	0.774	0.803	0.704	0.705	0.706	0.661
RF	0.83	0.774	**0.957**	0.847	0.859	**0.87**	0.755	0.809	**0.84**
ANN	0.648	0.746	0.648	0.654	0.68	0.548	0.615	0.697	0.531

application of the algorithm proposed in this paper, the APT model is used to predict the stock returns and achieve comparative analysis. The specific strategy is built as follows: the back-test time is from 2011.1 to 2018.1, and the stocks are traded at the end of each month. By ranking the probability that the A-share stocks are predicted to be "1", then the top 1%, 3%, or 5% of the stocks that are equally divided into the funds to buy are selected separately.

Table 10 shows the annualized return of different integrated models. The results demonstrate that the profitability of the SVM and RF models are better than APT, and regardless of whether using SVM or RF to predict stock price trends, the profitability is stronger than ANN. As the number of selected stocks decreases, the profitability of the model becomes stronger. Therefore, when choosing 1% of the total number of stocks, the higher returns can be obtained. The Sharpe ratio of the models is calculated in Table 11. Moreover, Table 12 and Table 13 shows the win rate and profit loss ratios of different models.

The above tables found that the best result can be obtained when selecting the features through RF and applying RF for stock price trend prediction. The RF-RF[1] integrated model is analyzed in detail as follows.

First of all, the specific strategy is built as follows: The back-test time is from January 2011 to January 2018, and the stocks are traded at the end of each month. By ranking the probability that the A-share stocks are predicted to be "1", all stocks are divided into 10 equal groups. For each group, all stocks that are equally divided into the funds to buy are selected. The next thing to do is to analyze the profitability of these ten groups.

[1]The RF-RF integrated model represents the random forest algorithm is used for both feature selection and stock price trend prediction.

TABLE 12. Win rate of different models.

	None	SVM-RFE	RF	None	SVM-RFE	RF	None	SVM-RFE	RF
APT	53.40%	54.30%	54.30%	54.30%	54.20%	54.30%	54.50%	54.30%	54.50%
SVM	57.20%	55.70%	56.40%	56.40%	56.10%	55.70%	55.80%	55.40%	55.30%
RF	55.80%	55.80%	**57.50%**	56.40%	56.00%	**56.80%**	56.10%	56.00%	**56.40%**
ANN	55.60%	56.00%	54.50%	54.90%	55.10%	54.30%	55.10%	55.30%	53.80%

TABLE 13. Profit loss ratio of different models.

	1%			3%			5%		
	None	SVM-RFE	RF	None	SVM-RFE	RF	None	SVM-RFE	RF
APT	1.58	1.504	1.592	1.584	1.576	1.59	1.598	1.596	1.602
SVM	1.653	1.639	1.699	1.582	1.624	1.579	1.556	1.553	1.555
RF	1.75	1.612	**1.802**	1.715	1.654	**1.715**	1.617	1.616	**1.669**
ANN	1.515	1.64	1.513	1.482	1.493	1.454	1.471	1.48	1.447

TABLE 14. Performance of different groups.

	Total Return	Annualized Return	Sharpe Ratio	Max Drawdown
Group1	339.21%	24.28%	0.744	45.95%
Group2	184.10%	16.58%	0.46	47.85%
Group3	143.87%	13.99%	0.37	46.77%
Group4	84.14%	9.38%	0.2	48.67%
Group5	47.38%	5.86%	0.069	50.87%
Group6	38.82%	4.94%	0.035	49.10%
Group7	8.38%	1.19%	-0.106	48.60%
Group8	-14.79%	-2.32%	-0.241	56.42%
Group9	-41.36%	-7.54%	-0.439	60.99%
Group10	-70.67%	-16.49%	-0.784	72.32%

Fig. 9 shows the net value of these ten groups over the period from January 2011 to January 2018. The red line represents the net trend of Group 1, while the purple and black lines represent the net value of HS300 and the Shanghai Composite Index. Table 14 shows the performance of different groups. Group 1 has the best results regardless of any evaluation indicators. In order to further illustrate profitability of the models, as shown in Table 15, the return for each year of the ten groups is calculated. Group 1 has the best performance compared to other models. Therefore, the RF-RF integrated model has strong profitability.

The above empirical results indicate that the RF-RF integrated model has the best performance in annualized return, Sharpe ratio, win rate and profit loss ratio. And the hierarchical combined back-testing also shows that the RF-RF integrated model has strong long-term predictability and profitability. At present, the proportion of quantitative investment in the field of financial investment is becoming larger and larger. Especially in recent years, with the poor market situation, traditional investment has been suffering a large loss in China stock market, and quantitative investment obtains stable income by controlling systematic risk, which makes quantitative investment products more and more trusted by investors. Based on the feature selection and machine learning algorithm, our empirical results find that the RF-RF integrated model can bring stable long-term return to investors

TABLE 15. Return of different combination in different years.

	2011	2012	2013	2014	2015	2016	2017
Group1	-26.23%	11.45%	44.29%	67.77%	118.68%	9.95%	-8.22%
Group2	-29.69%	5.49%	34.13%	54.53%	102.01%	1.54%	-9.90%
Group3	-29.36%	4.63%	28.83%	50.52%	107.15%	-3.75%	-14.66%
Group4	-33.78%	2.87%	26.16%	47.23%	92.16%	-10.16%	-15.70%
Group5	-31.37%	3.09%	17.29%	41.25%	75.73%	-13.89%	-16.90%
Group6	-31.64%	-0.81%	18.76%	39.14%	74.15%	-14.48%	-16.81%
Group7	-32.63%	-1.12%	15.13%	34.61%	52.39%	-17.80%	-16.20%
Group8	-33.23%	-3.09%	8.85%	34.04%	54.75%	-23.81%	-23.44%
Group9	-36.48%	-11.18%	8.52%	25.91%	43.76%	-28.26%	-26.24%
Group10	-41.52%	-24.25%	-2.73%	15.56%	30.02%	-28.73%	-36.45%

TABLE 16. Performance of the new portfolio.

Total Return	Annualized Return	Sharpe Ratio	Max Drawdown
285.37%	21.92%	2.86	13.58%

which is meaningful for guiding investment, as well as promoting investors' willingness to invest and improving the vitality of the capital market.

B. A LONG-SHORT TRADING STRATEGY BASED ON THE RF-RF INTEGRATED MODEL

Note that the Sharpe ratio of group 1 in Table 14 is 0.744 and the max drawdown is 45.95%. The main reason is that this model always holds stock fully at any time, so it is difficult to control the drawdown when the market falls substantially. However, when investors choose their investment strategy, they still have doubts about the relatively volatile return fluctuations.

The RF-RF integrated model in this paper helps investors in solving this problem by buying the stock of Group 1 and selling the stock of Group 10 at the same time. As can be seen from Fig. 10 and Table 16, the annualized return of this new long-short portfolio is 21.92% which is lower than the portfolio that selects the top 1% stocks, but the max drawdown of the new portfolio gets 13.58% which far below much lower than it, and the Sharpe ratio is 2.86, which is significant for investors.

V. CONCLUSION

This paper aims to analyze the profitability of various integrated stock selection models based on different feature selection and stock price trend prediction algorithms. The original features are filtered by feature selection methods. The time sliding window method is applied for cross-validation to determine the parameters of stock price trend prediction algorithms, which makes the model more practical in actual investment transactions. The empirical results show that the best performance can be obtained when the RF is applied for both feature selection and stock price trend forecasting. By selecting different stock numbers to build the model, it is also found that the RF-RF model has the highest return when it chooses top 1% of the stocks, achieving a 29.51% annualized return. The stratified back-testing method is used to further analyze the profitability of the RF-RF model, and the annualized return from 2011 to 2018 for the new long-short portfolio is 21.92% while the max drawdown is only 13.58%. Therefore, the RF-RF model is highly predictive of long-term stock price trends and can be used for guiding investment.

There are still some issues that need to be improved in this article: (1) no test in overseas markets such as US and UK, and (2) the feature selection algorithm still needs to be optimized, such as how to determine the number of features selected, and (3) there is still a need to continually explore more new features which have more predictability.

REFERENCES

[1] W. Huang, Y. Nakamori, and S.-Y. Wang, "Forecasting stock market movement direction with support vector machine," *Comput. Oper. Res.*, vol. 32, no. 10, pp. 2513–2522, Oct. 2005.

[2] C. Huang, D. Yang, and Y. Chuang, "Application of wrapper approach and composite classifier to the stock trend prediction," *Expert Syst. Appl.*, vol. 34, no. 4, pp. 2870–2878, May 2008.

[3] M.-C. Lee, "Using support vector machine with a hybrid feature selection method to the stock trend prediction," *Expert Syst. Appl.*, vol. 36, no. 8, pp. 10896–10904, Oct. 2009.

[4] Y. Kara, M. Acar Boyacioglu, and Ö. K. Baykan, "Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange," *Expert Syst. Appl.*, vol. 38, no. 5, pp. 5311–5319, May 2011.

[5] M. Ballings, D. Van Den Poel, N. Hespeels, and R. Gryp, "Evaluating multiple classifiers for stock price direction prediction," *Expert Syst. Appl.*, vol. 42, no. 20, pp. 7046–7056, Nov. 2015.

[6] D. Enke and S. Thawornwong, "The use of data mining and neural networks for forecasting stock market returns," *Expert Syst. Appl.*, vol. 29, no. 4, pp. 927–940, Nov. 2005.

[7] C.-F. Tsai and S.-P. Wang, "Stock price forecasting by hybrid machine learning techniques," in *Proc. Int. Multi-Conf. Eng. Comput. Scientists*, Mar. 2009, pp. 755–761.

[8] Y. Zuo and E. Kita, "Stock price forecast using Bayesian network," *Expert Syst. Appl.*, vol. 39, no. 8, pp. 6729–6737, Jun. 2012.

[9] E. Chong, C. Han, and F. C. Park, "Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies," *Expert Syst. Appl.*, vol. 83, pp. 187–205, Oct. 2017.

[10] B. Weng, L. Lu, X. Wang, F. M. Megahed, and W. Martinez, "Predicting short-term stock prices using ensemble methods and online data sources," *Expert Syst. Appl.*, vol. 112, pp. 258–273, Dec. 2018.

[11] A. Oztekin, R. Kizilaslan, S. Freund, and A. Iseri, "A data analytic approach to forecasting daily stock returns in an emerging market," *Eur. J. Oper. Res.*, vol. 253, no. 3, pp. 697–710, Sep. 2016.

[12] E. F. Fama and K. R. French, "Common risk factors in the returns on stocks and bonds," *J. Financial Economics*, vol. 33, no. 1, pp. 3–56, Feb. 1993.

[13] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques," *Expert Syst. Appl.*, vol. 42, no. 1, pp. 259–268, Jan. 2015.

[14] H. Y. Kim and C. H. Won, "Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models," *Expert Syst. Appl.*, vol. 103, pp. 25–37, Aug. 2018.

[15] V. Vapnik, "An overview of statistical learning theory," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 988–999, Sep. 1999.

[16] A. Fan and M. Palaniswami, "Stock selection using support vector machines," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2001, pp. 1793–1798.

[17] F. E. H. Tay and L. Cao, "Application of support vector machines in financial time series forecasting," *Omega*, vol. 29, no. 4, pp. 309–317, Aug. 2001.

[18] K.-J. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, nos. 1–2, pp. 307–319, Sep. 2003.

[19] R. Khemchandani, Jayadeva, and S. Chandra, "Knowledge based proximal support vector machines," *Eur. J. Oper. Res.*, vol. 195, no. 3, pp. 914–923, Jun. 2009.

[20] S. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, no. 3, pp. 660–674, Jun. 1991.

[21] Y. H. Cho, J. K. Kim, and S. H. Kim, "A personalized recommender system based on Web usage mining and decision tree induction," *Expert Syst. Appl.*, vol. 23, no. 3, pp. 329–342, Oct. 2002.

[22] V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, and B. P. Feuston, "Random forest: A classification and regression tool for compound classification and QSAR modeling," *J. Chem. Inf. Comput. Sci.*, vol. 43, no. 6, pp. 1947–1958, Nov. 2003.

[23] B. Lariviere and D. Vandenpoel, "Predicting customer retention and profitability by using random forests and regression forests techniques," *Expert Syst. Appl.*, vol. 29, no. 2, pp. 472–484, Aug. 2005.

[24] A. Prinzie and D. Van Den Poel, "Random forests for multiclass classification: Random multinomial logit," *Expert Syst. Appl.*, vol. 34, no. 3, pp. 1721–1732, Apr. 2008.

[25] M. Khashei and M. Bijari, "An artificial neural network (p, d, q) model for time series forecasting," *Expert Syst. Appl.*, vol. 37, no. 1, pp. 479–489, Jan. 2010.

[26] Q. Cao, M. E. Parry, and K. B. Leggio, "The three-factor model and artificial neural networks: Predicting stock price movement in China," *Ann. Oper. Res.*, vol. 185, no. 1, pp. 25–44, May 2011.

[27] R. CervellÓ-Royo, F. Guijarro, and K. Michniuk, "Stock market trading rule based on pattern recognition and technical analysis: Forecasting the DJIA index with intraday data," *Expert Syst. Appl.*, vol. 42, no. 14, pp. 5963–5975, Aug. 2015.

[28] W. You, Z. Yang, and G. Ji, "Feature selection for high-dimensional multi-category data using PLS-based local recursive feature elimination," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 1463–1475, Mar. 2014.

[29] X. Lin, F. Yang, L. Zhou, P. Yin, H. Kong, W. Xing, X. Lu, L. Jia, Q. Wang, and G. Xu, "A support vector machine-recursive feature elimination feature selection method based on artificial contrast variables and mutual information," *J. Chromatography B*, vol. 910, pp. 149–155, Dec. 2012.

[30] X.-Q. Hu, M. Cui, and B. Chen, "Feature selection based on random forest and application in correlation analysis of symptom and disease," presented at the Conf. IEEE Int. Symp. Med. Educ., Aug. 2009.

[31] D.-J. Yao, J. Yang, and X. J. Zhan, "Feature selection algorithm based on random forest," *J. Jilin Univ.*, vol. 44, no. 1, pp. 137–141, 2014.

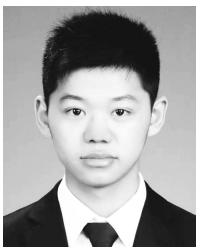
[32] X. Zhang, Y. Hu, K. Xie, S. Wang, E. Ngai, and M. Liu, "A causal feature selection algorithm for stock prediction modeling," *Neurocomputing*, vol. 142, pp. 48–59, Oct. 2014.

[33] J. Rohde, "Downside risk measure performance in the presence of breaks in volatility," *J. Risk Model Validation*, vol. 9, no. 2, pp. 31–68, Dec. 2015.

XIANGHUI YUAN received the B.Sc. degree in electrical engineering and its automation from Shaanxi Science and Technology University, Xianyang, China, in 2002, and the Ph.D. degree in control science and engineering from Xi'an Jiaotong University, Xi'an, China, in 2008.

From 2008 to 2018, he was a Faculty Member with the Department of Automation, Xi'an Jiaotong University. He is currently an Associate Professor with the Department of Financial Engineering, Xi'an Jiaotong University. His research interests include estimation and decision theory for stochastic systems, financial engineering, and machine learning for financial data analysis. He is also a CFA charter holder.

JIN YUAN was born in Anhui, China, in 1994. He received the B.S. degree from Northwest Polytechnic University, Xi'an, China, in 2015. He is currently pursuing the Ph.D. degree in applied economics with Xi'an Jiaotong University, Xi'an. His research interests include financial engineering, machine learning for financial data analysis, performance evaluation, and ranking.



TIANZHAO JIANG was born in Anhui, China, in 1993. He received the B.S. degree from Northwest University, Xi'an, China, in 2016, and the M.S. degree in control theory and science from Xi'an Jiaotong University, Xi'an, in 2019. He is currently working as a Researcher with Shanghai Foresee Investment Ltd., Liability Company. His research interests include information fusion, financial engineering, machine learning algorithm, and investment.

QURAT UL AIN received the master's degree in accounting and finance and the M.S. degree in finance from the University of Central Punjab, Lahore, Pakistan, in 2014 and 2017, respectively. She is currently pursuing the Ph.D. degree with the School of Economics and Finance, Xi'an Jiaotong University, China. Her areas of research interest are corporate finance, corporate governance, risk management, and technology innovation. Her current research work revolves around the governance role of women directors. Her research work has been published in very well-reputed journals.