Journal of Network Intelligence ©2024 ISSN 2414-8105
Taiwan Ubiquitous Information Volume **9**, Number **4**, November **2024**

Prediction of Financial Assets Income Based on Dynamic Weighted Ensemble Learning

Qing Yang*
Shaanxi Energy Institute, Xianyang 712000, P. R. China
16636066868@163.com

Yun-Peng Gong
Xi'an University of Architecture and Technology Huaqing College, Xi'an 710043, P. R. China
15229031702@163.com

Cai Gao
College of Management Science
University of Cyberjaya, Persiaran Multimedia, 63000 Cyberjaya, Selangor, Malaysia
uc3922@163.com

*Corresponding author: Qing Yang
Received December 21, 2023, revised March 21, 2024, accepted June 12, 2024.

Abstract. *The core of the asset return forecasting problem lies in accurately estimating future asset returns. However, volatility and uncertainty in financial markets complicate the forecasting of asset returns. Effective application of techniques such as statistics, data science and machine learning in asset return forecasting can improve the accuracy of the forecast. Therefore, this work proposes a method for predicting financial asset returns based on dynamic weighted ensemble learning. Firstly, an Arbitrage Pricing Theory (APT) model incorporating ensemble learning is proposed to address the shortcomings of traditional asset pricing models that cannot avoid high-dimensional problems when dealing with large quantities and complex statistical analyses. Then, an RVFL-Stacking model is proposed to mitigate the overfitting problem existing in traditional Stacking by introducing random weights. Next, the prediction of financial asset returns based on the RVFL-Stacking model is achieved by constructing a supervised regression task. Finally, the financial data of five higher education institutions were selected for return prediction. The data from February 2011 to February 2020 were selected based on data availability as well as reality. The experimental results show that the Sharpe ratio of the financial portfolio of the RVFL-Stacking model improves by about 41.2% based on the linear regression model.*
Keywords: Machine learning; Financial analysis; Stacking; Arbitrage Pricing Theory

1. **Introduction.** In finance, the prediction and interpretation of asset returns are key issues in the field of asset pricing. The key to the study of asset returns lies in the development of appropriate asset pricing models [1, 2]. Markowitz's mean-variance portfolio theory is the foundation of asset fixing, on which subsequent research models build [3, 4], using market risk to explain differences in cross-sectional returns of asset portfolios. By forecasting future returns on assets, investors can better optimise their portfolios to maximise expected returns [5]. Forecasting future returns also helps to reduce investment risk. Investors can better manage their portfolios by more accurately assessing the risk and return of assets. Forecasts of asset returns help to predict market volatility, enabling

firms to better develop risk management strategies and reduce the adverse effects of market fluctuations on financial performance [6]. Individuals and families can use forecasts of asset returns to develop more effective retirement planning and ensure adequate financial support at retirement. Firms can use asset return projections to assess the potential returns of different investment projects to make more informed investment decisions [7, 8]. Overall, financial asset return forecasting helps individuals, businesses and governments to make more informed decisions at different levels, optimise resource allocation, reduce risk and improve the effectiveness of financial planning.

The cross-fertilisation of artificial intelligence technology with other subject areas has been a big strategic direction nowadays. With the increasing development of computer technology and the growing amount of data, the improvement and application of machine learning algorithms have become a hot spot of academic research in recent years [9, 10, 11]. Machine learning models can extract relevant features from them based on labels and use the trained models to make predictions, and the redundancy between high-dimensional data has little effect on the training process of machine learning models. Redundancy has little effect on the training process of machine learning models. On this basis, machine learning becomes an effective alternative tool to address the challenges faced by traditional asset pricing. In the face of the reality that a large number of factors are constantly emerging and there are interactions among the factors, it is difficult for traditional linear models to aggregate effective pricing and return forecasting information from such data, while machine learning converts such a problem into a problem of learning the patterns among the data from the correlated data [12, 13]. Furthermore, the respective characteristics of machine learning and current asset pricing research make machine learning extremely advantageous for asset return forecasting. Machine learning can provide a large number of predictive function forms, both linear and nonlinear, providing a basis for achieving accurate predictions. Many machine learning methods such as tree models achieve nonlinear approximation by splitting the feature nodes.

While the performance of nonlinear models is currently debatable, deep learning models are superior in their ability to fit data [14]. In recent years, in the field of machine learning, ensemble learning models [15] have emerged from a wide range of models and have demonstrated their strong performance in industry and in various data competitions, showing results that can match deep learning in many tasks. Ensemble learning can reduce the risk of overfitting a single model by combining the predictions of multiple models. When training multiple different models and voting or weighted averaging across them, it reduces the over-reliance on training data and improves generalisation. Therefore, in order to effectively apply machine learning techniques in asset return forecasting, this work proposes a method for predicting financial asset returns based on dynamic weighted ensemble learning.

1.1. **Related Work.**

Ensemble learning is a popular direction of machine learning nowadays, and the application cases and related work in different scenarios can provide references for the research of financial asset return prediction.

In the field of machine learning and pattern recognition, pattern classification and regression problems are among the most important challenges. Pattern classification problems usually refer to classifying input data into different discrete classes, while pattern regression problems refer to mapping input data into a continuous output space. Ensemble learning is inherently favourable in improving the generalisation ability of learning models to efficiently solve real-world application problems. Therefore, the central concern is how to make the learning system have better generalisation ability. Therefore, the innovation of ensemble learning theory algorithms has been a popular area of machine learning in recent years. Based on feature selection and data classification, Hsieh et al. [16] proposed an ensemble learning algorithm that can be used for cancer diagnosis and scoring. Tajbakhsh and Suzuki [17] designed a computer-aided diagnostic system with multilevel feature selection and ensemble learning for lung nodule detection, and the use of the ensemble learning method effectively improved the diagnostic efficiency. Chen et al. [18] proposed an ensemble method for feature gene selection based on recursive classification trees, which not only has the ability to find disease-related genes, but also has strong data dimension compression capability. In the example, four pattern classification methods, such as support vector machine, confirmed that the method can significantly improve the accuracy of disease identification and classification. Yaman et al. [19] combined the ensemble learning technique with the task of face recognition, and carried out the research of multi-gesture, multi-ethnicity and multi-gender recognition, and the model used decision tree and Support Vector Machine (SVM). Tang et al. [20] also combined the ensemble learning technique with the multi-pose face recognition task, decomposed the prediction result of neural network into the weighted sum of multiple feature vectors, established multiple feature subspaces, trained a neural network for each feature subspace, and continuously combined another neural network into it, and achieved a better classification effect compared with the single neural network recognition.

Random weight neural network is a non-iterative training algorithm that randomly selects hidden weights and deviation values within a given range and keeps them constant throughout the training process, while the weights between the hidden and output layers are obtained by parsing. Compared with traditional BP-based training algorithms, random-weight neural networks can be trained faster with acceptable accuracy. Tang et al. [21] proposed a distillation feature approach to improve the interpolation performance of random-weight neural network models between training samples, i.e., to improve the generalisation ability. Compared with the previous work, this method enhances the understanding of the internal working mechanism and performs model optimisation, which makes random power neural networks more promising for machine learning applications. Hu and Suganthan [22] combined the Random Vector Functional Link (RVFL) in random power neural networks with ensemble learning, which on the training set RVFL is increasingly used in classification and prediction problems due to its better performance.

1.2. **Motivation and contribution.** The core of the asset return forecasting problem lies in accurately estimating the future rate of return on assets. However, volatility and uncertainty in financial markets complicate the forecasting of asset returns. Effective application of techniques such as statistics, data science, and machine learning in asset return forecasting can improve the accuracy of forecasting. Therefore, this work proposes a method for predicting financial asset returns based on dynamic weighted ensemble learning. The main innovations and contributions of this work include:

(1) The Arbitrage Pricing Theory (APT) [23] model incorporating ensemble learning is proposed to address the shortcomings of traditional asset pricing models that are unable to avoid high-dimensional problems when dealing with large quantities and complex statistical analyses.

(2) In the traditional Stacking model, the prediction results of the base learner are used as inputs to the meta-learner during the training process, which may lead to overfitting of the training data by the base learner. Therefore, this work proposes an RVFL-Stacking model that mitigates the overfitting problem by introducing random weights.

(3) The prediction of financial asset returns based on the RVFL-Stacking model is achieved by constructing a supervised regression task. The model input data are 135

pricing factors, while the output variable is the monthly return considering the actual financial strategy.

2. Ensemble learning theory.

2.1. Fundamental principle.

Ensemble learning is one of the most advanced, effective, and researched areas of machine learning, which improves the performance by learning from multiple weak learners and helps to learn features from large amounts of data. The basic idea of ensemble learning is to combine models with different features and then weight these models according to the prediction results. In ensemble learning, when training a classifier, each classifier is first trained as a small independent task. Each classifier learns a feature representation (e.g., number of categories and probabilities). Each classifier has its own weights (e.g., category number and probability), which are computed from the probabilities over the feature representation.

The idea of ensemble learning is to use integration to integrate processing of multiple single classifiers when training the model, common single classifier models mainly include Multilayer Perceptron (MLP) [24], Decision Tree Classifier (DTC) [25], K-Nearest Neighbors (KNN) [26], etc. The final classification category is determined by combining the classification results of multiple single classifiers to obtain a better performance improvement over a single classifier. The ensemble learning methods are mainly classified into three categories; Bagging, Boosting and Stacking. Bagging provides better variance reduction, Boosting provides better bias reduction and Stacking provides better prediction enhancement.

2.2. Generation method for weak learners.

(1) Bagging

As one of the simplest integration strategies currently available, bagging obtains a number of training subsets based on the original training set by bootstrap sampling method, and then a single classifier is selected to be trained on each of the training subsets, and multiple base classifiers are obtained through training [27]. One classification result is obtained for each base classifier, and finally a minority-majority vote is used to determine the final result of the output. For the sample data to be classified, the sample data is fed into the base learners obtained from the training phase. Each base learner separately discriminates the output for the category to which the samples to be classified belong. The outputs of all base learners are voted and the category with the most votes is the finalised category.

(2) Boosting

Each round of learning in the Boosting algorithm [28] adjusts the parameters based on the data to continuously improve the accuracy of the model. It first generates a weak learner based on the training samples, and then adjusts the sample distribution around the performance strengths and weaknesses of the weak learner, i.e., increasing the weights of samples that are detected incorrectly, so that more attention is paid to them subsequently. After adjusting the weighted training set, it continues to generate the next level of weak learners, and keeps cycling through this process until a certain number of weak learners are generated, and finally synthesises the outputs of these multiple weak learners based on some kind of combining strategy.

(3) Stacking

The Stacking algorithm [29] uses several different base models combined together and another model (meta-model) to perform a weighted average of their outputs to obtain more accurate predictions. It works as follows: in the first layer, M weak learners are generated using the training set; in the second layer, the weak learners of the first layer are used to generate predictions for some of the test sets by generalising the output predictions for training the secondary weak learners.

The data combination in the Stacking training process uses the weighted average probability method, which takes into account the predicted value for each category relative to the majority voting method and is suitable for use with a large number of base learners. For n base learners, m classification categories, and x is the financial dataset, the discriminative result of the i-th base learner can be expressed as:

$$C_i(x) = [p_{i1}, p_{i2}, p_{i3}, \ldots, p_{im}] \tag{1}$$

The selection of a suitable ensemble learning strategy based on the data type can avoid the occurrence of problems such as low prediction accuracy, overfitting, or underfitting of the learner to a certain extent, thus improving the overall performance of the learner. Stacking is able to reduce the risk of overfitting compared to Bagging and Boosting. Stacking does this by introducing a meta-model (or second-layer model) which is trained based on the output of the first-layer base model. This hierarchical structure helps to reduce the risk of overfitting, which is especially important in financial data, which typically has a high level of noise and uncertainty. In addition, Stacking provides greater flexibility to adjust different base models and meta-models according to specific financial data characteristics and demand choices. This flexibility is especially important for adapting to the complex and changing financial market environment. Therefore, this paper adopts Stacking to implement an ensemble learning model.

2.3. **Combining strategies.** For a classification or prediction task, suppose that T weak learners are included, where the predicted output of the weak learner h_i on sample x is an M-dimensional vector $\left(h_i^1(x), h_i^2(x), ..., h_i^M(x)\right)$, where $h_i^j(x)$ is the predicted output of h_i on the category label c_j. The voting method is a common ensemble learning combination approach that adopts the majority decision principle. Ideally, the voting method performance is no less than the classification performance of any of the base models.

(1) Absolute majority vote:

In the absolute majority voting method, each classifier votes for each disease category, and the final output predicted categories are those that receive more than 50% of the votes; if none of the category labels receives more than 50% of the votes, no classification result is output. The ensemble output category markers $H(x)$ can be expressed as:

$$H(x) = \begin{cases} c_j, & \text{if} \sum_{i=1}^{T} h_i^j(x) > 0.5 \sum_{k=1}^{M} \sum_{i=1}^{T} h_i^k(x) \\ \text{reject}, & \text{otherwise} \end{cases} \tag{2}$$

(2) Relative majority voting:

The category with the highest number of votes is output as the final classification result, and if more than 2 categories all receive the highest number of votes at the same time, one of them is randomly selected as the final prediction result. The ensemble output category labeling $H(x)$ can be expressed as:

$$H(x) = c_{\arg\max} \sum_{i=1}^{T} h_i^j(x) \tag{3}$$

(3) Weighted voting:

Weighted voting is a voting technique in which the weights are proportionally shared. The number of votes for a weak learner is multiplied by the weight coefficient to get the "weighted votes" for that learner, and the "weighted votes" of the same category are summed up, and the final category is the one that corresponds to the maximum number of votes. The final output is the category corresponding to the maximum value of votes. The ensemble output category token $H(x)$ can be expressed as follows:

$$H(x) = c_{\arg\max} \sum_{i=1}^{T} w_i h_i^j(x) \tag{4}$$

where w_i is the weight of h_i, $w_i \geq 0$, and $\sum_{i=1}^{T} w_i = 1$.

In the financial asset return prediction task, different kinds of weak learners with different tasks to fulfill may produce different types of $h_i^j(x)$. When using category-labelled voting, $h_i^j(x) \in \{0, 1\}$. If the class c_j predicted by the sample is 1 or 0, the vote at this point is a hard vote. When voting with class probability, $h_i^j(x) \in (0, 1)$, is an estimate of the posterior probability $P(c_j|x)$, the vote at this point is a soft vote. As mentioned above, the data combination in the Stacking training process uses the weighted average probability method, which takes into account the predicted values for each class relative to the majority voting method, and is suitable for use in cases where the number of base learners is large.

3. Asset pricing models based on ensemble learning.

3.1. Capital Asset Pricing Model (CAPM).

The CAPM is an important model used in finance to estimate the risk and expected return of an asset [30]. It is a central component of modern financial portfolio theory and was developed independently by William Sharpe, John Lintner and Jan Mossin in the 1960s.

The basic idea of the CAPM is that the expected return on an asset can be estimated by taking into account the risk-free rate of return, the systematic risk of the asset (market risk), and the excess return of the market as a whole (the market return minus the risk-free rate of return). The model is based on several key assumptions, including that financiers are risk averse, capital markets are perfectly competitive and informationally efficient, and that all financiers have the same expected rate of return and financial time horizon.

The mathematical expression for CAPM is given below:

$$E(R_i) = R_f + \beta_i \times (E(R_m) - R_f) \tag{5}$$

where $E(R_i)$ is the expected return on asset i, R_f is the risk-free rate of return (usually proxied by the yield on treasury bonds or other government-guaranteed bonds), β_i is the coefficient on asset i (reflecting the sensitivity of the asset's yield to market yields), $E(R_m)$ is the expected return on the market, and R_f is the risk-free rate of return.

The CAPM is widely used in areas such as asset pricing, estimation of the cost of capital, financial portfolio management and financial decision-making. Although the model has been criticised for the mismatch between its simplifying assumptions and actual market conditions, it remains an important tool for understanding and analysing the risk-return relationship in capital markets.

3.2. Arbitrage Pricing Theory (APT).

APT is a financial theory used to estimate the expected return of an asset [31]. The core idea of APT is that the expected return of an asset can be explained by a combination of a set of fundamental factors, not just market risk factors.

Unlike the CAPM, the APT assumes that the expected return on an asset is influenced by multiple factors, not just market risk factors. These factors can be macroeconomic variables (e.g., inflation rates, interest rate levels), industry-specific variables (e.g., industry growth rates, price-earnings ratios), or other factors that are correlated with asset returns. The core assumption of the APT is that asset returns are affected by a linear combination of multiple factors.

The mathematical expression for APT is as follows:

$$E(R_i) = R_f + \beta_{i1}F_1 + \beta_{i2}F_2 + ... + \beta_{in}F_n + \epsilon_i \tag{6}$$

where β_{ij} is the factor loadings (sensitivity of asset i to factor j), F_j is the risk premium for factor j, and ϵ_i is the idiosyncratic risk.

The portfolio return matrix consisting of multiple assets is denoted as:

$$\mathbf{R} = \mathbf{U} + \mathbf{F}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{7}$$

It is possible to convert the expected return to a weighted form of $\mathbf{I}$ and $\boldsymbol{\beta}$.

$$\mathbf{U} = \theta_1\mathbf{I} + \theta_2\boldsymbol{\beta} \tag{8}$$

The equation shows that the expected returns of the assets can all be expressed as compensation for their exposure to the various factors.

3.3. **APT model incorporating ensemble learning.** From the description of the APT model, it can be seen that a condition that must be satisfied for the APT model to hold is that there is a linear relationship between the expected return of the asset and the factors. This requires avoiding high-dimensional problems as much as possible when dealing with large quantities and complex statistical analyses.

The biggest advantage of artificial intelligence is its powerful feature prediction and analysis performance, so it is necessary to explore the APT asset pricing problem combined with nonlinear ensemble learning models. The representation of the APT model incorporating ensemble learning is shown below:

$$R_{t+1} = \alpha + \beta H_t + \epsilon \tag{9}$$

where H_t denotes that it is computed by the nonlinear ensemble learning model from the input stock features.

As an example, the Stacking algorithm used in this study can be represented as follows:

$$H_t = H_k(x_i) \tag{10}$$

where the functional relationship $H_k(\cdot)$ represents the mapping of the input $x_{i,t}$ to the final output.

The relationship between the layers of weak classifiers can be expressed as:

$$H_k(x_i) = H_{k-1}(x_i) + f_k(x_i), \quad k = 1, 2, ..., T \tag{11}$$

It can be seen that the ensemble learning model takes more account of the non-linear relationship between features, takes full advantage of the information contained in the high-dimensional dataset thereby improving the prediction accuracy, which in turn improves the factor Sharpe ratios and enhances the pricing ability of the model. Ensemble learning can improve the overall prediction accuracy by combining multiple individual prediction models. In the APT model, by constructing multiple individual predictive models using multiple underlying factors and factor loadings and combining them, changes in asset returns can be explained more comprehensively, thus improving prediction accuracy.

Ensemble learning improves model robustness, i.e. robustness to noise and outliers. By combining the prediction results of multiple models, the errors of individual models can be balanced, thus reducing the impact of outliers on the overall prediction. Taken together, the above discussion shows that the application of ensemble learning to asset pricing research is theoretically sufficient and necessary.

4. **Dynamically weighted Stacking ensemble learning model.**

4.1. **Random weight neural networks.** Common stochastic weight neural networks are Extreme Learning Machine (ELM) and Random Vector Functional link (RVFL).

ELM is a machine learning algorithm for a single hidden layer feed-forward neural network, where output weights can be computed analytically by randomly generating the parameters of the hidden units. Compared to the complex process of adjusting parameters in the BP algorithm, the ELM model does not require much repetitive training, and thus the training speed of the model is faster than that of the BP algorithm model. The ELM algorithm has a fast model training speed and is also better in terms of generalisation performance, but the shortcomings are that the accuracy is slightly lacking.

The RVFL network is built based on Functional link network [32]. The inputs to the output layer in RVFL have both linear original input features X from the input layer and nonlinear transformed features H from the hidden layer. Let d be the input data features and N be the number of hidden nodes, then each output node has $d+N$ inputs. This feature solves the problem of coexistence of linear and nonlinear relationships in the sample data. Since the hidden layer parameters are randomly generated and kept constant during the training phase, only the output weights β need to be computed. Because the input and output layers are directly connected, this network layout is more resistant to network overfitting. The structure of the RVFL is shown in Figure 1.

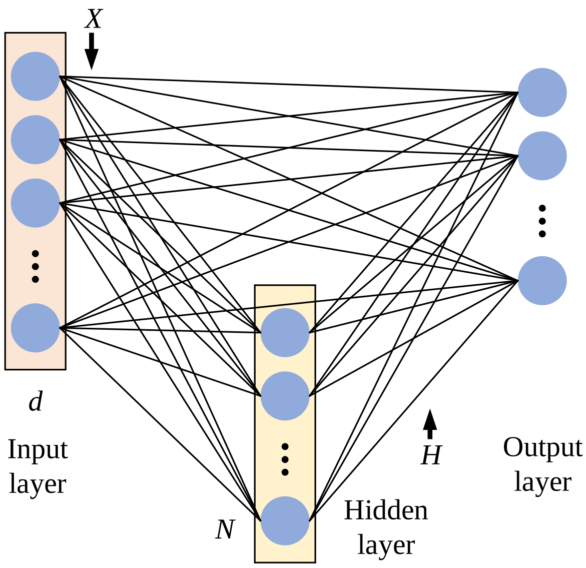

Figure 1. RVFL Network

RVFL network is a feed forward neural network which is based on the structure of a single layer feed forward neural network but with the addition of direct connections between the hidden and output layers. The key feature of RVFL network is that the weights and biases of its hidden layer nodes are randomly initialised and remain constant during the training process. The mathematical representation of the RVFL network can be presented in the following steps:

(1) Network structure: Suppose there is an RVFL network containing d input nodes, L hidden nodes and m output nodes.

(2) Input to the hidden layer: Let the input vector be $\mathbf{x} \in \mathbb{R}^d$, the weight matrix of the hidden layer be $\mathbf{W} \in \mathbb{R}^{L\times d}$ and the bias vector be $\mathbf{b} \in \mathbb{R}^L$. The output of the hidden layer $\mathbf{H} \in \mathbb{R}^{L\times N}$ (for N samples) can be expressed as:

$$\mathbf{H} = g(\mathbf{WX} + \mathbf{b}) \tag{12}$$

where $g(\cdot)$ is the activation function, commonly used activation functions include Sigmoid, ReLU and so on.

(3) Hidden Layer to Output Layer: The weight matrix of the output layer is $\boldsymbol{\beta} \in \mathbb{R}^{(L+d)\times m}$. The output of the RVFL network $\mathbf{Y}$ can be expressed as:

$$\mathbf{Y} = \boldsymbol{\beta}^T[\mathbf{H}; \mathbf{X}] \tag{13}$$

where $[\mathbf{H}; \mathbf{X}]$ denotes splicing the hidden layer output with the original input.

(4) Training Process: During the training process, the hidden layer weights $\mathbf{W}$ and bias $\mathbf{b}$ are randomly initialised and remain unchanged during the training process. The goal of training is to adjust the output layer weights $\boldsymbol{\beta}$ by minimising the output error. Typically, this can be achieved by minimising a loss function such as the Mean Square Error (MSE):

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(v_i - Y_i)^2 \tag{14}$$

where v_i, the target output of the i-th sample.

(5) Optimisation: The output layer weights $\boldsymbol{\beta}$ are usually solved by least squares or other optimisation algorithms.

4.2. **Logistic regression.** Logistic regression is a classic classification algorithm, through the training data in the positive and negative examples, learning sample features, to get and labels between the hypothesis function, logistic regression due to its algorithmic complexity is low, the effect is good, commonly used in classification prediction. The hypothesis function needs to use the sigmoid function, whose mathematical form is:

$$g(x) = \frac{1}{1 + e^{-x}} \tag{15}$$

In a logistic regression model, the function needs to be restricted to a certain condition which determines the assumption space of the model and the assumptions made by logistic regression are:

$$p(y = 1|x; \boldsymbol{\theta}) = g(\boldsymbol{\theta}^T x) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T x}} \tag{16}$$

where $g(\boldsymbol{\theta}^T x)$ is the sigmoid function.

4.3. **RVFL-Stacking.** Traditional Stacking model where the predictions of the base learner are used as inputs to the meta-learner during the training process, which can lead to the base learner overfitting the training data. The base learner performs well on the training set but has poor generalisation ability on unknown data. To address this problem, this work proposes an RVFL-Stacking model that mitigates the overfitting problem by introducing random weights, as shown in Figure 2.

When training the base learner on a subset of each training set, the RVFL-Stacking model uses random weights to reduce the base learner's dependence on specific training samples, thereby improving its generalisation ability. In addition, the RVFL-Stacking model based on randomly weighted neural networks helps to mitigate the information leakage problem that exists in traditional Stacking models by randomly selecting a subset of the training set during the training process so that the meta-learner does not have access to the complete base-learner prediction results during training.

The RVFL-Stacking model uses random-weight neural networks as the base learner and logistic regression as the meta-learner. The financial data was first divided into a training set and a test set, with 75% for the training set and 25% for the test set. Using the training set data in the random weight neural network for training, the prediction result of each RVFL network is used to get the final classification discrimination result by

Stacking integration strategy, and the trained model is tested on the test set to evaluate the model classification effect. In this paper, the model is used as a model for making predictions on financial asset returns.

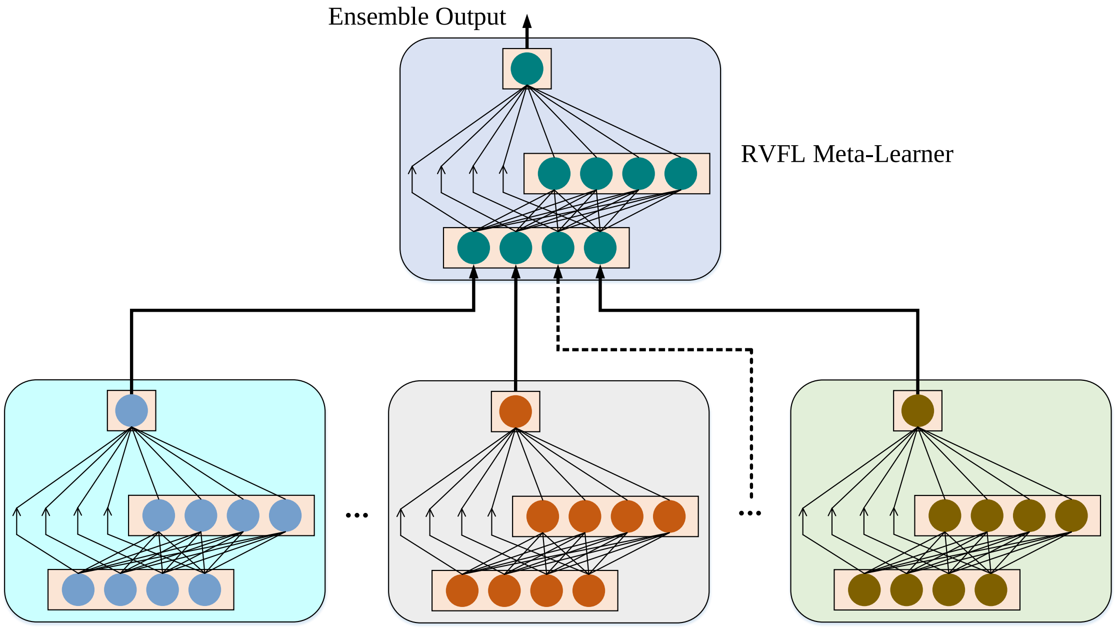

Figure 2. Structure of RVFL-Stacking

The pseudo-code for the RVFL-Stacking model is shown in Algorithm 1.

Algorithm 1 RVFL-Stacking Model

Input: Training set $(X_{\text{train}}, Y_{\text{train}})$, Test set (X_{test})

Output: Final prediction result Y_{final}

```
Set the parameters of the RVFL network (Regularization parameter λ, etc.)
Define the training function for the RVFL network Train_RVFL(X_train, Y_train)
Randomly initialize W and b for the hidden layer
H = g(X_train * W + b)
β = Y_train * H^T * inv(H * H^T + λ * I)
Return the output layer weights β
Define the prediction function of the RVFL network Predict_RVFL(X_test, β)
H = g(X_test * W + b)
Y_pred = H * β
Return prediction output Y_pred
Define the training function Train_Stacking(X_train, Y_train, models)
Initialize the input matrix of the meta-learner H_meta
for each model m in models do
    Predict the training set X_train using model m and obtain the output Y_pred_m
    Add Y_pred_m to H_meta as input to the meta-learner
end for
Train meta-learner using H_meta and Y_train to obtain weights β_meta
Return the weight of the meta-learner β_meta
Define the prediction function Predict_Stacking(X_test, models, β_meta)
Initialize the input matrix of the meta-learner H_meta_test
```

for each model m in models **do**
  Use model m to predict the test set X_{test} and get the output $Y_{\text{pred\_m\_test}}$
  Add $Y_{\text{pred\_m\_test}}$ to $H_{\text{meta\_test}}$ as input to the meta-learner
end for
$Y_{\text{pred\_final}} = H_{\text{meta\_test}} * \beta_{\text{meta}}$
Return predicted output $Y_{\text{pred\_final}}$
Train RVFL network using Train_RVFL function to obtain output layer weights β_{rvfl}
Train Stacking model using Train_Stacking function to obtain weights β_{meta}
Predicting output of an RVFL network using Predict_RVFL function and β_{rvfl}, Y_{rvfl}
Predicting Y_{stacking} of Stacking model using Predict_Stacking, models, β_{meta}
Dynamically weighted fusion of Y_{rvfl} and Y_{stacking} yields final prediction Y_{final}
Return final prediction result Y_{final}

5. Projections of return on financial assets based on the RVFL-Stacking model.

5.1. **Pricing factor selection.** In this paper, the financial data of five higher education institutions are selected for revenue forecasting. The data from February 2011 to February 2020 are selected based on data availability as well as reality. The pricing factor data selected in this paper contains a total of 135 factors in nine categories, such as quality factor, basis factor, sentiment factor, etc., and this part of the data will be used as the original data for model input.

The output variable selected in this paper is the monthly return considering the actual financial strategy (buy at the beginning of the month at the opening price and sell at the end of the month at the closing price). This is done by making the factor data of the current month as the model input and making the monthly return of the following month as the output, i.e., the time t risk factor data and the t+1 period monthly return are matched to become a sample.

5.2. **Data Preprocessing.** The raw data need to be preprocessed because of the influence of the magnitude, outliers, and industry market value of each factor. In this paper, Z-score standard method is chosen to solve the problem of outliers and different magnitudes of factor data.

Standardisation is a very common data preprocessing method in machine learning applications, and standardised data can make it easier for machine learning models to converge. Standardised data is more conducive to the convergence of machine learning models, but also to avoid the impact of model bias caused by different factors such as market capitalisation and net profit and revenue share. The specific operation is as follows:

$$Z_i = \frac{x_i - \bar{x}}{s} \tag{17}$$

where Z is the standardised score, x_i is the factor data, $\bar{x}$ is the standard value of the factor, and s is the standard deviation of the factor data.

5.3. **Definition of the prediction function.** In this work, the financial assets of higher education institutions are selected as the object of study, and the ensemble learning model (RVFL-Stacking) is used to aggregate the pricing factors to predict the future returns of the assets.

The asset return forecasting task in this paper is defined as a supervised regression task, which is presented as:

$$R_{it} = f\left(x_{i(t-1)}; \theta\right) + \epsilon_{it} \tag{18}$$

Where $f(\cdot)$ denotes the prediction function of the RVFL-Stacking ensemble learning model with parameter θ, R_{it} denotes the return of asset i at time t, $x_{i(t-1)}$ denotes the pricing factor of asset i at time $t-1$, and ϵ_{it} denotes the error term.

After determining the specific prediction model, it is necessary to construct the training method of the model. Since there is a before-and-after time relationship between the cross-sectional data of financial assets, the KFold cross-validation commonly used in machine learning certain data leakage problems, so this paper uses a sliding window approach to divide the data set. Since this paper simulates the financial situation at a real monthly frequency, positions need to be adjusted at the end of each month by selling assets that have been held for one month. The portfolio of financial assets selected by the model is bought and this operation is repeated at the end of the next month.

Compared with the KFold cross-validation method, this paper uses a sliding window to divide the data, which not only ensures that there is no risk of data leakage in the chronological order, but also ensures that it can be consistent with the actual financial process.

6. Experimental results and analyses.

6.1. Experimental environment and dataset. All prediction models in the experiments were run under the same computer configuration (Processor: 11th Gen Intel(R) Core(TM) i5-1135G7@2.40GHz 2.42GHz, Memory: 16.0GB, Windows version: 21H2, OS version: 22000.978)

In this paper, the financial data of three higher education institutions are selected for revenue forecasting. Data from February 2011 to February 2020 were selected based on data availability as well as reality. At the end of each month, the financial data are classified into 12 portfolios according to the model prediction results. The variability on the pricing factors of different dimensions was statistically analysed, where for each indicator the mean value of the indicator for each period was calculated first, and then averaged for the whole time t. The portfolios constructed by the RVFL-Stacking ensemble learning model were compared with the portfolios of the traditional linear model, XGBoost model, LightGBM model, and Catboost model.

6.2. Comparative analysis of the returns of the financial portfolios. The analysis of the returns obtained from the financial combinations of the four forecasting models is shown in Table 1.

Table 1. Comparison of Portfolio Returns for Different Forecasting Models

	RVFL-Stacking	**CatBoost**	**LightGBM**	**XGBoost**	**Linear**
Combination 1	0.0228	0.0217	0.0187	0.0165	0.0159
Combination 2	0.0182	0.0183	0.0177	0.0164	0.0154
Combination 3	0.0167	0.0162	0.0166	0.0153	0.0158
Combination 4	0.0151	0.0146	0.0146	0.0145	0.0139
Combination 5	0.0124	0.0126	0.0128	0.0128	0.0134
Combination 6	0.0109	0.0111	0.0119	0.0117	0.0120
Combination 7	0.0089	0.0084	0.0096	0.0105	0.0110
Combination 8	0.0067	0.0061	0.0072	0.0076	0.0079
Combination 9	0.0155	0.0134	0.0074	0.0063	0.0081
Combination 10	0.0090	0.0089	0.0073	0.0068	0.0073
Combination 11	0.0078	0.0058	0.0078	0.0068	0.0093
Combination 12	0.0068	0.0073	0.0076	0.0088	0.0087
Average value	0.0126	0.0120	0.0116	0.0112	0.0116

From Table 1, it can be seen that the ensemble learning model is generally higher than the return of the linear model, indicating that there is a non-linear relationship between the high-dimensional pricing factors and the expected return of the assets, and the ensemble learning model can better extract the non-linear information in it. Among them, based on the APT arbitrage pricing theory, the RVFL-Stacking model effectively improves the performance of the financial portfolio. Compared with other ensemble learning models, the RVFL-Stacking model has a higher mean value of returns for the financial portfolio.

The Sharpe ratios for the different forecasting models are shown in Figure 3.

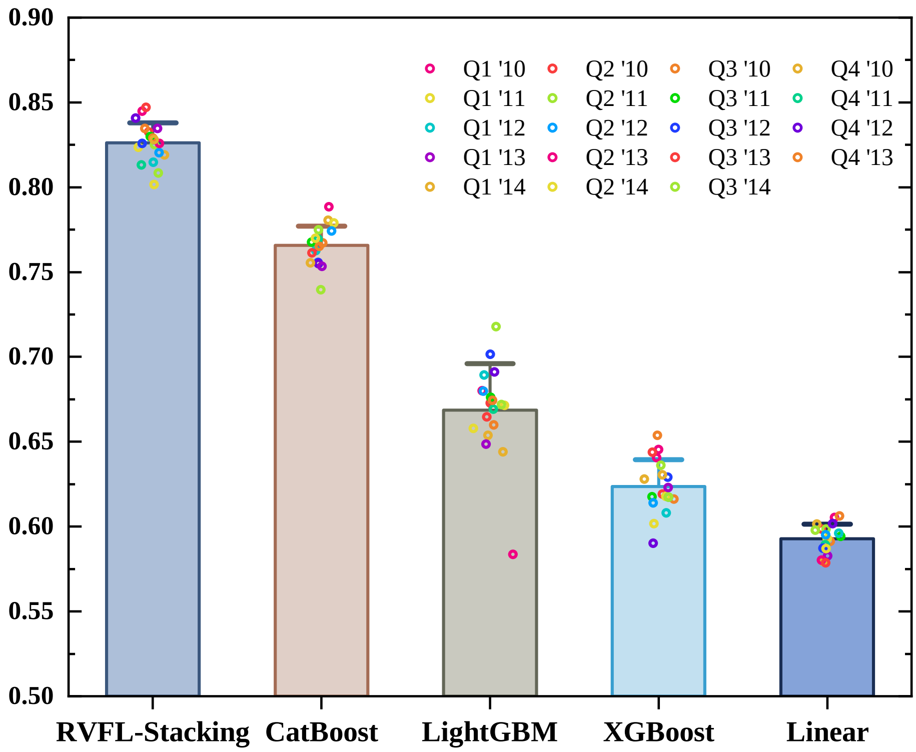

Figure 3. Sharpe ratio comparison

The best performance of the RVFL-Stacking model can be seen. Based on the linear regression model, the RVFL-Stacking model improves the Sharpe ratio of the financial portfolio by about 41.2%. The other three ensemble learning models do not improve as much as RVFL-Stacking, but they also still have a large improvement relative to the linear regression model.

7. **Conclusion.** In this work, an RVFL-Stacking based approach for predicting financial asset returns is proposed. Stacking is used to implement an ensemble learning model and an APT model incorporating ensemble learning is proposed. An RVFL-Stacking model is proposed to mitigate the overfitting problem of traditional single Stacking by introducing stochastic weights. Financial asset return prediction based on the RVFL-Stacking model is achieved by constructing a supervised regression task. The model input data are 135 pricing factors while the output variables are monthly returns considering actual financial strategies. A return prediction study was conducted using financial data from five higher education institutions and the experimental results showed that the Sharpe ratio of the financial portfolio of the RVFL-Stacking model was improved by about 41.2 per cent on the basis of the linear regression model.

Acknowledgement. This work supported by the key topics of humanities and social sciences for school-level youth in Shaanxi Energy Vocational and Technical College (No. 23QNRWZ02).

REFERENCES

[1] D. Avramov, and T. Chordia, “Predicting stock returns,” Journal of Financial Economics, vol. 82, no. 2, pp. 387-415, 2006.

[2] J. Kumari, and J. Mahakud, “Does investor sentiment predict the asset volatility? Evidence from emerging stock market India,” Journal of Behavioral and Experimental Finance, vol. 8, pp. 25-39, 2015.

[3] L. Forsberg, and E. Ghysels, “Why do absolute returns predict volatility so well?,” Journal of Financial Econometrics, vol. 5, no. 1, pp. 31-67, 2007.

[4] M. Guidolin, S. Hyde, D. McMillan, and S. Ono, “Does the Macroeconomy Predict UK Asset Returns in a Nonlinear Fashion? Comprehensive Out-of-Sample Evidence,” Oxford Bulletin of Economics and Statistics, vol. 76, no. 4, pp. 510-535, 2014.

[5] R. M. Sousa, A. Vivian, and M. E. Wohar, “Predicting asset returns in the BRICS: The role of macroeconomic and fundamental predictors,” International Review of Economics & Finance, vol. 41, pp. 122-143, 2016.

[6] A. Moghar, and M. Hamiche, “Stock market prediction using LSTM recurrent neural network,” Procedia Computer Science, vol. 170, pp. 1168-1173, 2020.

[7] A. Brav, R. Lehavy, and R. Michaely, “Using expectations to test asset pricing models,” Financial Management, vol. 34, no. 3, pp. 31-64, 2005.

[8] D. L. Thornton, and G. Valente, “Out-of-sample predictions of bond excess returns and forward rates: An asset allocation perspective,” The Review of Financial Studies, vol. 25, no. 10, pp. 3141-3168, 2012.

[9] M.-E. Wu, H.-H. Tsai, W.-H. Chung, and C.-M. Chen, “Analysis of Kelly betting on finite repeated games,” Applied Mathematics and Computation, vol. 373, pp. 125028, 2020.

[10] F. Zhang, T.-Y. Wu, Y. Wang, R. Xiong, G. Ding, P. Mei, and L. Liu, “Application of Quantum Genetic Optimization of LVQ Neural Network in Smart City Traffic Network Prediction,” IEEE Access, vol. 8, pp. 104555-104564, 2020.

[11] T.-Y. Wu, H. Li, and S.-C. Chu, “CPPE: An Improved Phasmatodea Population Evolution Algorithm with Chaotic Maps,” Mathematics, vol. 11, no. 9, 1977, 2023.

[12] T.-Y. Wu, A. Shao, and J.-S. Pan, “CTOA: Toward a Chaotic-Based Tumbleweed Optimization Algorithm,” Mathematics, vol. 11, no. 10, 2339, 2023.

[13] N. Burkart, and M. F. Huber, “A survey on the explainability of supervised machine learning,” Journal of Artificial Intelligence Research, vol. 70, pp. 245-317, 2021.

[14] I. Ibrahim, and A. Abdulazeez, “The role of machine learning algorithms for diagnosing diseases,” Journal of Applied Science and Technology Trends, vol. 2, no. 01, pp. 10-19, 2021.

[15] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, “A survey on ensemble learning,” Frontiers of Computer Science, vol. 14, pp. 241-258, 2020.

[16] S.-L. Hsieh, S.-H. Hsieh, P.-H. Cheng, C.-H. Chen, K.-P. Hsu, I.-S. Lee, Z. Wang, and F. Lai, “Design ensemble machine learning model for breast cancer diagnosis,” Journal of Medical Systems, vol. 36, pp. 2841-2847, 2012.

[17] N. Tajbakhsh, and K. Suzuki, “Comparing two classes of end-to-end machine-learning models in lung nodule detection and classification: MTANNs vs. CNNs,” Pattern Recognition, vol. 63, pp. 476-486, 2017.

[18] Q. Chen, Z. Meng, and R. Su, “WERFE: A gene selection algorithm based on recursive feature elimination and ensemble strategy,” Frontiers in Bioengineering and Biotechnology, vol. 8, 496, 2020.

[19] M. A. Yaman, A. Subasi, and F. Rattay, “Comparison of random subspace and voting ensemble machine learning methods for face recognition,” Symmetry, vol. 10, no. 11, 651, 2018.

[20] J. Tang, Q. Su, B. Su, S. Fong, W. Cao, and X. Gong, “Parallel ensemble learning of convolutional neural networks and local binary patterns for face recognition,” Computer Methods and Programs in Biomedicine, vol. 197, 105622, 2020.

[21] J. Tang, J. Zhang, G. Yu, W. Zhang, and W. Yu, “Multisource latent feature selective ensemble modeling approach for small-sample high-dimensional process data in applications,” IEEE Access, vol. 8, pp. 148475-148488, 2020.

[22] M. Hu, and P. N. Suganthan, “Experimental evaluation of stochastic configuration networks: Is SC algorithm inferior to hyper-parameter optimization method?,” Applied Soft Computing, vol. 126, pp. 109257, 2022.

[23] B. Solnik, “International arbitrage pricing theory,” The Journal of Finance, vol. 38, no. 2, pp. 449-457, 1983.

[24] M. M. A. Mia, S. K. Biswas, M. C. Urmi, and A. Siddique, "An algorithm for training multilayer perceptron (MLP) for Image reconstruction using neural network without overfitting," International Journal of Scientific & Technology Research, vol. 4, no. 2, pp. 271-275, 2015.
[25] S. R. Safavian, and D. Landgrebe, "A survey of decision tree classifier methodology," IEEE Transactions on Systems, Man, and Cybernetics, vol. 21, no. 3, pp. 660-674, 1991.
[26] E. Y. Boateng, J. Otoo, and D. A. Abaye, "Basic tenets of classification algorithms K-nearest-neighbor, support vector machine, random forest and neural network: a review," Journal of Data Analysis and Information Processing, vol. 8, no. 4, pp. 341-357, 2020.
[27] P. Bühlmann, and B. Yu, "Analyzing bagging," The Annals of Statistics, vol. 30, no. 4, pp. 927-961, 2002.
[28] R. E. Schapire, and Y. Freund, "Boosting: Foundations and algorithms," Kybernetes, vol. 42, no. 1, pp. 164-166, 2013.
[29] T. Jiang, J. P. Li, A. U. Haq, A. Saboor, and A. Ali, "A novel stacking approach for accurate detection of fake news," IEEE Access, vol. 9, pp. 22626-22639, 2021.
[30] N. Jegadeesh, J. Noh, K. Pukthuanthong, R. Roll, and J. Wang, "Empirical tests of asset pricing models with individual assets: Resolving the errors-in-variables bias in risk premium estimation," Journal of Financial Economics, vol. 133, no. 2, pp. 273-298, 2019.
[31] E. Renault, T. Van Der Heijden, and B. J. Werker, "Arbitrage pricing theory for idiosyncratic variance factors," Journal of Financial Econometrics, vol. 21, no. 5, pp. 1403-1442, 2023.
[32] C. Ren, and Y. Xu, "A fully data-driven method based on generative adversarial networks for power system dynamic security assessment with missing data," IEEE Transactions on Power Systems, vol. 34, no. 6, pp. 5044-5052, 2019.