Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 3938409, 13 pages
https://doi.org/10.1155/2021/3938409

Research Article

Better Effectiveness of Multi-Integrated Neural Networks: Take Stock Big Data as an Example

HangLin Lu and XiuYun Peng

Zhengzhou University, Zhengzhou, China

Correspondence should be addressed to HangLin Lu; luhanglinde@163.com

HangLin Lu and XiuYun Peng contributed equally to this work.

Received 9 August 2021; Revised 3 September 2021; Accepted 6 September 2021; Published 6 October 2021

Academic Editor: Thippa Reddy G

With the development of big data, in the financial market, the stock price prediction has many research directions from the perspective of big data. The classical time series prediction model cannot adapt to the high-latitude information of stock data in the era of big data. The development of deep learning provides a new idea for high-latitude stock data prediction. Four neural network models and three integrated learning models form different strategy sets, and the opening price of the next timestamp is predicted by backtracking information over the past 15 days with the characteristics of 12 indexes of the stock. The experimental results show that the prediction effect of the integration model based on the average weight policy and stacking policy is better than that of the single neural network, and the integration model based on stacking policy is expected to have the highest prediction accuracy and the minimum expected error. The accuracy was 80.2%, and the mean square error was 0.024. Compared with the single model, the accuracy is increased by 2%~7%, and the error is reduced by 0.01~0.03. The innovation of this article lies in the traditional machine learning thinking is applied to deep learning, as an individual with a variety of neural network to study, through the integration of learning strategies, fusion for the integration model, the experimental results show that the effect of the integrated model is better than that of a single model, to improve the robustness and accuracy of the model; the performance of the integrated model is more stable. For the utilization of big data resources, the integrated model of neural network has better prediction effect.

1. Introduction

In recent years, with the maturity and development of financial market, the theory of stock forecast is increasingly diversified. Early stock index predictions were based on market theories, such as Osborne's walking theory in 1959, which held that stocks could not predict the Brownian motion in physics [1]. Fama, winner of the Nobel Prize in Economics in 1970, puts forward the efficient market hypothesis, which believes that the trend of stock prices can be predicted under the condition of sufficient information [2].

In the beginning, the prediction of stock-related indexes also adopted statistical traditional regression model analysis, such as autoregressive conditional heteroscedasticity model (ARCH), autoregressive conditional heteroscedasticity model (ARIMA), and GARCH model [3].

With the development of machine learning and the improvement of the ability of machine learning in time series prediction, the relationship between machine learning and the prediction of financial-related indicators is getting closer and closer. Whether it is deep learning or basic machine learning model, it can constantly improve its performance during the training process. With the development of artificial intelligence and the improvement of computer performance in recent years, machine learning has been widely applied in the financial industry. In 1999, Allen and Karjalainen applied the genetic algorithm to the historical data of American stocks and deduced the trading rules from it [4].

Endemic learning has also been used in stock prediction. In 2003, Kim compared support vector machine and neural network to try the effect of support vector machine in stock index prediction process. In 2016, Khaidem used an integrated learning model of random forests to predict stock returns and reduce investment risk. Huang and Chen used support vector machine (SVM) to test the prediction of the model on the stock price of the Bank of China [5].

However, for stock index prediction, more scholars rely on a single model. Cui and Li used the GARCH model and BP network to carry out stock price prediction experiments, and BP network was better than the traditional statistical model [3]. However, the generalization ability of these networks is not strong, and it is easy to overfit the data of the training set, which makes the prediction effect of the test set worse. For time series, RNN, LSTM, and other cyclic neural networks have obvious advantages. Wang et al. compared the effect of RNN and LSTM in stock price prediction [6]. With the improvement of computer computing speed, some neural network algorithms have been more verified and applied. Yenidoğan et al. [7] used the LSTM model to predict the stock fluctuations of the CSI 300 Index and found that its prediction effect was better than the classical time series analysis method.

In 2017, Nelson et al. used LSTM neural network to build a stock index prediction model and predicted the rise and fall with the help of historical data. Meanwhile, by comparing the model with other machine learning methods, they found that deep learning could better extract data information and make more accurate predictions with stronger robustness. [8] immediately some integrated learning models such as forest. With the development of deep learning be replaced gradually, but the application of the integration strategy of integrated study on deep learning network can also make the performance of the model for promotion, Xie et al. than using LSTM neural network model, such as building-integrated learning model, in the stock of quantitative trading experiments achieved better effect [9].

The time series data used in this paper consists of 12 indicators, and each time, the data of the previous 15 trading days is backtracked to predict the opening price of the next time step. MLP, RNN, LSTM, and GRU were set as four basic models, and three combination strategies were adopted to form three integrated models. The first 70% data were taken as the training set and the last 30% as the prediction set. The training set data were formed into multiple batches of data by the self-help sampling method. In the training process, Adam was used as the optimizer of the deep neural network to predict the opening price of the next trading day and record the accuracy and error of the forecast rise and fall. We found that for the utilization of big data resources, the integrated model of neural network has better prediction effect.

2. Problem Raising and Theoretical Analysis

In the training setting of the deep learning model, in order to prevent overfitting on the test set data, the training samples of neural network are often randomly selected as training batches in the training set. Because of this randomness, the effect of the model in the process of training convergence is constantly fluctuating, and this fluctuation still exists in the training after model training convergence. This leads to a problem: the actual performance of the final model is unstable, and its effect has random error.

This error is reflected in the data of the verification set, which is actually the error caused by the different sensitivity of the model to different samples. In the test set, the fluctuation can be understood as the lack of feature learning of the model to the training data.

Then, different deep learning cell structures are determined, which determines the attributes of the corresponding network, which makes a model may have a better effect on a certain type of samples in the data set, but the learning effect on other types of samples is not ideal. That is, different network models may have different adaptive learning abilities to different samples. How to make the network adapt to the training of more types of samples as much as possible? We think of integrating neural networks with different structures to obtain a more robust model.

On the other hand, from the perspective of statistics, the fusion of prediction results of multiple models is similar to multiple sampling to take the mean, which can reduce the random error of prediction results and improve the stability of model prediction effect.

In conclusion, in order to reduce the randomness of this model training effect, capture the characteristics of the test set data as much as possible to improve the model performance. Here, we try to propose an integrated neural network model. Through the training and integration of deep learning networks with different structures, a more stable model can be obtained.

3. Individual Learner

Individual learner is one of the basic structures of the ensemble learning model, which can also be called the basic model. Individual learners also have their own learning prediction ability. In this paper, according to the characteristics of time series, we select 4 kinds of neural network models as individual learners, which are multilayer perceptron (MLP), recurring neural network (RNN), long-short-term neural network (LSTM), and gated recurring neural network (GRU).

3.1. MLP.

3.1. MLP. Multilayer perceptron is one of the most classical feedforward artificial neural network models.

The model has strong nonlinear fitting and generalization ability, and the weight and bias of each neuron are adjusted continuously with the help of the error back propagation algorithm, so as to reduce the error in the training set. However, the generalization ability of this model is insufficient, and it is easy to appear the phenomenon of overfitting for the data of the training set. Chen et al. proposed the problem of insufficient generalization ability of MLP [10].

3.2. RNN. Cyclic neural network, also known as recursive neural network, was proposed by Schuster and Paliwal [11] in 1997. It is a kind of neural network built for sequential data and can fully reflect the correlation of data at different time nodes [12]. Cyclic neural networks have some advantages in learning the nonlinear characteristics of sequences because of their memory in time. Recurring neural network has many applications in natural language processing [13], time series prediction, and other fields.

$$O_t = g(V \cdot S_t),$$
$$S_t = f(U \cdot X_t + W \cdot S_{t-1}), \tag{1}$$

where x_t represents the input value, s represents the value of the hidden layer, U represents the weight matrix from the input layer to the hidden layer, V represents the weight matrix from the hidden layer to the output layer, and o represents the output value. As can be seen from the figure, the weight matrix of the hidden layer of the cyclic neural network depends not only on the current input x but also on the value s of the hidden layer on the last timestamp.

3.3. LSTM. Long and short-term memory artificial neural network (LSTM) is a chain form designed to solve the long-term data dependence existing in the recursive neural network (RNN) [14], proposed by Hochreiter and Schmidhuber [13]. As a special cyclic neural network, LSTM is formed by repeating module chains. LSTM also has this chain-like structure, of which the most important basic structure is the cell. Each cell has a specific gate structure to realize selective information transfer. Through the information transfer of LSTM gate structure (forgetting gate, input gate, update gate, and output gate), each cell state can be updated according to the last output and the current input. The specific structure is shown in Figure 1.

3.4. GRU. Gated circulation unit is a variant of LSTM proposed by Chung et al., whose special structure can solve the phenomenon of gradient dispersion in the training process of standard RNN [15]. The GRU controls the input and memory of information through two gate structures, a reset gate and an update gate. The reset gate determines the combination of the new input information with the information previously memorized by the GRU cell, while the update gate is used to save the memorized information from the previous timestamp to the information retained by the current timestamp. This gate control structure can better preserve the information in the long-term time series and will not forget or erase the effective information because of the longer time series.

The basic structure of GRU is shown in Figure 2.

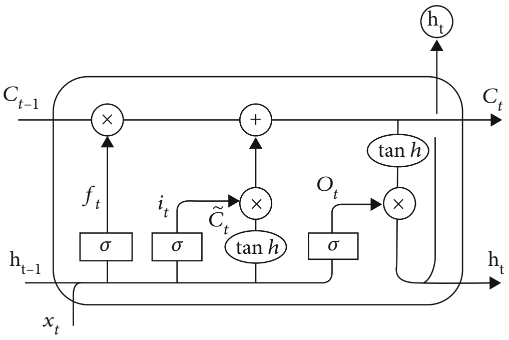

Figure 1: LSTM structure.

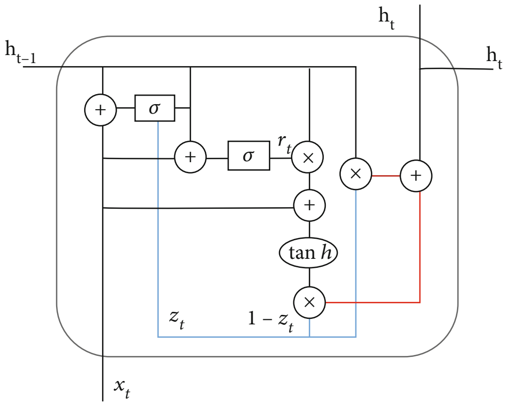

Figure 2: GRU structure.

4. Ensemble Learner

4.1. Basic Theory. Ensemble learning is a learning mode that constructs multiple individual learners and integrates them to achieve related classification or fitting tasks. Its basic structure is a single individual learner. Ensemble learning combines basic models through a combination strategy to achieve an integrated model that exceeds the effect of individual learners and improve the robustness of the model.

Common ensemble learning models are mostly built based on weak learners, such as random forest and boosting. In order to obtain a good integrated model, individual learners should be good but different, that is to say, individual learners have better performance, but at the same time, different learners have differences in principle or architecture. In this paper, we will use the model with strong learning ability as the basic learner. Multiple neural networks with different structures are used as individual learners, and MLP, RNN, LSTM, and GRU are selected as individual learners according to the characteristics of stock index prediction time series. Each individual learner itself has strong learning ability. Different types of individual learners are included in the integration, so the integration model constituted is "heterogeneous," and each basic model has a parallel relationship. The selection and structure setting of such an individual model can improve the robustness of the integration model in principle.

4.2. Integration Strategy

4.2.1. Average Weight Method. The combination strategy of the average weight method is a commonly used learning strategy for numerical regressions in ensemble learning. The method is to average the output of several individual learners to get the final predicted output.

The final forecast results are as follows:

$$H(x) = \frac{1}{T}\sum_{1}^{T} \text{model}_i(x). \tag{2}$$

When combining the average weight strategy in this paper, four models were selected, that is, the combining layer multiplied the output of each model by 0.25 and then summed up the predicted value. After the predicted value was compared with the target value to solve the mean square error (MSE), the error was transmitted in reverse and fed back to the four individual learners. In the error feedforward process, each weight of the binding layer is always maintained at 0.25, which does not change with the error feedforward. Model settings are shown in Figure 3.

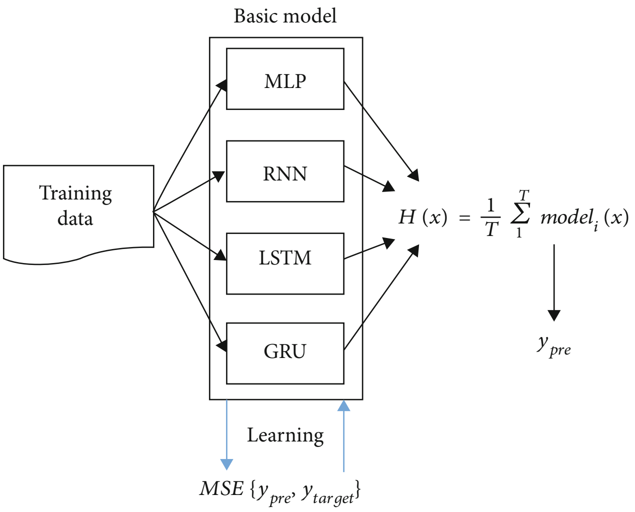
Basic model
MLP
RNN
LSTM
GRU
Training data
$H(x) = \frac{1}{T}\sum_{1}^{T} model_i(x)$
y_{pre}
Learning
$MSE\{y_{pre}, y_{target}\}$

Figure 3: Average weight method.

4.2.2. Stacking Method. The combination strategy of stacking is to regression integration of the output of each model through one or more metalearners. The whole training set is used to train the basic model, and the metamotor trains the predicted value of the basic model as the characteristic.

In the integration model of this paper, the basic model for different learning algorithms, stacking is heterogeneous integration. The model settings are shown in Figure 4.

The algorithm pseudocode is as follows:

4.2.3. Global Learning Method. The combination strategy of the global learning method refers to the connection between the basic model and the secondary learners, as shown in the figure. Then, the whole training set data is used to train the model. During the training process, the error feedback not only changes the internal network parameters of each basic model but also changes the internal weight of the metalearner. Model settings are shown in Figure 5.

5. Experimental

In this paper, four kinds of neural networks are integrated, and three kinds of integration learning strategies are adopted for model fusion. The whole process is shown in the figure. In the model, the first stage is data preprocessing, including removing missing values, data segmentation, and data normalization and then dividing the training set and the test set. In the second stage, the training model is constantly evaluated for the performance of the model in the training set and test set to determine the number of training iterations. Finally, the network is used to predict the results of the stock index. Step settings are shown in Figure 6.

The specific steps are as follows:

(1) Missing values were eliminated, minimum-maximum normalization method was used to normalize each stock index in its dimension to values within the range of [0,1], and then, the historical data were divided into two groups: training data set and test data set

(2) Batch the data of the training set. At the beginning of each training round, the order of the data of the training set is disrupted and 64 data are extracted as a batch

(3) Initialize the model network, including the weight and bias of parameters on each layer, determine the stochastic gradient descent algorithm as Adam, and set the maximum number of iterations as 100. The fluctuation data of error and accuracy were recorded after the model was stabilized

(4) Conduct training on different basic models and integrated strategy models

(5) Set the number of iteration training: 100. Evaluate the suitability of the model. If the model is suitable, save the model and record the mean square error and accuracy at the same time; if not, continue the error feedforward training

(6) Import the test set data into the model to confirm the optimal one for index prediction. Ensure that complete predictions are made after optimal test set predictions are made

(7) Evaluate the prediction accuracy of each model through five performance indicators

The experiment was conducted on a PC (CPU: AMD Ryzen 5 3500U,8 Gbps RAM). The development environment is Python 3.8, and Spyder is running on a Windows 10 operating system. The model is implemented using PyTorch.

5.1. Data Settings. In this study, we select Shenzhen stock: Ping An Bank (stock code: 000001), Ping An Bank Co., Ltd., is a national joint-stock commercial bank headquartered in Shenzhen (Shenzhen Stock Exchange: 000001). Its predecessor, Shenzhen Development Bank, is a national joint-stock bank publicly listed in the mainland of China. The experimental data set consists of daily historical data for the past 10 years as of January 4, 2010, BBB 0 and

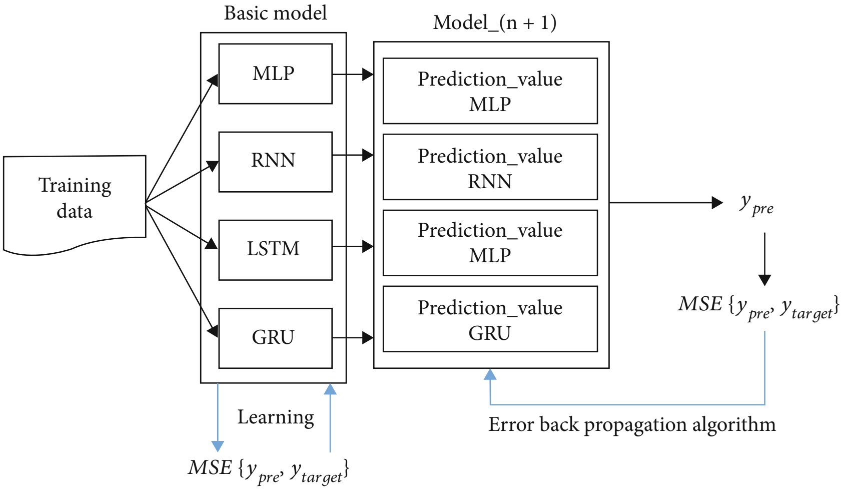

Figure 4: Stacking method.

```
Input: training data $D = \{x_i, y_i\}_{i=1}^{m}$
Output: ensemble classifier $H$
*Step1: learn base-level classifier*
**for** $t = 1$ to $T$ **do**
    learn $h_t$ based on $D$
**end for**
*Step2: construct new data set predictions*
**for** $i = 1$ to $m$ **do**
    $D_h = \{x_i, y_i\}$, where $x_i' = \{h_1(x_i), \cdots, h_T(x_i)\}$
**end for**
*Step3: learn a meta-classifier*
learn $H$ based on $D_h$
return $H$
```

Algorithm 1: Stacking.

December 31, 2019. All data are from the Oriental Fortune Market Center, the data as shown in the figure.

As shown in Table 1, each timestamp contains 12 characteristics, which are closing price, maximum price, minimum price, opening price, previous closing price, up/down amount, up/down amount, turnover rate, volume, transaction amount, total market value, and circulating market value.

5.2. Data Preprocessing. Data preprocessing is a crucial step in data analysis and model training. High-quality data leads to better models and predictions. First, a small number of missing values in the data set were eliminated, and then, each index was normalized to between. The normalization method is as follows:

$$X = \frac{X_t - X_{\min}}{X_{\max} - X_{\min}}, \tag{3}$$

where X_t is the value f a feature on the timestamp t, $X_{\max}$ is the maximum value on this feature dimension, and $X_{\min}$ is the minimum value on this feature dimension.

After data preprocessing, the original data set is divided into two independent sets at a fixed ratio. Among them, the first 70% of daily historical data is taken as the training set, and the remaining 30% of data is taken as the test set. As shown in the figure, the blue part is the training set data, and the orange part is the test set data, as shown in Figure 7.

The normalized data distribution of the training set and the test set is shown in the figure. As shown in Figure 8, it can be seen that the distribution domain of the training set is larger than that of the test set, that is, the training set and the test set of the data are properly divided.

5.3. Performance Index Evaluation. There are many ways to measure the effect of prediction. In order to properly evaluate the prediction ability of various models, the following five indicators are used in the experiment to measure the accuracy of the model: mean square error (MSE), mean absolute error (MAE), MAPE, and R squared.

$$\begin{aligned}
\mathrm{MSE} &= \frac{1}{N}\sum_{t=1}^{N}\left(y_{\text{test}}(t) - y\wedge_{\text{pre}}(t)\right)^2, \\
\mathrm{MAE} &= \frac{1}{N}\sum_{i=1}^{N}\left|y_{\text{test}}(t) - \hat{y}_{\text{test}}(t)\right|, \\
\mathrm{MAPE} &= \frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{y_{\text{test}}(t) - \hat{y}_{\text{test}}(t)}{y_{\text{test}}(t)}\right|, \\
R^2 &= 1 - \frac{\mathrm{MSE}\left(y_{\text{test}}(t) - \hat{y}_{\text{test}}(t)\right)}{\mathrm{Var}\left(y_{\text{test}}(t)\right)},
\end{aligned} \tag{4}$$

where N represents the number of samples of this group of data, while y_{test} and $\hat{y}_{\text{test}}$ represent the real and predicted values of the test set data, respectively.

Considering that the stock index pays more attention to the trend of rise and fall, the accuracy rate is introduced at the same time. The definition is if the index value of stock $T + 1$ is greater than that of stock T, and the predicted value of stock is also greater than that of stock T, it is considered

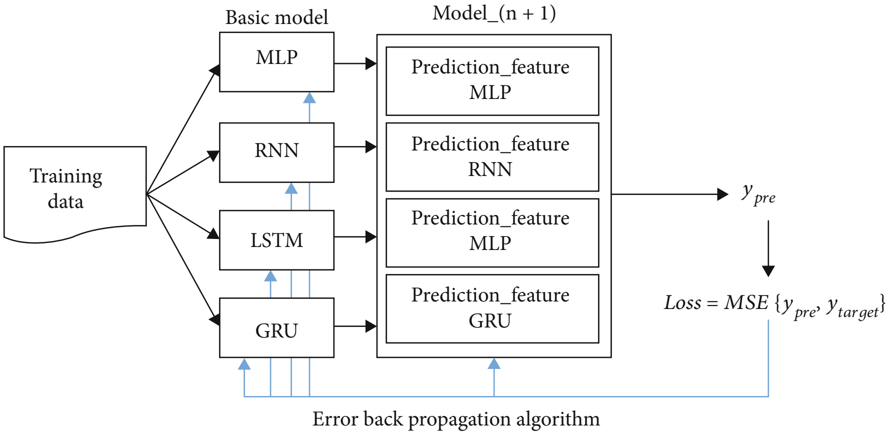

FIGURE 5: Global learning method.

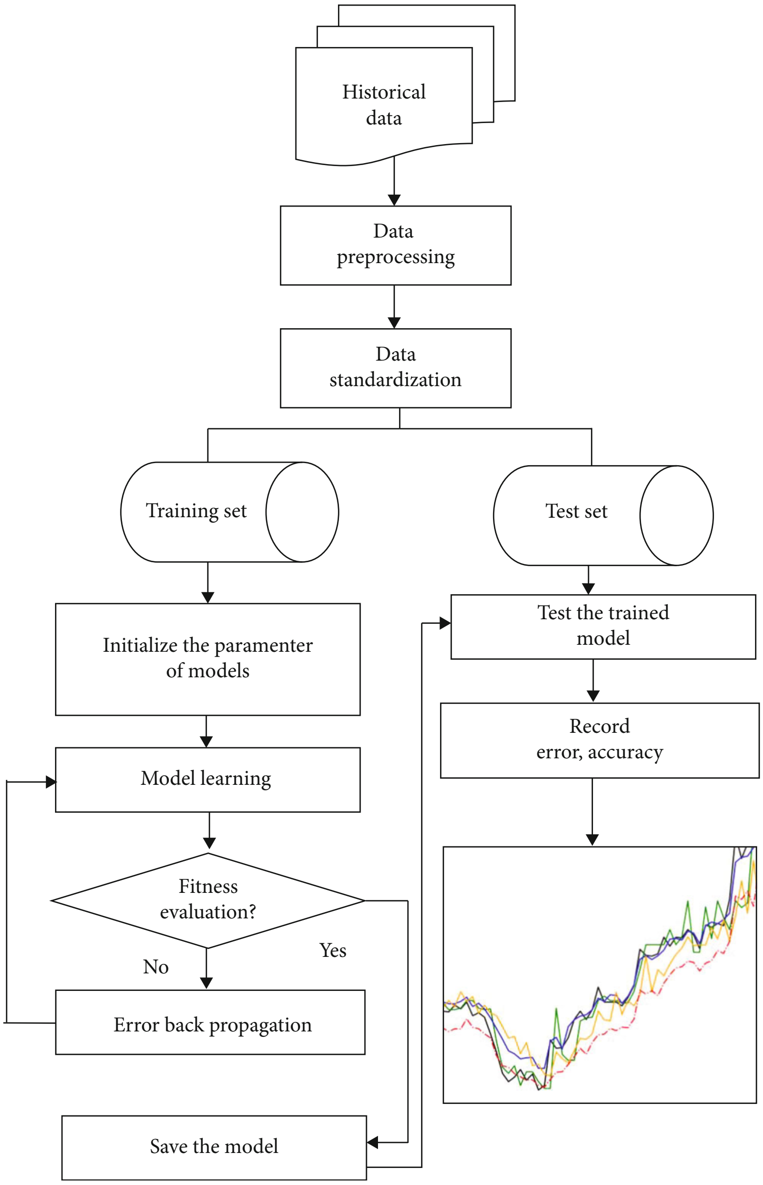

FIGURE 6: The experimental process.

Table 1: Stock data format.

Date	Close	Highest	···	Change (%)	Vol (share)	Vol (RMB)	EUR (RMB)	FAMC (RMB)
2010/1/4	23.71	24.58	···	-2.7082	24192276	580249472	73629834497	69330922520
2010/1/5	23.3	23.9	···	-1.7292	55649982	1293476939	72356606655	68132032674
2010/1/6	22.9	23.25	···	-1.7167	41214313	944453697	71114433150	66962384045
2010/1/7	22.65	23.05	···	-1.0917	35533685	804166316	70338074709	66231353651
2010/1/8	22.6	22.75	···	-0.2208	28854306	650667405	70182803021	66085147572

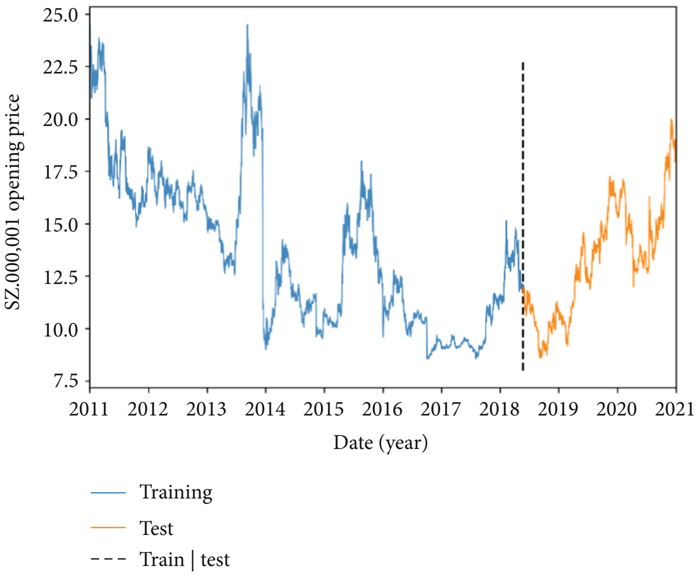

Figure 7: Training set test set partition.

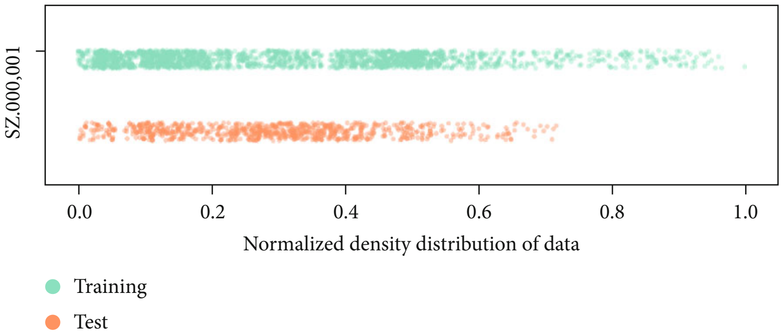

Figure 8: Data distribution of training set and test set.

that the forecast is correct. The reverse is the opposite. Namely, if the successful prediction of stock index trend is up or down, it is considered that the prediction is successful.

The formula is expressed as

$$\text{Accuracy} = \frac{n_{\text{right trend}}}{N_{\text{total}}}. \quad (5)$$

5.4. Parameter Setting of the Model. The setting of model parameters often affects the amount of information that can be recorded by the model, thus affecting the effect of

Table 2: Parameter settings.

Model	Parameter settings
MLP	Unit number = 32 Layer number = 2

RNN	Batch_size = 64
	Epoch = 100

LSTM	Parameter = Adam Learing rate = 5e − 4
GRU	

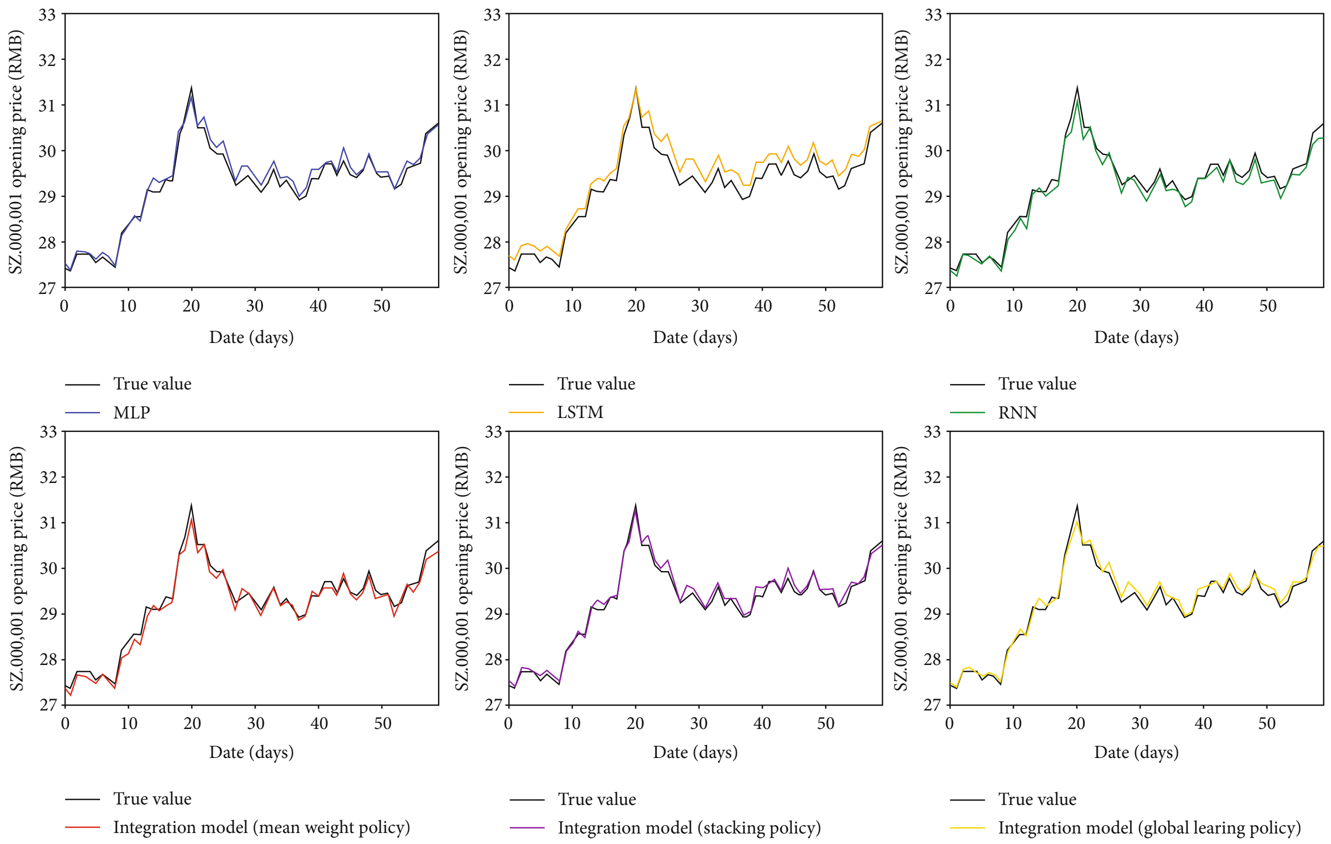

Figure 9: The prediction of several models.

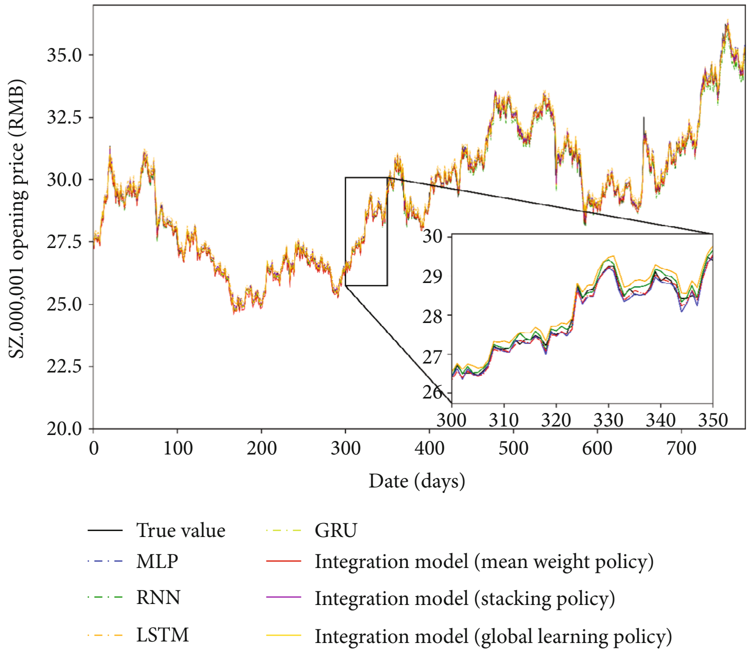

Figure 10: Complete data set prediction.

the model. In order to fairly compare the effects of different models, the parameters of the basic model will be set the same here. The model parameters are set as Table 2.

5.5. Forecast of Indicators. The whole data contains 2560 records, each timestamp records the closing price, the high, the low, the opening price, and other 12 characteristic

Table 3: Evaluation of prediction results.

Model	Accuracy	MAE	MAPE	MSE	R_2
MLP	0.758301	0.151200	0.011381	0.041578	0.993498
RNN	0.785069	0.133305	0.010259	0.032996	0.994840
LSTM	0.778255	0.143340	0.010028	0.042996	0.996720
GRU	0.776190	0.129919	0.009749	0.031959	0.995003
MEAN	0.788030	0.129757	0.009710	0.032805	0.994870
STACK	0.802240	0.118611	0.003991	0.028048	0.995695
LEARN	0.755701	0.154355	0.005160	0.043889	0.993264

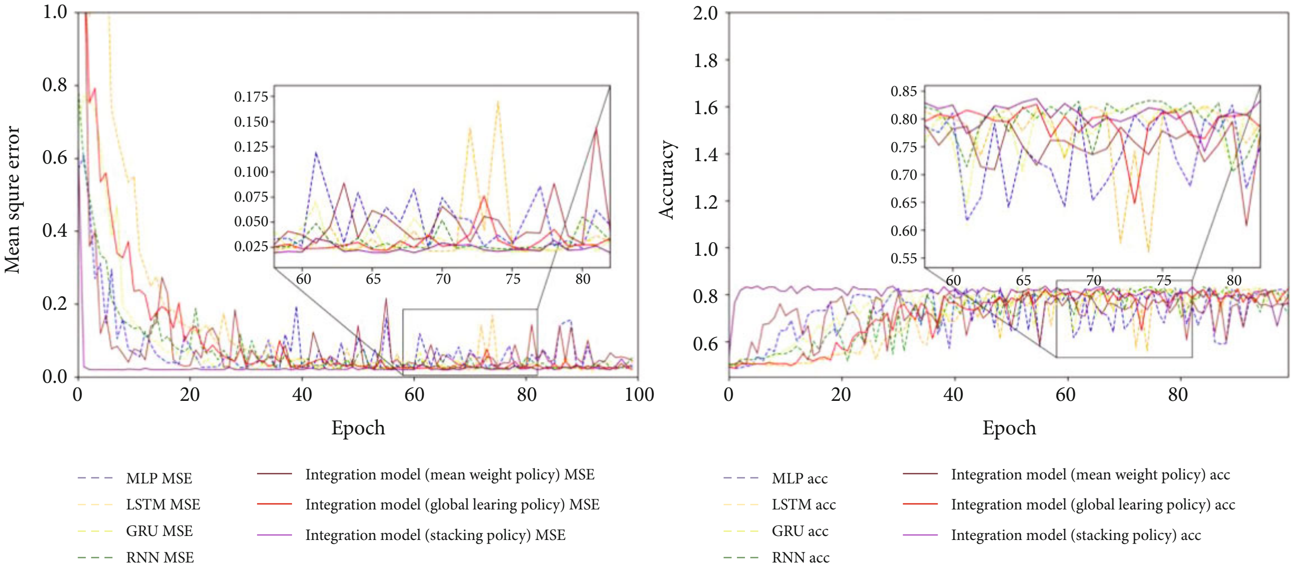

Figure 11: Models train batches with fluctuating stability and error conditions.

indicators. The first 1792 records were taken as the training set, 64 records were taken as a batch, and 28 batches were formed. The last 777 records were used as test set data. In order to comprehensively evaluate the prediction performance of the model, the prediction results are evaluated by the four performance indicators mentioned above.

Backtracking days refer to the data of the previous days used as the feature in the forecast, and the opening price of the next day is predicted by the feature data of the previous N days.

In order to evaluate the performance of various models, based on previous studies, we set the number of days for backtracking data to be 15.

6. Result Evaluation

6.1. Prediction Effect. Figure 9 shows the prediction of the opening price index of the test set during 60 days by the models under the three basic models and the three integration strategies. It is obvious from the figure that the curves of the integration strategy using the average weight method and the stacking strategy are the closest to the real value, as shown in Figure 9.

In order to reflect the prediction effect of the model more comprehensively, the prediction effect diagram of the model on the complete test set is drawn, as shown in Figure 10. It can be seen clearly from the enlarged subgraph in the figure that the integrated model of the average weight method and the stacking strategy are very close to the predicted value.

In order to evaluate the prediction accuracy of each model, the prediction accuracy of the model is evaluated through several evaluation indexes described above. The model of the iteration training batch (the last 50 batches) after the training results are stable is evaluated, and the average value of the evaluation indicators of each iteration is taken as the expected value of each indicator. The final results are shown in Table 3.

It can be seen from the table that among the single training models MLP, RNN, LSTM, and GRU, the evaluation results of RNN, LSTM, and GRU cycle networks are better than that of MLP network. The accuracy of RNN in predicting the rise and fall is higher, reaching 78%, and the R_2 of LSTM is higher, which indicates that the overall prediction fitting of LSTM is better. The MAE, MAPE, and MSE of GRU are smaller, which means the overall error of GRU prediction is smaller. From the analysis of the prediction and

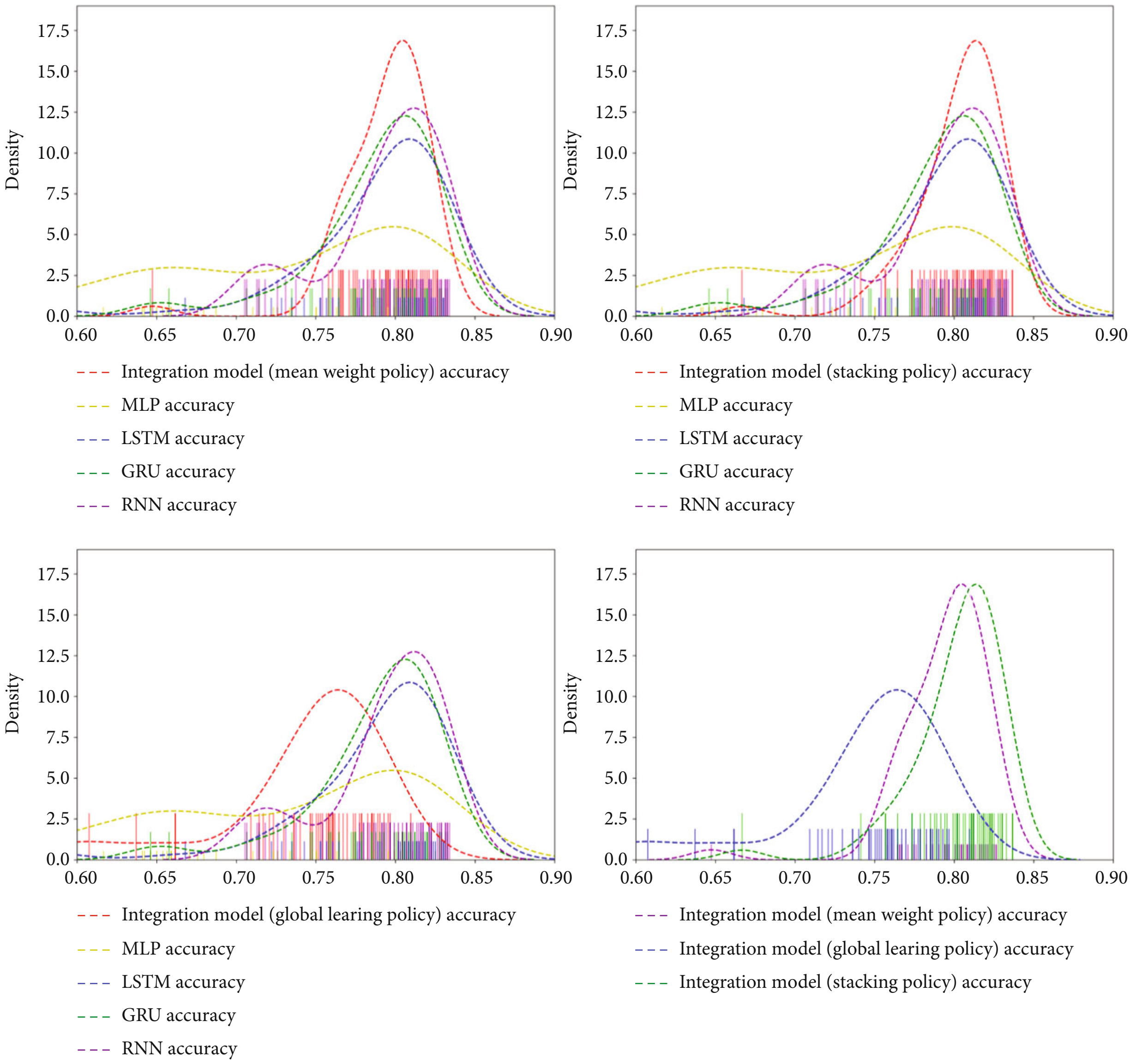

Figure 12: Prediction accuracy distribution after stable model training.

evaluation of a single model, it can be seen that the sensitivity of different single models to data has different characteristics.

In the integration model, the prediction effects under the three integration strategies are also different. Using the integrated model of average weight strategy and stack strategy, the prediction effect is better. The accuracy is 78.8% and 80.2%, respectively. Meanwhile, the overall prediction error of the integrated model of stacking strategy is smaller, and its MAPE is as low as 0.003991, and R_2 is as low as 0.9956. But on the other hand, the prediction effect of the integrated model of global learning strategy is not very good, with an accuracy of 75.5% and a MSE of 0.0438, which is the largest error among the tested models, and the prediction effect is even worse than that of the single model.

6.2. The Stability and Validity of the Model. In the process of experiment, we set the maximum training session as 100 times, but in the process of model training, it is often difficult to get the optimal model.

In most cases, it fluctuates at a certain level after the model training is stable. Whether the model parameters can be stabilized around the optimal parameters is an important reference to evaluate the stability of the model effect, as shown in Figure 11.

The figure reflects the convergence process and stability of each model after convergence. It can be seen from the figure that each model has basically converged after 30 rounds of training, but each model has fluctuations. In the MSE curve, the value is small and stable: the integrated model of the average weight method integration strategy and the stacking strategy, and the MSE is stable at [0.025,0.035]. In the accuracy figure, the more stable accuracy is also the integrated model of the average weight method integration strategy and the stacking stack strategy, and its accuracy is stable at [0.75,0.83].

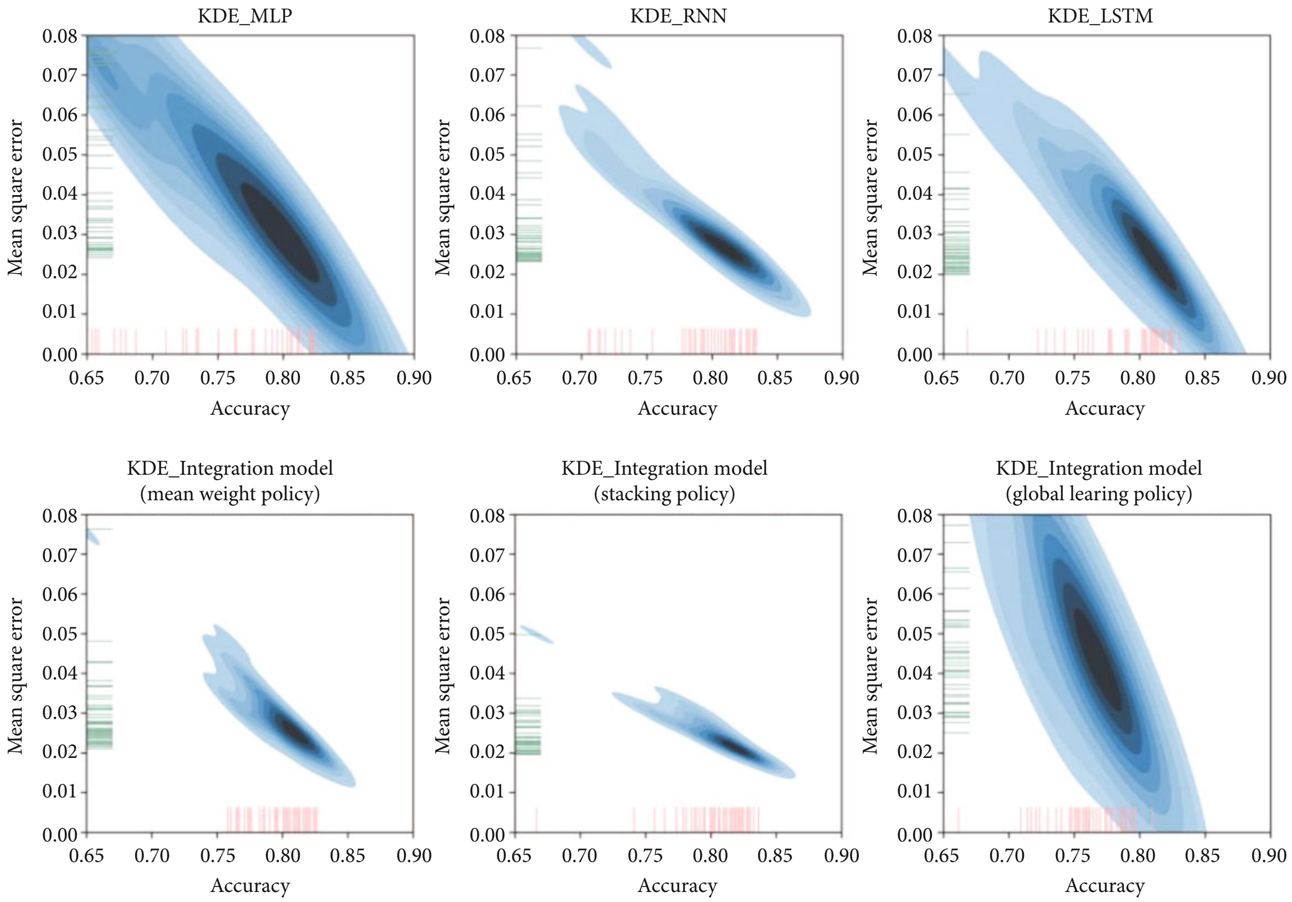

Figure 13: Two-dimensional distribution of prediction accuracy and error after stable model training.

In order to more accurately evaluate the stability of model training, we recorded the changes of MSE and accuracy of different models during the training process. Meanwhile, in the training round (the last 50 rounds) after confirming the stable fluctuation of the model, we drew the kernel density estimation curves of MSE and accuracy of each model. It reflects the effectiveness and stability of individual learner and three kinds of integrated learning models.

As can be seen from Figure 12, the density peak position of the integration model of the integration strategy of the average weight method and the integration model of the stacking strategy is close to the density peak position of the basic model, while the density of the integration model of the integration strategy of the average weight method and the integration model of the stacking strategy is more concentrated at the peak. That is to say, the accuracy error of these two integrated models is smaller, and the possibility of getting a better model is greater. However, the accuracy of the integrated model of global learning strategy is worse than that of each basic model. The expected accuracy of the integrated model of global learning strategy is worse than that of the single model, and the accuracy of the corresponding density peak is even lower than that of any single model.

According to the fourth picture in the figure, it can be seen that among the three kinds of integration strategies, the density peak of accuracy of the integration model of the integration strategy of the average weight method and the integration model of the stacking strategy is similar, that is, the error of stability is smaller in the training process and the effect is better. Relatively speaking, the density peak of the integration model of stacking strategy is more to the right, and the accuracy of the corresponding peak is higher than that of the integration model of the average weight method integration strategy. This means that the integrated model for the stacking policy works better.

As shown in Figure 13, for the training round (the last 50 rounds) after the stable fluctuation of the model, a two-dimensional kernel density estimation diagram of its accuracy and mean square error was drawn, as shown in the figure. The larger the area of blue shadow in the figure is, the more dispersed the density is, which means that the accuracy and mean square error of the model fluctuate more and become more unstable during the training process. As can be seen from the diagram, the use of the average weight strategy integration model and integrated model of the stacking strategy of the shadow area is small, intensity bigger, which means that the model is more stable in the process of training, and maximize the density of these two models (i.e., the deepest shadow color) for greater accuracy, the corresponding smaller mean square error (MSE). This means that the expected performance of these two models is better than that of the other models.

Table 4: Expectation and standard deviation of models, accuracy, and error.

Model	Accuracy		Mean square error	
	Mean	Standard deviation	Mean	Standard deviation
MLP	0.73756	0.075862	0.05546	0.036732
RNN	0.79318	0.036592	0.03173	0.011244
LSTM	0.78512	0.054487	0.03355	0.026722
GRU	0.78798	0.039606	0.02939	0.011710
Integration model (mean)	0.79393	0.027936	0.02919	0.008925
Integration model (learning)	0.74497	0.049791	0.05828	0.037212
Integration model (stacking)	**0.80224**	**0.028740**	**0.02441**	**0.005195**

In order to concretely evaluate the error fluctuation, the expectation and standard deviation of the mean square error and accuracy of the training round (the last 50 rounds) after the stable fluctuation of the model were calculated. The calculation formula is as follows:

$$E(X) = \frac{1}{n}\sum_{i=1}^{n} x_i,$$
$$\sigma = \sqrt{D(X)} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (x_i - x)^2}. \tag{6}$$

In Table 4, the expected values of model accuracy and model mean square error are shown, in which the best effect is shown in bold. The integrated model with stacking strategy has the highest expected accuracy and the lowest standard deviation of accuracy. Meanwhile, the expected mean square error and the standard deviation of the mean square error are the least. The above indicates that this model expects the best effect, and its effect is the most effective and stable.

7. Conclusion

In this paper, four deep learning network models are constructed and three strategies are adopted to combine them into three integrated networks. Through the training of the model, the performance of the model is reflected by the data of the test set. The accuracy and error of the model were recorded during the training process. Through the experiment, we found the following:

(1) To take the average weight strategy composed of the integrated model and to take the stacking strategy composed of the integrated model, the effect of these two models is better than the effect of a single neural network. By recording the fluctuation of accuracy and error during the training of each model, we analyzed the stability of the model. It is found that the value of the peak accuracy density of the ensemble model composed of the average weight strategy and the ensemble model composed of the stacking strategy is higher than that of the basic model and the ensemble model adopting the global learning strategy. The density peak of the integration model adopts the first two integration strategies simultaneously. It is higher than the peak density of single strategy and global learning strategy. This means that the first two models, performance and stability, are expected to be better than other models. At the same time, these two models have stronger robustness. By comparing the integration model formed by the two combined strategies, we find that the integration model with stacking policy has the highest expected accuracy. At the same time, the expected error of the model is minimum, which means that the model takes into account the stability while optimizing the effect

(2) The integrated model composed of global learning strategy has the worst effect among all models, and its performance and stability are even inferior to that of the single base model. Through the analysis of the weight layer of this network, it is found that in the process of training, this strategy not only promotes the training of each single basic model in the process of error back propagation but also affects the weight allocation of each basic model by the metalearner. This leads to a complex game relationship between individual learner and metalearner, that is, when an individual learner with poor performance is optimized, the metalearner reduces the weight of the output of the individual learner. In this case, the better performance of the learner is constantly changing, which makes the metalearner unable to confirm an optimal weight strategy. In this case, the larger fluctuation of the whole model during the training process can also be explained theoretically

(3) Disadvantages of the model. Through the above experiments, we find that a more stable model can be obtained by integrating a variety of neural network models. However, the improvement of this stability requires a large time cost. For example, for a model integrated with four individual learners, four models need to be trained separately. This increases the time cost by four times, but the performance improvement is small

Future research will try to use the overlay strategy module integration model built in this paper. This multimodal model will be applied to more fields. We will forecast the indicators of financial products such as foreign exchange, commodities, bonds, and futures. We plan to use an integration strategy that combines several different deep learning networks to build a better, more stable model. With the help of these integration strategies, the application of traditional machine learning thinking to the field of deep learning has great inspiration for the construction of multipattern deep learning models. With the development of the era of big data, the advantages of this new integration approach will be gradual.

Data Availability

The source of the data has been declared and is readily available on the network.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Authors' Contributions

Hanglin Lu and Xiuyun Peng contributed equally to this work.

Acknowledgments

National Innovation Training Program for College Students. (Item No.202110459137) Website: http://gjcxcy.bjtu.edu.cn/Index.aspx. The paper is funded by the "national innovation training program for college". The fund aims to encourage college students to innovate and apply for it through the project plan.

References

[1] M. F. M. Osborne, "Brownian motion in the stock market," *Operations Research*, vol. 7, no. 2, pp. 145–173, 1959.

[2] B. Chairman and E. F. Fama, "Efficient capital markets: a review of theory and empirical work†," *The Journal of Finance*, vol. 25, no. 2, pp. 418–420, 1970.

[3] C. Jianfu and L. Xingxu, "Stock price forecasting: a comparison between GARCH model and BP neural network model," *Statistics and Decision Making*, vol. 6, pp. 21-22, 2004.

[4] F. Allen and R. Karjalainen, "Using genetic algorithms to find technical trading rules (revision of 20-93)," Wharton School Rodney L. White Center for Financial Research, 1999.

[5] H. Tongxing and C. Fangfang, *Journal of Chongqing University of Technology (Natural Science)*, vol. 30, no. 2, 2016.

[6] W. Jun, Z. Peng, and Y. Shuai, "Comparison of SEQ2SEQ RNN and LSTM models based on stock forecasting," *Time Financial*, vol. 35, pp. 381-382+392, 2018.

[7] I. Yenidoğan, A. Çayir, O. Kozan, T. Dağ, and Ç. Arslan, "Bitcoin forecasting using ARIMA and PROPHET," in *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, pp. 621–624, Sarajevo, Bosnia and Herzegovina, 2018.

[8] D. Nelson, A. Pereira, and R. Oliveira, "Stock market's price movement prediction with LSTM neural networks," in *2017 International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, USA, 2017.

[9] Q. Xie and X. X. Cheng Gengguo, "Research on stock forecasting model based on neural network enrollment learning," *Computer Engineering and Applications*, vol. 55, no. 8, pp. 238–243, 2019.

[10] X. W. Chen, W. Y. Zhu, X. M. Qian et al., "Estimation of surface layer optical turbulence using artificial neural network," *Acta Optica Sinica*, vol. 40, no. 24, article 2401002, 2020.

[11] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning (Vol. 1)*, MIT Press, Cambridge, 2016.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] A. A. Ariyo, A. O. Adewumi, and C. K. Ayo, "Stock price prediction usingthe ARIMA model," in *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, pp. 106–112, Cambridge, UK, 2014.

[15] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, https://arxiv.org/abs/1412.3555.