# Quantitative Investment Model Based on Bidirectional LSTM

Sijia Chen*

*Corresponding auther E-mail address: chensijia@whut.edu.cn

School of International Education, Wuhan University of Technology, Wuhan 430070, China

**ABSTRACT**: With the development of big data technology, quantitative investment has become more and more important in the global financial trading market. To discuss the price prediction effects of different machine learning methods under the unit of trading time of minutes, three models based on integration learning, LSTM and bidirectional LSTM are developed in this paper. To evaluate the effectiveness of each model, this paper uses historical data of opening price, closing price, high price, low price, volume and amount for every 5 minutes from July 14, 2021 to January 28, 2022 in China's A-share market, and takes VMA, VMACD, BBI, MA, EXPMA and other indicators with strong correlation with volume and closing price for model construction and backtest analysis, and The regression prediction effect of the model is evaluated by the goodness-of-fit and other indicators. Finally, it is evaluated that the bidirectional LSTM-based time series forecasting model is better in predicting the volume every 5 minutes and can assist investors in making minute-based buy-sell decisions.

**Keywords:** Quantitative Investment; Bidirectional LSTM; Integrated Learning

## 1. Background of the Study

Quantitative investment is a trading method that aims to obtain stable returns by issuing buy and sell orders through quantitative methods and computer programming. With features such as historical data dependence, discipline and fine-grained risk control, quantitative investment can improve the efficiency of market price discovery and information efficiency. [1] The continuous development of digital finance, the development of big data technology and the reduction of trading fees have driven the continuous development of quantitative investment. [2]

In recent years, quantitative investment has become more and more important in the global financial trading market, and quantitative investment in China has also entered a phase of rapid development after 2021. One of the main approaches to quantitative investment is to use machine learning and deep learning as tools for time series forecasting of data, and the other is to use impact factors to classify high quality and poor quality stocks in order to adjust investment strategies. However, market information is very heterogeneous and there are many factors that affect the price of a product, so it is challenging to extract effective indicators from the huge amount of market information to develop a trading strategy. Thus, it is necessary to build a model to forecast future prices and guide trading strategies based on various investment indicators.

ICBBEM 2023, May 19-21, Hangzhou, People's Republic of China

DOI 10.4108/eai.19-5-2023.2334234

## 2. Literature Review

The financial market in China has not been established for a long time, and China's quantitative investment industry started late, and there are limitations such as absence of hedging tools, trading system restrictions, and sentiment in the market, which objectively provide some adverse conditions for the growth of the quantitative investment industry. The quantitative investment industry in China started relatively late, but the research on related topics in academia and industry has not lagged behind.

Domestic scholars explored BP neural networks, simple models of SVR in the early days, and thereafter, there has been a gradual emergence of research results based on integrated learning and deep learning (e.g., random forests). [3]

The interest in machine learning in the industry started roughly after 2015. In 2016, the first research report based on Adaboost appeared, and its machine learning method was framework-level boosting. [4] Post-2017, a large number of machine learning and deep learning application research reports published by power brokers, and some of the brokers combined the deep accumulation of brokerage firms in the financial industry in a more in-depth way, and experimental explorations of classical integrated learning methods and deep learning methods were conducted.

Most of the current research uses daily closing prices as data to analyze the predictive effects of integrated learning and deep learning, but quantitative investments are often traded in minute time units, so comparing the effects of machine learning models that trade in minute time units has some practical significance.

## 3. Integrated Learning Algorithm Based Regression Prediction

### 3.1 Integrated Learning Framework Establishment

In this paper, we use an integrated learning framework to build a two-layer model. The first layer consists of multiple primary learners, and we use Random Forest, Bag KNN, Xgboost, LightGBM and other machine learning algorithms; the second layer uses LightGBM as the secondary learning algorithm. In this paper, we use the stacking strategy, i.e., the training result of the training set on the primary learner will be used as the input sample of the secondary learner, and then the secondary learner will be trained and predicted once to get the final prediction result.

In this paper, two integrated learning frameworks, Bagging and Boosting, and the Staking model fusion strategy are used. The overall architecture of the model is shown in Figure 1.

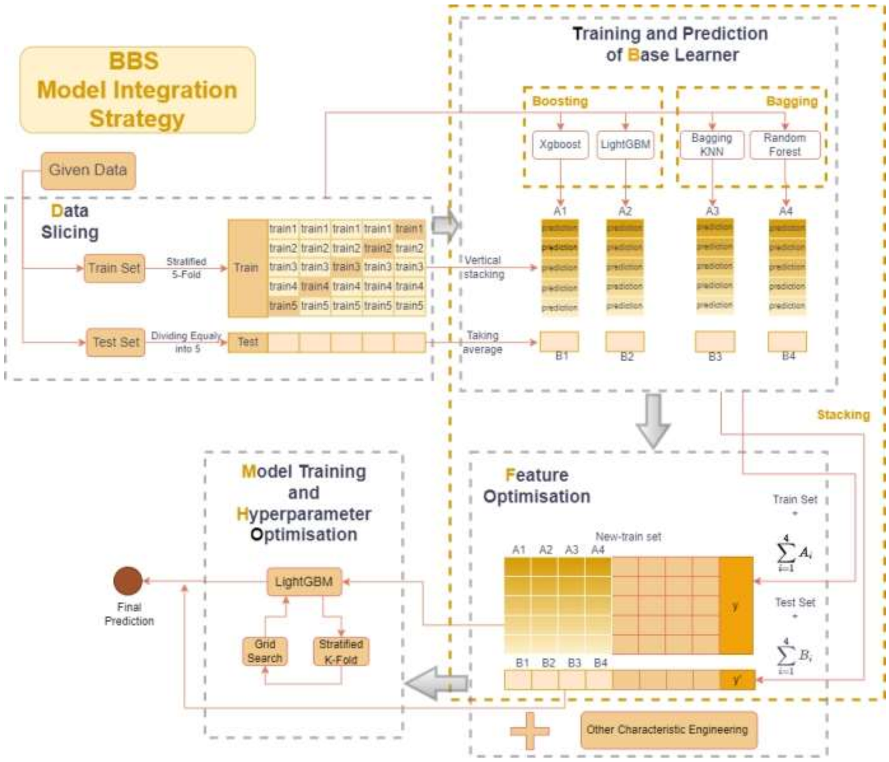


**Figure 1** BBS Model Frame

### 3.2 Cross-validation

In order to verify the performance of the classification evaluator, this paper adopts the K-fold cross-validation method, which is to divide the original data into K groups. Each subset of data is used as a validation set, with the remaining K-1 subsets of data are used as a training set, resulting in K models. The classification accuracy of the K models in the final validation set is averaged as a performance metric of the regressor's prediction accuracy.

### 3.3 Primary Learner Selection

In this paper, we choose Random Forest, Bag_KNN, Xgboost, LightGBM as the primary learner and LightGBM as the secondary learner, and we mainly introduce the first three models below, leaving the LightGBM model to be introduced later.

#### 3.3.1 Random Forest

Random forest algorithm is inherited and improved from the traditional decision tree, which is able to analyze complex interactive features, learn faster, and have higher robustness in dealing with missing value data. [5]

The random forest regression model is based on the following principle:

Input: training dataset D.

Output: regression tree f(x).

Each region in the input space on which the training dataset is located is divided recursively into two sub-regions and the output values of each sub-region are identified in order to construct a binary decision tree:

1. Choose the optimal tangent variable j and tangent point s, and solve

$$\min_{j,s}\left[\min_{c1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2\right] \tag{1}$$

An iteration is performed on the variable j. The cut point s of the fixed-point cut variable j is scanned and the pair (j,s) that minimizes the above equation is selected.

2. Delineate the region using the selected pair (j,s) and determine the appropriate output value;

3. Continue to call steps (1), (2) for both subregions until the stopping condition is satisfied.

$$R_1(j,s)=\{x|x^{(j)}\le s\},R_2(j,s)=\{x|x^{(j)}>s\} \tag{2}$$

$$\widehat{c_m}=\frac{1}{N_m}\sum_{x_i\in R_m(j,s)}y_i\,,x\in R_m,m=1,2 \tag{3}$$

4. Partition the input space to M regions, R1, R2, ... , RM, to yield a decision tree;

$$f(x)=\sum_{m=1}^{M}\widehat{c_m}\,I(x\in R_m) \tag{4}$$

### 3.3.2 KNN

KNN ( k-Nearest Neighbor) is one of the widely used data mining classification algorithms that uses the k nearest neighbors to reflect the present data When performing regression prediction, the weighted mean of the K neighbors is used as the prediction result. [6]

The distance metric is as follows:

$$D(X,Y)=\left(\sum_{i=1}^{n}|x_i-u_i|^p\right)^{1/P} \tag{5}$$

When p is 1, the distance is the Manhattan distance, and when p is 2, the distance is the Euclidean distance (Euclidean distance).

### 3.3.3 XGBoost

The Xgboost (Extreme Gradient Boost) model is a gradient boosting decision tree (GBDT) that strives for extreme efficiency.

It is essentially a model based on tree structure and integrated learning. The principle is to keep adding trees according to the splitting of features, and each time a tree is added, the residuals of previous predictions are fitted to obtain a new output.

The objective function equation is:

$$L(\phi) = \sum_i l\left(\widehat{Y}_i - Y_i\right) + \sum_k \Omega(f_k) \tag{6}$$

where $l\left(\widehat{Y}_i - Y_i\right)$denotes the prediction error of the i-th sample，$\sum_k \Omega(f_k)$represents a function of the complexity of the tree, the smaller the function, the lower the complexity and the stronger the generalization ability of the model, the expression is:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda|g|^2 \tag{7}$$

where T denotes the number of leaf nodes and g denotes the value of the node.

### 3.4 Secondary Learner Selection

#### 3.4.1 LightGBM

LightGBM is an improved gradient boosting decision tree framework. It is based on the idea of linear combination of M weak regression trees into strong regression trees as shown in the following equation.

$$F(x) = \sum_{m=1}^{M} f_m(x) \tag{8}$$

where F(x) is the final output value; $f_m(x)$ is the output value of the mth weak regression tree.

The major refinements to the LightGBM model are the histogram algorithm and the leaf-wise strategy of growth with depth restrictions. In the histogram algorithm, the continuous data is divided into K integers and constructs a histogram of width K. During the traversal process, the discrete values are accumulated as indexes in the histogram and the optimal decision tree split points are searched.The leaf-wise strategy with depth restriction is to find the leaf with maximum gain for each split and loop through. At the same time, the depth of the tree and the number of leaves are limited to reduce the complexity of the model and prevent overfitting. With the above improvements, the LightGBM model greatly reduces the memory consumption. The model's training time is merely 1/10 of the training time of the XGBoost model, with improved accuracy, making it well suited for dealing with high data-volume problems such as prediction.

## 4. LSTM-based Multivariate Time Series Forecasting Model

### 4.1 RNN

#### 4.1.1 Forward Propagation

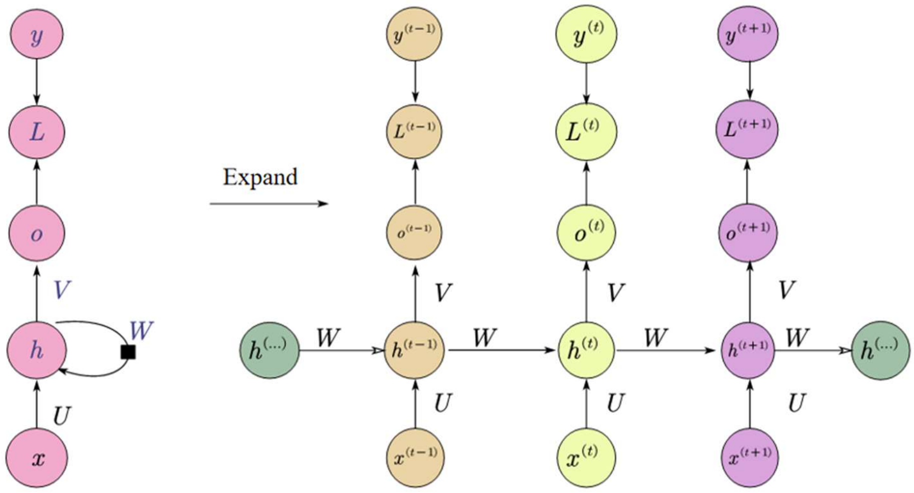


**Figure 2** RNN forward propagation schematic

As shown in Figure 2, where x is the input, h is the hidden layer unit, o is the output, L is the loss function, and y is the label of the training set. The t carried in the upper right corner of these elements represents the state at the moment t.

It is easy to see from the forward propagation schematic that the behavior of the hidden layer h at moment t depends not only on the inputs at that moment, it is also affected by the moments prior to moment t. In the figure, V, W, and U are weights, which are the same value for the same type of power connection.

Then for moment t

$$h^{(t)} = \phi\left(Ux^{(t)} + Wh^{(t-1)} + b\right) \tag{9}$$

where $\phi(x)$ is the active function, generally chosen as the tanh function, and b is the bias.

The output at moment t is much simpler:

$$o^{(t)} = Vh^{(t)} + c \tag{10}$$

The predicted output of the final model is:

$$\hat{y}^{(t)} = \sigma\left(o^{(t)}\right) \tag{11}$$

where $\sigma$ is the active function.

### 4.1.2 Reverse propagation training

Since the data are time-series data, in this paper, the time back propagation algorithm BPTT is used.

The central idea of BPTT is relatively simple, that is, search for better points in the direction of the negative gradient of the parameter to be optimized until it converges.

From the forward schematic, it can be seen that among the three parameters to be optimized, W and U need to be related to the previous data.

V does not need to be considered. Therefore, this paper discusses the optimization of parameters in categories.

Both W and U require reference to historical data for optimization, therefore, this paper is simplified to illustrate three moments.

$$\frac{\partial L^{(3)}}{\partial W} = \frac{\partial L^{(3)}}{\partial o^{(3)}}\frac{\partial o^{(3)}}{\partial h^{(3)}}\frac{\partial h^{(3)}}{\partial W} + \frac{\partial L^{(3)}}{\partial o^{(3)}}\frac{\partial o^{(3)}}{\partial h^{(3)}}\frac{\partial h^{(3)}}{\partial h^{(2)}}\frac{\partial h^{(2)}}{\partial W} + \frac{\partial L^{(3)}}{\partial o^{(3)}}\frac{\partial o^{(3)}}{\partial h^{(3)}}\frac{\partial h^{(3)}}{\partial h^{(2)}}\frac{\partial h^{(2)}}{\partial h^{(1)}}\frac{\partial h^{(1)}}{\partial W} \tag{12}$$

$$\frac{\partial L^{(3)}}{\partial U} = \frac{\partial L^{(3)}}{\partial o^{(3)}}\frac{\partial o^{(3)}}{\partial h^{(3)}}\frac{\partial h^{(3)}}{\partial U} + \frac{\partial L^{(3)}}{\partial o^{(3)}}\frac{\partial o^{(3)}}{\partial h^{(3)}}\frac{\partial h^{(3)}}{\partial h^{(2)}}\frac{\partial h^{(2)}}{\partial U} + \frac{\partial L^{(3)}}{\partial o^{(3)}}\frac{\partial o^{(3)}}{\partial h^{(3)}}\frac{\partial h^{(3)}}{\partial h^{(2)}}\frac{\partial h^{(2)}}{\partial h^{(1)}}\frac{\partial h^{(1)}}{\partial U} \tag{13}$$

It should be noted that the activation function is also nested in the equation, so this is a compound derivative process. Also in this process, the loss of the RNN increases with time, so it cannot be simply partial derivatives for time t only.

The generalizations of W and U can be derived from the above equations and analysis.

$$\frac{\partial L^{(t)}}{\partial W} = \sum_{k=0}^{t}\frac{\partial L^{(t)}}{\partial o^{(t)}}\frac{\partial o^{(t)}}{\partial h^{(t)}}\left(\prod_{j=k+1}^{t}\frac{\partial h^{(j)}}{\partial h^{(j-1)}}\right)\frac{\partial h^{(k)}}{\partial W} \tag{14}$$

$$\frac{\partial L^{(t)}}{\partial U} = \sum_{k=0}^{t}\frac{\partial L^{(t)}}{\partial o^{(t)}}\frac{\partial o^{(t)}}{\partial h^{(t)}}\left(\prod_{j=k+1}^{t}\frac{\partial h^{(j)}}{\partial h^{(j-1)}}\right)\frac{\partial h^{(k)}}{\partial U} \tag{15}$$

The optimization process of parameter V is slightly simpler and the idea is the same as that of W and U.

$$L = \sum_{t=1}^{n} L^{(t)} \tag{16}$$

$$\frac{\partial L}{\partial V} = \sum_{t=1}^{n}\frac{\partial L^{(t)}}{\partial o^{(t)}}\cdot\frac{\partial o^{(t)}}{\partial V} \tag{17}$$

#### 4.1.3 Problems with RNN

Further analysis of the above equation, since, as stated earlier, the activation functions are nested within it, taking out the relevant parts from the equation for observation shows that

$$\prod_{j=k+1}^{t} \frac{\partial h^j}{\partial h^{j-1}} = \prod_{j=k+1}^{t} \tanh' \cdot W_s \tag{18}$$

Cumulative multiplication of activation function derivatives leads to "gradient vanishing" and "gradient explosion" problems. The gradient explosion can be solved by setting a threshold and scaling the gradient vector, i.e. gradient pruning. For the gradient disappearance problem, we can improve the problem by replacing the Relu activation function, but this does not solve the problem fundamentally.

To sum up, we use LSTM (Long Short Time Memory Network) to solve the problem.

### 4.2 LSTM

LSTM is a sequential-based recurrent neural network structure, which is an improved version of RNN and can solve the problem of gradient disappearance and gradient explosion very well. Its structure is shown in Figure 3.

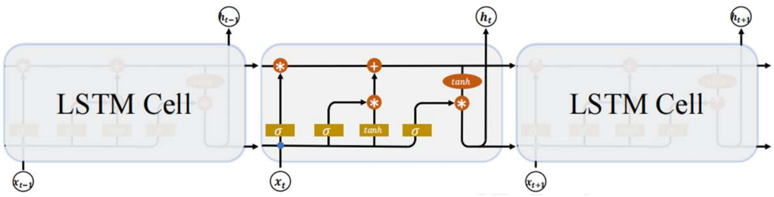


**Figure 3** LSTM schematic

Here in Figure 4, one of the inputs and outputs of the LSTM at moment t is extracted for analysis in this paper.

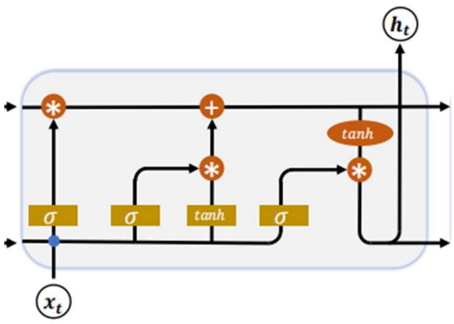


**Figure 4** Input-output diagram at time t

It can be seen that LSTM has one more cell state than RNN, which is used to save important information. It can also be seen from the figure that it runs directly on the chain, making important information to be preserved during the run. As for how the unimportant information is handled, the LSTM sets up three types of gate mechanisms to solve this problem.

#### 4.2.1 Oblivion Gate

This gate is responsible for controlling the oblivion of the state of content of cells in the previous layer, with $h_{t-1}$ of the previous sequence and $X_t$ of the present sequence as inputs, through a sigmoid activation function, to retain the valid information in the previous layer of cell states and forget the rest of the unimportant information. It is worth noting that the inputs are in vector form, which is evident from the figure that when $f_t$ is 0, the information is forgotten in its entirety, when $f_t \in (0, 1)$, some of the information is forgotten, and when $f_t$ equal to 1, the information is retained in its entirety. Therefore, in this paper, the sigmoid function is used as the activation function since the result of this function approaches 0 or 1 for most of the range of values, which is consistent with the forgetting gate mechanism.[7]

#### 4.2.2 Update Gate

The role of this gate is to process the input of the current sequence. There are two parts to this process, one of which is, this part determines which new information can be added to the current information state. The other part is the $\tilde{c_t}$ that can be considered to be brought by the current input and is restricted to the range (-1, 1) by the tanh function.

The new information is then updated by multiplying the forgetting gate given by, thus selectively forgetting the past information, and the same is done on the right side by multiplying the new $\tilde{c_t}$ and $i_t$ to take the new information, and finally combining the two to complete the state update.

#### 4.2.3 Output Gate

Finally, the output $h_t$ of the LSTM is determined by the content of the information stored in the cell state. At this point, after the first two gates have been updated, the sigmoid activation function determines the part of the output that needs to be scaled by the tanh activation function content, and then the two are multiplied together to create the final output $h_t$.

In this way, LSTM solves the problem of gradient explosion and gradient disappearance through its own forgetting mechanism and information updating mechanism

## 5. Bidirectional LSTM forecasting model

Both the above RNN model and LSTM model can only give the output of current timing based on the information of past timing. However, it is not known whether the future timing information affects the input of the current timing, so a bidirectional LSTM model is built in this paper.

The bidirectional LSTM is equivalent to having two recurrent neural networks, one trained from front to back and one trained from back to front, and the two networks are connected by the same output layer, so the output layer can get the past information as well as the future information.

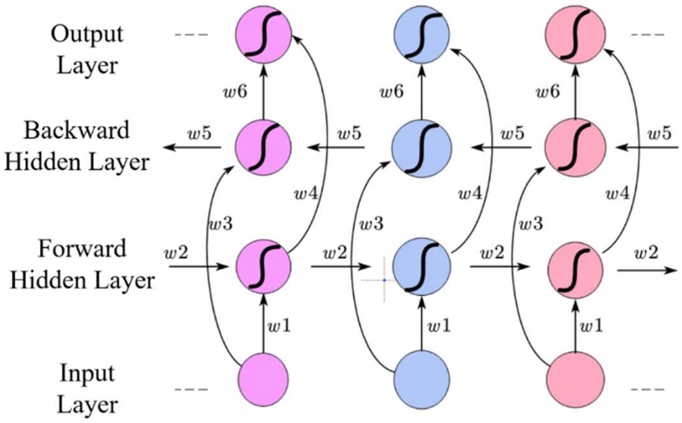


**Figure 5** Bidirectional LSTM schemetic

The above Figure 5 shows an unfolded bidirectional RNN network, where six weights are reused, with the forward hidden layer performing forward computation and the backward hidden layer performing backward computation, and combining the results of the two layers to obtain the final output. It is important to note that there is no information flow between the forward hidden layer and the backward hidden layer.

The principle equation is as follows:

$$\mathrm{h_t} = f(\omega_1 x_t + \omega_2 \mathrm{h_{t\text{-}1}}) \tag{19}$$

$$h'_t = f(\omega_3 h_t + \omega_5 h'_{t+1}) \tag{20}$$

$$o_t = g(\omega_4 h_t + \omega_6 h'_t) \tag{21}$$

## 6. Empirical Analysis

Based on the opening price, closing price, high price, low price, volume and amount for every 5 minutes from July 14, 2021 to January 28, 2022, this paper uses VMA, VMACD, BBI, MA, EXPMA and other indicators that have strong correlation with volume and closing price to forecast prices by the three forecasting methods introduced above, respectively, and obtains the following model effects.

**Table 1** Integrated learning model effect

| Data Set | MSE | RMSE | MAE | MAPE | $R^2$ |
|---|---|---|---|---|---|
| Train Set | 91874232795905.95 | 9585104.74 | 266300.78 | Inf | 0.93 |
| Test Set | 1477143744016117.00 | 38433627.78 | 18978292.65 | 172.99 | -23.72 |

**Table 2** LSTM model effect

| Data Set | MSE | RMSE | MAE | MAPE | $R^2$ |
|---|---|---|---|---|---|
| Train Set | 6134295788286.02 | 2476751.05 | 2517158.59 | 10.66 | 0.41 |
| Test Set | 47018076364907.86 | 6856972.83 | 1755436.25 | 20.09 | 0.21 |

**Table 3** Bidirectional LSTM model effect

| Data Set | MSE | RMSE | MAE | MAPE | $R^2$ |
|---|---|---|---|---|---|
| Train Set | 4877136184561.87 | 2208423.91 | 783134.03 | 15.22 | 0.55 |
| Test Set | 28880878806819.02 | 5374093.30 | 1568298.50 | 20.69 | 0.51 |

The Tables 1,2,3 show that the integrated learning performs well on the training set, but performs poorly on the test set, indicating that the model is severely overfitted. The two-way LSTM time series prediction volume model has lower MSE, RMSE, MAE and higher R2 than the LST M time series prediction volume model, and the MAPE is only slightly higher than the LSTM time series prediction volume model, so the two-way LSTM time series prediction is the best.
The backtesting results of the model using two-way LSTM for volume forecasting are shown in Figure 6 and Figure 7.

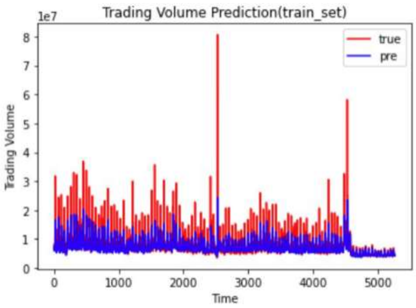


**Figure 6** Prediction of Train Set

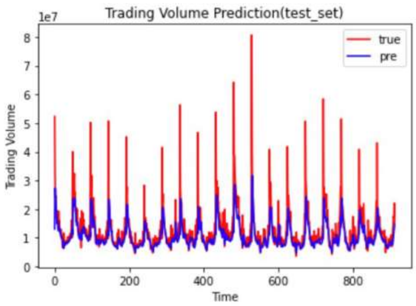


**Figure 7** Prediction of Test Set

## 7. Conclusion

When the trading time of quantitative investment is unit, the amount of data is large, and among the integration learning and deep learning methods selected in this paper, bidirectional LSTM has the best prediction effect. The bidirectional LSTM quantitative investment model established in this paper can help investors to make decision aids, and the model backtesting effect is good and has some practical significance.

## Reference


[1] Zhang Xiaoyan, Zhang Yuan. Analysis of the development and impact of quantitative investment in China. Tsinghua Financial Review. (2022) No.1:P44

[2] He Longchao. Analysis of the development potential of quantitative investment in China. Technology and Economic Market. (2020)No.06:P60

[3] Dong Yanjun, Wang Xiaotian, Ma Hongming etc. Electricity load forecasting method based on random forest and long and short-term memory network. Global Energy Internet. (2022) No.5 Vol.2:P148-P149

[4] Zeng Jingxiang, Zhang Jinxi, Cao Dandan etc. Intelligent detection of asphalt pavement leveling using KNN method. Journal of South China University of Technology (Natural Science Edition). (2022) No.50 Vol.03:P51

[5] Ouyang H.B., Huang Q., Yan H.J.. Financial time series forecasting based on LSTM neural network. China Management Science. (2020) No.4:P27-35.

[6] Su, Z., Lu, M., Li, D. X.. Empirical financial applications of deep learning: dynamics, contributions and perspectives. Finance Research. (2017) No.5:111-126.

[7] Kim H Y., Chang H W. Forecasting the Volatility of Stock Price Index: a Hybrid Model Integrating LSTM with Multiple GARCH-Type Models.Expert Systems with Applications.(2018) 103(01):25-37