INTERNATIONAL JOURNAL OF CIRCUITS, SYSTEMS AND SIGNAL PROCESSING
DOI: 10.46300/9106.2022.16.79 Volume 16, 2022

Financial Time Series Prediction Based on XGBoost and Generative Adversarial Networks

Jialing Xu, Jingxing He, Jinqiang Gu, Huayang Wu, Lei Wang, Yongzhen Zhu, Tiejun Wang, Xiaoling He*, Zhangyuan Zhou
School of Science, Zhejiang University of Science and Technology,
Hangzhou, 310023
China
*Correspondence: 1520217202@qq.com

Received: July 1, 2021. Revised: December 30, 2021. Accepted: January 14, 2022. Published: January 15, 2022.

Abstract—Considering the problems of the model collapse and the low forecast precision in predicting the financial time series of the generative adversarial networks (GAN), we apply the WGAN-GP model to solve the gradient collapse. Extreme gradient boosting (XGBoost) is used for feature extraction to improve prediction accuracy. Alibaba stock is taken as the research object, using XGBoost to optimize its characteristic factors, and training the optimized characteristic variables with WGAN-GP. We compare the prediction results of WGAN-GP model and classical time series prediction models, long short term memory (LSTM) and gate recurrent unit (GRU). In the experimental stage, root mean square error (RMSE) is chosen as the evaluation index. The results of different models show that the RMSE of WGAN-GP model is the smallest, which are 61.94% and 47.42%, lower than that of LSTM model and GRU model respectively. At the same time, the stock price data of Google and Amazon confirm the stability of WGAN-GP model. WGAN-GP model can obtain higher prediction accuracy than the classical time series prediction model.

Keywords—GRU, LSTM, Stock Forecasting, WGAN-GP, XGBoost.

I. Introduction

THE stock market is a complex system, and its volatility is closely related to the economy. Effective forecasting is of great significance to the government, investment institutions and individual investors. In the stock market forecast, researchers have conducted a lot of research on the nonlinear and chaotic characteristics of financial time series, and proposed Auto-Regressive and Moving Average model (ARMA), Auto-Regressive Integrated Moving Average Model (ARIMA), Generalized Auto-Regressive Conditional Heteroskedasticity (GARCH) and other traditional statistical prediction methods [1], these methods use the principle of random process, relying on the past values of the sequence to analyze and predict the future values of the data. In the field of traditional machine learning, researchers have proposed a variety of financial time series prediction methods including Support Vector Machine (SVM), XGBoost and LSTM [2]-[5]. Zhang et al. [6] compared the SVM method with the Back Propagation (BP) method and found that the former is better. Huang et al. [7] consider that the LSTM model is more accurate than the XGBoost model. Sawon et al. [8] found that the XGBoost method was superior to the traditional linear regression method and LSTM in the prediction of large data sets. More research on traditional machine learning for predicting financial time series is available in the literature [9]-[11]. Most of the traditional machine learning methods above use supervised learning to process financial time series data, which needs to classify the high-dimensional financial data subjectively, therefore, there are some defects in using supervised learning method. In the field of deep learning, the generation of adversarial networks using unsupervised learning has become an important branch in this field with its unique idea of network game, has been widely used in the fields of video, image, fraud detection and natural language processing [12]-[14]. Yu et al. [15] proposed a network architecture for the detection of credit card fraud, Zhou et al. [16] proposed a GAN network structure for medical fraud, Xie et al. [17] proposed a counter-network model to improve user profile quality, Wang et al. [18] used GAN model to predict Shanghai-shenzhen stock index, and found that it can effectively improve the prediction accuracy. Ricardo and Carrillo [19] proposed to use GAN model to predict the trend of stock price, and constructed a multi-layer perception (MLP) as a generator, the recurrent neural network is used as the network structure of the discriminator. The results show that the GAN model can improve the prediction performance, but the gradient collapse will occur during the error training, it is found that the original GAN model is used in most of the above-mentioned literatures, and the feature variables are not well optimized, so there are some problems such as unstable training and low prediction accuracy, more research on deep learning prediction methods can be found in [20]-[24].

Considering that GAN has few applications in financial time series forecasting and has the characteristics of modeling high-dimensional nonlinear data, this paper is devoted to apply GAN to financial time series forecasting. Firstly, the XGBoost method in traditional machine learning is used to optimize the feature factor, and the optimized factor is applied to the gate current unit (GRU) to overcome the gradient explosion and gradient disappearance of the Recurrent Neural Network (RNN) [25], a WGAN-GP model with GRU as generator (G) and convolutional neural network (CNN) as discriminator (D) is established, and the prediction performance of the WGAN-GP model is compared with that of the classical time prediction model (LSTM) and GRU, to explore whether WGAN-GP can further improve the prediction effect.

II. Algorithm Design

A. Basic Theory of the Model

(1) Extreme gradient boosting model

XGBoost [26] is composed of classification and regression tree (CART), and uses gradient tree boosting (GTB) to implement a multi-tree ensemble learning algorithm. The sum of the predicted values for the sample for each CART is the predicted values for the XGBoost model, which is defined as follows:

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i) \tag{1}$$

Among them, K represents the total number of trees, $\hat{y}_i$ refers to the prediction result of the i sample, and $f_k(x_i)$ refers to the prediction result of the k sample in the x_i number.

The objective function for XGBoost is:

$$\text{Obj} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t)}\right) + \sum_{k=1}^{K} \Omega(f_k) \tag{2}$$

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda\|\omega\|^2 \tag{3}$$

The objective function consists of two parts, the loss function and the regularization term, $\sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t)}\right)$ is the loss function, used to measure the difference between the real result and the predicted result, and $\sum_{i=1}^{t} \Omega(f_k)$ is the regularization term, which can effectively prevent the tree structure from being too complicated and avoid the model from over fitting. In the regularization term, the first term γT is used to control the complexity of the tree, and the second term $\frac{1}{2}\lambda\|\omega\|^2$ is used to control the weight fraction of the leaf nodes.

Next, we discuss how XGBoost finds the optimal eigenvalues in the process of training objective function optimization from the angle of formula. The process of generating XGBoost tree is top-down, starting from the root node and splitting by feature selection, that is, one node is split into two sub-nodes, in the training process of XGBoost model, Gain before and after splitting is calculated to determine whether the node is split or not. Finally, after training, the importance of the output features of the model, that is, the importance of features score. Any machine learning problem can be started with an objective function. We first give the objective function of XGBoost in the s iteration as follows. The relevant derivation can be found in the references [27], [28]:

$$\text{Obj}^{(s)} = \sum_{j=1}^{T}\left[\left(\sum_{i\in I_j} g_i\right)\omega_j + \frac{1}{2}\left(\sum_{i\in I_j} h_i + \lambda\right)\omega_j^2\right] + \gamma T \tag{4}$$

In (4), I_j is the sample set of the leaf node j, that is, all the samples that fall on the leaf node j, ω_j is the fraction of the leaf node j, the first derivative $g_i = \partial_{\hat{y}^{(t-1)}} L\left(y^i, \hat{y}^{(t-1)}\right)$ is the first gradient statistics of the loss function, the second derivative $h_i = \partial^2_{\hat{y}^{(t-1)}} L\left(y^i, \hat{y}^{(t-1)}\right)$, γT is the coefficient and the number of leaf nodes, respectively, which is used to control the complexity of the tree. The larger the value is, the larger the objective function is, and thus the complexity of the model is restrained, the optimal ω_j^* of leaf node j is:

$$\omega_j^* = -\frac{G_j}{H_j + \lambda} \tag{5}$$

In (5), $G_j = \sum_{i\in I_j} g_i$, $H_j = \sum_{i\in I_j} h_i$, the fraction of the leaf node not only depends on the second derivative information of the first order, but also relates to the coefficient λ. The smaller the λ, the lower the fraction of the leaf node. The optimal objective function value is:

$$\text{Obj}^{(s)} = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j + \lambda} + \gamma T \tag{6}$$

In (6), The optimal objective function value is not only affected by λ, but also related to the γT of the complexity of control tree. Next is how to determine the optimal feature and cut point, assuming that the current split node is j, then its objective function is:

$$\text{Obj}_j = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j + \lambda} + \gamma \tag{7}$$

When the node splits, the objective functions of the two child nodes are:

$$\text{Obj}_s = -\frac{1}{2}\left(\frac{G_{jL}^2}{H_{jL} + \lambda} + \frac{G_{jR}^2}{H_{jR} + \lambda}\right) + 2\gamma \tag{8}$$

The objective function changes before and after the node splitting are obtained by subtracting the above formula (7) from (8), and the formula is as follows:

$$\mathrm{Obj}^{*}=\frac{1}{2}\left(\frac{G_{\mathrm{L}}^{2}}{H_{\mathrm{L}}^{2}+\lambda}+\frac{G_{\mathrm{R}}^{2}}{H_{\mathrm{R}}^{2}+\lambda}-\frac{G^{2}}{H+\lambda}\right)-\gamma \tag{9}$$

In (9): G and H are the first and second order statistics of the current node, $\mathrm{G_L}$ and $\mathrm{G_R}$ are first-order statistics of left and right nodes respectively, $\mathrm{H_L}$ and $\mathrm{H_R}$ are second-order statistics of left and right nodes respectively. In the process of splitting each node, all the features and cut points of Obj^{*} are calculated, and the largest features and cut points of Obj^{*} are selected as the optimal features and cut points, so the model can output the importance of features after training. Using XGBoost for feature selection, we can not only explain the features, but also select the features after sorting to improve the fitting effect of the following models.

(2) Generative adversarial network model

GAN [29] consists of two parts, generator G and discriminator D, whose job is to produce examples that look as real as possible, and whose goal is to distinguish between real and fake examples (data generated by G). The two cooperate to achieve Nash Equilibrium, that is, the generator finally generates the same data distribution as the real sample, and the discriminator can not distinguish whether the data is from the real data or the generated data. The WGAN-GP [30] model used in this paper adds a penalty term (gradient penalty, GP) [31] to the GAN to solve the gradient collapse of GAN. The objective function is:

$$\begin{aligned}L_{\mathrm{WGAN\_GP}}=&E_{x\sim P_g}\left[D\left(x\right)\right]-E_{x\sim P_r}\left[D\left(x\right)\right]\\&+\lambda E_{x\sim P_x}\left[\left(\left\|\nabla_x D\left(x\right)\right\|_2-1\right)^2\right]\end{aligned} \tag{10}$$

In the formula, P_r is the actual data distribution, P_g is the generated data distribution, $\lambda E_{x\sim P_x}\left[\left(\left\|\nabla_x D\left(x\right)\right\|_2-1\right)^2\right]$ is the gradient penalty, λ is the gradient penalty coefficient, $\left(\left\|\nabla_x D\left(x\right)\right\|_2-1\right)^2$ is a non-negative number, our goal is to minimize loss function, Then make $\left\|\nabla_x D\left(x\right)\right\|_2$ stable near 1. If the gradient norm of the model deviates from its target norm value 1, the model will be punished. This method can effectively solve the problems of gradient dispersion, slow and unstable training process.

In the training process of G and D in this article, the loss function of D is defined as:

$$L_{\mathrm{D}}=\frac{1}{m}\sum_{i=1}^{m}\left[D\left(y^{i}\right)-D\left(G\left(x^{i}\right)\right)+\lambda E\left(\left\|\nabla D_{y^{j}\,x^{i}}\right\|_2-1\right)^2\right] \tag{11}$$

In(11), m is the size of the small batch of samples, that is, the sample size that enters the training at one time, x is the input data of G, $G\left(x^{i}\right)$ represents the data generated from G, and y is the target input of the real data set. In this article, y refers to Alibaba's closing price sequence, the loss function of G is [32]:

$$L_{\mathrm{G}}=-\frac{1}{m}\sum_{i=1}^{m}D\left(G\left(x^{i}\right)\right) \tag{12}$$

In the formula, the loss function of G only relies on the output value of D to adjust the parameters. When D is judged incorrectly, the data of G will be biased, and the loss function will become unstable. In order to solve this problem, we need first to train well discriminator D, so train D many times before training G to improve the stability of the model.

B. Algorithm Network Structure

The structure of the algorithm is shown in Figure 1. It includes Data pre-processing, feature optimization of the XGBoost model, generator and discriminator. Generator G is a gated loop unit. The difference between Generator G in this paper and that in the original GAN model is that the preprocessed data is not directly input into generator G, instead, the XGBoost model is used to optimize the evaluation index and technical index and input it into G. Finally, the output of G is given to discriminator d together with the original data set. The advantage of this method is that the feature factor input into g can reduce the complexity of g and improve the generalization ability of the model. At the same time, in a multi indicator evaluation system, different evaluation indicators usually have different dimensions and orders of magnitude, that is, the unit or value range of different characteristics may be different, for example, the range of trading volume is million to ten million, however, the closing price ranges from 10 to 100. When the level of each index varies greatly, if the original index value is directly used for analysis, the role of the index with higher values in the comprehensive analysis will be highlighted, relative weakening of the role of lower numerical level indicators. Therefore, in order to ensure the reliability of the results, reduce the noise of abnormal data in the sequence, and improve the convergence speed of the model iteration, all feature sequences and target sequences are normalized in the data preprocessing part:

$$X_{\mathrm{n}}=\frac{X-X_{\min}}{X_{\max}-X_{\min}} \tag{13}$$

In (13), X_{n} is the standardized data, X is the optimized data after the XGBoost algorithm, $X_{\min}$ and $X_{\max}$ are the minimum and maximum of the optimized data respectively.

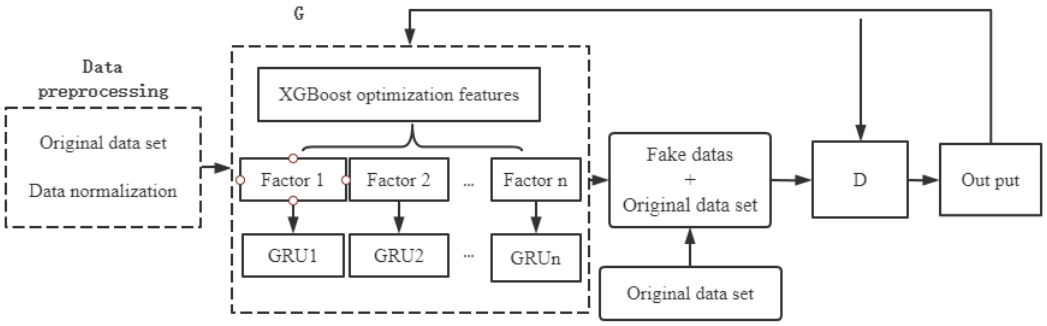

Figure 1. Algorithm diagram

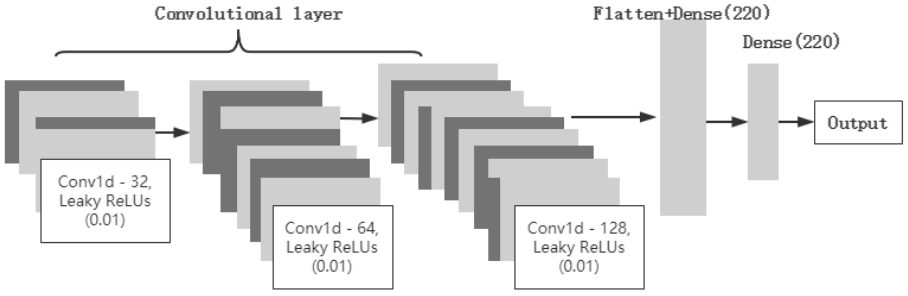

Figure 2. Convolutional neural network

Discriminator D is a recurrent neural network whose network structure is shown in Figure 2. CNN is a deep neural network consisting of an input layer, a plurality of hidden layers and an output layer. The hidden layer consists of a convolutional layer, a flattened layer and a fully connected layer, then reduce the dimensions to the full connectivity layer output. The function of D is equivalent to a binary classifier, used to distinguish the input data is generated by the generator of false data or real data, so the discriminator must have a good classification ability, CNN has a good performance in classification accuracy, it is widely used in image classification task [33], so the discriminator is CNN.

C. Parameter Training

The XGBoost model uses the exact greedy algorithm to find the optimal features and the optimal cut-off points [34-37]. The training process is as follows:

(1) Initialize the split payoffs and gradient statistics, $gain = 0, G = \sum_{i \in I} g_i, H = \sum_{i \in I} h_i$;

(2) For each feature $k = 1,2,\ldots,m$, the following steps are performed:

First initialize the first-order gradient statistics and the second-order gradient statistics of the left subnode, $G_{\mathrm{L}} = 0, H_{\mathrm{L}} = 0$, and then sort the values of all samples contained in the node under this feature, after sorting according to $G_{\mathrm{L}} = G_{\mathrm{L}} + g_j, H_{\mathrm{L}} = H_{\mathrm{L}} + h_j$ and $G_{\mathrm{R}} = G - G_{\mathrm{L}}, H_{\mathrm{R}} = H - H_{\mathrm{L}}$ this way to the left sub-node and the right sub-node of the gradient statistics, and then calculate the final split after the income, choose the largest income.

$$gain = \max\left(gain, \frac{G_{\mathrm{L}}^2}{H_{\mathrm{L}} + \lambda} + \frac{G_{\mathrm{R}}^2}{H_{\mathrm{R}} + \lambda} - \frac{G^2}{H + \lambda} \right) \tag{14}$$

(3) Finally, find the most profitable feature and split the split point. The MODEL iterates the process until it stops.

The generated adversarial network model and LSTM and GRU models are all trained by Adam algorithm [38]. The specific steps are:

(1) First initialize the learning rate of generator G and discriminator D, the weight parameter θ and the hyper parameter λ in the penalty term,

(2) And then determine whether the model has converged, if there is no convergence, update the weight θ in G according to the Adam algorithm, $\theta \leftarrow \left(\nabla_\theta L_{\mathrm{G}}^{new} \right)$,

(3) And then update the weight $w \leftarrow \left(\nabla_{\mathrm{w}} L_{\mathrm{D}} \right)$ in D,

(4) And repeat this process until the model converges to the end.

The WGAN-GP model is generated by GRU. The number of GRU neurons is 1024 and 512. To avoid over fitting, the drop out value is 0.02, the learning rate is 0.0001, the small sample size is 128, the number of training of epochs is 100 iterations, then two full-connection layers are added, and the number of full-connection layers in the last layer is consistent with the output step to be predicted. The discriminator consists of three convolution layers, the convolution layer consists of three one-dimensional convolution layers, the number of neurons are 32, 64 and 128 respectively, the flat layer and the fully

connected layer consist of 220 neurons. All layers except the output layer are set to Leaky Rectified Linear Units (Leaky ReLUs) as the activation function, the alpha is 0.01. The output layer sets a linear activation function and gives a scalar score at the end, at the same time, the same discarded value as the generator is used to prevent over-fitting. In the LSTM model, the learning rate is 0.0001, the batch size is 64, and the number of training of epochs 60 iterations. In the GRU model, the learning rate is 0.0005, the batch size is 256, and the number of training of epochs 50 iterations.

The generator G is a gated loop unit. The difference between the generator G in this article and the G in the original GAN model is that instead of directly inputting the preprocessed data into the generator G, the XGBoost model is used to estimate the index and the technical indicators are optimized and input into G, and finally the output of G is given to the discriminator D together with the original data set. The advantage of this is that the feature factors filtered by the XGBoost model are input into G, which can reduce the complexity of G and improve the generalization ability of the generated model.

This article uses CNN as the discriminator D, and its network structure is shown in Figure 2. CNN is a deep neural network that includes an input layer, multiple hidden layers, and an output layer. The hidden layer in this article includes convolutional layer, flattening and fully connected layer. First, input the feature sequence, then extract the features through the convolutional layer, and then reduce the dimension and output through the fully connected layer. Among them, the convolutional layer is composed of three one-dimensional the convolutional layer is composed of 32, 64, and 128 neurons. Finally, a flattened and fully connected layer is added. It is composed of 220 neurons. All layers except the output layer are set to Leaky Rectified Linear Units (Leaky ReLUs) as the activation function, the alpha is 0.01. The output layer sets a linear activation function and gives a scalar score at the end. The function of D is equivalent to a binary classifier, which is used to distinguish whether the input data is fake data generated by the generator or real data, therefore, the discriminator must have a good classification ability, CNN has a good performance in the classification accuracy, is widely used in the image classification task, so we choose CNN as the discriminator.

D.Evaluation Indicators

In order to compare the out-of-sample prediction effects of the LSTM, GRU model and the generative adversarial network model, this paper uses the root mean square error to express, the evaluation index calculation formula is as follows:

$$R_{\mathrm{MSE}}=\sqrt{\frac{\sum_{i=1}^{n}\left(x_i-x_i'\right)^2}{n}} \tag{15}$$

In (15), n is the number of real data samples, x_i is the real closing price, and x_i' is the predicted closing price.

III. Data Interpretation And Preprocessing

A. Data Set

The financial time series data collected in this paper are all from the financial WIND database. The data set is Alibaba's stock price data, valuation indicators, technical indicators, and stock price trends extracted according to Fourier transform from September 2014 to September 2021.

The stock price data is the opening price, closing price, highest price, lowest price and trading volume.

Valuation indicators include price earnings ratio (PE) and price-to-sales ratio (PS), PE is the ratio of stock price to earnings per share and is one of the most important measures of the value of a stock. PS is the price of a stock divided by sales per share and can be used to determine the value of a stock relative to past performance.

Technical indicators include overbought and oversold factors and trend factors. Overbought and oversold factors include price momentum and Bollinger bands in different time periods in the past, trend factors including moving averages of similarity and differences, moving averages of different lengths of time, and exponential moving averages.

Based on the daily closing price, we created the Fourier transform to extract the long term and short term trends of Alibaba shares. The Fourier transform takes a function and creates a series of sine waves that, when combined, approximate the original function, which helps the GRU choose its predicted trends more accurately.

The total number of features collected in the original data is 23, we need to pre-process the original data, for the downloaded financial stock data, we need to check the outliers and missing values. For the missing values, the method we use is to take the average of two adjacent numbers, then normalize the data, and divide the training set and the test set. The predicted stock in the model is Alibaba, with a total of 1747 observations and 23 characteristic variables, and the training set is separated from the test set by a dashed line, as shown in Figure 3. The test data set started on September 3, 2019.

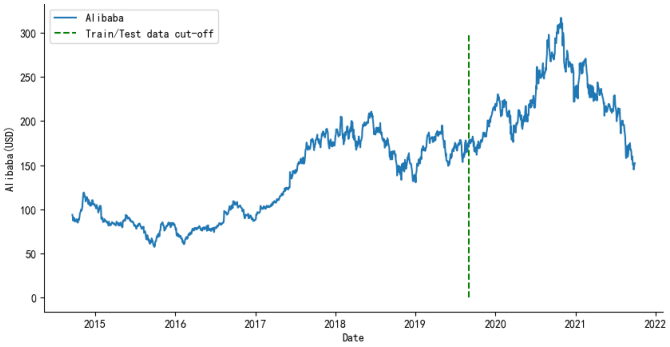

Figure 3. Training set and test set

B. Data Structure

In this section we will show the data set structure of the input to generator g and the output structure after generator G. The original data is a two-dimensional structure with $x_0, x_1, x_2, \ldots, x_m$

as the number of features and $t_0,t_1,t_2,\ldots,t_m$ as the time. The original shape of the data is (1747,23), where 1747 is the number of samples and 23 represents the number of features per sample, in order to predict the future trend of stock prices, the original data must be reconstructed into a three-dimensional structure, that is, an additional dimension of the time step. The data set after the creation of the time step is transformed into a three-dimensional structure (1747,30,23), where 30 refers to the time step parameter, to use data from the past 30 days to predict future data, we use a scroll window equal to 1 to divide the data set, as shown in Figure 4.

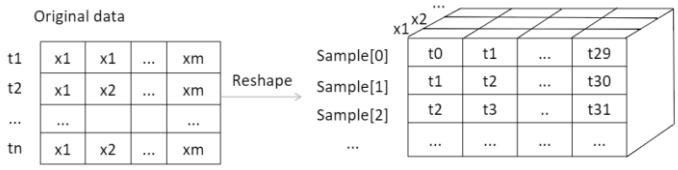

Figure 4. Input data structure

IV. Test Results And Analysis

A. Model Training Results

The following results were obtained by training XGBoost and WGAN-GP in the manner described in section 1.3. First of all, the model of XGBoost selects the feature factor. The parameters of the model are adjusted by grid search method [39]. Some parameters are: the maximum depth of the tree is 6, the learning rate is 0.1, the maximum number of iterations is 50, and the objective function is logistic, the maximum depth of the tree means that the decision tree will stop splitting when the number of splits is greater than Max. After training, the model gets the importance rank of each feature, as shown in Figure 6. Feature importance indicates the importance or value of each feature in the model. The more a feature makes key decisions in the decision tree, the more important the feature is, that is, the more times a feature is used to build a decision tree in the model, the higher its score, with the vertical axis showing the names of the features and the horizontal axis showing the importance of the features, with the MACD as a split feature 290 times.

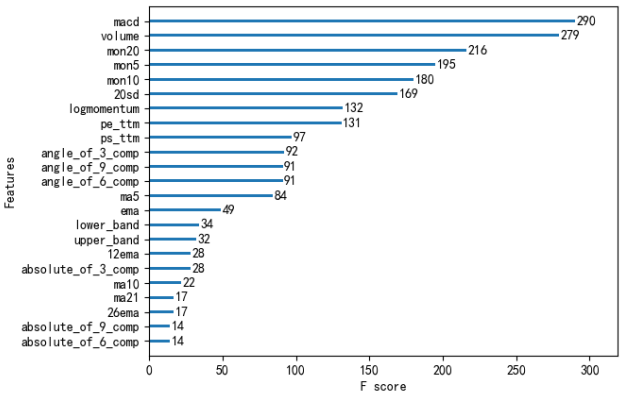

Figure 5. Feature importance score ranking

After each feature importance score is obtained, the feature importance can be selected by testing several thresholds and by measuring the area under the curve (AUC) [40]. There are two kinds of parameters to set the threshold, the median value and the average value of feature importance. When the value of feature importance is greater than or equal to the threshold value, the feature is extracted, otherwise it is discarded. The final result is shown in Figure 6. As you can see, when the number of features is 19, the AUC value of the test set is 83.01%, thus reducing the number of original features from 23 to 19.

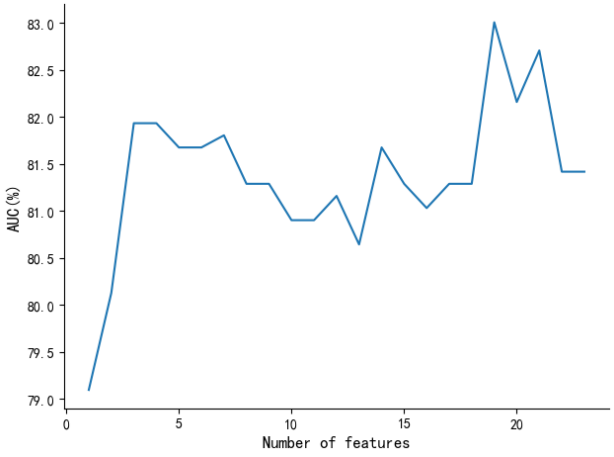

Figure 6. Model prediction accuracy under different number of features

Figure 7 below shows the training errors of generator G and discriminator D in the WGAN-GP model, and the blue line is the loss path of the discriminator, and the orange line is the loss path of the generator. We can see that the loss function of discriminator decreases to zero as the number of iterations increases.

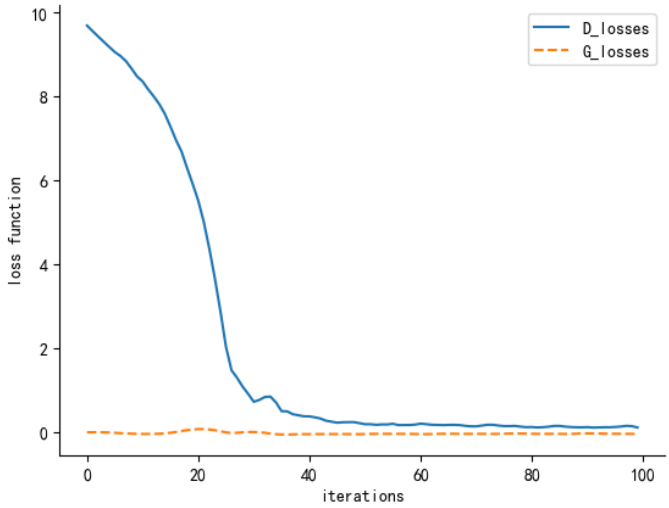

Figure 7. Loss function of WGAN-GP

B. Comparison of Results of Different Methods

Figure 8. LSTM model test set results

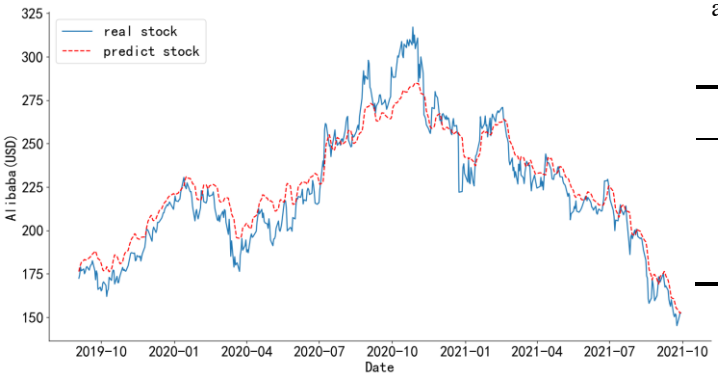

Figure 9. GRU model test set results

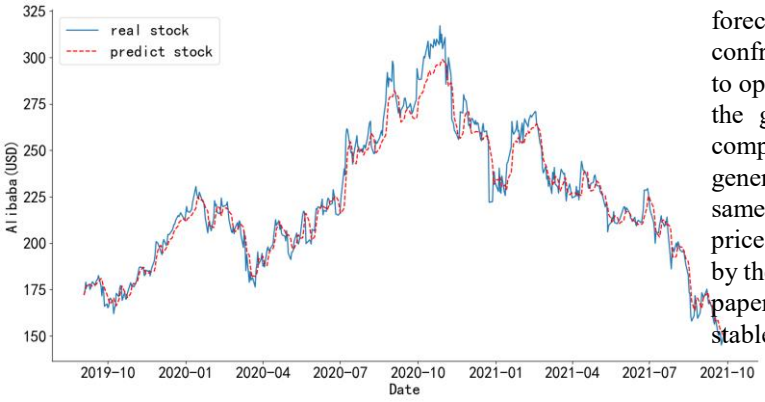

Figure 10. WGAN-GP model test set results

Figures 8, 9 and 10 show the predicted and true values of the LSTM, GRU, and WGAN-GP model, respectively. It can be seen that the prediction trend of the three models outside the sample is basically the same as that of the real data, but WGAN-GP model can better fit the trend of the real data as a whole compared with the LSTM and GRU. The fitting effect of LSTM model and GRU model is far less than that of WGAN model. With the increase of the amount of data out of the sample, in a period of 2020, the three models all deviate from the real data, but the LSTM and GRU models both deviate more than the WGAN model. After 2020, the prediction error of the three models decreases, the predictive power of the WGAN model is still the best. In addition, compared with the GRU model, the GRU model is slightly better than the LSTM model. After 2020, the prediction error of the GRU model is less than that of the LSTM model. The results show that the prediction data of WGAN-GP model fits the real data as a whole, and there is no major deviation, it proves a strong predictive ability.

C. Model Evaluation

In order to verify the effectiveness of this model, the stock price data of Google and Amazon are selected for further empirical research according to the same method. Table 1 below compares the root-mean-square errors of the three stocks outside the samples of the three models. It can be seen that the WGAN-GP model can still best capture the nonlinear characteristics of financial sequences, and the prediction accuracy is the highest and stable outside the sample.

Table 1. RMSE summary of different models

Model	Alibaba	Google	Amazon
GRU	0.97	0.95	0.98
LSTM	1.34	1.05	0.81
WGAN-GP	0.51	0.49	0.54

V. Conclusion

This paper uses the XGBoost algorithm to optimize the characteristics of the index factors, and further compares the out-of-sample forecasting capabilities of the classic time series forecasting model LSTM, GRU and the generative confrontation network model. We used the XGBoost algorithm to optimize the features of the input value of the generator G in the generative confrontation network model to reduce the complexity of the generative model and improve the generalization ability of the model outside the sample. At the same time, to prove the universality of this model the stock prices of Google and Amazon are selected for the demonstration by the same method. The test results show that the model in this paper has high prediction accuracy, and the model is relatively stable, and the root mean square error is stable at about 0.5.

References

[1] J. J. Wang, J. Z. Wang, Z. G. Zhang and S. P. Guo, “Stock index forecasting based on a hybrid model”, Omega, vol. 40, no. 6, pp. 758, 2012.

[2] L. Han, Z. Su and Z. D. Liu, “Co-dynamic prediction model of Financial Market: DWT-SVM method”, Systems Science and mathematics, vol. 40, no. 12, pp. 2342-2356, 2020.

[3] Y. Wang and Y. K. Guo. “Application of the improved XGBoost model in stock forecasting”, Computer Engineering and Applications, vol. 55, no. 20, pp. 202-207, 2019.

[4] C. Liu and J. P. Du, "A financial data forecasting method based on depth-LSTM and attention mechanism", Computer Science, vol. 47, no. 12, pp. 125-130, 2020.
[5] H. Ying and H. J. Yang, "Financial time series forecasting based on XGBoost and LSTM Models", Technology and Industry, vol. 21, no. 8, pp. 158, 2021.
[6] W. Q. Zhang, "Research on Financial Data Prediction Based on Support Vector Machine Learning", Dalian: Dalian University of Technology, 2020.
[7] W. Huang, Y. Nakamori and S. Y. Wang, "Forecasting stock market movement direction with support vector machine", Computers and Operations Research, vol. 32, no. 10, pp. 2513-2522, 2005.
[8] M. Sawon, M. N. Hosen and S. Chandra, "Sales Forecasting for Retail Chains", India, 2015.
[9] Q. F. Hu, "Credit risk prediction model of financial company based on machine learning algorithm", Electronic Technology and Software Engineering, no. 10, pp. 146-147, 2021.
[10] L. L. Chen, "Application of Machine Learning in Financial Time Series Prediction", Hangzhou Dianzi University, 2020.
[11] T. Xue, S. H. Qiu, H. Lu and X. S. Qin, "Exchange rate prediction based on EMD and multi-branch LSTM network", Journal of Guangxi Normal University, vol. 39, no. 2, pp. 41-50, 2021.
[12] L. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative adversarial nets", Advances in Neural Information Processing Systems, vol. 2, pp. 2672–2680, 2014.
[13] M. K. Yu, "Research on Credit Card Fraud Prediction Model Based on Generative Confrontation Network", Zhengzhou: Henan University of Economics and Law, 2020.
[14] R. R. Chen, "Research on Credit Card Transaction Fraud Detection Model Based on GAN-XGBoost", Hangzhou: Hangzhou Normal University, 2019.
[15] M. K. Yu, "Research on Credit Card Fraud Prediction Model Based on Generative Countermeasure Network", Zhengzhou: Henan University of Finance and Economics, 2020.
[16] T. T Zhou, "Research on Medical Insurance Anomaly Detection Based on Deep Learning", Chengdu: University of Electronic Science and Technology of China, 2019.
[17] J. Z. Xie, "Research on Customer Portrait Model of a Commercial Bank Based on Machine Learning", Chongqing: Chongqing University of Technology, 2020.
[18] J. Wang, H. M. Zou, D. D. Qu and L. Bai, "Prediction of financial time series based on empirical mode decomposition generating antagonism network", Computer applications and software, vol. 37, no. 5, pp. 293, 2020.
[19] A. Ricardo and R. Carrillo, "Generative adversarial network for stock market price prediction", Stanford University, vol. 32, 2019.
[20] Y. Ibrahim, J. Y. Liu, X. X. Yang, H. W. Sha, P. Li and H. B. Wang, "Analyzing the impact of soft errors in deep neural networks on GPUs from instruction level", WSEAS Transactions on Systems and Control, vol. 15, pp. 699-708, 2020.
[21] K. Zhang, G. Zhong, J. Dong, S. Wang and Y. Wang, "Stock market prediction based on generative adversarial network", Procedia Computer Science, vol. 147, pp. 400-406, 2019.
[22] H. J. Yan, "Financial Time series data integration prediction based on deep learning", Forum on Statistics and Information, vol. 35, no. 4, pp. 33-41, 2020.
[23] R. Y. Yuan, "Deep learning based stock forecasting analysis", China's Collective Economy, no. 24, pp. 105-106, 2021.
[24] J. Xu, Y. K. Zhu and C. X. Xing, "Research on financial transaction algorithm based on deep reinforcement learning", Computer Engineering and Applications, pp. 1-11, 2021.
[25] T. Ma, J. Wang, S. W. Ding, J. M. Pan and J. M. Zhu, "A combination forecasting model of interval financial time series from multi-scale perspective", Management and Technology of Small and Medium-sized Enterprises, no. 9, pp. 59-61, 66, 2021.
[26] E. Chong, C. Han and F. C. Park, "Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies", Expert Systems with Applications, vol. 83, 2017.
[27] Y. Y. Zuo, M. F. Wang, J. W. Hong and D. Ma, "A multivariate time series prediction method based on ensemble learning", Journal of Small Microcomputer Systems, vol. 41, no. 12, pp. 2475-2479, 2020.
[28] H. H. Yu and Q. Dai, "Prediction of non-stationary financial time series based on adaptive incremental ensemble learning", Data Acquisition and Processing, vol. 36, no. 5, pp. 1030-1040, 2021.
[29] D. M. Nelson, A. C. Pereira and R. A. Oliveira, "Stock market's price movement prediction with LSTM neural networks", In: International Joint Conference on Neural Networks (IJCNN), 2017, pp. 1419.
[30] J. Wang, H. M. Zou, D. D. Qu and L. Bai, "Financial time series forecast based on empirical mode decomposition generative adversarial network", Computer Applications and Software, vol. 37, no. 5, pp. 293, 2020.
[31] K. Cho, B. Van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation", Association for Computational Linguistics, vol. 1, pp. 1724-1734, 2014.
[32] X. J. Li, C. R. Cui, G. L. Song, Y. Q. Su, T. Z. Wu and C. Y. Zhang, "A stock trend forecasting method based on time series hypergraph convolutional neural network", Journal of Computer Applications, 2021, http://kns.cnki.net/kcms/detail/51.1307.tp.20210806.1415.008.html..
[33] S. Hochreiter and R. Schmidhuber, "Long short-term memory", Neural Computation, vol. 9, no. 8, 1997.
[34] L. Kanaan, J. Haydar, M. Samaha, A. Mokdad and W. Fahs, "Intelligent bus application for smart city based on LoRa technology and RBF neural network", WSEAS

Transactions on Systems and Control, vol. 15, pp. 725-732, 2020.
[35] S. Gochhait, Y. Rimal and S. Pageni, "The comparison of forward and backward neural network model – a study on the prediction of student grade", WSEAS Transactions on Systems and Control, vol. 16, pp. 422-429, 2021
[36] Q. Smith and R. Valverde, "A perceptron based neural network data analytics architecture for the detection of fraud in credit card transactions in financial legacy systems", WSEAS Transactions on Systems and Control, vol. 16, pp. 358-374, 2021.
[37] Q. Liu, "Research on the Prediction Method of Stock Price Changes Based on Neural Network Model", Beijing: Beijing University of Posts and Telecommunications, 2020.
[38] Y. X. Wang, "Stock Forecast Based on Convolutional Neural Network", Tianjin: Tianjin Polytechnic University, 2019.
[39] Y. Hu, "A stock market timing model based on convolutional neural networks-take the shanghai stock exchange index as an example", Financial Economics, no. 4, pp. 71, 2018.
[40] Y. S. Liu, "Empirical Study of Internet Financial Fraud Prevention Based on CNN-XGBoost", Central China Normal University, 2020.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

Jialing Xu simulated and optimized the XGBoost and WGAN-GP models
Jingxing He collected the relevant data.
Jinqiang Gu implemented the Algorithm 2.1.1
Huayang Wu implemented the Algorithm 2.1.2
Lei Wang collected the relevant data and Data pre-processing
Yongzhen Zhu organized and implemented the XGBoost model training
Tiejun Wang organized and conducted the XGBoost experiment on feature selection
Xiaoling He trained the GAN generator
Zhangyuan Zhou did the simulation and statistics.