The Comparsion of Stock Return Prediction for Random Forest, Ordinary Least Square, and XGBoost

Junsheng Wang*

College of Mathematics and Statistics Chongqing University, Chongqing, China

*Corresponding author: 20201867@cqu.edu.cn

Abstract. With the stock market growing larger and the violent fluctuation becoming more frequent after the COVID-19 pandemic broke out, investors and researchers urgently need a method to predict the behavior of the stock market accurately. This research is determined to find out the performance of random forest (RF), XGBoost and ordinary least square (OLS) models in terms of predicting the return of given subjects. This research uses tushare to collect data and Jupyter Notebook to run the models. Libraries such as numpy, pandas, scikit-learn, and stockstats are also used in this paper. According to the analysis, XGBoost and RF model outperformed OLS model in all three subjects and the difference between RF and XGBoost model is subtle. Meanwhile, the results also revealed that the choice of subjects may affect the performance of model. Finally, only technical indicators were included in the process of model setup and this may negatively impact the results. These results shed light on the performance difference of the three models and lay a foundation for future high-efficiency hybrid models.

Keywords: Stock price prediction; Xgboost; Random Forest; Ordinary Least Square; bigdata analysis.

1. Introduction

As the stock market thrives worldwide, finding ways to predict the stock price or the volatility of stock has become unprecedentedly urgent. Generally, investors desire to know whether the price of a stock is going to rise or plummet by analyzing historical data, but the task is challenging due to the fluctuating intrinsic of the stock market and the impact of many factors on it. No trader will buy a stock if he knows that the price will go down tomorrow, thus there is a need for methods to accurately predict the stock price. Many researchers have dedicated their time to solving this problem. Existing prominent methods can be roughly divided into two categories, statistical and soft computing techniques [1].

Two major methods in statistical techniques are autoregressive integrated moving average and (ARIMA) and generalized autoregressive conditional heteroskedasticity (GARCH) while machine learning and deep learning are methods widely used in soft computing techniques. Ayodele revealed in his research in 2014 that the ARIMA model and ANN model achieved similar results considering that their forecast error is both quite low. However, it is also found that the ARIMA model is better at predicting the direction of a stock price movement while the ANN model is better at predicting the value. Yang used twenty factors, including market factors and emotional factors, to set up and compare three models in deep learning (CNN, DNN, LSTM) and three models in classic machine learning (SVM, LR, NB). It found that DNN models outperformed other models [2]. Cao et al. constructed a comprehensive index (CEUI) using the entropy value method to enable the GARCH-MIDAS model to reflect the influence of multi-dimensional economic uncertainty on the exchange rate of RMB [3]. Chen et al proposed to use the Pearson correlation coefficient to extract features of the close price of a stock [4]. The empirical results indicated that using the Pearson correlation coefficient to extract features can significantly lower the prediction error. The research also showed that it is wrong to assume that there is any correlation between two stocks that belong to one parent company. Khaidem et al. proposed to consider the forecasting problem as a classification problem in order to eliminate the forecasting error and claimed that the algorithm of the ensemble of multiple decision trees was shown to outperform existing algorithms found in the literature [5]. Wang & Hao used five technical factors to construct a random forest model and applied grid search with cross-validation to optimize

parameters [6]. Based on the analysis, the GS-RF model can achieve better results than previous trading strategies using technical factors. Xu &n Zhou compared GARCH (1, 1) model, TARCH (1, 1) model, and EGARCH (1, 1) model in their research on the stock price of new energy [7]. They concluded that GARCH (1,1) and TARCH (1, 1) can fit the volatility of the stock price better. Yang et al. combined XGBoost and LightGBM model with 1:1 fusion and found this hybrid model can predict the stock price more accurately than a single XGBoost or LightGBM model [8]. Mitchell & Frank [9] reported implementation of the library XGBoost based on compute unified device architecture (CUDA). They used a graphics processing unit (GPU) to accelerate the XGBoost algorithm and found that GPU made the algorithm three to six times faster than using multicore CPUs on desktop machines. In the research of discrimination power of model selection criteria, Dell' Aquila & Ronchetti mainly focused on the discrimination power of Akaikes' Information Criterion (AIC) and Schwarz's Bayesian Information Criterion (BIC) when R^2 is low [10]. Their research revealed none of these methods performed well in discriminating models when the R^2 is low. Oomen used ARFIMA to model the realized variance time series and found that it outperformed conventional GARCH type models [11]. Engle proposed the dynamic conditional correlation (DCC)model which is as flexible as univariate GARCH but is not as complex as multivariate GARCH in his research in 2000 [12]. According to the results, this model performed well in many times varying correlation processes and often outperformed simple multivariate and other models regardless of the criterion. Zuo et al. proposed to use one Bayesian network to predict the stock price return and add the prediction error data of the first algorithm for determining a new Bayesian network [13]. They chose prediction accuracy and the correlation coefficient to compare these two algorithms with conventional time-series prediction algorithms. It is found that the two algorithms performed 26% and 30% better than conventional time-series prediction respectively. From the literature, it can be seen that little research has been conducted to compare different models' prediction effects on two or more subjects. Meanwhile, filling these blank benefits both investors and researchers. It enables investors to choose appropriate models to gain higher profit and lays a foundation for the possible hybrid models which researchers might develop in the future. Therefore, research on the differences and analysis of the difference is needed. This paper will set up three models (OLS, RF, XGBoost) to predict the prices of three stocks in the A stock market and analyze their difference in the forecasting effects.

The rest part of the paper is organized as follows. The Sec. II will introduce the methodology used in this paper to calculate and sift the technical indicators will be given as well as the processing of the models. Subsequently, the Sec. 3 will present the empirical results. Afterwards, the discussions, explanations as well as limitations of the results will be demonstrated. Eventually, a brief summary will be given in Sec. 5.

2. Methodology

2.1 Data collection and processing

This paper chose three stocks in the A stock market (NDSD, BYD, ZHGD) as subjects. K line data of these companies from 25th, June 2019 to 11th, February 2022 (640 days in total) was collected. Missing values were deleted since their proportion is low in the data set. All data used in this paper was collected from Tushare. Numpy, Pandas, Matplotlib, Sklearn, Stockstats were used to process the data. The program was written and tested on Jupyter Notebook.

The original data obtained from tushare included open price, high price, low price, close price, and volume. Stockstats was applied then to calculate technical indicators, including six-period relative strength index (RSI), 6-period commodity channel index (CCI), six-period KDJ, energy index, moving average convergence/divergence (MACD), and the high price delta between current and two days later. RSI is used to measure buyers' strength and sellers' strength. CCI and KDJ can detect the overbought and oversold trend. MACD is capable of reflecting the volatility of the stock price. Formulae to calculate these indicators are presented as follow.

$$RSI = 100 \times \frac{u}{u+d} \quad (1)$$

Where u means the average value of the rise in stock price during the given period and d means the average value of the drop in stock price during the given period.

$$DIF = EMA(close, 12) - EMA(close, 26) \quad (2)$$

$$DEA = EMA(DIF, 9) \quad (3)$$

$$MACD = DIF - DEA \quad (4)$$

Here, EMA (x, y) means y-period exponential moving average of x.

$$RSV = \frac{Close_t - Low_{[t-n+1,t]}}{High_{[t-n+1]} - Low_{[t-n+1,t]}} \quad (5)$$

Where $Close_t$ means the close price at t day,$High_{[t-n+1]}$ means the highest price from t-n+1 day to t day, $Low_{[t-n+1,t]}$ stands for the lowest price from t-n+1 day to t day. n represents the time span.

$$K = \frac{2}{3} \times K_{t-1} + \frac{1}{3} \times RSV_t \quad (6)$$

Here, K_t and RSV_t represent the K value and RSV at t day, respectively.

$$D_t = \frac{2}{3} \times D_{t-1} + \frac{1}{3} \times K_t \quad (7)$$

Where D_t and K_t mean D value and K value at t day, respectively.

$$J_t = 3 \times K_t - 2 \times D_t \quad (8)$$

Here, J_t, K_t, and D_t stand for the J value, K value, and D value respectively.

The original data usually cannot be fed into models directly. Procedures such as normalization are necessary. This paper used zero-mean normalization. The formula is presented as follow:

$$x^* = \frac{x - \bar{x}}{\sigma} \quad (9)$$

Where $\bar{x}$ represents the mean value of the feature and σ represents the standard deviation of the feature.

Table 1. Pearson correlation coefficient of indicators for NDSD

RSI	CCI	RSV	K	J	CR	MACDh
0.511	0.561	0.603	0.256	0.371	0.206	0.176
CRshift	MACDshift	KMDshift	high-2-d	RSIshift	CCIshift	
0.516	0.664	0.716	-0.257	0.858	0.697	

Table 2. Pearson correlation coefficient of indicators for BYD

RSI	CCI	RSV	K	D	J	CR	MACDh
0.493	0.557	0.606	0.292	0.134	0.396	0.204	0.196
CCIshift	CRshift	MACDshift	KMDshift	Volume	high-2-d	RSIshift	
0.671	0.533	0.653	0.684	0.198	-0.259	0.862	

After normalization, a correlation test was conducted on the dataset. The first selection was completed by calculating every indicator's Pearson correlation coefficient with the return and retaining those indicators that have a Pearson correlation coefficient with the return above 0.1. The correlation coefficient of collected indicators for the three subjects were listed in Tables I-III. Indicators for NDSD and ZHGD are six-period RSI, six-period CCI, six-period RSV, six-period K value, six-period J value, energy index, MACDh, the high price delta and two days later (high_2_d), and the first order difference of RSI, CCI, RSV, K value, J value, energy index and MACDh. Indicators for BYD are final indicators for NDSD and ZHGD plus volume of the day. During the process of setting up OLS model, there will be an autocorrelation test on these factors to satisfy the fundamental hypothesis of the model.

Table 3. Pearson correlation coefficient of indicators for ZHGD

RSI	CCI	RSV	K	J	CR	MACDh
0.490	0.545	0.596	0.242	0.362	0.140	0.156
CRshift	MACDshift	KMDshift	high-2-d	RSIshift	CCIshift	
0.461	0.716	0.723	-0.227	0.890	0.702	

2.2 Model setup

2.2.1 Random Forest model

Random forest is a kind of ensemble learning that uses multiple independent decision tree models and combines their conclusions to make the forecast. There are two features of the Random Forest model. First, it conducts random sampling on the training data so that every Decision Tree model in it is highly possible to have different conclusions. Second, Random Forest trains multiple decision trees on different subspaces of the feature space but only increased bias slightly [5]. These two characteristics give the Random Forest model strong resistance to noise and overfitting. Besides, the Random Forest model generates an internal unbiased estimate of the generalization error as the forest building progresses. Moreover, it has an effective method for estimating missing data and maintains accuracy when a large proportion of data of the data are missing.

In this paper, the Random Forest model is constructed using grid search with cross-validation to optimize the number of decision trees and the max depth of the trees. The range of the number of decision trees is set at five to one hundred and fifty with the step length of five. The range of max depth of trees is set at five to fifty with the step length of five. Gini impurity measure is applied as the splitting criterion. Gini impurity at a node N is given by:

$$G(N) = 1 - \sum_j p[j|N]^2 \tag{10}$$

Where $p[j|N]$ is the ratio of label j in the node N and j is the number of labels. The best parameters for each Random Forest model are listed in table IV.

Table 4. Best parameters for Random Forest model

Stock name / parameters	NDSD	BYD	ZHGD
n-estimators	150	10	20
max-depth	5	15	30

2.2.2 XGBoost model

XGBoost is a kind of ensemble learning model. In this paper, the decision tree is used as the weak classifier. The core of XGBoost model is to repeatedly split the features to create a new tree. The process of creating a new tree is a process of learning a new function to fit the residual of last prediction. XGBoost algorithm has a regular term to punish complex trees which weaken the impact

of overfitting. Meanwhile, XGBoost algorithm supports column sampling. It not only saps overfitting, but also improves calculation efficiency.

Setting the object function in round t as $Obj^{(t)}$, XGBoost models in this paper apply mean squared error (MSE) as the loss function.

$$Obj^{(t)} = \sum_{i=1}^{n} l\left(y_i - \hat{y}_i^{(t)}\right) + \sum_{i=1}^{t} \rho(f_i) \tag{11}$$

Therefore,

$$\sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \rho(f_t) + C \tag{12}$$

Hence,

$$\sum_{i=1}^{n} [2(\hat{y}_i^{(t-1)} - y_i) f_t(x_i) + f_t(x_i)^2] + \rho(f_t) + C \tag{13}$$

Where $\hat{y}_i^{(t)}$ is the prediction at round t, $\rho(f_t)$ is the regular term to punish complex model. The term $\hat{y}_i^{(t-1)} - y_i$ is the residual between true value and prediction value. The goal is to find a f_t to minimize the $Obj^{(t)}$ without its constant term. Noticeably, the MSE is used as the loss function to obtain the object function. If another loss function is applied, the XGBoost model will use Taylor's Formula to expand it into a two-degree polynomial in order to calculate the residual approximately. This is what differentiates XGBoost from gradient boost decision tree (GBDT). The parameters of the XGBoost model for three subjects is listed in table V.

Table 5. Best parameters for Boost model

Stock name / parameters	NDSD	BYD	ZHGD
max depth	6	6	5
Learning rate	0.04	0.05	0.1
n-estimators	85	50	60
subsample	0.9	0.8	0.6

2.2.3 Multiple Linear Regression model

Multiple Linear Regression is a regression analysis method. This model is superior to the other two models discussed in this paper in the aspect of explanatory since it assumes that the dependent variable has a linear relationship with independent variables. Nevertheless, it also has more limited application in reality due to the same reason. Few subjects have pure linear relationship with their influential factors. The Linear Regression model is based on five fundamental hypotheses:

- The expectation of random error term is zero.
- The explanatory variables are homoscedastic and each one of them is not autocorrelated.
- The co-variance between random error term and explanatory variable is zero.
- One explanatory variable is not linearly correlated with another explanatory variable.
- The random error term is subject to normal distribution.

The Linear Regression model is constructed on one basic equation:

$$y = \theta_0 + \omega_1 x_1 + \omega_2 x_2 + \cdots + \omega_n x_n \tag{14}$$

Where x_i (i=1,2…n) is the independent variables, $\omega_i (i = 1,2 \cdots n)$ is the weight of each independent variable, θ_0 is the bias term. Method of least squares can be used to estimate $\omega_i (i = 1,2 \cdots n)$. MSE is the loss function for linear regression model in this research. The expression of loss function is as below:

$$loss = \sum_{i=1}^{n}(\theta_0 + \theta_1 x_i - y_i)^2 \quad (15)$$

Here, θ_0 is the bias term. $\theta_1 = (\omega_1, \omega_2, \cdots, \omega_n)$ is the vector consists of weight, x_i represents the vector consists of observed value of features, y_i is the true value. The loss function can also be interpreted as:

$$loss = \left|\left|\hat{Y} - Y\right|\right|^2 = \left|\left|X\theta - Y\right|\right|^2 = (X\theta - Y)^T(X\theta - Y) \quad (16)$$

The goal is to find θ_0 and θ_1 that minimize the loss function. By method of least squares, the ideal θ_0 and θ_1 can be obtained by partial derivation of the loss function.

$$loss = \theta^T X^T X\theta - \theta^T X^T Y - Y^T X\theta + Y^T Y \quad (17)$$

Thus, one derives the

$$\frac{\partial loss}{\partial \theta} = 2X^T X\theta - 2X^T Y = 0 \quad (18)$$

Hence,

$$\theta = (X^T X)^{-1} X^T Y \quad (19)$$

From the deduction above, it can be seen that the advantage of method of least squares is the capability of obtaining the ideal parameters directly. Factors' autocorrelation may have significant impacts on the model and distort the forecast. To minimize the impact of factors' autocorrelation, autocorrelation coefficient (nflags=637) of each factor is calculated and is used to sort the factors out. Only the ten factors that have the lowest autocorrelation coefficient were used to train the OLS model. The autocorrelation coefficient of factors for each subject is listed in table VI-VIII.

Table 6. Autocorrelation coefficient for NDSD

high-2-d	MACDh	MACDshift	J	K	CCIshift	CR	KMDshift	RSV	CCI
0.000016	0.000044	0.000112	0.000118	0.000339	0.000595	0.000812	0.001253	0.001794	0.001822

Table 7. Autocorrelation coefficient for BYD

MACDh	RSIshift	MACDshift	K	D	high-2-d	CRshift	J	CCIshift	Volume
0.000002	0.000013	0.000018	0.000029	0.000032	0.000044	0.000164	0.000301	0.000473	0.000841

Table 8. Autocorrelation coefficient for ZHGD

MACDh	RSIshift	MACDshift	K	D	high-2-d	CRshift	J	CCIshift	Volume
0.000002	0.000013	0.000018	0.000029	0.000032	0.000044	0.000164	0.000301	0.000473	0.000841

3. Results

Table 9. R2 of the RF, XGBoost, OLS model for subjects

Stock name / Model name	NDSD	BYD	ZHGD
RF	0.871	0.885	0.899
XGBoost	0.846	0.863	0.921
OLS	0.730	0.832	0.825

In this paper, coefficient of determination (R^2) and mean squared error (MSE) are used as metrics to evaluate the effects of models. R^2 reflects the extent of the volatility for the dependent variable can be explained by independent variables. R^2 is given by:

$$R^2 = \frac{SSR}{SST} \tag{20}$$

where SSR is regression sum of squares, SST is total sum of squares. MSE can show the degree of prediction price deviating from the real price. MSE is given by:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(RP_i - PP_i)^2 \tag{21}$$

where RP_i is the real price at i day and PP_i is the prediction price at i day.

Three models discussed in this paper and their coefficient of determination for three subjects are listed in Table IX while their mean squared errors are listed in Table X.

Table 10. MSE of the RF, XGBoost, OLS model for subjects

Stock name / Model name	NDSD	BYD	ZHGD
RF	0.155	0.106	0.098
XGBoost	0.142	0.104	0.070
OLS	0.324	0.154	0.170

The outcome of using stock return of NDSD, BYD and ZHGD to fit RF, XGBoost and OLS model is shown in Figs. 1-3. Overall, all three models perform quite well on the test set, judging from their coefficient of determination and mean squared error. In details, even the same model has different performance when the subject is different.

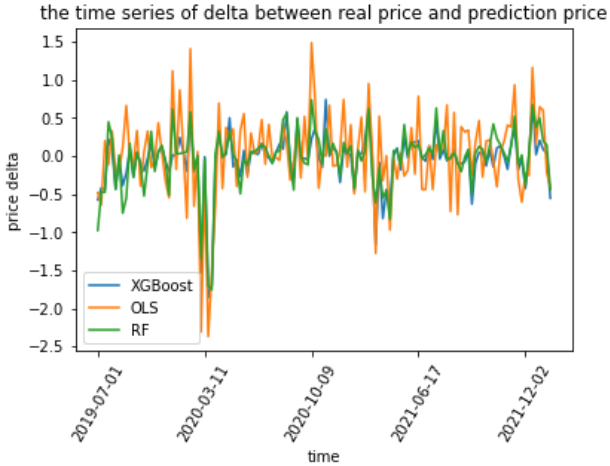

Figure 1. Delta between real price and prediction price (NDSD).

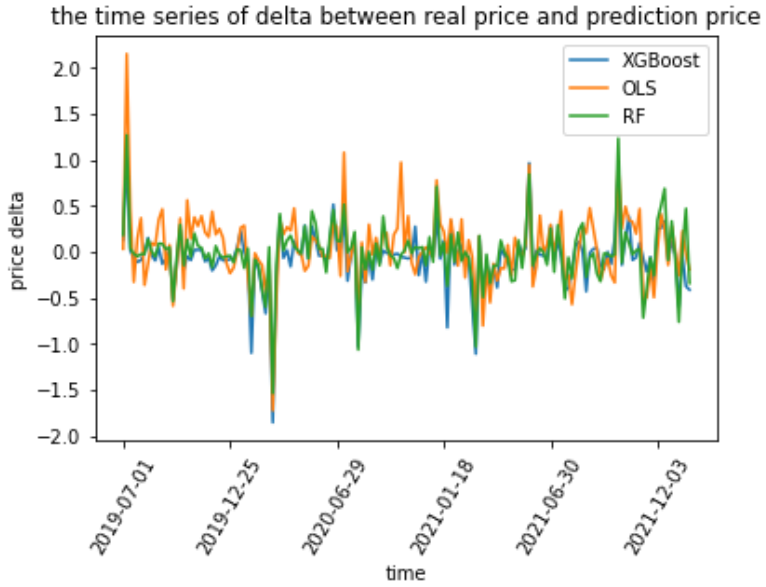

Figure 2. Delta between real price and prediction price (BYD).

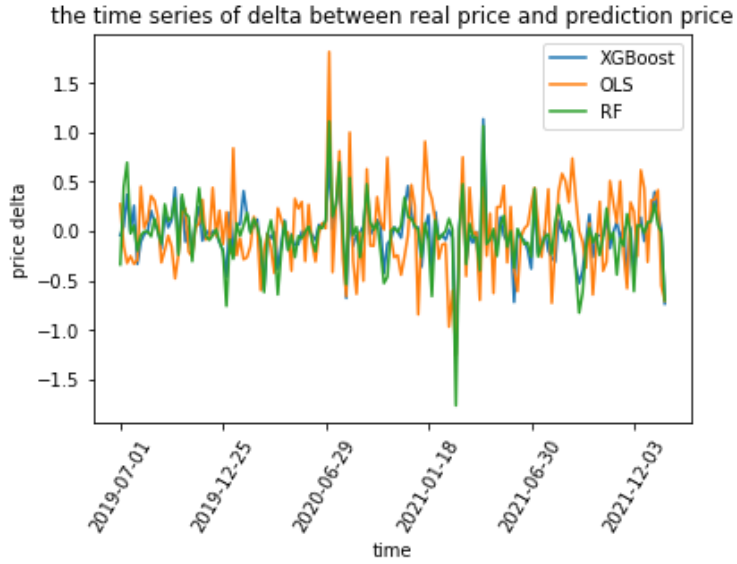

Figure 3. Delta between real price and prediction price (ZHGD).

4. Discussion

Seen from Table IX, random forest model performed slightly better that XGBoost model while the multiple linear models ranked lowest in the coefficient of determination. Many factors can impact this difference between models. Specifically, the sample number may not be large enough to embody performance difference of the models. Despite that, all three models achieved good results as illustrated in Figs. 1-3. It can also be seen that the RF and XGBoost model outperformed the OLS model in predicting the value.

However, Table IX also indicated that the RF and OLS model achieved best and poorest fitting results respectively When NDSD's return is fixed as the subject; when BYD's return is fixed as the subject, the RF and OLS model achieved best and poorest fitting results respectively; when ZHGD's return is fixed as the subject, the XGBoost and OLS model achieved best and poorest fitting results respectively. This phenomenon revealed the fact that the choice of subjects may influence different models' performance and thus explain the performance difference between the three models discussed in this paper.

Nevertheless, there are some limitations and drawbacks of above analysis. In this research, only the technical indicators were calculated and considered, which means that the choice of indicators is not comprehensive enough and it may impact the accuracy of results. Moreover, the parameters of random forest are optimized by grid-searching. Due to the limitation of processing speed and capability of the equipment, the length cannot be set shorter, which results in the imperfection of parameters. The same limitation also exists for XGBoost model because its parameters are adjusted by researcher's experience. The lack of data quantity (640 days in total) may also lead to the inaccurate outcome and this can be improved by adapting a larger dataset.

5. Conclusion

In summary, the comparisons of the stock price forecasting performance for random forest, XGBoost, OLS model are presented in this paper. In feature engineering, stockstats and Pearson correlation coefficient are applied to calculate and select factors respectively. The models are evaluated by coefficient of determination. According to the analysis, random forest model performed best followed by XGBoost model and OLS ranked lowest considering the three subjects comprehensively. When setting up XGBoost model, the parameters were adjusted by the researcher's experience which means that the parameters may not be the ideally optimized parameters. Future research can focus on the difference between methods of time series prediction models (e.g., ARIMA or GARCH) and methods of deep learning (e.g., convolutional neural networks, CNN). This research optimized the investors' choice of models by comparing their performance in A stock market and prepared for the possible hybrid model in the future. These results offer a guideline for future research to differentiate models.

References

[1] A. A. Adebiyi, A. O. Adewumi, and C. K. Ayo, "Comparison of ARIMA and artificial neural networks models for stock price prediction," Journal of Applied Mathematics 2014, 2014.

[2] W. Yang, "Multiple-Factor Stock Selection Model Based on Neural Networks," Huazhong University of Science&Technology 32, 2018.

[3] W. Cao, C. Cui, W. Zhu, "Research on influence of multi-source economic uncertainty on the exchange rate of RMB-analysis based on the multi-factor GARCH-MIDAS model," Financial Theory and Practice 2021: 59 - 69.

[4] Y. Chen, Z. Tang, Y. Luo, J. Yang, "Research on stock price prediction based on Xgboost algorithm and pearson optimization," Information Techonology, 2018.

[5] L. Khaidem, S. Saha, and S. R. Dey, "Predicting the direction of stock market prices using random forest," arXiv preprint arXiv:1605.00003 2016.

[6] H. Wang, Y. Hao, "Stock Prediction Algorithm Based on Random Forest and Technical Factors", Modern Computer, 2021.

[7] W. Xu, Y. Zhou, "Research on Stock Price of New Energy Based on VAR Model and GARCH Model," China Economic & Trading Herald, 2021, 98 - 100.

[8] Y. Yang, et al. "Stock Price Prediction Based on XGBoost and LightGBM," E3S Web of Conferences. Vol. 275. EDP Sciences, 2021.

[9] R. Mitchell, and E. Frank, "Accelerating the XGBoost algorithm using GPU computing," PeerJ Computer Science 3, 2017: e127.

[10] R. Dell'Aquila, and E. Ronchetti, "Stock and bond return predictability: the discrimination power of model selection criteria." Computational statistics & data analysis50.6, 2006, 1478 - 1495.

[11] R. C. A. Oomen, "Using high frequency stock market index data to calculate, model & forecast realized return variance." European Univ., Economics Discussion Paper 2001/6, 2001.

[12] R. Engle, "Dynamic conditional correlation: A simple class of multivariate generalized autoregressive conditional heteroskedasticity models." Journal of Business & Economic Statistics 20.3 (2002): 339 - 350.

[13] Zuo, Yi, et al, "Application of bayesian network for nikkei stock return prediction," 2011 International Conference on Technologies and Applications of Artificial Intelligence. IEEE, 2011.