Predict Gold Yields with ARIMA and LSTM

Ruoqi Zhang *
Department of Economics, China University of Political Science and Law, Beijing, China
*Corresponding author: 210622038@cupl.edu.cn

Abstract. The volatility of the gold price has a profound effect on many financial activities around the world. Developing a reliable forecasting model can provide insight into the volatility, behavior and dynamics of the gold price and can ultimately provide the opportunity to make significant profits. Common forecasting models fall into two categories: classical linear models and deep learning models using neural networks. In this paper, the ARIMA model and the LSTM model are selected as representatives of the two types mentioned above to forecast the gold return on May 1, 2020.The experimental results show that the 90% confidence interval of ARIMA includes the actual gold return, but there is a substantial discrepancy between the point forecast and the gold return rate itself. And the prediction accuracy of the LSTM model depends heavily on the data size of the training set. Finally, by comparing the forecasting results with the actual results, this paper analyzes the shortcomings of each model and proposes improvement directions for future gold return rate forecasting research.

Keywords: ARIMA; LSTM; Time series; Gold yields prediction.

1. Introduction

Gold is a popular choice for risk management and hedging for market participants. Traders can benefit greatly from its strong value preservation features, fast capital settlement and other characteristics. However, gold always makes investing riskier, and its volatile market fluctuations can lead to substantial investment and financial losses while delivering substantial gains. Therefore, it is crucial for investors to choose a reasonable and effective forecasting model to reduce the potential risk of their investments.

With the continuous development in the field of mechanical learning, forecasting models regarding gold returns can be broadly classified into two categories: classical forecasting models and deep learning forecasting models. Among the classical linear models, the emergence of autoregressive integrated moving average models (ARIMA) has led to significant advances in the study of predicting gold returns. For example, Sharma A. M. and Baby. S. used ARIMA model to forecast and study the gold price in India. In numerous studies, traditional forecasting methods have been shown to be successful in predicting the trend or dynamic characteristics of gold prices [1-5].

In the field of machine learning and deep learning, many scholars have started to simulate stock market forecasting using LSTM models [6,7]. The LSTM technique is a model that extends the memory of RNN (recurrent neural network). As a recurrent neural network, LSTM has "short-term memory", which uses previous information for the current task, and then analyzes and compares the prediction results of these models. Some financial series forecasting using LSTM models have achieved promising forecasting results.

This paper is innovative in that it analyzes the similarities and differences between LSTM and ARIMA models and compares the results and operating speeds of LSTM and ARIMA models, filling a relevant gap in the field.

2. Method

2.1 Classical Time Series (ARIMA)

ARIMA models predict future changes in random variables from past and present values by describing the autocorrelation of time dependent (t) variables. Depending on whether the original series of the model is smooth and the part included in the regression, it can be divided into the autoregressive model AR(p), the moving average model MA(q), the autoregressive moving average process ARMA (p, q). When the original series contains seasonal trends, the ARIMA multiplicative seasonal model is used, and its basic structure is: ARIMA(p, d, q)(P,D,Q)s, where d and D denote ordinary difference and seasonal difference orders; p and q denote the autoregressive and moving average orders in the model; P and Q are the autoregressive and moving average orders of the seasonal model, and S is the seasonal period [8-10].

2.2 Deep Learning Time Series (LSTM)

LSTM is a modified version of the traditional recurrent neural network RNN. It was proposed by Hochreiter and Schmidhuber in 1997. Compared with the normal RNN, LSTM adds a memory cell to determine whether the information is useful or not, and solves the problem of gradient disappearance and gradient explosion during the training of long sequences. This improvement enables it to have better performance in longer sequences.

The cell state is the key of LSTM. To protect and control the state of the memory cell, three control gates are placed in a memory cell, called input gate, forget gate and output gate. The structure of the LSTM memory cell is shown in the figure 1. Details of the three gates are as follows.

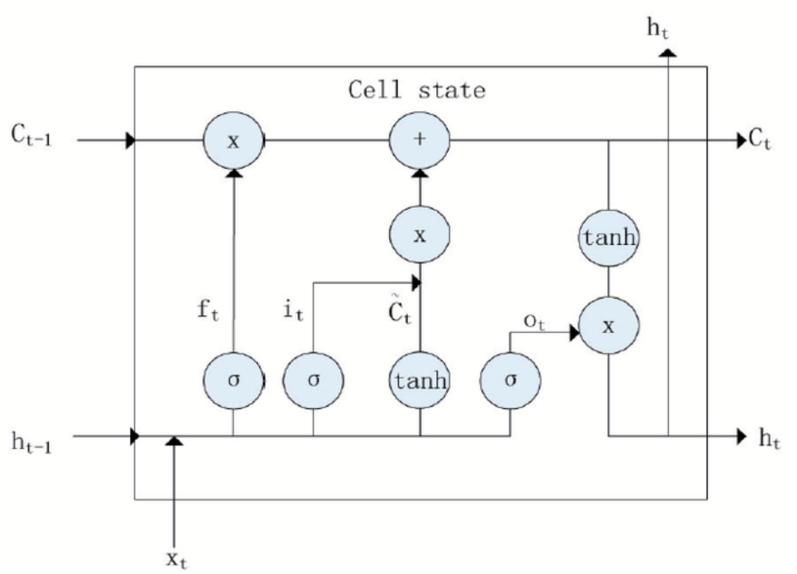

Fig. 1 LSTM’s structure

The forgetting gate determines whether the information meets the algorithm's authentication by activating the function Sigmoid, which ultimately decides which information is saved and forgotten. The input gate first updates the information and determines if the information is valid, then generates a vector to update the state of the memory cell and stores the determined new information in the memory cell. In this way, the current cell state and the long-term cell state can be combined into a new cell state. The output gate step determines what to output, and this output will be based on the filtered cell state. First, a sigmoid layer is used to determine which values of the cell state are output. Then, the cell state is mapped to values from -1 to 1 through the tanh layer and the result is multiplied by the result of the output gate so that only that part of the output is needed.

3. Empirical Analysis

3.1 Data splitting

The closing U.S. dollar gold prices used in this study are information collected by Yahoo Finance, a website that provides financial data and commentary. For both ARIMA and LSTM models, the paper choose the data from January 4, 2000 to April 30, 2020 as the training set to predict the gold price return on May 1, 2020.

3.2 ARIMA model

3.2.1 Data processing

In order to turn the gold price time series into the time series of the gold return rate, it is required to do a series of transformations for the data. In brief, after guaranteeing that the data is numeric (if it is not numeric, use as.numeric(unlist()) to turn it into a numeric form), a combined function diff(log()) is utilized to turn the gold price into the log return, which is similar to the rate of return. One thing about this process should be noted: it would not work if the function becomes log(diff()).

3.2.2 Verifying the sequence stationarity

Figure 2 plots the time series after the transformation. The absence of any trend, periodicity, or seasonality in the plot suggests that the time series is stationary. Additionally, since the ndiffs() function returns 0, the time series no longer requires difference calculation. Finally, the author strongly rejects the null hypothesis, the time series is not stationary, in the Augmented Dickey-Fuller Test, because the p-value is 0.01. The conclusion drawn from this is that the series is wholly stationary.

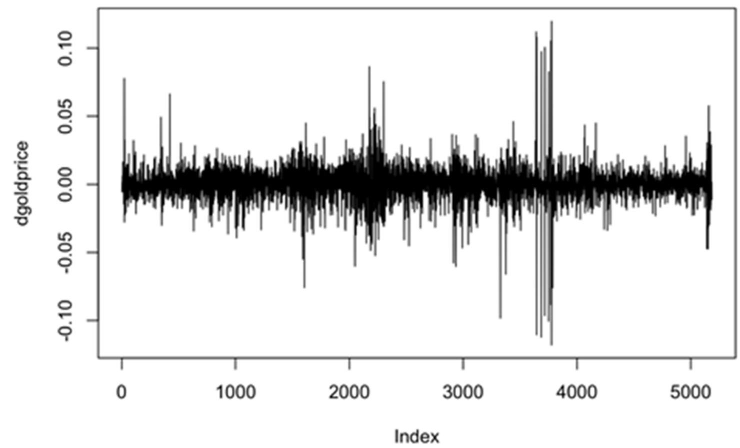

Fig. 2 Differenced time series with the log scale

3.2.3 Check for stationarity

The auto.arima() function could be useful for choosing the parameters p and q in order to find an acceptable model. The auto.arima() function suggests that the best model in this situation is the ARIMA(2,0,2) with a non-zero mean (since the time series have already been differentiated once, the d value is 1 for the gold price), which has the smallest AIC (Akaike information criterion, an indicator to measure the goodness of statistical model fitting) value in comparison to other models. Additionally, the p and q parameters are also non-zero according to the ACF (Figure 3) and PACF (Figure 4) plots since their trail is off to zero. It is reasonable to select the model ARIMA provided by auto.arima() as a result.

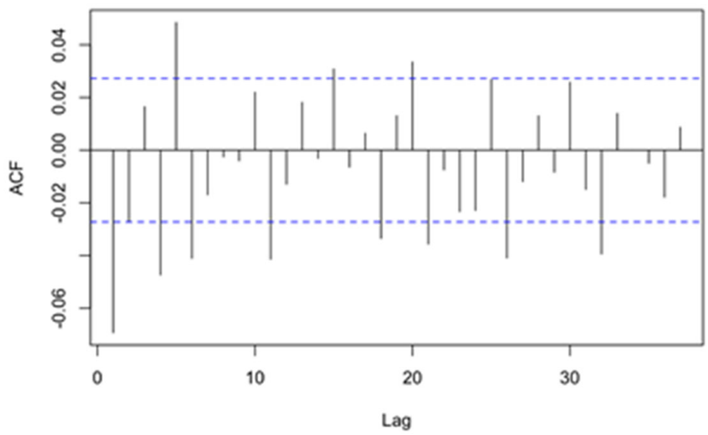

Fig. 3 ACF

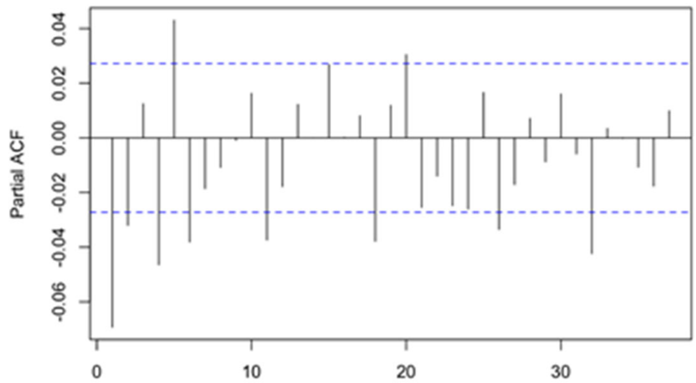

Fig.4 Partial ACF

3.2.4 Evaluating the model fit

It is important to evaluate the model's fitness after choosing it. The normal Q-Q plot that is displayed above was created in this instance by applying the qqnorm() and qqline() functions. It demonstrates that the residuals appear to have a thick tail distribution, which indicates that the occurrence of the extreme value is more likely (Figure 5). Therefore, the point forecast should be accurate because the extreme value below and above the line is symmetric, whereas the confidence interval anticipated by the auto.arima() function—assuming that the residual has a normal distribution—would be erroneous.

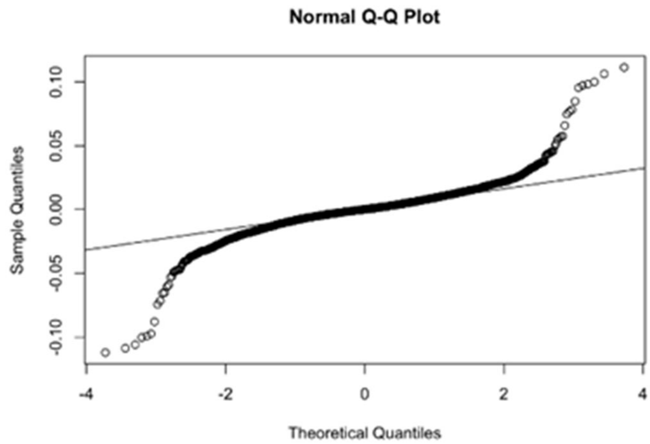

Fig.5 Normal Q-Q Plot

Additionally, the histogram generated by the hist () function and displayed on the following page shows that the peak of the residuals appears to be substantially higher than the normal distribution. Therefore, the kurtosis () function is used to assess the kurtosis of the residual distribution to verify the aforementioned claim. This function's output is 15.35884, which suggests that it has a fatter tail than the usual distribution and that its peak is substantially higher and sharper (Figure 6).

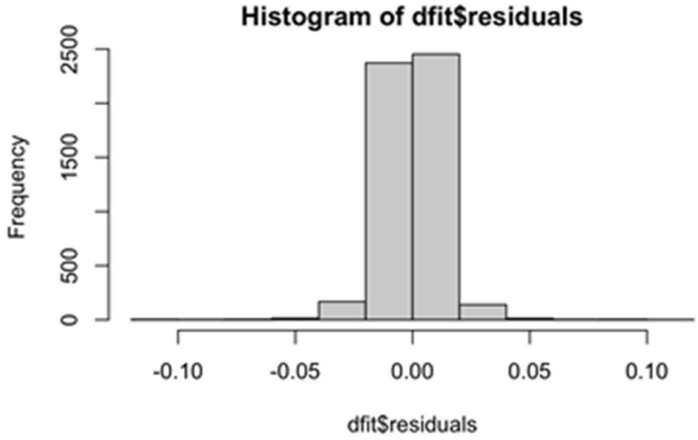

Fig. 6 Histogram of dift$residuals

As a result, in the process that follows, it is necessary to recreate the confidence interval that fits with the residual distribution. Returning to the topic, the Box-Ljung test could be used to demonstrate that all residual autocorrelations are zero. The results are not significant in this instance, the p-value is greater than 0.05, indicating that the autocorrelations are equal to zero. Due to these extremes, the ARIMA model looks to suit the data used to make the point forecast well, but it needs to be changed when determining the confidence interval.

3.2.5 limitations and solutions

As the paper mentioned above in 'Evaluating the model fit', since the residuals are not normally distributed, the predicted confidence interval below is invalid. Therefore, it is necessary to revise the 90% confidence interval (the point forecast separately adds lower and upper 95% percentiles of the residuals). First, it is indispensable to locate the lower and upper 95% percentiles of the residuals. In detail, since the total amount of residuals is known, which is 5185, such multiplication the paper shown below could be used to know which number of the sequence (a sequence arranging the residuals from large to small) is the lower and upper 95% percentiles 259, 4927, respectively.

Therefore, the upper and lower 95% percentiles are separately the 259th number and the 4927th number of the sequence (Figure 7).

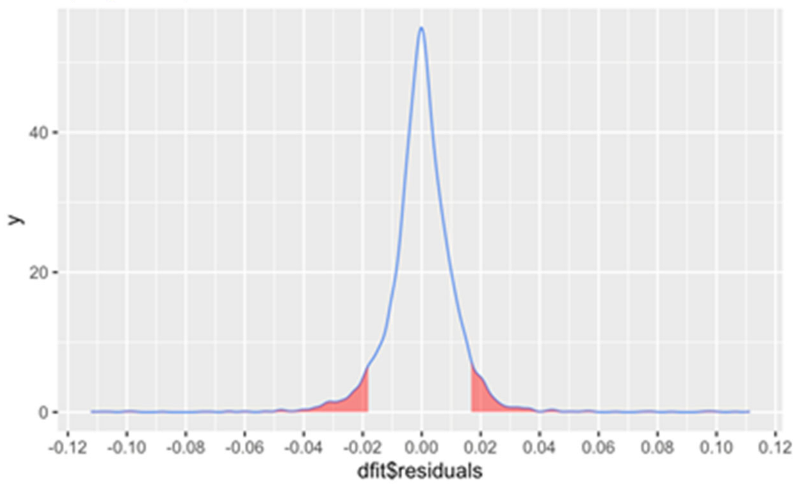

Fig. 7 Dift$residuals

Then the sort () function is used to arrange the sequence—including all the residuals with their numerical value from large to small—as shown above. In such a series, the authors locate the 259th number and the 4927th number with ease, that is -0.018062325 and 0.01652954, the value of lower and upper 95% quantiles of the residuals (Figure 8).

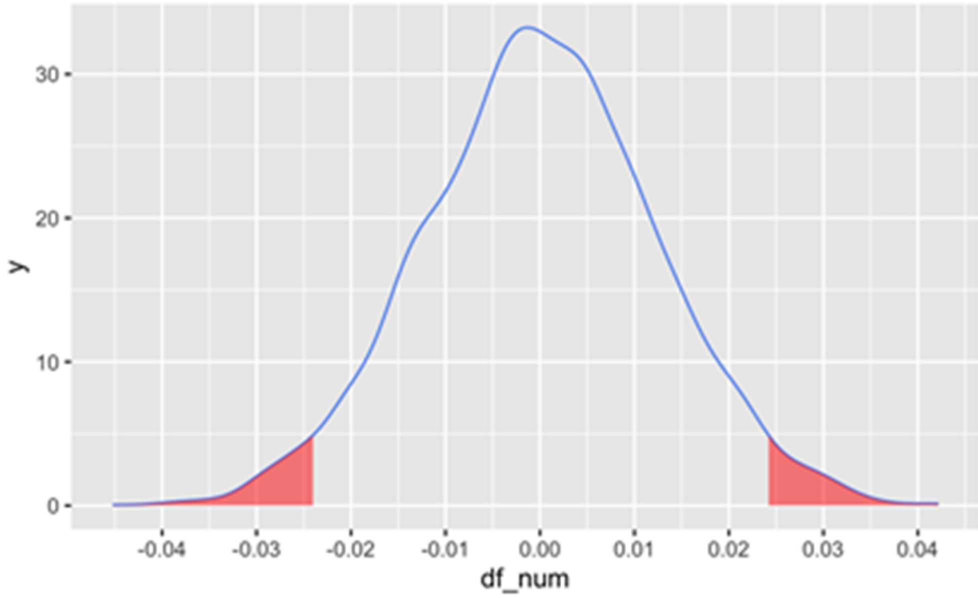

Fig. 8 Df-num

Finally, since the point forecast (7.251787e-06) and certain percentiles are known, the confidence interval is (0.0165368, -0.01805507), which is narrower than the confidence interval predicted by the auto.arima() function. For explaining this phenomenon, the histogram of the normal and fat-tail distributions of the residuals are plotted and compared. The Figure 7 is the fat-tail distribution of the residuals, which is much more concentrated in the middle but more dispersed in the periphery compared with the normal distribution on the right side.

To be specific, the frequency of 0 (almost the mean value which is the center of the whole graph) in the figure 7 is about 55, which is much greater than the frequency of 0 in the figure 8 which is about 33. What's more, between x=-0.25 to x=0.25, the left graph is much steeper than the right graph, which also indicates that the fat-tail distribution is much more concentrated in the middle. In other aspects, focusing on the periphery of the two graphs, the left graph has a greater range (from x=-0.11 to x=0.11) than the range in the right graph (from x=-0.05 to x=0.05). Hence, no doubt that the fat-tail distribution of the residuals is more dispersive in the periphery than the normalone.

In conclusion, since most of the residuals, in this case, are concentrated around the middle, the gap between the lower and upper 95% percentiles including those residuals would be narrow as the graph shows: the area between two red shadows has a high and sharp mountain which contains 90% percent of the residuals in a short range. Also, since the gap between two red shadows is the same as the difference between the upper and lower confidence interval, the range of the real confidence interval is much less than the one constructed by the normal distribution.

The analysis above is the solution of resetting the confidence interval when the residuals are not normally distributed. After resetting, the prediction of the ARIMA model of the gold return is 7.251787e-06 with a 90% confidence interval (0.0165368, -0.01805507), which is more accurate than the result predicted with the model established by assuming that the residuals are normally distributed.

3.3 LSTM model

3.3.1 Data processing

The original data collection must first be transformed into time series prediction-ready data. In order for the LSTM model to predict stock returns, the data also be adjusted when it is used. This includes normalizing the variables, both input and output values, and converting the dataset into a supervised learning problem (t-1).

3.3.2 Training setting

In this paper, the LSTM model has seven layers.The dropout layers (with dropout rate 0.2) prevent the network from overfitting. The unit number for each LSTM layer is set 50. The dense layer is used to reshape the output.

Table 1. Model framework

Layer(type)	Output Shape	Param #
lstm (LSTM)	(None, 1, 50)	10400
Dropout(Dropout)	(None, 1, 50)	0
lstm-1 (LSTM)	(None, 50)	20200
Dropout-1 (Dropout)	(None, 1)	0
lstm-2 (LSTM)	(None, 50)	20200
Dropout-2 (Dropout)	(None, 1)	0
Dense(Dense)	(None, 1)	51

4. Result

The principle and construction process of ARIMA and LSTM have been introduced above. This section will show the gold yeilds prediction results obtained by the model in this paper, and then make comparative analysis.

4.1 Prediction of ARIMA model

Unquestionably, the ARIMA's 90% confidence interval (0.0165368, -0.01805507) includes the actual gold return rate; yet, there is a substantial discrepancy between the point forecast (7.251787e-06) and the gold return rate itself (3.94e-03).

4.2 Prediction of LSTM model

In order to forecast the return on May 1st, 2020, the paper used the return determined from the data of April 30th, 2020. The value of RMSE (0.00019669) the final model fit test indicates that the model does not fit well.

5. Discussion

From the above conclusions, it is evident that both ARIMA and LSTM models have unsatisfactory forecasting results on gold returns. In the following, this paper will specifically analyze the reasons.

5.1 ARIMA Model

The reasons for the inaccuracy of forecasting gold returns using ARIMA are as follows. First, Gold returns can have huge residuals as it is quite volatile. Its confidence interval would be rather large as a result in order to take into account earlier data with significant variations that did not fit the model.

The variation of gold futures is closely related to a variety of factors such as politics, economy, investor psychology, business conditions, etc. It is especially subject to extremely significant economic influences, leading to a great randomness and chance of price differences in different periods, which determines the tendency of gold returns to be non-linear, and the ARIMA model cannot take this into account, so there will often be high values of residuals. As a result, the confidence interval would be so large that it would be unable to provide us with an accurate forecast.

5.2 LSTM Model

The unsatisfactory prediction results of the LSTM model may be caused by the following points. First, it puts a lot of strain on the hardware to train them. A lot of resources are needed even when it is unnecessary to train these networks quickly. Similar to running these models locally, running them in the cloud likewise uses a lot of resources. Because they require storage bandwidth-bound processing, which is the worst nightmare for hardware designers and ultimately limits the usability of neural network solutions, LSTMs are challenging to train. In order to execute in each sequential time step, LSTM needs 4 linear layers per unit (MLP layers). To the point where they could not use many computational units, linear layers need a lot of storage bandwidth. Usually, this is because the system doesn't have enough storage bandwidth to support the computational units. Adding more computing units is also simple, but increasing storage bandwidth is more difficult (note that there are enough wires on the chip, long wires from the processor to the storage, etc.). Hence, hardware acceleration is not a good fit for LSTM and its variations.

The LSTM and its variations somewhat address the gradient problem of the first RNN, but it is still insufficient. Although it could handle sequences of 100 magnitudes, it is still challenging for sequences of 1000 magnitudes or longer.

Second, the computation takes a while. If the network is very deep and the time span of the LSTM is long, the computation would be quite big and time-consuming. Each cell of the LSTM implies 4 completely connected layers (MLP).

6. Conclusion

This paper use ARIMA and LSTM models to forecast the gold return on May 1, 2020 based on gold futures data from January 4, 2000 to April 30, 2020, and evaluate the forecasting performance of the models by comparing them with the actual data. The conclusions of this paper are as follows. (1) neither the ARIMA model nor the LSTM model alone is suitable for forecasting gold yields. (2) The large residuals and nonlinear characteristics of gold returns lead to inaccuracy in predicting gold returns using ARIMA. (3) The size of the data selection and the heavy reliance on laddered forecasts lead to inaccuracy in predicting gold returns using LSTM. It is concluded that future research can consider two efforts: first, using ARIMA models to forecast gold return time series with significant trends; second, increasing the data size to improve the forecast accuracy in LSTM models.

References

[1] Qiang Li, Liwen Chen, and Yong Zeng. The Mechanism and Effectiveness of Credit Scoring of P2P Lending Platform: Evidence from Renrendai.com. China Finance Review International, 2018, 8(3):256–274.

[2] Charles T Carlstrom and Timothy S Fuerst. Agency Costs, Net Worth, and Business Fluctuations: A Computable General Equilibrium Analysis. The American Economic Review, 1997, 893–910.

[3] Barro, R. and Misra, S. Gold Returns. The Economic Journal, 2016, 126(594):1293-1317.

[4] Dixon M F, Halperin I, Bilokon P. Machine learning in Finance. New York, NY, USA: Springer International Publishing, 2020.

[5] Not Use LSTM's to Predict the Stock Market. Bluesky Capital. https://www.blueskycapitalmanagement.com/, last accessed 2022/08/22.

[6] GitHub. 2022. Gold-Price-Forecasting-Model-/gold. https://github.com/, last accessed 2022/08/22.

[7] Gold Price. 2022. Gold Price History. https://goldprice.org/gold-price-history.html, last accessed 2022/08/22.

[8] Hu, Y. Gold market volatility characteristics and risk measures. New Finance, 2021, (02):57-62.

[9] Livieris, I., Pintelas, E. and Pintelas, P. A CNN–LSTM model for gold price time-series forecasting. Neural Computing and Applications, 2020, 32(23):17351-17360.

[10] Xu L. P.,Luo M. C. Short-term analysis of gold price forecast based on ARIMA model[J]. Finance and Economics Science, 2011, (01):26-34.