

Research on Quantitative Trading Investment Strategy Based on LSTM and Dynamic Programming

Jiayuan Mao¹, Kexin Wang², Wei Zhao³, Ningning Liu⁴, Youyu Wang¹,
Wei Fan^{5, *}

¹College of Science, Chongqing University of Technology, Chongqing, 400054, China

²Chongqing Key Laboratory of Public Big Data Security Technology, Chongqing, 401420, China

³College of materials science and Engineering, Hebei University of Science And Technology, Shijiazhuang, Hebei, 050091, China

⁴IT service center, Heibei Open University, ShiJiazhuang, HeBei, 052360, China

⁵College of Computer Science and Engineering, Chongqing University of Technology, Chongqing, 400054, China

*Corresponding author. Email: fanwei220412@163.com

Abstract. As financial markets matured, more standardized and quantitative research emerged. In this paper, we model the historical price returns of U.S. gold and BTC over a five-year trading period from September 11, 2016, to September 10, 2021, and estimate the total investment returns as of September 10, 2021. First, we build a variety of price-prediction models to fit the image to the data to get the most accurate image of the actual price trend. It was found that the LSTM neural network-based multi-interval segmentation price prediction model, which has the best prediction effect by comparison with the actual value. Therefore, it was used as the core prediction model for this paper. Then, we use the planning model to find the optimal investment plan by using the difference between the price change of gold and bitcoin trading in the next three days as the decision variable, the highest Sharpe ratio on the third day as the objective function. At the same time, the variance of the forecast value in the next three days is used as the risk characterization, and different weights are assigned to the risk and return as the objective function to characterize the investment strategy under different investment personalities. In order to accelerate the convergence of the planning model, we added a particle swarm algorithm for optimization. In the end, we obtained the results for the specific daily investment scenarios for aggressive investors, intermediate investors, and steady investors.

Keywords: quantitative trading, LSTM neural network, Sharpe ratio, Dynamic Programming.

1. Introduction

Satoshi Nakamoto originally proposed the concept of Bitcoin as a P2P form of digital currency. Bitcoin has an open and transparent transaction record and is a decentralized payment system. From the very beginning, as an unclaimed digital currency, it has become a high-yield, high-risk financial product that is the choice of many investors seeking high returns.[1] Moreover, gold is a general value preservation product with low risk and low return and is preferred by many risk-averse financial managers. Furthermore, the so-called quantitative trading means using computer technology to allocate with the help of mathematical and statistical thought models with the background of financial knowledge to get a high percentage of return.[2]

2. Price Prediction Model Based on LSTM Neural Network

In the early days, through the efforts of many researchers, many models on stock forecasting were developed, such as the ARIMA model [3] and the generalized autoregressive conditional heteroskedasticity model (GARCH). However, many researchers also used their statistical properties for time series forecasting. They can only extract the linear features, but financial time series are very complex, and financial time series arise from People's economic and financial activities. Naturally,

there will be apparent perturbations in the data, which makes the financial time series not satisfy the assumptions of linearity and smoothness. The extraction of highly nonlinear features such as stock change curves, machine learning, and deep learning algorithms is gradually taking the mainstream [4], but support vector machines (SVMs) still present difficulties in kernel function selection, parameter tuning, and extraction of surface features. [5] However, machine learning algorithms can still achieve good financial time series analysis forecasting results. Therefore, we will use LSTM artificial neural network among machine learning algorithms for forecasting analysis. [6]

Use 30 days to train the neural network, predict the price of the next three days, the input is the price data of the first 30 days, the output is the price data of the next three days, and one day at a time to keep rolling back the prediction

The data used in the LSTM neural network should be the time-series features of the time series data to be extracted. The window length is set to 30 days, the rolling window length is set to 1 day, and the prediction period is 3 days. It means that we will predict the stock price 3 days later by looking at the data from the previous 30 days based on the current time point of each day. These settings also apply to the window length and forecast length for CNN time series data.

LSTM is a recurrent neural network suitable for extracting temporal features from time series and can learn long-term time series dependencies. Long short-term memory (LSTM) units are the units of a recurrent neural network (RNN). An RNN consisting of LSTM units is often called an LSTM network (or just LSTM). The unit remembers the values in an arbitrary time interval.

The structure of the LSTM is shown in the figure below. Generally, the LSTM consists of input gates i_t , forgetting gates f_t , and output gates O_t are a total of three gates. Three gates control the flow of information in and out of the cell. C_t is the state of the current cell, h_t is the state of the hidden layer, x_t is the input data.[7]

The input determines the value of the forgetting gate x_t at moment t together with the output h_{t-1} at the moment $t-1$. The activation function used is the sigmoid function, which is expressed as:

$$f_t = \sigma(W_f \cdot [h_{t-1}, X_t] + b_f) \quad (1)$$

When the value of the forgetting gate is obtained, the new information obtained is added to the state using the input gate. Thus the replacement of the old information is achieved in the past with the expression:

$$i_t = \sigma(W_i \cdot [h_{t-1}, X_t] + b_i) \quad (2)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, X_t] + b_c) \quad (3)$$

Let i_t be multiplied after adding it to the information of the forgotten gate to obtain a new C_t , whose expression is given by:

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \quad (4)$$

The result of the output gate is the latest state C after the forgetting gate and the input gate, and the output value h_{t-1} at time $t-1$ and the current input value x_t at time t . At this time, the activation function σ is no longer a sigmoid function but a tanh function so that the required information can be output from the output gate.

$$o_t = \sigma(W_o \cdot [h_{t-1}, X_t] + b_o) \quad (5)$$

The forgetting gate f_t determines which information the neural unit discards, and this gate layer can output a value between 0 and 1 from the forgetting gate by reading the states of h_{t-1} and x_t , with 0 representing complete discard and 1 representing complete retention. The input gate determines the value to be updated by the neural unit i_t , which takes the information finished filtered by the forgetting gate and uses the tanh function to update the neural unit state. The final output gate o_t determines the state of the neural unit output, which is the first generic sigmoid layer that determines the states of the neural unit to be output, which is compressed between -1 and 1 using the tanh function according to these states.

Based on our prediction results, we predict the price trend for the next three days, then repeat the operation for each day and select the results for three days. In order to verify the accuracy of the forecasting model, we can calculate the error using the results of days N+1, N+2, and N+3, roughly as follows:

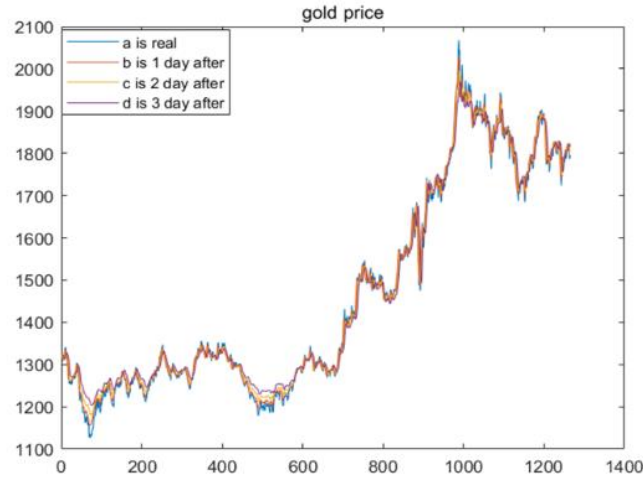


Figure 1 Gold Forecast Error

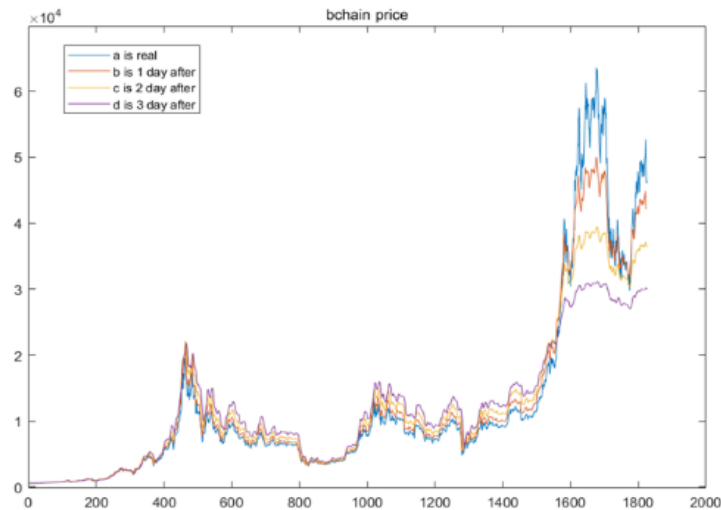


Figure 2 Bitcoin Forecast Error

By calculation, the forecast error R2 is about 0.99 for gold and 0.92 for bitcoin, so the forecast effect is good.

It has been found that this prediction model has been very accurate so that we can make our future financial investment strategy based on this prediction model

3. An Investment Model Based on Dynamic Programming

We have all the price data before the current day, and now we want to find a suitable trading strategy by mathematical modeling to ensure profitability afterward. The next day, we will add today's data to the historical data and again work backward in price prediction to determine whether to buy, sell, or trade. This dynamically changing trading process is a classic dynamic planning process, where we use the previous forecasts to adjust today's trading strategy and then recursively until the last day.

In order to facilitate the calculation of the planning model, it is assumed that the proportion of cash, gold, and Bitcoin held in the total assets on the Nth day is recorded as $[c_N, g_N, b_N]$. We have the relational formula:

$$c_N + g_N + b_N = 1 \quad (6)$$

We set the decision variable as the two-tuple $[\Delta G_N, \Delta B_N]$, which are the changes in the proportion of gold and bitcoin, respectively.

It is supposed that the increase of cash, gold, and Bitcoin on the Nth day is $[0.004657\%, \overline{G_N}, \overline{B_N}]$ respectively. According to the forecast result, on the N day, we will have the Nday data and the forecast result of the $N + 1, N + 2, N + 3$ day.

To better quantify the relationship between daily investment risk and return, we introduce the Sharpe Ratio in finance. Based on modern asset portfolio theory, the Sharpe Ratio focuses not only on the return of an asset but also on the asset's risk, measuring the asset's risk-adjusted return, which indicates the price per unit of risk.

$$\text{Sharpe Ratio} = \frac{E(R_p) - R_f}{\sigma_p} \quad (7)$$

Where $E(R_p), \sigma_p, R_f$ represent the expected return, the standard deviation of the return, and the return of the risk-free asset during the observation period, respectively.[9]

According to the planning model, on the N -day, we set the objective function $f(N)$ to be the value of the Sharpe ratio on the third day when the position is adjusted today and the next three days remain unchanged. When this objective function reaches the maximum, it is the state where the risk and return are balanced.

Taking N -day as an example, the proportion of cash, gold, and Bitcoin in total assets is $[c_N, g_N, b_N]$

At the end of each day, that is, the ups and downs need to be settled first at the end of the N th day.

If the increase of cash, gold, and bitcoin on that day is $[0.004657\%, \overline{G_N}, \overline{B_N}]$ respectively, the proportion will become $[(1 + 0.004657\%)c_N, (1 + \overline{G_N})g_N, (1 + \overline{B_N})b_N]$. It can be seen that the sum of the proportions here may no longer add up to 1. It is due to the total holding of cash. The amount of gold and bitcoin has changed. After we have made all the changes, we will re-normalize and then repeat the operation for day $N+1$.

Then, the trade will be adjusted to the position, and the transaction fee will be settled.

When only the purchase of gold is required, the change in the amount of gold is $\Delta G_N > 0$. In this case, the change in the amount of cash is $1.0101 * \Delta G_N$ (this is because the buy rate is 1%, and 101 dollars does not buy 100 dollars of gold). When only the sale of gold is required, the change in the amount of gold is $\Delta G_N < 0$, and the change in the cash ratio is $0.99 * \Delta G_N$.

When only purchases of bitcoins are required, the change in bitcoin amount is $\Delta B_N > 0$, and the change in cash amount is $1.0204 * \Delta B_N$. When only selling bitcoins is required, the change in the number of bitcoins is $\Delta G_N < 0$, and the change in the amount of cash is $0.98 * \Delta G_N$. After the transaction fee is settled, the final cash, gold, and bitcoin shares of total assets for the day are

$$\begin{bmatrix} (1 + 0.004657\%)c_N - (1 \pm 0.01)\Delta G_N \\ -(1 \pm 0.02)\Delta B_N \\ (1 + \overline{G_N})g_N + \Delta G_N \\ (1 + \overline{B_N})b_N + \Delta B_N \end{bmatrix} \quad (8)$$

Then, we normalize it by multiplying the sum of the three percentages by yesterday's total amount, which gives us today's total amount and return, as well as today's percentages of cash, gold, and bitcoin

According to our assumptions, keep the cash, gold, and bitcoin amount constant for the next three days of the function. Therefore, the latter process ΔG_{N+1} and ΔB_{N+1} directly equal to 0, and wait until the third day, then go to calculate the Sharpe ratio of the third day, and then proceed to the next step of detailed analysis.

The natural constraint of this model is that after all operations are performed on that day, the cash, gold, and bitcoin share of total assets should be greater than or equal to 0, i.e.

$$\begin{cases} (1 + 0.004657\%)c_N + (1 \pm 0.01)\Delta G_N \\ \quad + (1 \pm 0.02)\Delta B_N \geq 0 \\ (1 + \overline{G_N})g_N - \Delta G_N \geq 0 \\ (1 + \overline{B_N})b_N - \Delta B_N \geq 0 \end{cases} \quad (9)$$

Notice that in the model building process above, the objective function $f(N)$ was set to require that the position be adjusted today and remain unchanged for the next three days, but in reality, it is the position that remains unchanged that causes problems in the daily decision making. In the case of frequent fluctuations, each time we decide, we only consider the current position condition until the third day, which can easily lead to selling at the low point and buying at the high point, which leads to unexpected results. Therefore, we can adjust our investment plan for the next three days based on our forecast results to achieve a higher profit. Therefore, in the optimization process, we set the value of $[\Delta G_{N+1}, \Delta B_{N+1}, \Delta G_{N+2}, \Delta B_{N+2}]$. In the middle of the process, leave the constraints unchanged, and reuse the above model to solve.

However, in the process of doing so, the convergence speed is too slow due to a large number of decision variables and the difficulty of determining the range. Therefore, we need to use intelligent algorithms to help speed up the convergence and continue the planning model with initial values, which will greatly speed up the convergence speed. We choose to use a particle swarm optimization algorithm.

Particle Swarm Optimization (PSO), first proposed by Eberhart and Kennedy in 1995, is a group-wise algorithm that simulates the cooperation mechanism of the foraging behavior of a flock of organisms in nature to find the optimal solution to the problem. The basic concept is based on studying the foraging behavior of bird flocks. Imagine a scenario where a flock of birds is randomly searching for food, and there is only one piece of food in the area. All birds do not know where the food is, but they know how far their current position is from the food. The simplest and most effective strategy? The PSO algorithm is inspired by this behavioral characteristic of biological populations and is used to solve optimization problems. [10]

According to the above model process, when using the Sharpe rate of the third day as the objective function to take the maximum value, we can find that the final \$1000 can get \$4278.70

In the above modeling process, we used the Sharpe ratio of finance as the objective function, and the investment strategy was too conservative, investing as much as possible in gold and not in bitcoin. Because bitcoin is different from traditional financial items, it has a massive range of daily fluctuations, and the variance is so large that it rejects such a large risk as a Sharpe ratio. Therefore, the final return is not high.

In fact, for investors, different personalities will lead to different investment strategies. In addition to using a single Sharpe ratio as the objective function, we can also use other methods to characterize different people's mental expectations of risk and return. We set three types of people as aggressive (who do not consider the existence of risk), prudent (who use the Sharpe ratio as the objective function), and conservative (who consider risk and return to have other weights), and develop different investment plans.

We use the standard deviation of the forecast for the next three days, σ_N , as an assessment of risk, and if the forecast price fluctuates more in the next three days, it means that the risk is higher in the future.

For aggressive personalities, we set the objective function f_N to maximize the return after three days, for robust personalities, we still set the objective function f_N to maximize the Sharpe ratio after three days, and finally, for conservative personalities, we set the objective function f_N to be the return after three days minus 0.75 times the risk assessment value

Finally, our daily yield is shown below:

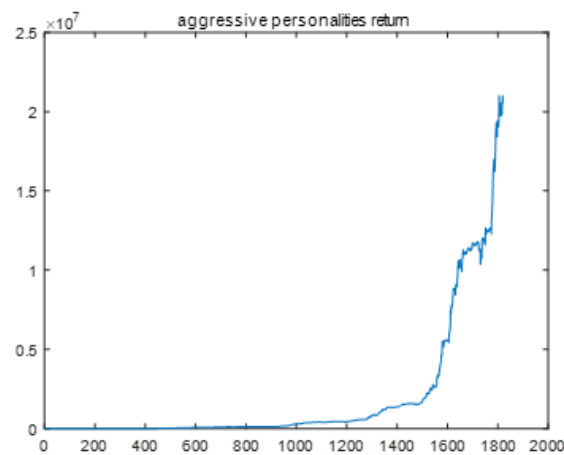


Figure 3 Aggressive personalities

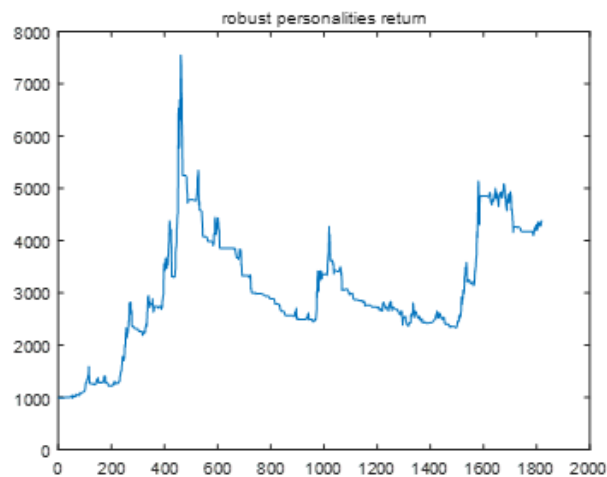


Figure 4 Robust personalities

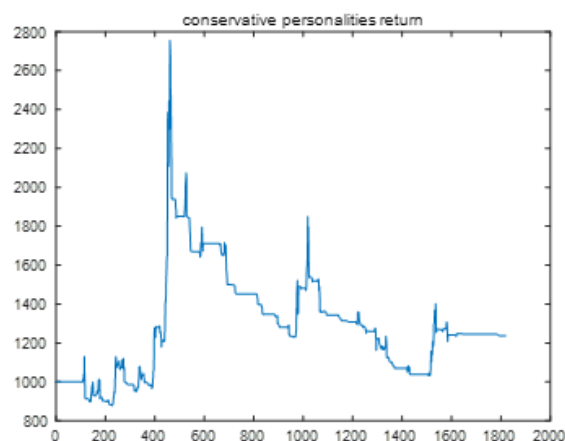


Figure 5 Conservative personalities

The final results are \$70,265.49 for the aggressive investor, \$4,278.70 for the prudent investor, and \$1,146.93 for the conservative investor.

4. Conclusion

This paper has built a gold and BTC price prediction model based on LSTM neural network to help investors easily predict and track stock prices. The model can evaluate whether the end of each

trading day is investable by only the daily closing price to date. It can then comprehensively analyze its future valuation value and probability based on the algorithm training to give investors a rational decision recommendation. We believe that such comprehensive and rationalized decision-making is crucial to investors' investment judgment when investors are dealing with a large amount of stock price fluctuation information. The judgment of the neural network data model under machine learning is free from any emotional and personal psycho-physiological factors of investors and usually provides more efficient input for investors' judgment with data support and real-time analysis. Therefore, it is more valuable in stock market investment analysis under time constraints and many disturbing factors.

We then build neural network-based models to predict increases or decreases in the value of stocks in the online marketplace. Our model can accurately predict the precise short-term changes in the product over the next one to three days. This accurate prediction helps investors avoid greater losses before the stock's valuation decreases or develop a better strategy before it increases in value.

Based on the Sharpe ratio, we use the degree of volatility of the equation to judge the risk of the product. Our analysis shows that, on balance, gold is a low-risk, low-return product, while bitcoin has a higher risk and return.

Based on the results of our analysis, we have developed a sound quantitative portfolio investment strategy for investors.

References

- [1] Xie Wenhao, Cao Guangxi. Research on the multi-fractal characteristics of Bitcoin's volume-price relationship based on MF-ADCCA method [J/OL]. Chinese Journal of Management Engineering: 1-10[2022-03-22].DOI:10.13587/j.cnki.jieem.2022.05.001.
- [2] Ruiqi Sun. Research on LSTM neural network-based price trend forecasting model for U.S. stock index [D]. Capital University of Economics and Business.
- [3] Bangzhu Zhu and Yiming Wei. Carbon price forecasting with a novel hybrid ARIMA and least squares support vector machines methodology [J]. Omega, 2013, 41(3) : 517-524.
- [4] Fang Yichou, Lu Zhuang, Ge Junwei. Joint RMSE loss LSTM-CNN model for stock price prediction [J/OL]. Computer Engineering and Applications: 1-10 [2022-02-21].
- [5] Zhang Lei. Research on stock market trend analysis and prediction based on support vector machine [D]. Nanjing University of Posts and Telecommunications.
- [6] Zhang Libiao, Zhou Chunguang, Ma Ming, et al. Particle swarm-based algorithms for solving multi-objective optimization problems [J]. Computer Research and Development, 2004, 41(7):6.
- [7] Perez, Ruben E., Peter W. Jansen, and Joaquim RRA Martins. "pyOpt: a Python-based object-oriented framework for nonlinear constrained optimization." Structural and Multidisciplinary Optimization 45.1 (2012): 101-118.
- [8] Li, Minqiang. Basic theory and applications of genetic algorithms [M]. Science Press, 2002.
- [9] Han, C.G.. Mathematical modeling methods and their applications [M]. Higher Education Press, 2009.
- [10] Ernest Chen. Quantitative trading [M]. Northeast University of Finance and Economics Press, 2014.