Problem Chosen **C** | **2022 MCM/ICM Summary Sheet** | **Team Control Number** **2223483**

Simple Template for MCM Contest by latexstudio

Summary

People are always looking for good investment methods to improve their asset position. However, human perception is sometimes limited and emotionally biased. Quantitative investment uses programming, big data analysis, and mathematical models to build trading strategies and apply them to actual trading investments. Based on the advantages of it, such as the rapid data processing response, the significant amount of value created by the transaction, and the relatively greater objectivity, we derive quantitative investment strategies through modeling to maximize trader's benefits.

Several models are established: Model I: Price Forecasting Model based on Long Short-Term Memory Network; Model II: Cross-Species Arbitrage Model.

Firstly, considering the large amount of data given in the question, we make data processing before all the models are established, which includes data cleaning ,data classification and data normalization.

For Model I: We built a neural network prediction model with LSTM as the main body. Among all the data,90% of the data is used as the training set to train the model, and 10% of the data is used as the test set for prediction. Our LSTM model can predict the daily price of bitcoin and gold, which can provide a data base for subsequent trading strategies.

For Model II: We use a cross-species spread arbitrage trading strategy, make a time series graph of the spread of two futures, analyze the spread, find a reasonable spread range, and determine how to operate when the spread moves beyond or below the reasonable spread range, and use matlab to build a cross-species spread arbitrage trading strategy model, which results in a daily trading strategy.

After the eatablishment of the models, we changed the relevant parameters and analyzed and evaluated the model by annualized return, Sharpe ratio and other indicators to prove that the strategy we gave is the best one. Meanwhile, the actual prices of gold and bitcoin were brought into the model for backtesting to calculate the annualized return, and the error analysis of the annualized return obtained by applying the prediction data showed that the error is small, which indicates that the prediction model is more accurate and further proves the effectiveness of our strategy.

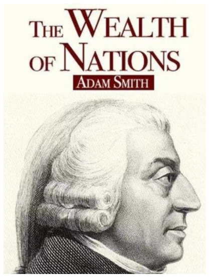
What's more, in order to determine the sensitivity of the strategy to transaction costs, we change the transaction costs several times, obtain the better trading strategy with different transaction costs by the above model and estimate its return, and draw the return curve. Analysis of the data shows that our strategy is less sensitive to transaction costs and proves to be more stable.

Keywords: Quantitative investing; Statistical Arbitrage; Long Short-Term Memory Network Algorithm

Contents

1 Introduction

1.1 Problem Background



"Behavior without rational guidance is at least inefficient," said Adam Smith, the father of economics. Indeed, according to the famous Hypothesis of Economic Man , people tend to seek to maximize their own interests and utility, which also holds in the trading market.

However, personal investment decisions are highly subjective and cannot be scientifically predicted based on available data. Just a small decision can lead to huge losses, which is completely contrary to the goal of a self-interested economic man.

As a result, driven by modern technological tools such as artificial intelligence, big data, and cloud computing, quantitative investment strategies have emerged and has been developing rapidly. Quantitative investment uses programming, big data analysis, and mathematical models to build trading strategies and apply them to actual trading investments.

Compared to traditional strategies, quantitative investment is not only more efficient and transferable,but also has the advantages of overcoming human weaknesses thanks to the objective historical evaluation tools. Currently, quantitative fund management has become one of the most important investment tools for global asset management companies. It is also meaningful to adopt quantitative investment strategies in personal investment decisions to achieve stable financial appreciation and reasonable risk avoidance.

1.2 Restatement of the Problem

With the 1000 dollar given at the beginning, which can be held or trade for bitcoin or gold, a trading strategy is required to be developed, which aims at making the trader's total return to the maximum within a five-year period.

Considering the specific constraints given, the restate of the problem can be expressed as follows:

- Providing a model which instructs the trader's behavior by simply using the data up to that very day and calculate the final return when the five-year period ends.
- Prove the strategy to be the most effective way to earn profits
- Make sensitivity analysis of the strategy to transaction costs and find out the effect of the changes in transaction costs on the established model
- Based on the results attained above, write one to two pages of memo to the trader to convey our approach.

1.3 Literature Review

Long Short-Term Memory Network put forward by Alex Grave[1] is quite good at predicting continuous variables. Besides, RNN and LSTM mentioned by Alex Sherstinsky[2] can also well enrich the LSTM system and incorporate these extensions into the Vanilla LSTM network, producing the most general LSTM[3] variant to date. All the previous explorations inspire us to use the long Short-Term Memory Network in terms of predicting the daily price of bitcion and gold.

When it comes to statistical arbitrage, the classic literature in this field about pair trading strategies is Gatev Goetzmann and Rouwenhorst(2006), whose abbreviation is GGR[4]. The basic theory of the distance method was introduced in detail in their pioneering study. They also found the advantage of the method is that it is robust and easy to implement, and the disadvantage of is that the mean reversion is weak. Chen(2012)[5] improved the correlation measure to a Pearson correlation coefficient and improved the strategy model using quasi-multivariate. Besides, Girma and Paulson(1999)[6] built a trading strategy that includes transaction costs based on the cointegration approach.

To conclude,the distance method is the most simple and easy way and avoids in-sample data mining. Moreover, since it is applicable to a wide range of assets and markets, it has the basis for wide application. Of course, since the distance method has relatively small payoffs due to the SSD criterion, there is room for further optimization, and the combination of cointegration methods for asset portfolio selection is one of the directions for improvement. The cointegration method is more robust and has good mean reversion, and the relevant research is abundant.

1.4 Our Approach

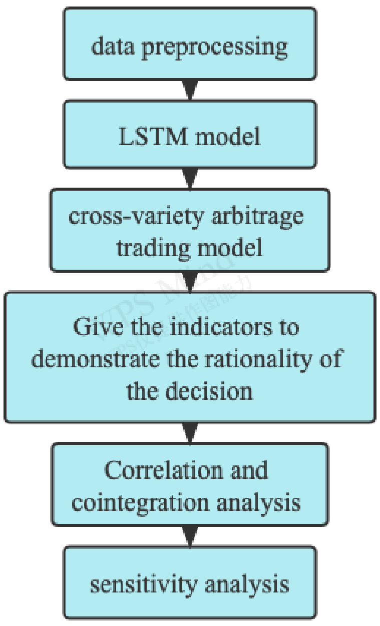

data preprocessing
LSTM model
cross-variety arbitrage trading model
Give the indicators to demonstrate the rationality of the decision
Correlation and cointegration analysis
sensitivity analysis

Figure 1: The framework of our work

- Data pre-processing
- Build predictive models
- Correlation and cointegration analysis of gold and bitcoin price series
- Build a cross-country arbitrage model to obtain daily trading strategies
- Prove that the model gives the best strategy by giving indicators to justify the decision
- Increase or decrease the percentage of transaction cost, run the model again, and perform sensitivity analysis

2 General Assumptions

As we all know, various kinds of complex factors do have their own specific effect on the practical problems ,in which condition,reasonable assumptions to simplify the model are what we need to clarify at the very beginning. In this part, we list the main hypotheses in our approach and each hypothes is followed by its corresponding explanation:

- Assumption1: Gold and Bitcoin are traded in whole multiples only, without regard to buying or selling fractional shares.
  Explanation: We only buy or sell integer multiples of gold bitcoin for daily trading
- Assumption2: Quantify daily asset values to derive a return curve
  Explanation:

$$\begin{aligned}\text{Asset value on the day} = {} & \text{cash on the day + number of gold shares held} \\ & \times \text{ gold price on the day + number of bitcoin shares held} \\ & \times \text{ bitcoin price on the day}\end{aligned} \tag{1}$$

- Assumption3: There is no limit on the time to hold the asset and there is no cost to hold the asset.
  Explanation: We can hold the position for a long period of time in order to find the best time to sell and increase the rate of return.
- Assumption4: Sell all the products on the last day.
  Explanation: We liquidate all assets on the last day in order to obtain a larger return during the trading period and to better reflect the return of the strategy model.
- Assumption5: The novel coronavirus continues to spread, which will inevitably affect personal earnings in the trading market
  Explanation: Although the outbreak of novel coronavirus has been effectively controlled in some areas, the global outbreak continues to spread and the epidemic has a trend of normalization definitely. The epidemic not only poses a threat to our health, but also has an impact on the trading market.Taking this into consideration, we finally set the risk-free rate of return at 2.5%.

3 Model preparation

3.1 Notations

Some important mathematical notations used in this paper are listed in Table 1.

Table 1: Notations used in this paper

Notations	Meaning
S_0	Initial capital
S_i	Assets on day i
R	Rate of return
r_i	Annualized rate of return
ΔP_i	Variance in price
SR	Sharpe Ratio
$E(R_p)$	Expected rate of portfolio Return
Rf	Risk-free rate
σp	Standard Deviation of Portfolio

3.2 The Data

3.2.1 Data Overview

The question has provided us with data directly, therefore we shouldn't need to make other data collections. What we need to do is to process the data by some particular algorithms and some other methods.

3.2.2 Data Cleaning

We found that the series of daily price data for gold is missing due to the fact that gold can't be traded on holidays. With the intention to simplify the model and improve the efficiency of our algorithm,we use SPSS to let missing values made up using the average value of neighboring points.

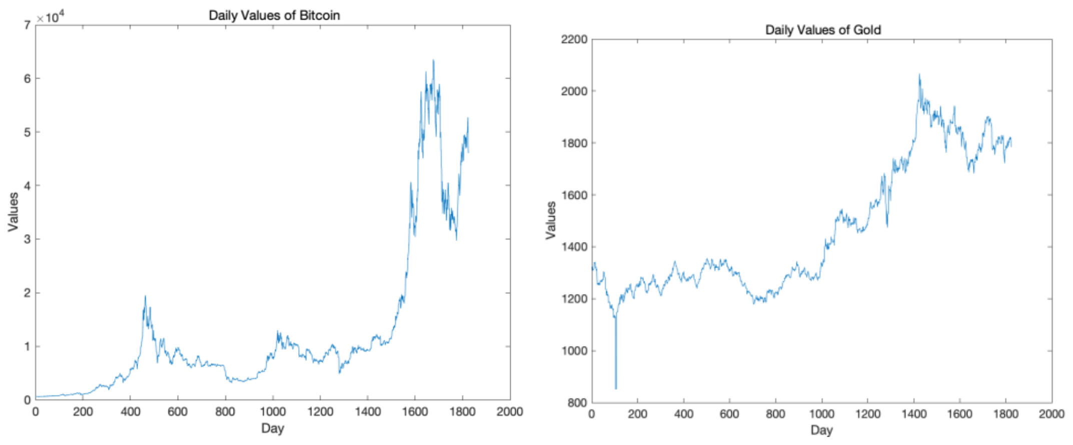


Figure 2: Daily Values of Bitcoin and Gold

4 Model I: Price Forecasting Model based on Long Short-Term Memory Network

4.1 Pre-conditions

- The operation of the financial market is not a purely random process, which inspires us to explore the hidden pattern of the time series. Therefore, the LSTM networks become the leading candidates for the prediction.
- The price prediction is based on the data given in the question. Some external Factors, such as political climate and market environment are not taken into consideration.

4.2 Pre-processing

4.2.1 Data Classification

We separate the original data into two different parts: training set and test set. The former is for the in-sample training, and the latter is for out-of-sample predictions.

4.2.2 Data normalization

The linear function conversion we used can be seen in the equation:

$$x_{scale} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{2}$$

Where x, x_{scale} represent the values before and after conversion respectively, $x_{\max}$ and $x_{\min}$ indicate the maximum and minimum value of the sample.

After doing so, the faster convergence when running programs can be well ensured. Otherwise, the loss of the training model will drop very slowly.

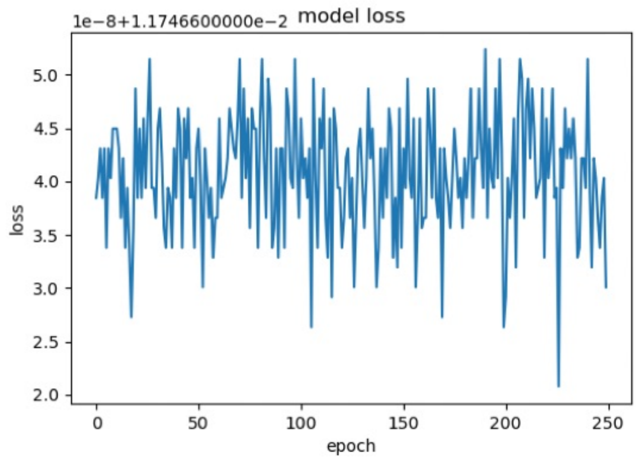


Figure 3: model loss

4.3 Model establishment

Considering the lack of the other features of the data, such as opening and closing price, the feature space and targets required for training and prediction are easy to determine. Our input and output are both one-dimensional, which is consistent with the given price data.

Then, we have a further discussion about the details in the Long Short-Term Memory Deep Learning Network Model.

Compared with RNN, which only has one state of the hidden layer, i.e. h, LSTM has another cell state for the preservation of long-term state.

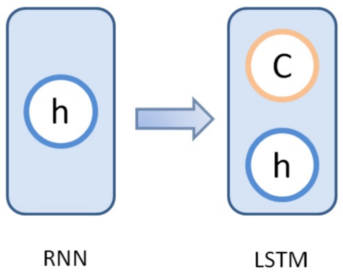


Figure 4: comparison between RNN and LSTM

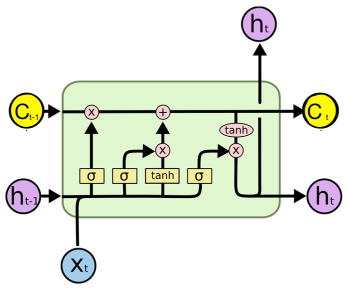


Figure 5: cell state

The Model has a gate structure consisting of forget gate, input gate and output gate.

Firstly, the forget gate determines how much of the cell state C_{t-1} of the previous moment is retained to the current moment C_t, its expression is as follows:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{3}$$

Where σ denotes the sigmoid function, whose output is between 0 and 1 ,b_f and $W_f$respectively represent the bias term and weight matrix of the forget gate, $[h_{t-1}, x_t]$ denotes the joint of two vectors into a longer vector.

$\tilde{C}_t$ and i_t construct the input gate, which determines how much of the input x_t of the network at the current moment is saved to the cell state C_t. The former can be seen as the information brought by the new input, and the latter, with the same structure as the forget gate , functions as the reserved section of new information. The process can be calculated as follows:

$$\tilde{C}_t = \tanh\left(W_c \cdot [h_{t-1}, x_t] + b_c\right) \tag{4}$$

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \tag{5}$$

When it comes to updating C_t, we choose to use the formula:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{6}$$

Where $f_t * C_{t-1}$ represents the selective forgetting or retention of the previous information, while $i_t * \tilde{C}_t$ represents the selective forgetting or retention of the new information. One plus the other, and then a new state can be established.

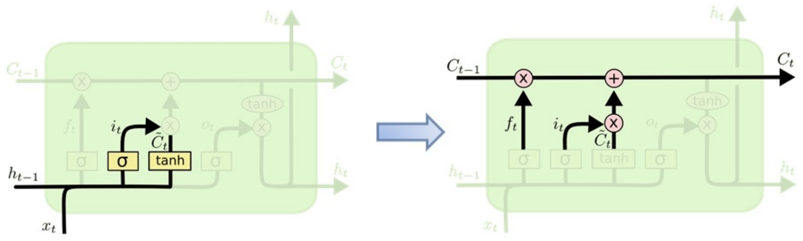


Figure 6: input and update

The output gate controls how much of the cell state C_t is output to the current output value of the LSTM h_t. The expression is:

$$o_t = \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right) \tag{7}$$

Finally, we can get the output of timestep, which is defined:

$$h_t = o_t * \tanh\left(C_t\right) \tag{8}$$

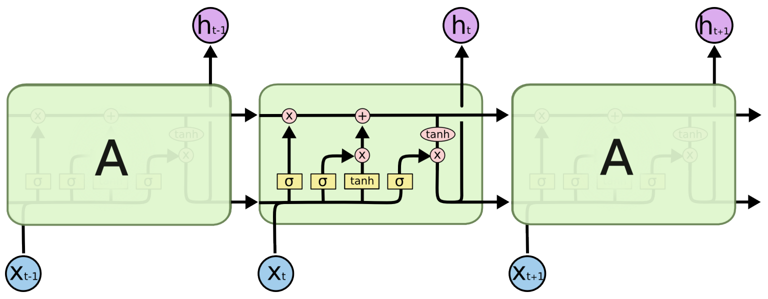


Figure 7: general structure

After repeated debugging, the final setting is listed as follows:

Table 2: Final setting in the algorithm

Types	Descriptions
Number of neurons	128, 256
Number of training rounds	250
Learning Rate	0.05, 0.001
activation function	relu
Key Parameter	loss=means_quared_error

Part of the training results are reflected as follows:

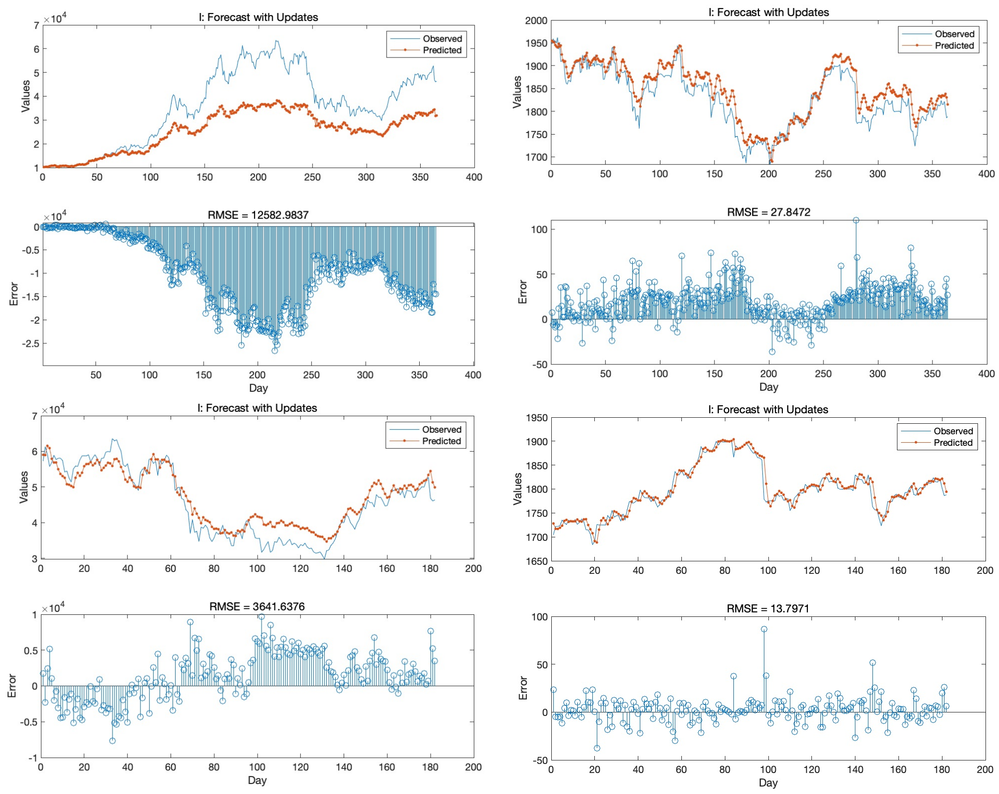


Figure 8: training procedure

The left column is about Bitcoin and the right column is about gold.

The first row selects 80% of the data as the training set, while the second row selects 90%.

Based on the principles and settings mentioned above, once we input data, the model will make its own predictions.

4.4 Results and Analysis

Based on the concept of time series, our model make the self-connection between hidden layers into realization through multiple input and output of time series in turn. After inputting the financial time series data into the model, it can process the data with the optimization principle, thus contributing to a better prediction .

We use python to do the price forecasting. The pseudocode is as follows:

Algorithm 1 LSTMLayer

```
Input: training set:
        $\text{Train} = \{x_t\}_{t=1}^{m}$
    test set:
        $\text{Test} = \{x_t\}_{t=1}^{m+1}$
Output: Predicted value
        $Prediction = \{P_t\}_{t=1}^{n}$
for all $x \in$ Train do
    • Determine the information that needs to be discarded by the cell by calculation
        $f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$   (9)
    • Determine the information that needs to be added to the cell by calculation
        $i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)\tilde{C}_t$
        $= \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$   (10)
    • Update cell state
        $o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$   (11)
    • Determine the output information by calculation
        $h_t = o_t * \tanh(C_t)$   (12)
end for
for all $x \in$ Test do
    predict
end for
Draw prediction+observed
```

Through simulation, the predicted daily price of gold and bitcoin from 2016.9.11-2021.9.10 is shown in the Figure:

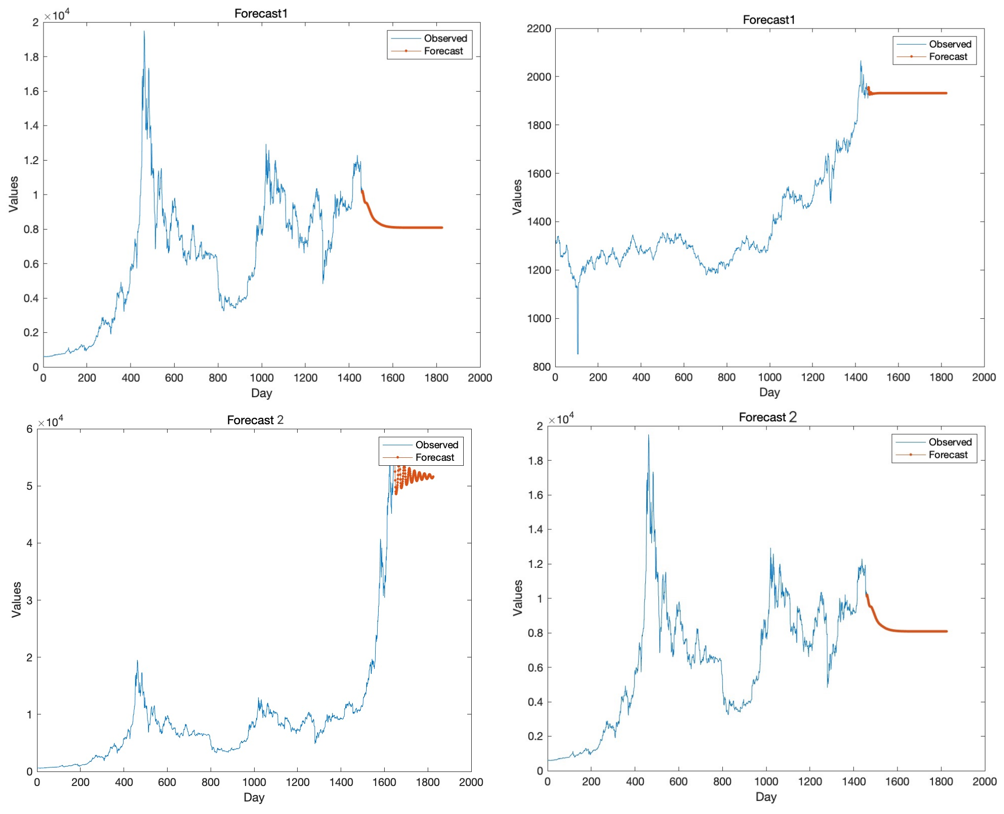


Figure 9: price forecast for bitcoin and gold

5 Model II: Cross-Species Arbitrage Model

5.1 Our preparation

First of all , we have had a deep insight into Cross-Species Arbitrage mainly through the literature[7]: Cross-species arbitrage generally occurs between two or more commodity futures that are somewhat correlated but not identical, and its profitability depends on the deviation of the futures product spreads from their normal spread levels. Since each commodity futures may be mispriced, this creates room for investors to profit by buying low and selling high or selling high and buying low in the range before the spread returns to its normal level.

We can logically conclude from it that a stable cointegration relationship is a vital prerequisite for cross-species arbitrage. Thus, before building the model, we further verify whether there is a relationship between the prices of gold and bitcoin especially at a statistical level and have a discussion about their long-term robustness in detail. In order to analyze the correlation between the prices of gold and bitcoin , we use spss to do a Pesrson test on the price data of the two products and then obtain the following figure:

		Bitcoin	Gold
Bitcoin	Pearson correlation coefficient	1	.645**
	Sig.		.000
	Number of cases	1825	1825
Gold	Pearson correlation coefficient	.645**	1
	Sig.	.000	
	Number of cases	1825	1825

**. At the 0.01 level , the correlation is significant

Figure 10: results of Feasibility Analysis

It is seen in the graph that Pearson correlation coefficient is 0.645 ,which is significantly larger than 0.5.Moreover, with the sig. smaller than 0.05, there is no sigh of contingency ,therefore it successfully passed the correlation test.

In conclusion, our Feasibility Analysis indicates that the price series of bitcoin price and gold are highly correlated, therefore Cross-Species Arbitrage between the two products is fully feasible.

Looking at the things from a different perspective, we have a brief analysis of the risks in the bitcoin and gold market, which is based on constant coefficient functions and constant coefficient derivative functions[8]. The results indicate that the respective market risks of the two show roughly opposite trends, which also confirms our conclusion that the two portfolios can be hedged.

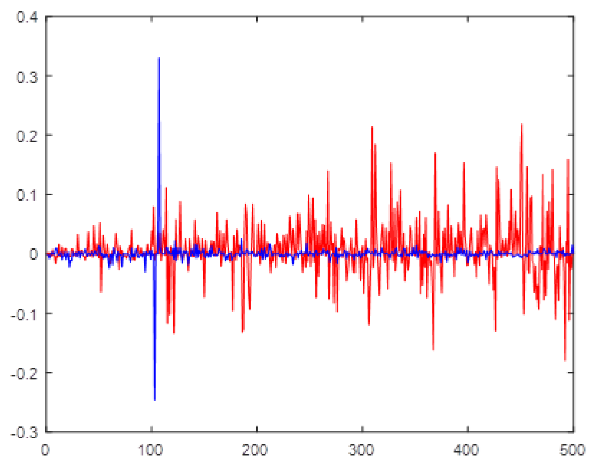


Figure 11: Market Risk Measurement

5.2 Model establishment

The basic processing route of statistical arbitrage is to use statistical analysis tools to study and analyze the historical data of the relationship between a set of related prices, study the historical stability of the relationship, estimate its probability distribution, and determine the extremes in the distribution, the extreme region of the distribution, i.e. the negative domain .

When the price relationship in the real market enters the negative area, it is considered that the price relationship cannot be maintained for a long time, and arbitrageurs have a high probability of success to enter the arbitrage market.

The basic processing route can be vividly reflected in the following figure:

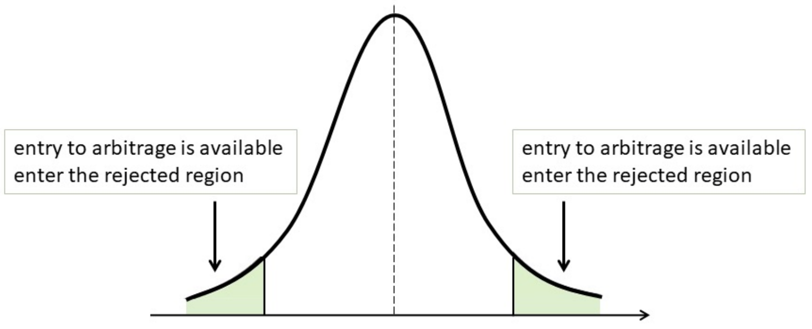

entry to arbitrage is available
enter the rejected region
entry to arbitrage is available
enter the rejected region

Figure 12: the basic processing route of statistical arbitrage

Then comes the Principle of Arbitrage.

When the spread deviates from the equilibrium spread and is greater than the equilibrium spread, if the deviation exceeds n times the standard deviation, reverse arbitrage is carried out, that is, shorting the spread.

When the spread deviates from the equilibrium spread and is smaller than the equilibrium spread, if the deviation exceeds -n times the standard deviation, a forward arbitrage is performed, that is, going long on the spread.

On the basis of the mentioned above, we established our own cross-species arbitrage trading model focusing on gold and bitcoin. According to the data acquired from the prediction model,that is Model I: Price Forecasting Model based on Long Short-Term Memory Network,we calculate the variances in price with the total number of 1825.

The specific calculation formula is as follows:

$$\Delta P = P_b - P_g \tag{13}$$

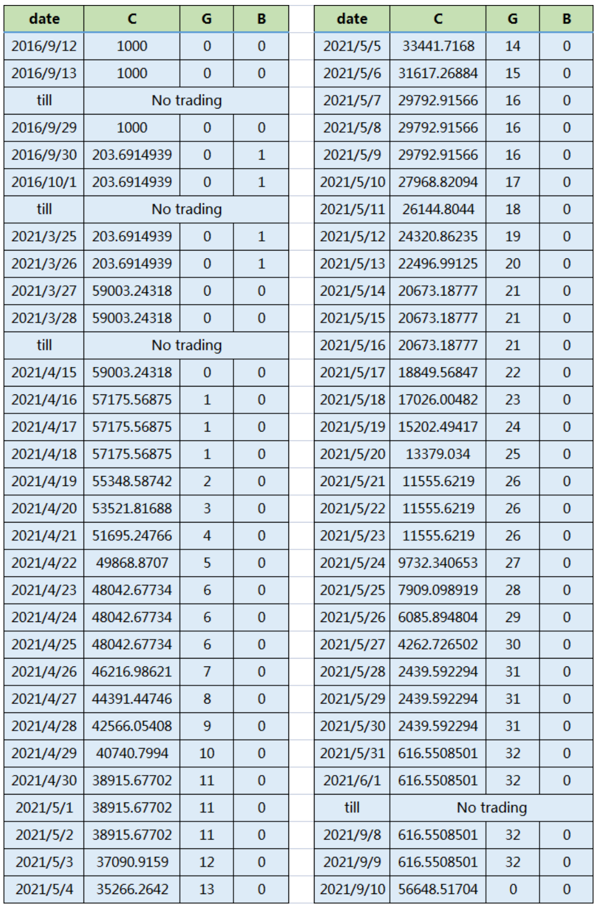
Where P_b represents the daily price of bitcoin, P_g represents the daily price of gold, and ΔP stands for variance in price, which is the value of the subtraction of the two.

Then, find the 1%, 35%, 65% and 99% quantiles of ΔP. In this step, we sort the 1825 spreads data sequentially and then find the target data with the serial number1825*1%, 1825*35%, 1825*65% and 1825*99%.

- The following are the decision-making principles we have developed.
  - When 65% quantile $< \Delta P <$ 99% quantile, We choose to sell bitcoin and buy gold
  - When 1% quantile $< \Delta P <$ 35% quantile, We choose to sell gold and buy bitcoin.
  - When $\Delta P >$ 99% quantile, selling all the bitcoin turns into our option.
  - When $\Delta P <$ 1% quantile,it is reasonable to sell all the gold.
- At the same time, it is well worth mentioning that we have considered the different trading time limits for bitcoin and gold because bitcoin can be traded daily, while gold is only traded on the opening day.

5.3 Results and trading strategy

With the help of matlab, we finally turned our idea into reality. Ultimately, we derive the following trading strategy from the model:



date	C	G	B
2016/9/12	1000	0	0
2016/9/13	1000	0	0
till	No trading		
2016/9/29	1000	0	0
2016/9/30	203.6914939	0	1
2016/10/1	203.6914939	0	1
till	No trading		
2021/3/25	203.6914939	0	1
2021/3/26	203.6914939	0	1
2021/3/27	59003.24318	0	0
2021/3/28	59003.24318	0	0
till	No trading		
2021/4/15	59003.24318	0	0
2021/4/16	57175.56875	1	0
2021/4/17	57175.56875	1	0
2021/4/18	57175.56875	1	0
2021/4/19	55348.58742	2	0
2021/4/20	53521.81688	3	0
2021/4/21	51695.24766	4	0
2021/4/22	49868.8707	5	0
2021/4/23	48042.67734	6	0
2021/4/24	48042.67734	6	0
2021/4/25	48042.67734	6	0
2021/4/26	46216.98621	7	0
2021/4/27	44391.44746	8	0
2021/4/28	42566.05408	9	0
2021/4/29	40740.7994	10	0
2021/4/30	38915.67702	11	0
2021/5/1	38915.67702	11	0
2021/5/2	38915.67702	11	0
2021/5/3	37090.9159	12	0
2021/5/4	35266.2642	13	0
2021/5/5	33441.7168	14	0
2021/5/6	31617.26884	15	0
2021/5/7	29792.91566	16	0
2021/5/8	29792.91566	16	0
2021/5/9	29792.91566	16	0
2021/5/10	27968.82094	17	0
2021/5/11	26144.8044	18	0
2021/5/12	24320.86235	19	0
2021/5/13	22496.99125	20	0
2021/5/14	20673.18777	21	0
2021/5/15	20673.18777	21	0
2021/5/16	20673.18777	21	0
2021/5/17	18849.56847	22	0
2021/5/18	17026.00482	23	0
2021/5/19	15202.49417	24	0
2021/5/20	13379.034	25	0
2021/5/21	11555.6219	26	0
2021/5/22	11555.6219	26	0
2021/5/23	11555.6219	26	0
2021/5/24	9732.340653	27	0
2021/5/25	7909.098919	28	0
2021/5/26	6085.894804	29	0
2021/5/27	4262.726502	30	0
2021/5/28	2439.592294	31	0
2021/5/29	2439.592294	31	0
2021/5/30	2439.592294	31	0
2021/5/31	616.5508501	32	0
2021/6/1	616.5508501	32	0
till	No trading		
2021/9/8	616.5508501	32	0
2021/9/9	616.5508501	32	0
2021/9/10	56648.51704	0	0

6 Test the model

6.1 Sensitivity analysis

To determine the sensitivity of the strategy to the transaction cost, we vary the transaction cost several times, obtain the better trading strategy with different transaction costs and estimate its return by the above model, and draw the return curve.

By reviewing the information, we learned that the transaction cost of gold is generally slightly higher than that of bitcoin.

- let $\alpha_{bitcoin} = 0.01$, test $\alpha_{gold} = 0.01, 0.02, 0.03, 0.04, 0.05$ seperately.

  Here are the results:

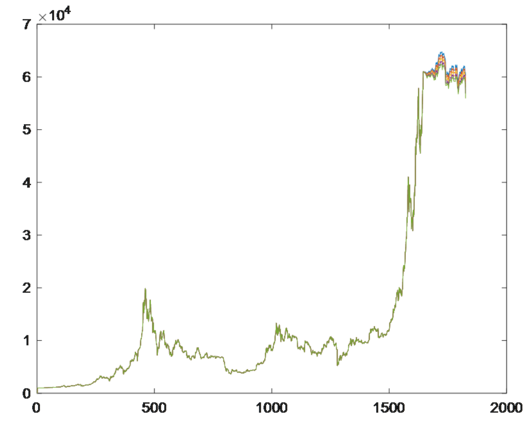


Figure 13: sensitivity analysis—when α_{gold} changes

The graph shows that the impact of transaction costs of α_{gold} on the strategy is mainly concentrated after day 1600, so we capture the graph after 1600 in order to visualize the sensitivity of the strategy to transaction costs. The intercepted images are shown as follows:

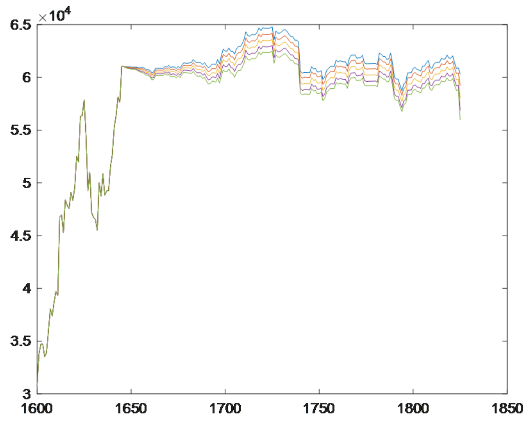


Figure 14: Partial sensitivity analysis—when α_{gold} changes

- let $\alpha_{gold} = 0.02$, test $\alpha_{bitcoin} = 0, 0.01, 0.02, 0.03, 0.04$ seperately. The results can be seen in the figure below:

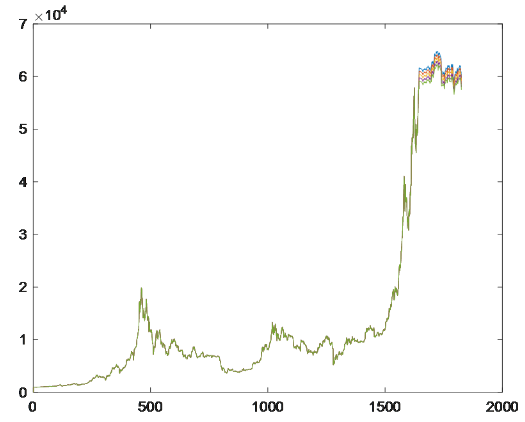


Figure 15: sensitivity analysis—when $\alpha_{bitcoin}$ changes

- As mentioned above, the impact of transaction costs of $\alpha_{bitcoin}$ on the strategy is mainly concentrated after day 1600, so we still capture the chart after 1600:

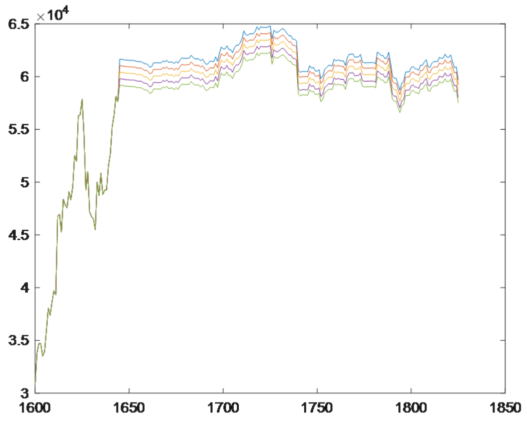


Figure 16: Partial sensitivity analysis—when $\alpha_{bitcoin}$ changes

By analyzing the graphs, we conclude that as the transaction cost increases, the return decreases, but the effect on the rate of return is low. Therefore, the model is less sensitive to transaction costs.

6.2 Prove the effectiveness of the strategy

In order to prove the strategy based on the model to be the relatively effective way in terms of earning profits,

To prove that our resulting trading strategy is better, we use the indicators such as the earning Rate and Sharpe ratio. The simulated profit can be calculated as follows:

$$S_{1825} - S_0 = 56648.51 - 1000 = 55648.51,$$

Besides, through calculation we find the earning Rate id 55.65%. For evaluation purposes, we use the average annualised rate of return, which is 11.1297%. Therefore we can conclude that our strategy has the ability to achieve a higher annualised rate of return.

The Sharpe ratio is calculated by subtracting the risk-free rate from the average of the fund's NAV growth rate and dividing it by the standard deviation of the portfolio return, which represents the excess return per unit of total risk and also reflects the extent to which the fund's NAV growth rate exceeds the risk-free rate per unit of risk.

Based on the Sharpe Ratio formula, we arrive at:

$$SR = [E(Rp) - Rf]/\sigma p = 1.02,$$

where E(Rp) represents the expected return of the portfolio; Rf represents risk-free rate, in our calculation we let Rf=1.6%; op represents standard deviation of the portfolio.

With a Sharpe Ratio greater than 1, we can conclude that the return on investment outweighs the risk.

At the same time, we bring the actual price of gold and bitcoin into the model for backtesting calculations and come up with an annualised return and calculate the error:

$$S_{1825} - S_0 = 57663.2 - 1000 = 56663.2,$$

$$e = \frac{\text{actualprofit} - \text{simulatedprofit}}{\text{actualprofit}} = 1.8\%,$$

according to the standard , 1.8% is a small error, indicating that the prediction model is more accurate and further proves that our strategy is the best.

7 Conclusion

7.1 Strengths

Our model offers the following strengths:

Accuracy: According to the results of the Price Forecasting Model based on Long Short-Term Memory Network, the accuracy of the predicted price can be assured through the test of the model.

Stability: Through the sensitivity and robustness analysis, . In other words, our models have high stability and extensive applicability.

Preciseness: The establishment of our models is not from subjective assumption, but after the analysis of multiple indicators, such as the feasibility analysis of Cross-Species Arbitrage between the two products. What's more, we make our trading strategies as close to reality as possible, for example, we do not set up transactions during the holidays.

Effectiveness: When it comes to the effectiveness of our model, annualized rate of return can serve as a good measure. After continuous correction and improvement, the final annualized rate of return of our model is 11.1297%, which means our trading strategy has a high return on investment and strong investment feasibility.

7.2 Possible Improvements

We cannot deny that our model has the following limitations and related improvements:

- Despite several rounds of training and debugging, we have not yet found a good solution to the hysteresis in prediction.
- Our model can only identify a few particular factors.We haven't give due attention to the invisible influences from the external environment.

References

[1] Alex Graves. "Long Short-Term Memory" (2012)

[2] Alex Sherstinsky. "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network" Physica D: Nonlinear Phenomena 404 (2020): 132306.

[3] Felix A. Gers, Jürgen Schmidhuber and Fred Cummins. "Learning to Forget: Continual Prediction with LSTM" Neural Computation 12 (2000): 2451-2471.

[4] Gatev E,Goetzmann W N, Rouwenhorst K G. Pairs trading: Performance of a relative-value arbitrage rule[J]. REVIEW OF FINANCIAL STUDIES,2006,19(3):797-827.

[5] Chen H, Chen S, Li F. Empirical Investigation of an Equity Pairs Trading Strategy[J].Social Science Electronic Publishing,2012.

[6] Girma P B, Paulson A S. Risk arbitrage opportunities in petroleum futures spreads[J]. Journal of Futures Markets, Wiley Online Library, 1999, 19(8):931-955.

[7] Ye W.Y., Sun L.P., Miao B.Q.. A dynamic cointegration study of gold and bitcoin based on a semi-parametric MIDAS quantile point regression model[J]. Systems Science and Mathematics, 2020, 40(7):16.

[8] Yang, Pingle. Design of cross-species arbitrage strategy for stock index futures[D]. University of Electronic Science and Technology, 2021.

Appendices

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import math
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout, LSTM
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
#from keras import optimizers
import time
import xlwt

def creat_dataset(dataset, look_back=1):
    dataX, dataY = [], []
    for i in range(len(dataset)-look_back-1):
        a = dataset[i: (i+look_back)]
        dataX.append(a)
        dataY.append(dataset[i+look_back])
    return np.array(dataX), np.array(dataY)

dataframe = pd.read_csv('/Users/liang/PycharmProjects/pythonProject6/
   dataset/LBMA-GOLD.csv',
                        header=0, parse_dates=[0],
                        index_col=0, usecols=[0, 1], squeeze=True)
dataset = dataframe.values
#dataframe.head(10)

'''plt.figure(figsize=(12, 8))
dataframe.plot()
plt.ylabel('price')
#plt.yticks(np.arange(0, 300000000, 100000000))
plt.show()'''

scaler = MinMaxScaler(feature_range=(0, 1))
dataset = scaler.fit_transform(dataset.reshape(-1, 1))

train_size = int(len(dataset)*0.8)
test_size = len(dataset)-train_size
train, test = dataset[0: train_size], dataset[train_size: len(dataset)
   ]

look_back = 1
trainX, trainY = creat_dataset(train, look_back)
testX, testY = creat_dataset(test, look_back)

trainX = np.reshape(trainX, (trainX.shape[0], trainX.shape[1], 1))
testX = np.reshape(testX, (testX.shape[0], testX.shape[1], 1))

# create and fit the LSTM network
model = Sequential()
model.add(LSTM(4, input_shape=(None, 1)))
model.add(Dense(1))
model.add(Activation('relu'))
model.compile(loss='mean_squared_error', optimizer='adam')
```

```
history = model.fit(trainX, trainY, epochs=250, batch_size=1, verbose
    =2)
#model.save(os.path.join("DATA","Test" + ".h5"))
# make predictions

trainPredict = model.predict(trainX)
testPredict = model.predict(testX)

trainPredict = scaler.inverse_transform(trainPredict)
trainY = scaler.inverse_transform(trainY)
testPredict = scaler.inverse_transform(testPredict)
testY = scaler.inverse_transform(testY)

trainScore = math.sqrt(mean_squared_error(trainY, trainPredict[:, 0]))
print('Train Sccore %.2f RMSE' %(trainScore))
testScore = math.sqrt(mean_squared_error(testY, testPredict[:, 0]))
print('Train Sccore %.2f RMSE' %(testScore))

trainPredictPlot = np.empty_like(dataset)
trainPredictPlot[:] = np.nan
trainPredictPlot = np.reshape(trainPredictPlot, (dataset.shape[0], 1))
trainPredictPlot[look_back: len(trainPredict)+look_back, :] =
    trainPredict

testPredictPlot = np.empty_like(dataset)
testPredictPlot[:] = np.nan
testPredictPlot = np.reshape(testPredictPlot, (dataset.shape[0], 1))
testPredictPlot[len(trainPredict)+(look_back*2)+1: len(dataset)-1, :]
    = testPredict

fig1 = plt.figure(figsize=(6, 4))
plt.plot(history.history['loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.show()

fig2 = plt.figure(figsize=(10, 8))
plt.plot(scaler.inverse_transform(dataset))
plt.plot(trainPredictPlot)
plt.plot(testPredictPlot)
plt.ylabel('price')
plt.xlabel('date')
plt.show()

fig3 = plt.figure(figsize=(10, 8))
plt.plot(np.arange(train_size+1, len(dataset)+1, 1), scaler.
    inverse_transform(dataset)[train_size:], label='dataset')
plt.plot(testPredictPlot, 'g', label='test')
plt.ylabel('price')
plt.xlabel('date')
plt.legend()
plt.show()
```