New Times Gold Diggers: Mining Profits in Price

With the rise of information technology, an increasing number of independent traders attempt to derive their strategies using statistical tools, and mining information from market price fluctuations rightfully becomes the core part of it. In trying to get excess returns, a bunch of strategies exist to combat the return on investment, such as VaR, CVaR, ARIMA, etc. However, a more targeted strategy is necessary for independent investors in optimizing their portfolio individually, which draws forth the problem we need to solve.

We have developed a model for a specific trader which used the data of gold and bitcoin price for a five-year trading period from September 11, 2016 to September 10, 2021, operating from $1,000 and calculated the estimated investment value on September 10, 2021. In doing so, we:

-Built a price prediction model based on **LSTM neural network** and trained the model using the trading prices of Bitcoin and gold for 365 days prior to 11/9/2016. The test lengths were taken as 30 days (one month) and 365 days (one year), respectively, and the trend of the test set prediction was consistent with high accuracy. Therefore, we successfully made a prediction for the 5-year gold price and bitcoin forecast from 2016 to 2021.

-Next, we combined the prediction with our **risk-built in DP model**. We took the gold and bitcoin transaction changes in a 3-days-period as the decision variable, taking the highest gross profit of the target in the cycle as the objective function, and established the risk factor participation model. In this process, we creatively proposed **three types of risk factors** and solidified them in constraints, replacing the methodology of maximizing the Sharpe Ratio. Then, we double validated the **stability of the model** through internal stability test and external model comparison, meanwhile gave the forecast trend after adjusting the transaction cost to provide more reference to the trader.

-Finally, we summarized the pros and cons of the model and attached a 2-page memo.

The principal advantage of our strategy is that it can enable investors who under the condition of information asymmetry (lack of market psychology analysis, international situation forecast, etc.) to make a pot of gold **merely by analyzing the price trend**. This is a strategy that is not only suitable for ordinary retail investors, but also provides some inspiration for giants.

Moreover, the DP model was developed from scratch, so it is **unique** in the market. It is dynamically iterative, meaning that each day a new achievable portfolio can be derived independently of the previous day's profit and loss. This strategy is not only suitable for T+1 trading, but also consorts with the intraday high-speed trading after modifying the parameters.

Contents

1 Introduction

Trading strategy is a topic that the world has been investigated for centuries. The previous study was done in 1827, when Robert Brown, a Scottish biologist, gave his own name to the phenomenon of tiny particles moving freely through liquids: Brownian motion. Louis Bachelier was the first to quantify the movement in the field of econometrics. In his published paper later in 1900, he argued that the reasons affecting the rise and fall of stock prices were endless and could not be predicted dynamically and accurately by probability theory models, nor was it an exact science. But at a static moment in the market, a mathematical model can be established to analyze the probability of the market [1]. This opened up a precedent for future generations to use mathematical models to pursue market profits through the random behavior of the massive.

1.1 Problem Summary

The background of the problem assumes that a market trader holding a principal of 1000 USD was ready to enter the spot market for trading on September 11, 2016. The spot available includes spot gold provided by the London Bullion Market Association and bitcoin provided by global electronic currency trading platforms. We need to develop a trading strategy that maximizes investment profits over a 5-year period, using only the past steam of daily prices to date to decide when we will increase, decrease or keep the spot. The special displayed portfolio consisting of cash, gold, bitcoin **[C, G, B]** and the differed commission rate of each transaction are also notable.

In addition to continuously optimizing the maximum return on the principal after 5 years, we also need to provide evidence of strategy dominance which is the most likely to be demonstrated in the comparison of other models and the inner stability of the model itself. Finally, we must test the strategy sensitivity based on various levels of changes in transaction costs.

1.2 Data Sources

Our model is informed by the gold daily prices sourced by London Bullion Market Association and the bitcoin daily prices by the source of NASDAQ, each with their own dimensions. The heads include the dates and the daily opening prices of gold and bitcoin over the five-year period 2016-2021.

1.2.1 Data Cleaning

The census data provided had missing and partially filled in data that would have been challenging to effectively utilize. We did the following to sanitize the data-set:

-Deal with the factors that had incomplete data, such as the missing days (represented by the head 'date') in gold and bitcoin transactions other than the closed days. Incomplete data for a factor would inhibit the creation of a proper model, as there could be hidden trends that are not apparent because of missing data. Thus, we fill the short-term gap with interpolation method. Because the prices of bitcoin and gold do not fluctuate significantly in the short term, the interpolation method has little effect on the data itself.

-Sort out the date format. The date formats of data sets varies, which brings unnecessary programming trouble. Therefore, we uniformly process the data into the format of dd-mm-yyyy.

-Decompose the data into multiple partial signals such as linear, nonlinear, and periodic signals. After removing noise, the data is reconstructed for use by the neural network model. Next, normalize the reconstructed data into the 0-1 interval.

1.3 Existing Models

-Markowitz's portfolio theory established when the American economist Markowitz (1952) published the paper "*The Choice of Portfolio*"[2], marking the beginning of modern portfolio theory. He uses mean-variance model analysis to draw the conclusion that risk reduction can be effectively achieved through investment portfolios.

The basic assumptions of Markowitz's portfolio theory are:

(1) Investors are risk-averse and pursue maximizing expected utility;

(2) Investors choose portfolios based on the expected value and variance of returns;

(3) All investors in the same single-period investment period. Markowitz proposed to use the expected return and its variance (E, δ^2) to determine the efficient portfolio.

$$\begin{gathered} min\delta^2(r_p) = \sum\sum w_i w_j \text{cov}\,(r_i, r_j) \\ E(r_p) = \sum w_i r_i \end{gathered} \tag{1}$$

In the formula, r_p: combination income;

r_i: Income of the i and j assets;

w_j: The weight of asset i and asset j in the combination;

$\delta^2(r_p)$: The variance of the portfolio return is the overall risk of the portfolio;

cov (r_i, r_j): The covariance between the two assets.

In Markowitz's model, the variance is used to describe risk, and the distribution of returns is symmetrical. Many scholars have put forward different opinions, especially the **CVaR** model. CVaR is a better risk measurement technology than VaR proposed by RockafeUar and Uryasev in 1997[3], which counts the average loss value of the portfolio under the condition that the loss of the portfolio exceeds a given VaR value.

1.4 Our Thoughts

From the perspective of existing trading models, there is not only the problem of unsatisfied price prediction accuracy, but also the trading time rely heavily on human manipulation. How to come up with a trading model that is **accurate in prediction** and can **achieve dynamic self-iteration** has become our main focus.

First of all, the date-price graphs of gold and bitcoin are linear graphs that evidently follow the time logic, so we first consider classical models such as exponential smoothing. Next, we consider the use of neural network models such as BP network or RNN, which can greatly improve the prediction accuracy through iteration and convolution.

More importantly, traditional financial models (such as **VaR\CVaR**) maximize returns by constraining internal risks and setting risk limits. They pay more attention to determining the time point to stop losses. But in our **DP model**, we are not intimidated by making short-term losses, because algorithms naturally act at the highest point of predicted price by mandatory formulating a new investment portfolio every cycle (through dynamic state transition), and avoid the superposition of internal risks.

2 Background

2.1 The Quantitative Trading

Quantitative trading refers to replacing human subjective judgments with advanced mathematical models, using computer technology to select various "high probability" events that can bring excess returns from huge historical data to formulate strategies, which greatly reduces investor sentiment. The impact of volatility and avoid making irrational investment decisions when the market is extremely manic or pessimistic [4].

2.1.1 Strategy Processing Logic

The strategy processing logic needs to consider factors such as spot selection, timing, position management, take profit and stop loss.

-Quantitative spot selection is to use quantitative methods to select a certain investment portfolio, hoping that such a portfolio would obtain investment returns that exceed the market index.

-Quantitative timing refers to the use of quantitative methods to determine the buying and selling points. If expected a rise, buy and hold, otherwise sell to clear the position.

-Position management is the decision time to invest in a certain portfolio or the time of entering/leaving the market (in batches).

-Take Profit and Stop Loss. Take profit, as the name implies, means selling in time when gaining profits; while stop loss represents selling goods in time when price drop to avoid greater losses. Timely decision is effective in obtaining stable profits.

2.2 Spot

Spot refers to physical goods available for shipment, storage and manufacturing. Spot available for delivery can be exchanged for cash on a near-term or forward basis, or prepaid, a general term for goods that the buyer pays for within a very short period of time. Spot is the symmetry of futures.

-For this question, old is classified a physical spot, and Bitcoin a conceptual spot.

2.2.1 T+D and T+0

In T+0 method, the spots must be delivered at the time of the transaction happens. However in T+D method, you can defer spot delivery to any specific time at the price on the previous contract, not the date of delivery.

-For this question, gold is officially recognized as T+D spot while Bitcoin uses the T+0 delivery method.

3 Nomenclature

Symbol	Definition
n	Total length of investment
$[C_i, B_i, G_i]$	Portfolio on day i in USD
P_i	Gross profit on day i

T	Prediction and traversal cycle
P_j	Maximum gross profit in the cycle
β_i	Perceptible risks factor
$[C_i, G_i, B_i]_s$ and $[C_i, G_i, B_i]_g$	Safe-position in USD

4 Assumptions

(1) The data attached is daily based, which means we don’t need to consider the possibility of multiple transactions in the same day for Bitcoin transactions as legalized in the T+0 mode, that is, the shortest Bitcoin trading cycle is COUNTED one day.

(2) Requirements of only relying on price to formulate strategies lead to the assumption that only the information that price itself can bring to traders is considered, and the impact of external factors such as political fluctuations, emergencies and market psychology on gold and bitcoin prices are not important.

(3) Assume that traders prefer long-term quantitative strategies (once developed, sustainable for 5 years or more), rather than short-term unsustainable strategies that are prone to failure.

(4) Assume that cash holdings will increase at a fixed annual interest rate of 2%, and the ‘short selling’ of our spot is not allowed.

5 Model Development

5.1 Model Background of LSTM

RNN is a special neural network structure, which itself is a network containing loops, allowing information to be passed between neurons. Although RNN can theoretically retain the information of all historical moments, in actual use, the transmission of information tends to be gradually attenuated because the time interval is too long, and the effect of the information is greatly reduced after a period of transmission. Therefore, ordinary RNNs do not have a good solution to the long-term dependence of information. In order to overcome this problem, Hochreiter improved RNN in 1997 and proposed a special RNN model - LSTM network, which can learn long-term dependent information Long Short Term Memory (LSTM) network is a special The RNN model, whose special structural design makes it possible to avoid long-term dependency problems, remember the information of very early moments is the default behavior of LSTM, and there is no need to pay a high price for it [5].

In this question, our requirement is: try to predict the closing price of the next day according to the short-term fluctuations of Bitcoin and gold prices(using only the price up to the day), but in the long run we don’t want to lose the information by long-term historical prices, such as seasonal price fluctuations etc. All things considered, LSTM became our preferred method.

5.2 The Establishment of Price Forecasting Model through LSTM

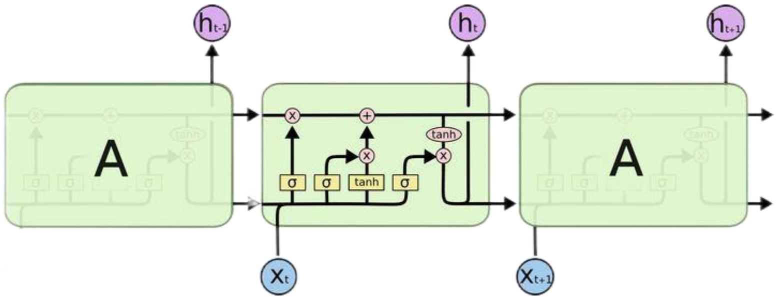

Figure 1: Flowchart representation of LSTM neural network layers

LSTM has 4 neural network layers and interact in a special way. Some basic module representation methods are shown in the up figure.

-Yellow square: represents a neural network layer (Neural Network Layer)

-Pink circles: Indicates bitwise operations or pointwise operations, such as vector sum, vector product, etc.

-Single arrow: indicates signal transfer (vector transfer)

-Confluence arrow: indicates the connection of two signals (vector concatenation)

-Shunt arrows: Indicates that the signal is duplicated and passed to 2 different places.

LSTM mainly includes three different gate structures: **forget gate, memory gate and output gate**. These three gates are used to control the information retention and transmission of the LSTM, which is finally reflected to the cell state C_t and output signalh_t.

The **forget gate** consists of a sigmoid neural network layer and a bitwise multiply operation, which can be expressed as:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2}$$

The memory gate consists of an input gate and a tanh neural network layer and a bitwise multiplication operation, which can be expressed as:

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh\,(W_C \cdot [h_{t-1}, x_t] + b_C) \end{aligned} \tag{3}$$

With the forget gate and the memory gate, we can update the **cell state**C_t.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4}$$

The output gate works with the $tanh$ function and the bitwise multiplication operation to pass the cell state and the input signal to the output.

$$\begin{gathered} o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \\ h_t = o_t * \tanh\,(C_t) \end{gathered} \tag{5}$$

Among them, the $tanh$ function is a kind of activation function, and the function image is:

$$f(x) = \frac{e^x - e^{-x}}{e^{-x} + e^x} \tag{6}$$

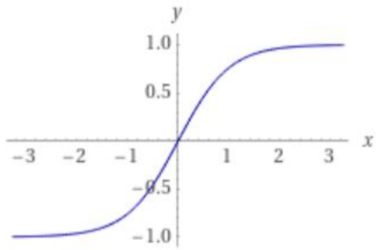

Figure 2: Plot of the activation function using Walfram

5.2.1 The Training of the LSTM Model

The title requires that we can only predict today's closing price **through the previous day,** so we consider:

-First, use the price data of 365 days before September 10, 2016 (data source[6][7]: Microsoft's official electronic currency API, https://cryptosheets.com/;London Bullion Market Association website https://www.lbma.org.uk/) to train the LSTM model using MATLAB;

-Then verify the accuracy of the time series fit using the 5-year price data from 2016-2021;

-Finally, use the model to forecast the current closing price daily from September 10, 2016. In order to **incorporate the daily new price into the prediction model to realize dynamic iteration**, we added a **FOR loop** to LSTM and successfully solved the problem.

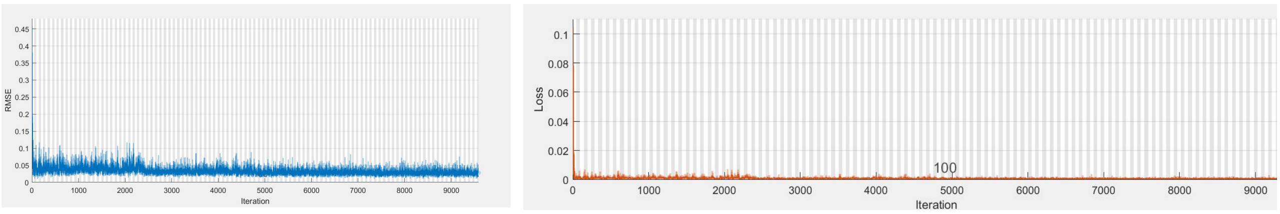

Figure 3: The Training Process of LSTM (using gold price)

It can be seen from the blue line that the training achieved a high precision, and the yellow line that the systematic loss is close to 0, which means the training effect of the model is excellent.

5.2.2 The Testing of the LSTM Model

The test lengths are taken as **30 days (one month) and 365 days (one year)**, respectively. As is shown in the chart below, the trend of the test set prediction is consistent with high accuracy, indicating that the prediction is reliable. Therefore, the model can surely be used to predict future prices trend.

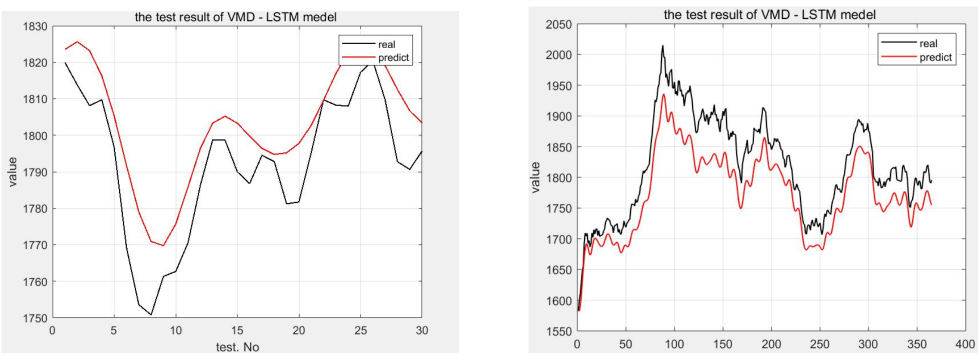

Figure 4: Test Result for One Month & One Year

5.2.3 The Application of the LSTM model

We trained the price of Bitcoin in the same way, then applied the trained model to the price trend of the month before September 10, 2016, and successfully made a prediction for the 5-year **gold price and bitcoin forecast** from 2016 to 2021. NOTICE that we posted the complete chart. In fact, in the actual operation, the API is updated iteratively and the trader cannot see the data after that day.

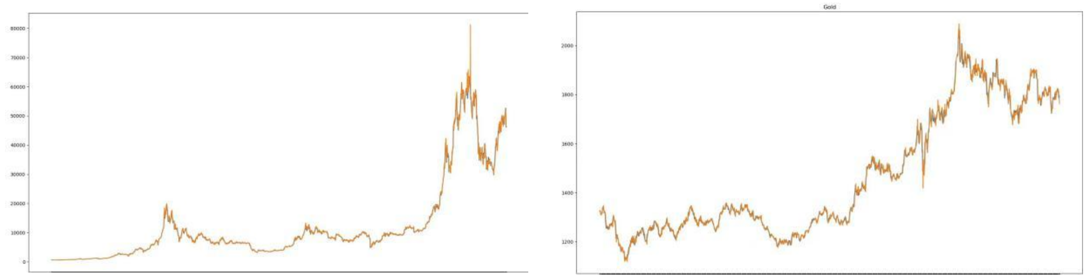

Figure 5: Final Prediction Result (Bitcoin the left)

5.3 Model Background of DP

Dynamic programming (DP) is a method used in mathematics, management science, computer science, economics, and bioinformatics to solve complex problems by decomposing the original problem into relatively simple sub-problems Methods. Dynamic programming is often applied to problems with overlapping subproblems and optimal substructure properties.

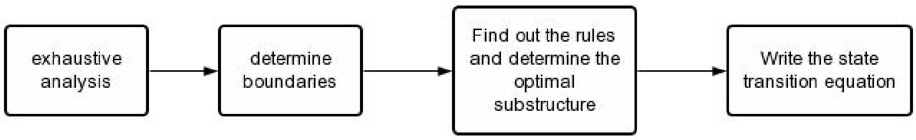

Figure 6: The Process of Dynamic Programming

Characteristics of problems that can be solved with DP include:

-The problem has the property of optimal substructure, which means the optimal solution of a problem consists of optimal solutions to subproblems.

-No aftereffect: the process after a certain state will not affect the previous state.

-There are overlapping sub-problems: dependent sub-problems may be used multiple times in the next stage of decision-making.

For this question, the problem requires us to formulate a position strategy currently only by predicting tomorrow's price. So the problem has the property of optimal substructure: the 5-year global optimization can be divided into daily optimizations, and the systematic average of daily profit and loss becomes the total profit and loss. Taking all into consideration, DP becomes our best choice.

5.4 The Establishment of Dynamic Iterative Trading Strategy

We mainly use ***python*** to implement dynamic programming models. The specific logic is as follows:

(1) Set up the total length of investment as n and the date as n_i

(2) Set up the initial state of the portfolio $[C_i, B_i, G_i]$, $i = 1 \ldots n$

(3) Set up the gross profit P_i on day i (P_i=profit R_i+principal D_i)

(4) Set up a traversal cycle$T$, traverse the maximum gross profit P_j ($P_j = max(P_i)$) of the bitcoin and gold investment portfolio within T days, and perform dynamic iteration. At every $i + 1$ day, calculate $max(P_j, P_{i+1})$ as the newP_j, and get the corresponding investment portfolio$[C_j, B_j, G_j]$. **At this point the portfolio is the best option to open the next day.**

5.4.1 Introduce Risk Factors

Gold and Bitcoin are high-leverage and high-yield investment products. Investors can use small funds to make large-scale transactions, thereby improving the utilization rate of funds. It also lowers the threshold for transactions.

However, in the 1960s, the **Capital Asset Pricing Model** was proposed, which revealed the relationship between the expected rate of return of assets (expected excess rate of return) and risk. The formula points out that the expected excess return of an asset is determined by the exposure of the asset to market risks, that is, the excess expected return of an asset can be completely explained by market factors.

$$E[R_i^e] = \beta_i \lambda + \alpha_{\mathrm{i}} \tag{7}$$

The purpose of investing is to find factors that explain the return on assets. Notable is that, as natural safe-haven assets, gold and bitcoin are different from listed companies in the stock market, and the exposure of assets to market risks is relatively stable, therefore the factors are easy to summarize. **We derive three manifest and perceptible risks β_{i} that the price itself can bring to investors:**

-The First Type of Risk: Transaction Risk caused by Random Price Fluctuations

Over the past five years, gold's annual price volatility has been close to 40% and peaked at more than 50%. Intraday volatility is generally between 1% and 3%. This is pertaining to the relatively high price of gold and the unstable economic and political environment in the world [5].

Solution strategy:

(1) Always set the safe-position $[\mathrm{C_i}, \mathrm{G_i}, \mathrm{B_i}]_s$ for each operation to avoid exacerbating losses.

(2) Limit the number of holdings $[\mathrm{C_i}, \mathrm{G_i}, \mathrm{B_i}]$, and each transaction volume ratio should be restrained to a specific range, set as $(\frac{1}{6}, \frac{1}{3})$.

(3) When making a profit, set the lowest profit level $[\mathrm{C_i}, \mathrm{G_i}, \mathrm{B_i}]_g$for automatically closing the deal.

-The Second Type of Risk: Seasonal Fluctuations in Bitcoin and Gold Prices

In the short and medium term of the long-term trend, the price of gold shows slight seasonality. The probability of the price peak appearing in the second season soars, and the annual lows basically appearing at the beginning of the year, probably owing to the changes in the market demand.

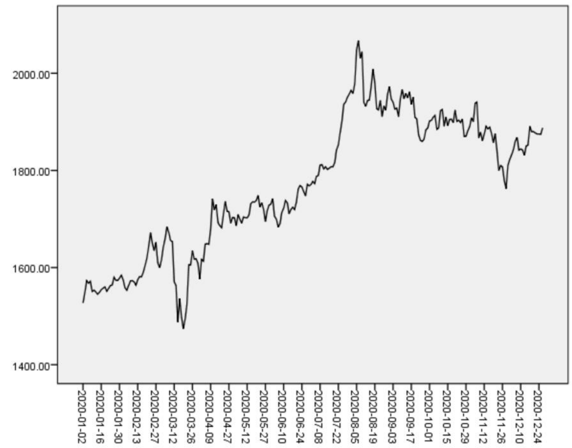

Figure 7: The Gold's seasonal price swings in 2020, as an example

Solution strategy: Add a seasonal risk factor β_i to gold earnings in the first quarter that conversely constrains the holding.

The Third Type of Risk: Asset Allocation Risk Caused by Fluctuations in the Correlation between Bitcoin and Gold

Historically, Bitcoin and gold have not had a strong correlation. But at certain times of the year, Bitcoin and gold appear some kind of fugacious correlation trend. Therefore, when there is a negative correlation between the price of gold and Bitcoin, the trader need to increase the gold holdings and reduce the Bitcoin holdings; otherwise, we should control the positions and be careful about the possibility of liquidation.

Solution strategy: The algorithm we combined with our DP can naturally avoid the third type of risk, because we set a tendency to diversify investment.

5.4.2 The Strategies for Non-Tradable Days

Gold cannot be traded on weekends. If encountered with the closing days, we fix our gold position and observe **whether the price of Bitcoin rises**. If it does not rise, sell bitcoin for cash, and if it rises, buy bitcoin with cash. When only the transaction of buying Bitcoin is required, record the change in the Bitcoin amount as $\Delta B > 0$, at this time, the change of cash is $1.02 * \Delta B$. When only the transaction of selling Bitcoin is required, the change in Bitcoin amount is $\Delta B < 0$, at this time, the change in the amount of cash is $0.98 * \Delta B$.

5.4.3 The Application of the DP model

We built the DP model from the scratch using *python* that implements an iterative strategy which incorporates all of the above objective risk factor considerations. The core state transition function and the objective function is uncomplicated:

$$\begin{cases} P_{j+1} = max\ P_j \\ P_j = max\ (P_{Ci}, P_{Gi}, P_{Bi}) \end{cases} \tag{8}$$

$$\begin{aligned} & max\ P_j = \beta_i P_i + \alpha_i \\ & s.t. \begin{cases} C_i - \Delta C_i \geq 0 \\ B_i - 0.02\Delta B_i \geq 0 \\ G_i - 0.01\Delta G_i \geq 0 \end{cases} \end{aligned} \tag{9}$$

Meanwhile we built in the risk factor through algorithm implementations to the constraints and set the final objective function through **greedy methodology** to maximize **total return$\boldsymbol{TR}$.** At the same time, we calculated **total return ratio $\boldsymbol{TRR}$, Sharpe ratios $\boldsymbol{SR}$, and annualized rate of return$\boldsymbol{ARR}$**.

$$TRR = \frac{cumulative\ income - current\ value}{purchase\ cost} \tag{10}$$

$$ARR = \frac{TR}{\text{holding time}} * 365 \tag{11}$$

$$SR = \frac{AR - annualized\ risk - free\ rate}{Standard\ Deviation\ of\ Portfolio} \tag{12}$$

Our forecast period is 3 days ($T = 3$). IT SHOULD BE NOTED THAT since we predicted the highest price in the next 3 days and bought in accordance with the expectation of future earnings, therefore short-term losses are inevitable. Ultimately, however, we can guarantee that the strategy will get the optimal profit. The final yield curve and several financial index are as follows:

TRR	ARR	SR
342571%	685142%	2.45

Table 1: The values of important financial index

USD	GOLD/USD	BTC/USD	DATE
0	68502.62765	1380.017785	2019.9.1
684.9883081	67700.91182	1404.293762	2019..9.2
719.2650729	68196.04112	1491.148349	2019.9.3
70089.87293	0	7.32814E-13	2019.9.4
70082.72368	0	7.008987284	2019.9.5

Table 2: Example of the portfolio [C,G,B] in USD

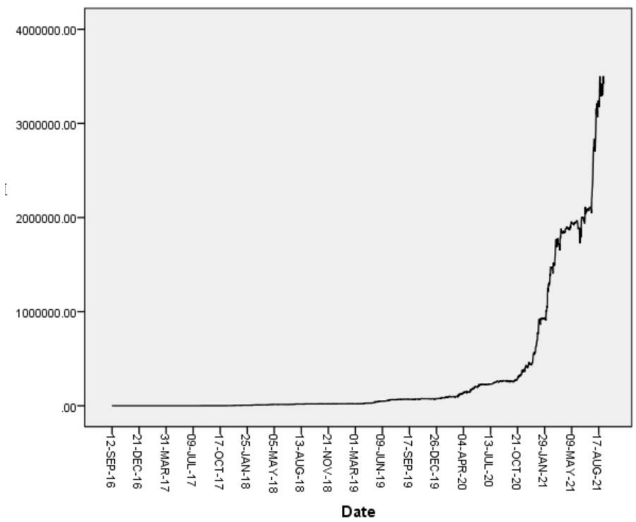

Figure 8: The Yield Curve of our Strategy(y-axis of 10 USD)

6 Results

6.1 Part 1: Trend-Prediction-Based Strategy

For the first question, we adopted an unusual **DP-greedy algorithm model**. The characteristic of the model is that, assuming that relatively accurate predictions are provided, then according to the characteristics of the greedy algorithm (whichever target is profitable, increase the position), the continuous improvement of the profit must be ensured. In order to optimize the accuracy of our prediction, we used the convolutional neural network LSTM model, and after 12,000 iterations and 4 trainings, the MSE was finally controlled to a satisfactory level. This ensures the feasibility of real-time iterative trading strategies.

Our results also perfectly reflect the advantages of the algorithm which is different from common financial models established with the objective function optimizing the Sharpe Ratio. It creatively incorporates the risk factor into the **constraints** instead and meanwhile boldly exploits the leverage to dig the latent profit in the future, safeguarded by the precise prediction. **The profit curve**, **as shown in the figure, eventually reaches a positive peak after short-term fluctuations, which is 34257109.15 USD.**

6.2 Part 2: Comparison & Stability Evidence

6.2.1 Validation of the LSTM Model

To validate the reliability of the model, we decided to split the process into **internal stability validation** and **model comparative advantage validation**. First, in order to verify the internal stability, we first choose to adjust the parameters. If the model can still get relatively good prediction results after adjusting the flexible parameters, it proves that the LSTM model is not affected by internal disturbances. In our LSTM model, there are these parameters for us to adjust:

Parameter	Values
The number of neurons in the first hidden layer	/
Te number of neurons in the second hidden layer	/
Maximum number of LSTM iterations and the initial learning rate	Range $(0.01 - 0.2)$
Batch size of samples	Range $(10 - 450)$

Table 3: The Parameters for Adjustments

We chose to adjust the learning rate, tried 0.1 and 0.2 respectively, and re-plotted the results as shown in the figure (taking the price of gold as an example)**:**

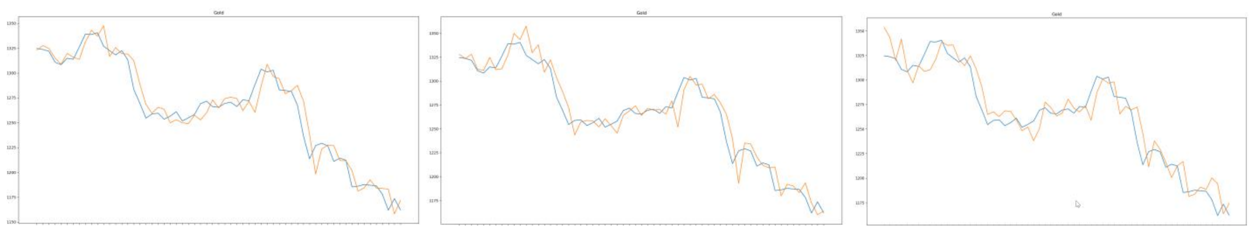

Figure 9: Different Trends under 0.08,0.1 and 0.2 Learning Rate

We took a period of time (10/9/2016-12/9/2016) to zoom in and observe, and found that the prediction changed little after the number of iterations is changed, indicating that the model is not sensitive to data and has good internal stability.

Next, we introduce **ARIMA** (Autoregressive Integrated Moving Average model), one of the time series forecasting analysis methods favored by classical quantitative finance, to provide a comparison. We calculate and compare the mean squared error of the two models, or **MSE**. The larger the MSE, the more the predicted value deviates from the true value, and the more underperforming the model is.

$$MSE = \frac{\sum_{i=1}^{r}(n_i-1)\, s_i^2}{N-r} \tag{13}$$

The comparison between ARIMA prediction results is as follows. The MSE of LSTM is **256 USD**, while the MSE of ARIMA reaches **399 USD**, which shows that LSTM as a machine learning algorithm has a advantage in this problem compared to traditional methods.

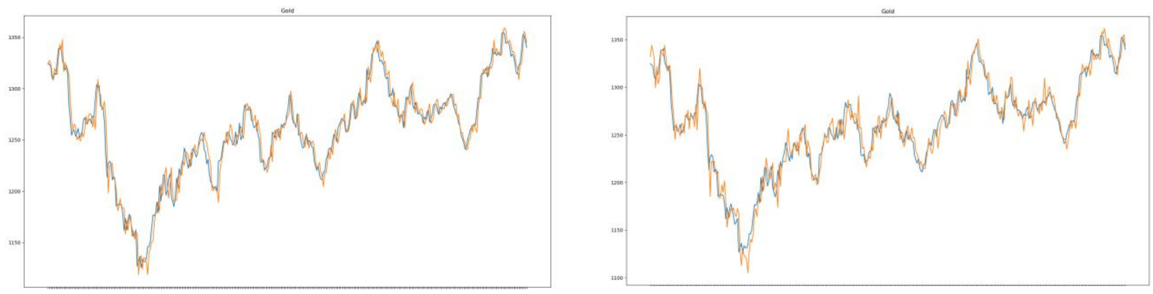

Figure 10: Difference between the Prediction Performance in LSTM and ARIMA

6.2.2 Validation of the Dynamic Programming Model

We need to determine whether the trading strategy will be sensitive to conditions. To prove that the solution we choose is the optimal, consider a certain disturbance to the portfolio. Add a fluctuation of **8%** to the above-mentioned trading volume of gold and bitcoin. If the result after the disturbance is not optimal, then it proves that our original investment portfolio is the optimal solution.

Total Return	34257109	31812066	31611744	31755325
Sharpe Ratio	2.45	2.2773	2.2719	2.269

Table 4: Various Financial Index Results after Adding Fluctuations

The final result illustrated that the financial index value did not exceed the original plan. Therefore, the investment plan we chose is the optimal.

6.3 Part 3: Sensitivity Analysis after Adjustments

In a real environment, transaction costs may be constantly adjusted with changes in the international environment. In Gold we set:

α_1%	1%	2%	3%

In Bitcoin we set:

α_2%	2%	3%	4%

Table 5: The Transaction Costs for Adjustments

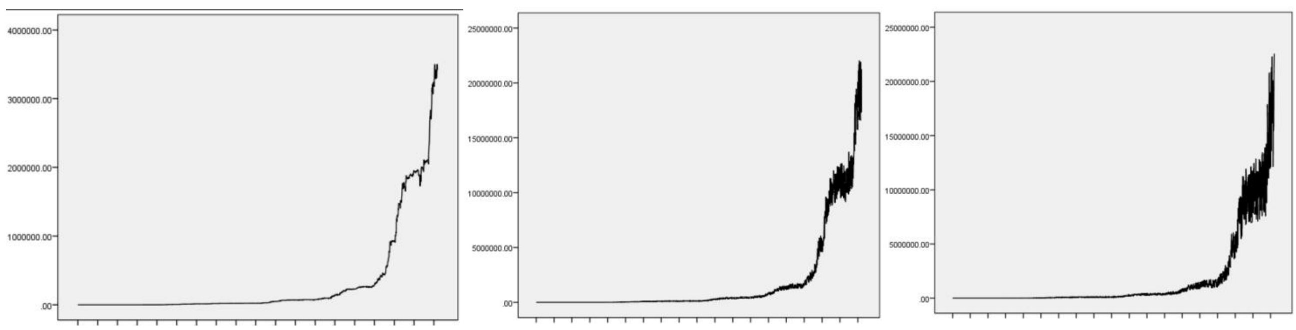

Figure 11: Difference between the Prediction Performance under the Transaction Costs of 3 Sets

The results are shown above. Obviously, as the commission increases, the transaction volume of gold and the final income decreases simultaneously. As the commission became lower, the transaction volume of gold and the final income increases simultaneously. The Bitcoin analysis performs the same.

7 Conclusion

With the help of statistical and mathematical tools, ecometrics models are commonly used nowadays to conduct transactions and obtain an excess return. This article uses data from the United States for a five-year trading period from September 11, 2016 to September 10, 2021, operating from $1,000, and the estimated investment value on September 10, 2021.

First, we established a price prediction model based on the LSTM neural network, and trained the network through the trading prices of Bitcoin and gold before 2016. From September 11, 2016 to September 10, 2021, the price forecast for the next three days is carried out every afternoon. The prediction effect of the model is satisfying through the comparison with the real value. Next, using the DP model, we took the gold and bitcoin transaction changes in the next three days as the decision variable, taking the highest gross profit of the target in the next three days as the objective function, and established the risk factor participation constraints. Finally, we performed a sensitivity test against the model and determined that the scheme is an optimal one.

7.1 Model Strengths

-The principal advantage of our strategy is that it can enable investors who under the condition of information asymmetry (lack of market psychology analysis, international situation forecast, etc.) to make a pot of gold **merely by analyzing the price trend**. This is a strategy that is not only suitable for ordinary retail investors, but also provides some inspiration for giants.

-Our strategy is dynamically iterative, meaning that each day a new achievable portfolio can be derived independently of the previous day's profit and loss. This strategy is not only suitable for T+1 trading, but also very suitable for intraday short-term trading after simply modifying the parameters.

-The model is easily expandable to other portfolios, such as stock portfolios and futures portfolios. Theoretically, the number of objects can be increased to dozens or hundreds.

-Our prediction model was tested for sensitivity and achieved good accuracy and internal stability. Compared with other models, neural networks using Hheuristic algorithm greatly speed up model convergence and perform better in big data processing problems.

-The DP model was developed from scratch, so it is **unique** in the market.

7.2 Model Weaknesses and Limiting Assumptions

-Time consuming to calculate. Each cell of LSTM consists of 4 fully connected layers. If the time span of LSTM lasts long with numerous dimensions instead of 2, the costs might become unbearable.

-The impact of external factors such as political fluctuations, emergencies and market psychology on gold and bitcoin prices are not considered.

8 Memo to the Trader

Date: February 22, 2022

To: The Trader

From: MCM Team #2229018

Subject: Trading Strategy Insights and Recommendations

Dear Friend,

In recent years, mild monetary inflation and quantitative easing are still in place, needless to say that your choice of gold and bitcoin as safe havens is advisable. Although we cannot predict the uncertainty of the future, in an effort to maximize returns from your portfolio, we have modeled how spot bitcoin and gold prices will fluctuate in 5 years.

Unlike the quantitative trading strategies you have seen in the past, such as the double moving average strategy (ARIMA) and VaR, our strategy is not set primarily on risk hedging and risk control, because the market information you now have is only the market price. After careful consideration, we developed a whole new strategy based on **dynamic programming** which takes advantage of all your needs:

-1. Ability to predict the trend in the next 3 days based on historical prices, so as to seize the opportunity;

-2. Ability to achieve dynamic iteration. The strategy will instruct you how much cash, gold or bitcoin to increase or decrease the next day after the close of each day;

- 3. Try to maximize profits and only focus on the final result after 5 years.

This is how the model implements your needs: The model breaks long-term decisions into several short-term investment decisions.

-First, we set a forecast period to predict the one with the highest price among Bitcoin, gold, and savings in 3 days, and increase the position of this object through our algorithm.

-Whenever the closing price of a new day is updated, the model will automatically incorporate the new price and enter a new forecast period.

-In the long run, we will increase our positions in commodities with high yields and reduce our positions in commodities with low yields. The daily investment portfolio is a dynamic equilibrium.

From the results, though you only invested $1,000 in the early stage, after the profits roll in, a narrow price fluctuation will bring hundreds of thousands of profits and losses, eventually reach a profit of **34257109.15 USD** in 9/10/2021.

Reminders: • Our model does not account for external risks like the international turmoil, therefore the strategy natually inclines to be aggressive.

• The forecast period of our model is 3 days. For higher profits, you can appropriately prolong the cycle. But there are also increased risks that come with it, mainly in the likelihood of inaccurate forecasts.

Insights: • Our model detected that transaction costs not only affect market volume, but are also amplified by predicted prices, resulting in sharp swings in returns. Therefore, it is recommended to **increase the position appropriately when the transaction cost decreases** to gain greater benefits.

- The global economy has entered a period of unprecedented uncertainty. Bitcoin can be a relatively safe asset, but the future is unpredicted. Bitcoin's high returns relative to gold also come with high risks. Given a number of complex factors, it's hard to draw a conclusion right now whether we should invest in virtual currencies.

Best Regards,

Team #2229018

References

[1]Bachelier, L.. "Theory of speculation.".

[2] Myron, S. , and P. The." SAMUELSON, Paul Anthony, "Intertemporal Price Equilibrium: to the Theory of Speculation," Weltwirtschaftliches Archiv, 79. SCHOLES. The Market for Securities: Substitution Versus Price Pressure and the Effects of Information on Share of Business, 179-211. " patents finance (1972).

[3] Markowitz, Harry M." Portfolio Selection: Efficient Diversification of Investment." The Journal of Finance 15.3(1959).

[4] Rockafellar, R. T., and S. Uryasev. "Optimization of conditional valueat-risk." Journal of Risk 2.3(1999).

[5] Hochreiter, S. , and J. Schmidhuber." Long Short-Term Memory." Neural Computation 9.8(1997):1735-1780.

[6] (n.d.). R-Breaker. https://www.myquant.cn/docs/python_strategyies/425. Accessed in July 29, 2021

[7] Sha, Lou. Get an easy look at LSTM and its Python code implementation. https://zhuanlan.zhihu.com/p/104475016?utm_source=wechat_session&utm_medium=social&utm_oi=1217085389032296448&utm_campaign=shareopn. Accessed in January 14, 2022.

[8](n.d.).Portfoliotheory.https://wiki.mbalib.com/wiki/%E6%8A%95%E8%B5%84%E7%BB%84%E5%90%88%E7%90%86%E8%AE%AAccessed in March 11, 2009.

[9](n.d.). Quantitative Trading (Investment Methods). https://baike.baidu.com/item/%E9%87%8F%E5%8C%96%E4%BA%A4%E6%98%93/5266581?fr=aladdin Accessed in July 15, 2016.

Appendix

LSTM CODE:

```
% lstm time - series prediction , based on vmd
clear
close all
clc
load var modes
t = sum(modes);
% LSTM process
testNumber=365;
prediction_length = 100;
predictValues=zeros(size(modes, 1),prediction_length);
for k = 1 : size(modes, 1)
    data = modes(k, :)';
    disp(' ')
    disp(['IMF - ', num2str(k)])
    lag=4;
    for i=1:length(data)-lag
        deal_data(i,:)=data(i:i+lag)';
    end
    input=deal_data(:,1:end-1);
    output=deal_data(:,end);
    N=length(output);
    trainNum=N-testNumber;
    p_train = input(1:trainNum,:)';
    t_train =output(1:trainNum)';
    p_test =input(trainNum+1:trainNum+testNumber,:)';
    t_test =output(trainNum+1:trainNum+testNumber)';
    [pn_train ,ps]= mapminmax(p_train, 0, 1);
    [tn_train, ts] = mapminmax(t_train, 0, 1);
    pn_test = mapminmax('apply', p_test, ps);
    tn_test = mapminmax('apply', t_test, ts);
    for i = 1:length(t_train)
        P_train{i,1} =    pn_train(:,i);
    end
    for i = 1:length(t_test)
        P_test{i,1} =    pn_test(:,i);
```

```
    end
    numFeatures = size(p_train, 1);
    numHiddenUnits1 = 50;
    numHiddenUnits2 = 80;
    numResponses = 1;
    layers = [ ...
        sequenceInputLayer(numFeatures)
        lstmLayer(numHiddenUnits1,'OutputMode','last','name','hidden1')
        dropoutLayer(0.3,'name','dropout_1')
        lstmLayer(numHiddenUnits2,'OutputMode','last','name','hidden2')
        dropoutLayer(0.3,'name','dropout_2')
        fullyConnectedLayer(numResponses,'name','fullconnect')
        regressionLayer('name','out')];
    options = trainingOptions('adam', ...
        'MaxEpochs',200, ...
        'GradientThreshold',1, ...
        'InitialLearnRate',0.01, ...
        'LearnRateSchedule','piecewise', ...
        'LearnRateDropPeriod',50, ...
        'LearnRateDropFactor',0.3, ...
        'MiniBatchSize',20,...
        'Verbose',0, ...
        'Plots','training-progress');
    %% train LSTM
    net = trainNetwork(P_train,tn_train',layers,options);
    %% test LSTM
    testn_simu = predict(net,P_test);
    simu(k, :) = mapminmax('reverse',testn_simu', ts);
    %% LSTM prediction
    futureInput=t(end-lag+1:end)';
    for i=1:prediction_length
        fn_in=mapminmax('apply',futureInput,ps);% 瀵硅緭鍏ユ暟鎹   繘琛屽綊涓?鍖?
        futureInputn{1,1} = fn_in(:,1);
        predictValue = predict(net,futureInputn);
        predictValues(k,i) = mapminmax('reverse',predictValue',ts);
        futureInput=[futureInput(2:end);predictValues(k,i)];
    end
end
PredictValues = sum(predictValues);
```

```
t_test = t(end-testNumber+1 : end);
test_simu=sum(simu);
error=test_simu-t_test;
figure
plot(t_test,'k-','linewidth',1)
hold on
plot(test_simu,'r-','linewidth',1)
grid on
legend('real','predict')
xlabel('test. No'),ylabel('value')
title('the test result of VMD - LSTM medel ')
figure
plot(1:length(t), t,'k-','linewidth',1)
hold on
plot(length(t) - length(test_simu) + 1:length(t), test_simu,'r-','linewidth',1)
grid on
legend('real','predict')
xlabel('All. No'),ylabel('value')
title('the prediction result of VMD - LSTM medel ')
figure
plot(1:length(t),t,'k-','linewidth',1)
hold on
plot(length(t):length(t)+length(PredictValues),[t(end),PredictValues],'r-','linewidth',1)
grid on
legend('real','predict')
xlabel('No'),ylabel('value')
title('the future trend prediction result of VMD - LSTM medel ')
%calculate error
rr = corrcoef(t_test, test_simu);
R2 = rr(1, 2)^2;
disp(['the value of R-square is: ', num2str(R2)])
```

DP CODE

```
def get_rate(gold_prices, Bit_prices):
    global gold_prices
    global Bit_prices
    global dp
    global rate
    rate = []
```

```
dp = [[0 for _ in range(3)] for _ in range(len(gold_prices))]
rate1 = [[0 for _ in range(len(gold_prices))] for _ in range(3)]
rate = [[0 for _ in range(3)] for _ in range(3)]
for i in range(len(gold_prices)):
    if i == 0:
        dp[0][0] = 1
        dp[0][1] = -0.01*gold_prices[i]
        dp[0][2] = -0.02*Bit_prices[i]
        rate1[0][i]=[1,0,0]
        rate1[1][i]=[1,0,0]
        rate1[2][i]=[1,0,0]
    else:
        dp[i][0] = max(
            dp[i - 1][0],
            dp[i - 1][1] + (dp[i-1][1]/gold_prices[i-1])*(0.99*gold_prices[i]-gold_prices[i-1]),
            dp[i - 1][2] + (dp[i-1][2]/Bit_prices[i-1])*(0.98*Bit_prices[i]-Bit_prices[i-1])
            )
        if dp[i][0] == dp[i-1][0]:
            rate1[0][i] = [1,0,0]
        elif dp[i][0] == dp[i - 1][1] + (dp[i-1][1]/gold_prices[i-1])*(0.99*gold_prices[i]-gold_prices[i-1]):
            rate1[0][i] = [0,1,0]
        else:
            rate1[0][i] = [0,0,1]
        dp[i][1] = max(
            dp[i - 1][0] + (dp[i - 1][0]/gold_prices[i-1]*0.99)*(gold_prices[i]-gold_prices[i-1]),
            dp[i - 1][1] + (dp[i-1][1]/gold_prices[i-1])*(gold_prices[i]-gold_prices[i-1]),
            dp[i - 1][2] + (dp[i - 1][2]/Bit_prices[i-1]*0.98)*(gold_prices[i]-gold_prices[i-1])
            )
        if dp[i][1] == dp[i - 1][0] + (dp[i - 1][0]/gold_prices[i-1]*0.99)*(gold_prices[i]-gold_prices[i-1]):
            rate1[1][i] = [1,0,0]
        elif dp[i][1] == dp[i - 1][1] + (dp[i-1][1]/gold_prices[i-1])*(gold_prices[i]-gold_prices[i-1]):
            rate1[1][i] = [0,1,0]
        else:
            rate1[1][i] = [0,0,1]
        dp[i][2] = max(
            dp[i - 1][0] + (dp[i - 1][0]/Bit_prices[i-1]*0.98)*(Bit_prices[i]-Bit_prices[i-1]
```

ARIMA

```
from statsmodels.tsa.holtwinters import ExponentialSmoothing
from sklearn.metrics import mean_squared_error
```

```
Gold_data = pd.read_csv(r'C:\Users\JUA4SZH\Desktop\LBMA-GOLD.csv')
Gold_data.fillna(method='pad',inplace=True)
Gold = pd.read_excel(r'C:\Users\JUA4SZH\Desktop\lbma_gold_am_usd_2016-06-01_2016-09-10.xlsx')
Gold_train = Gold.iloc[0:72,:]
Gold_test = Gold.iloc[72:144,:]['USD'].values
test_length = len(Gold_train)
SEASONAL_PERIODS = 12
Gold_train = Gold_train['USD'].values
# Holt-Winter's model with exponential trend
Gold_predict = []
for i in range(0,1265):
    Gold_train = Gold.iloc[0+i:72+i]['USD'].values
    hw_1 = ExponentialSmoothing(Gold_train,
                        trend='mul',
                        seasonal='add',
                        seasonal_periods=SEASONAL_PERIODS).fit()
    hw_forecast_1 = hw_1.forecast(1)
    Gold_predict.append(hw_forecast_1)
Gold_p = []
for i in Gold_predict:
    Gold_p.append(i[0])
pd1 = pd.DataFrame(Gold_predict)
Gold_d = Gold_data['Date']
Gold_Predict = pd.concat([Gold_d,pd1],axis=1)
Gold_Predict.to_excel(r'C:\Users\JUA4SZH\Desktop\Gold_predice.xlsx',index = False)
A =Gold_data['USD']
B = Gold_Predict[0]
MSE = mean_squared_error(A, B)
print(MSE)
import matplotlib.pyplot as plt
Gold_actual = pd.read_csv(r'C:\Users\JUA4SZH\Desktop\LBMA-GOLD.csv')
Gold_actual.fillna(method='pad',inplace=True)
Gold_pred = pd.read_excel(r'C:\Users\JUA4SZH\Desktop\Gold_predice.xlsx')
a = Gold_actual['USD']
b = Gold_pred['Date']
c = Gold_pred[0]
Gold = pd.concat([b,a,c],axis=1)
Gold = Gold.iloc[0:366,:]
Gold.columns = ['Date','actual','pred']
```

```
plt.figure(figsize=(20,10))

plt.title('Gold')

plt.plot(Gold['Date'],Gold['actual'])

plt.plot(Gold['Date'],Gold['pred'])
```