Problem Chosen	2022 MCM/ICM Summary Sheet	Team Control Number
C		2215271

Gold or Bitcoin: An Investment Strategy Based on LSTM and VaR

Summary

A wise investment depends on two important factors: a precise judgment of price movements and a keen awareness of potential risks. The purpose of this paper is to build a model to predict the price of gold and bitcoin as well as to hedge the risks of trading, thus giving sound trading decisions.

In TASK 1, we first used the **Long Short-Term Memory Network (LSTM)** to predict the daily closing prices of gold and bitcoin on day 31 based on the last 30 days of data, and plotted the prediction curves of the prices. The results show that our prediction curves fit the real price curves very well, which means our predictions are very accurate. Moreover, we performed a risk assessment using a **Value at Risk (VaR) Model**, combining profit and risk indicators for modeling trading decisions. Then we constructed a **Finite Automata** with four states, which determines each trading decision throughout the trading process. Finally we conclude that the asset value reaches $\$116,296.4$ on September 10, 2021.

In TASK 2, we established a trading strategy evaluation model based on **Sliding Window Mechanism**, setting its window size to 80 days and sliding steps to 30 days. Then we defined three metrics as the evaluation criteria of the model: final asset value (FAV), downside risk (DR), and max drawdown (MD). Compared to the **Cubic Exponential Smoothing Model**, the LSTM-only Model, or the VaR-only Model, our model performs best on a combination of three metrics, which proves that it provides the optimal investment strategy.

In TASK 3, we analyzed the sensitivity of our trading strategy to transaction costs and explored the way in which transaction costs influence trading decisions. By varying the value of the transaction cost portfolio, we observe the changes in three metrics (FAV, DR, and average max drawdown MD_{avg}) and conclude that: when transaction costs decrease, the model's ability to capture profits increases, but this also means that risk is increasing, so the asset value curve fluctuates more; and when transaction costs increase, the model focuses more on risk avoidance, but it is still keen to finds profit margins and makes wise trading decisions.

Finally, we conduct a sensitivity analysis in order to gain some deep understanding of our model, and verify the robustness of the model in many cases. Additionally, we analyze the strengths and weaknesses of our model.

Keywords: Long Short-Term Memory, Value at Risk, Finite Automata, Sliding Window, Exponential Smoothing

Contents

1 Introduction

1.1 Problem Background

Investing is always a topic of interest. However, how to invest is a challenge. To gain more profit, market traders often buy and sell assets, such as gold and bitcoin. So the right investment strategy is crucial for them. Advances in computer technology have given us a powerful ability to obtain data and perform calculations. Therefore we can analyze and extract the essential factors that influence the price movements of gold and bitcoin based on the given data to conclude the best investment strategy.

In this problem, we are given the pricing data files containing daily troy ounce closing prices and individual bitcoin prices from September 2016 to September 2021. Based on the historical price data, we will accomplish the following tasks:

- Consturct a model that gives the best trading decision based only on the day's price data and calculate what the initial $1,000 is worth on 9/10/2021.
- Build a evaluation model to prove that our strategy is the best.
- Explore the sensitivity of the strategy to transaction costs and how transaction costs affect the trading decision.
- Communicate our strategy, model, and results to the trader in a memorandum of at most two pages.

1.2 Our work

The work we have done is mainly shown in the following **Figure**1.

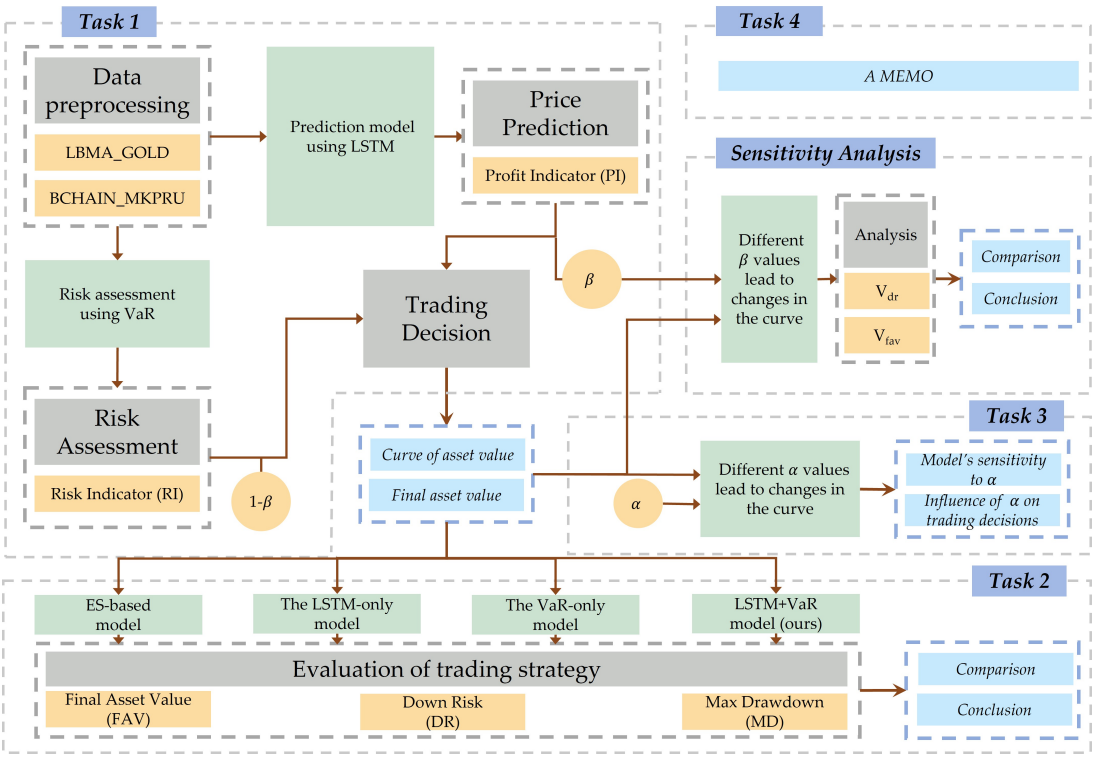


Figure 1: Our work

2 Preparation of the Models

2.1 Basic Assumptions

We made the following assumptions to help us with our modeling. These assumptions are the promise of our subsequent analysis.

- We assume that future prices are not influenced by other external factors, that is, gold and bitcoin prices are only influenced by historical prices.
- We assume that gold is traded in the smallest unit of one ounce (because the gold market limits the minimum trading unit to one ounce) and bitcoin is traded in the smallest unit of 0.001 bitcoins.
- We assume that the prices of gold and bitcoin satisfy some distribution, which will be used as a premise for our risk assessment.

2.2 Notations

The primary notations used in this paper are listed in Table 1.

Table 1: Notations

Symbol	Definition
p	The predicted price
r	The real price
d_k	The k^{th} trading day
$PI(k)$	Profit indicator on day d_k
$RI(k)$	Risk indicator on day d_k
ρ	The threshold for trading
F	Decision indicator
M_k	Total value of assets on day d_k
β	Dynamic indicator weight
FAV	Final asset value
DR	Downside risk
MD	Max drawdown
V_{fav}	Volatility of FAV
V_{dr}	Valatility of DR

2.3 Data Preprocessing

Before constructing the model, we gave an overview of the data and found that the price of gold had null values on some days. So we removed them to prevent any effect on the model.

Further, we found inconsistent date formats in both *LBMA_GOLD* and *BCHAIN_MKPRU* datasets. To ensure that we can track the correct dates in the investment process, we modified all date data to the same format.

3 TASK 1: Quantitative Trading Decision Model Based on LSTM and VaR

A sound trading decision depends on many factors. A critical factor that influences the decision is the expectation of profit. Therefore, we established a price prediction model in *section 3.1*, which allows us to make decisions based on the predicted price and the current price. Considering that any investment is risky, we constructed another risk evaluation model in *section 3.2*, the results of which will serve as a further basis for the trading decision.

3.1 Prediction Model Using LSTM

3.1.1 Long Short-Term Memory Network

The price movements of gold and bitcoin are both time-series data. To handle these time-series data, we use the Long Short-Term Memory[1] (LSTM) Network , which has self-feedback neurons and solves the long-range dependence problem of traditional recurrent neural networks (RNN). The basic structure of LSTM is shown in the following **Figure2**.

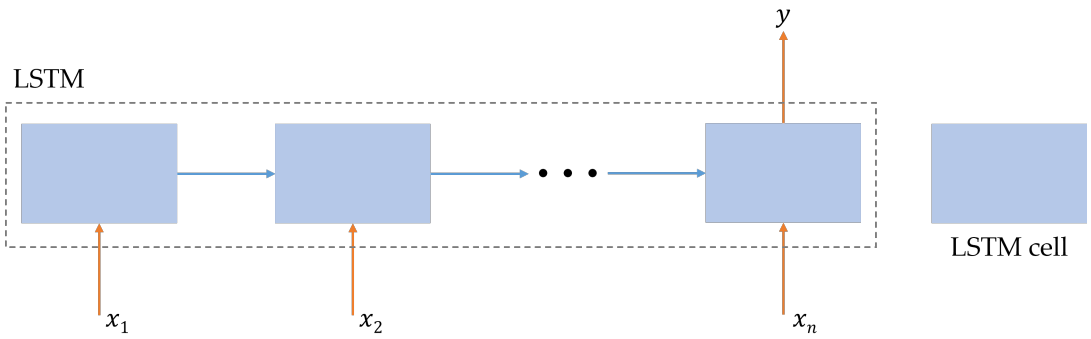


Figure 2: Basic structure of LSTM

Formally, each cell in LSTM is computed as follows:

$$X = \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix}, \tag{1}$$

$$f_t = \sigma(W_f \cdot X + b_f), \tag{2}$$

$$i_t = \sigma(W_i \cdot X + b_i), \tag{3}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_c \cdot X + b_c), \tag{4}$$

$$o_t = \sigma(W_o \cdot X + b_o), \tag{5}$$

$$h_t = o_t \odot tanh(c_t). \tag{6}$$

where W_i, W_f, $W_o \in \mathbb{R}^{d \times 2d}$ are the weighted matrices and b_i, b_f, $b_o \in \mathbb{R}^d$ are biases of LSTM to be learned during training, parameterizing the transformations of the input, forget and output gates respectively. σ is the sigmoid function and $\odot$ stands for element-wise multiplication. x_t includes the inputs of LSTM cell unit in **Figure2**. The vector of hidden layer is h_t.

Considering that people usually measure the price movement of an asset in months, we will predict the price of a given day based on the previous 30 trading days. Thus, we set the loop unit of LSTM to be 30 and take the price data of the first 30 days as the input sequence to get the prediction price. Additionally, we use gradient descent as the optimization method, and the loss function is defined as follows:

$$L(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2 \tag{7}$$

In equation7, y is the true price and $\hat{y}$ is the prediction price.

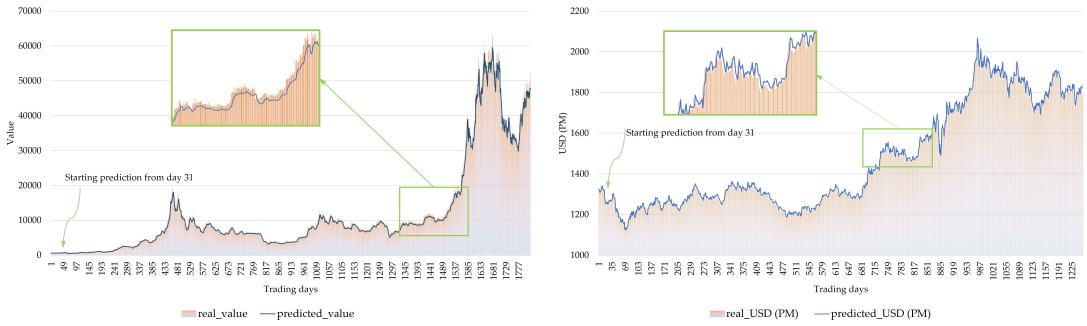
3.1.2 Analysis of Predicted Results

Since we base our trading decisions on the price of a period (set to previous 30 days), we start predicting from day 31, using the LSTM model. The calculation of the predicted price p_{k+1} is shown by the Equation8:

$$p_{k+1} = LSTM(X, \theta) \tag{8}$$

where $X = [r_{k-29}, r_{k-28}, ..., r_k]^T$ is the vector containing the real price of the previous thirty days and θ is the hyperparameter of the LSTM model.

In this subsection, our goal is to analyze whether the results of the prediction model are precise for a given 30 days of data. Therefore, we used an autoregressive approach for prediction and plotted the forecast curve of the price and compared it with the real curve as shown in the **Figure**3.



(a) bitcoin (b) gold

Figure 3: Comparison of predicted curve and real curve

From **Figure**3, we can see the prediction curve overlaps with the true curve for the first 30 trading days. This is because we make our trading decisions based on the price movements of the previous 30 trading days. Since investing is risky, a wise investment strategy needs to be informed. The historical data available for the first 30 trading days is too little to make it suitable for investment. So it makes sense that we do not trade in the first 30 trading days. Also, from the zoomed-in details in the graph, the difference between our prediction curve and the real curve is still small. Overall, our curves fit the true curve well, which indicates that the predictions are accurate.

3.2 Risk Assessment Using VaR

A successful investment can not only look at the expected profit. In this subsection, we will make a risk assessment and use the risk factor as one of the bases for trading

decisions later in section 3.3.

In order to assess the possible risks associated with the investment, we use the idea of Value at Risk[2] (VaR), considering the maximum probable loss of a specific portfolio at a particular confidence interval in the future. We assume that the prices of gold and bitcoin satisfy some distribution and c is the confidence interval of that distribution. The VaR value corresponds to the value of the left-hand side confidence interval at $1-c$ of the profit-and-loss distribution. Our measurement of risk can be divided into three steps as follows:

***Step* 1:** We let the profit distribution at moment k be $D(k)$, and according to the law of large numbers we take the price data for the 30 days before moment k and subtract them to get the daily profit, then the profit distribution can be represented by these data.

***Step* 2:** Then we sort the profits, let the length of the profits be len and the confidence interval be c. Then the $\lfloor len * c \rfloor^{th}$ element of the ordered profits array gives the data as the VaR value at the confidence level of c. It represents the probability that this asset will lose VaR value on the next day as c.

***Step* 3:** Then we divide the VaR value by the current true value to get the loss rate LR for the next day.

Thus, we obtain the loss rate LR, which will be further used in section 3.3.

3.3 Trading Decisions

3.3.1 Indicators for Decision Making

Based on the prediction model and risk evaluation model, we can obtain the predicted price and calculate the risk factor. Correspondingly, we define the profit indicator PI and the risk indicator RI, which jointly form the basis of the trading decision.

- **Profit Indicator**

To maximize the profit, we use LSTM and output the results to continue as input, thus predicting the price trend in the future period under current conditions and obtaining the trading day with the highest expected profit. Then, we define the profit indicator $PI(k)$:

$$PI(k) = \frac{\max\limits_{k \le i \le k+15}\{p_i\} - r_k}{r_k} \tag{9}$$

where r_k is the real price of the designated day d_k and $\max\limits_{k \le i \le k+15}\{p_i\}$ indicates the highest predicted price for the next fifteen days. The buy and sell date of the transaction can be expressed as $[d_k, d_m]$, and d_m can be calculated by equation10:

$$d_m = \arg\max_{k \le i \le k+15}\{p_i\} \tag{10}$$

Note that we perform the calculation of PI and d_m values on every trading day, rather than taking a circular strategy. This is because if a circular strategy is adopted, i.e., no trades wound be made in the time period from d_k to d_m, then the true value may

differ significantly from the value predicted at dk during this period. This situation will cause us to miss out on benefits that could have been gained or take losses that could have been reduced. However, our method avoids it. Because on eachtrading day, we reconsider the true value of the day as well as the re-predicted value for the next 15 trading days, which leads us to a wiser trading dicision.

- **Risk Indicator**

To measure the total risk over the $[d_k, d_m]$ time period, we use the loss rate LR calculated in section 3.2 to define the risk indicator $RI(k)$:

$$RI(k) = LR \cdot (d_m - d_k) \tag{11}$$

Based on the profit indicator PI and the risk indicator RI, in other words, after taking into account the desired return as well as the potential risk, we are ready to make a sensible trading decision.

3.3.2 The Process of Making Each Trading Decision

Combining the profit indicator PI and risk indicator RI, we define the F-score F:

$$F = \beta \cdot PI + (1 - \beta) \cdot RI - \rho,\ 0 < \beta < 1 \tag{12}$$

where β, ρ are hyperparameters. The value of ρ depends on the transaction costs α, and we find their values by equation13.

$$\rho = \frac{1}{(1-\alpha)^2} - 1 \tag{13}$$

Then the vector form of equation12 can be expressed as:

$$\begin{bmatrix} F_g(k) \\ F_b(k) \end{bmatrix} = \begin{bmatrix} PI_g(k) & RI_g(k) \\ PI_b(k) & RI_b(k) \end{bmatrix} \cdot \begin{bmatrix} \beta \\ 1-\beta \end{bmatrix} - \begin{bmatrix} \rho_g \\ \rho_b \end{bmatrix},\ 0 < \beta < 1 \tag{14}$$

where $F_g(k)$ and $F_b(k)$ represent the F-score of gold and bitcoin on trading day d_k respectively. $[\rho_g, \rho_b]^T$ can be calculated using equation13, based on the transaction costs of gold and bitcoin $[\alpha_g, \alpha_b]^T$, respectively.

Then, we define four states based on the values of F_g and F_b and construct a finite automata[3] (FA) as shown in the **Figure**4.

Each state transfer in the FA is the transaction action we will perform. Here, each of our trading decisions is fully bought or sold as we combine profit indicators and risk indicators and only make trading moves when there is high profit and low risk. The detailed description of each state and transfer is as follows:

state 1: If the day is a weekend, then we jump directly to state 2. When $F_g < 0$, we sell gold and jump to state 3; when $F_g > 0$ and $F_b - F_g > 0$, we sell gold and buy bitcoin, then jump to state 2.

state 2: If the day is a weekend, we sell bitcoins when $F_b < 0$ and jump to state 3. When $F_b < 0$, we sell bitcoins and jump to state 3; when $F_b > 0$, we sell bitcoins and buy gold when $F_g - F_b > 0$ and we can afford gold, and then jump to state 1; if the condition "when fg-fb>0 and we can afford gold" is not met, we sell bitcoins and buy

gold. Otherwise, we maintain the state, because at this point $F_b > 0$, which means we can still make a profit.

state 3 (initial state): If $\max(F_g, F_b, 0) \leq 0$, then no trade will be made that day; if $\max(F_g, F_b, 0) > 0$, we will buy the asset corresponding to the larger of F_g and F_b, and jump to state 1 if it is gold, and to state 2 if it is bitcoin. If the obtained trading day dm exceeds the last day, then jump to state 4. (Note that there are special cases when the minimum trading unit cannot be met, e.g., if F_g is higher than F_b but we cannot afford one ounce of gold, we can only buy bitcoins, and if it's the weekend, we can only buy bitcoins, too.)

state 4 (finite state): When this state is reached, the state transfer stops and the final result is obtained.

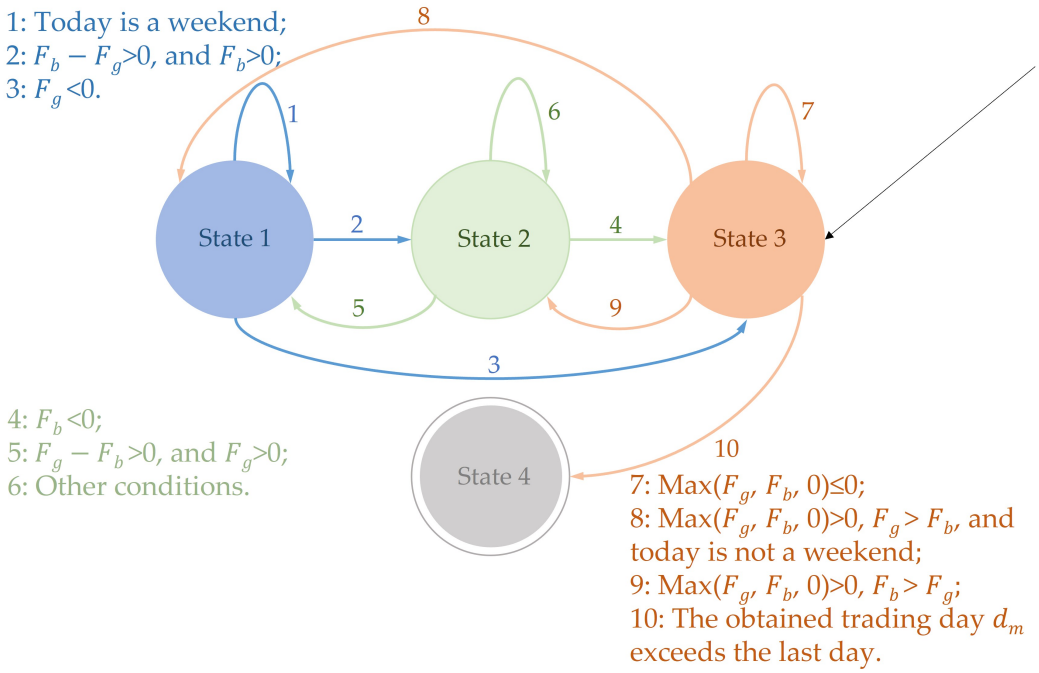


Figure 4: Finite Automaton with four states

Considering that our trading strategy is full trading, the more assets we own, the higher the risk is. Therefore, the weight of the risk indicator $1 - \beta$ should rise dynamically with the increase of total assets, and the corresponding weight of the profit indicator β will fall dynamically, as shown in equation15:

$$\begin{cases} M_k = (C_k + G_k \cdot r_{gk} + B_k \cdot r_{bk}) \\ \beta = \beta_0 - \frac{3}{10^6} M_k \end{cases} \tag{15}$$

where $[C_k, G_k, B_k]$ is the asset holding portfolio on day d_k, and M is the total value of it. This equation indicates that every time the assets we own rise by \$10,000, the value of β decreases by 0.03. Here, we set β_0 to 0.6, and in the sensitivity analysis in section 6 we will explore the effect of the value of β_0 on the model.

The pseudo-code for our decision process is shown as follows:

algorithm 1 Framework of investment strategy.

Input: The dataset of gold, $dataGold$; The dataset of bitcoin, $dataBit$; Hyperparameters; f(), a function that gets the value of f, the input is the dataset and a hyperparameter, and the output is f; getDayData(), a function that gets the data of the previous n days from the dataset, the input is n and the dataset $data$, and the output is the data of the previous n days

Output: Funding triple, ans;

1: Initialize related parameters, let ans=[1000,0,0], day=10/9/2016, $state$=3;
2: If day==9/10/2021, jump to 9;
3: $data_g$=getDayData(n,$dataGold$), $data_b$=getDayData(n,$dataBit$)
4: fg=f($data_g$, β), fb=f($data_b$, β);
5: If $state$==1. If it is not a weekend, there are two cases: (1) When $fb - fg > 0$ and $fb > 0$, sell gold to buy bitcoin, update the funding triple ans and set $state$=2. (2) When $fg < 0$, sell the gold, update the funding triple ans and set $state$=3. Jump to 8;
6: If $state$==2. If it is a weekend, sell bitcoins when $fb < 0$, update the funding triple ans and set $state$=3; if it is not a weekend, there are two situations: (1) When $fg - fb > 0$ and $fg > 0$, sell bitcoins to buy gold, update funding triples ans and set $state$=1. (2) When $fb < 0$, sell bitcoins, update funding triples ans and set $state$=3. jump to 8;
7: If $state$==3. If it is a weekend, buy bitcoin when $fb > 0$, update the funding triple ans and set $state$=2; if it is not a weekend, there are two situations: (1) When fb>fg and $fb > 0$, buy bitcoin, update the funding triple ans and set $state$=2. (2) When $fg > fb$ and $fg > 0$, sell gold, update funding triples ans and set $state$=1. Jump to 8;
8: day=day+1, jump to 2;
9: return ans;

3.4 Results of the Model

We used the above strategy and algorithm and calculated from September 2016 to September 2021 to finally obtain the change curve of asset value. We concluded that the value of our assets reached $116296.4 on September 10, 2021.

In addition, we compare the results brought by the strategy of using static β and dynamic β, as shown in **Figure**5.

From the figure, we can see our strategy using dynamic β ended up with an asset value of $116296.4 compared to $73,899.42 for the strategy using static β. Moreover, we can conclude that adopting the dynamic β strategy successfully stops the continuous decline in value in two crucial periods, which is the effect of the weighting of the risk indicator with the change in asset value.

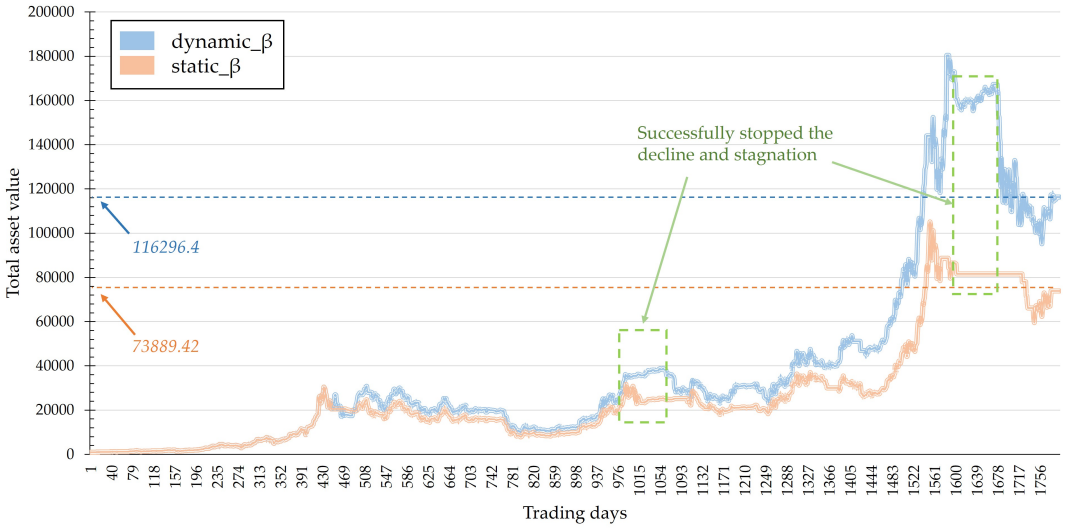


Figure 5: Total asset value

4 TASK 2: Sliding-window Based Evaluation Model For Trading Strategy

4.1 Construction of the Evaluation Model

To prove that our strategy is optimal, we build an evaluation model based on a sliding window[4] and evaluate several different strategies based on three essential indicators which will be defined in section 4.1.2.

4.1.1 Sliding Window Mechanism

Although different models will cause variance in trading strategies, the asset value trends they form are all time-series data. Therefore, we adopted the sliding window mechanism.

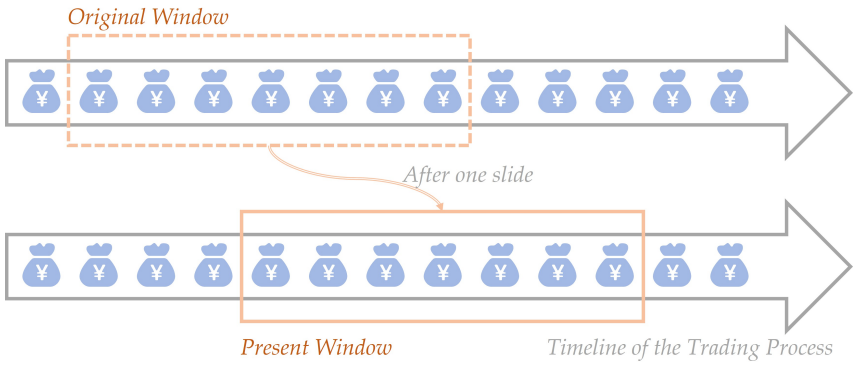


Figure 6: Schematic of sliding window mechanism

Sliding window mechanism is a method to frame a time series based on a specified length of units to calculate the statistical metrics within the frame. It is equivalent to a slider of a specified length sliding on top of a scale, with each unit sliding providing feedback on the data within the slider. A schematic of the sliding window mechanism is as shown in **Figure**6.

We will set an identical sliding window with a window size of 80 days and a sliding step of 30 days for the asset value curves obtained for the different strategies.

4.1.2 Essential Metrics

After establishing the sliding window, we need to choose the appropriate evaluation metrics to quantify the trading strategy in order to judge its good or bad performance. An essential metric is the final asset value, which is the most intuitive evaluation criterion of an investment process. In addition, to evaluate the entire trasaction process, we have borrowed two common metrics from the financial sector and applied them to our evaluation model.

- Final Asset Value

Using different decision models will result in different trading strategies, which means our final asset values will likely be different. Therefore, the final asset value FAV is an important measurement basis.

- Downside Risk

Generally, the downside risk[6] is an estimate of the likelihood that a security will decline in value if market conditions change, or the amount of loss that could be suffered as a result of a decline. It explains the "worst case" scenario for an investment, or how much an investor could lose.

Here, we define the indicator DR to represent the downside risk at different stages of the entire trading process. To calculate DR, we should find the expected return of gold and bitcoin first: doing a differential on the price per day to get the daily return. Then we average the daily returns to obtain the daily expected return EG_{daily}, EB_{daily} and use the following equation16 to find the expected $E_{i,j}$ returns from trading day d_i to trading day d_j:

$$\begin{cases} EG_{i,j} = M_i + (j-i) \cdot EG_{daily} \\ EB_{i,j} = M_i + (j-i) \cdot EB_{daily} \\ E_{i,j} = \lambda \cdot EG_{i,j} + (1-\lambda) \cdot EB_{i,j} \end{cases} \tag{16}$$

where M_i is the total value of the assets we hold on the transaction date d_i (which can be obtained from equation15) and λ is a hyperparameter (we set it to 0.8). Then we derive the downside risk DR from the asset values M_k obtained from the decision model:

$$DR_{i,j} = \sqrt{\frac{\Sigma_{k=i+1}^{j} \min\{0, E_{i,j} - M_k\}^2}{n}} \tag{17}$$

The euqation17 means that when the expected asset value M_k is greater than the expected return E_{ij}, it will be taken as 0. Otherwise the square of its offset is calculated, and finally the average of all the sample values is squared. Then we set d_i to be the first trading day and d_j to be the last trading day, i.e., we calculate the downside risk for the entire trading process.

- Max Drawdown

The max drawdown[7] is the maximum value of the retracement of the return when the value curve goes to its lowest point at any historical point backward in the selected

period. It describes the worst-case scenario that can occur after trading.

To calculate the max drawdown rate MD in the time period $[d_i, d_j]$, we utilize the sliding window mechanism. We choose $d_{\lfloor \frac{i+j}{2} \rfloor}$ as the dividing line, find the peak M_{max} in the time period $[d_i, d_{\lfloor \frac{i+j}{2} \rfloor}]$ and the trough M_{min} in the time period $[d_{\lfloor \frac{i+j}{2} \rfloor}, dj]$. Then we obtain the max drawdown MD:

$$MD = \frac{M_{min}}{M_{max}} \tag{18}$$

Based on these three metrics, we can then evaluate the different models in section 4.2.

4.2 Evaluation of Different Trading Strategies

In this subsection, we will evaluate four decision models. They are the exponential smoothing[5]-based model, the LSTM-only model, the VaR-only model, and the LSTM+VaR model which we adopted in TASK1.

Here we briefly introduce the cubic exponential smoothing, which serves to predict the next trend of the curve. It assumes that the curve has three characteristics: linearity, smoothness and seasonal volatility, and therefore three variables are used to represent these three characteristics. This we use the cumulative multiplicative formula of the cubic exponential smoothing, which is given by the following equations.

$$S_t = \sigma \frac{y_t}{I_{t-L}} + (1 - \sigma) \cdot (S_{t-1} + b_{t-1}) \tag{19}$$

$$b_t = \gamma \cdot (S_t - S_{t-1}) + (1 - \gamma) \cdot b_{t-1} \tag{20}$$

$$I_t = \epsilon \frac{y_t}{S_t} + (1 - \epsilon) \cdot I_{t-L} \tag{21}$$

where S_t represents the smooth feature at moment t, b_t represents the linear feature at moment t, I_t represents the seasonal feature at moment t, L is the seasonal period, and y_t is the actual value at moment t. σ, ϵ, and γ are three hyperparameters that take values in the range $[0, 1]$. It can be seen that the features of each new moment t_k will contain the features of the previous moment as well as y_k, so by recursion we can get that the features of the new moment are actually the weighted sum of the features of the previous moment and y_k. After that, at day d, we can use the following formula for the predicted value of day m:

$$F_{d+m} = (S_d + mb_d) \cdot I_{d-L+m} \tag{22}$$

In this application, we need to initialize the values of S, b, I before L. Due to the transient nature of the cubic exponential smoothing memory, we set:

$$\begin{cases} S_i = y_i \\ b_i = y_{i+1} - y_i \quad , \ i <= L \\ I_i = 1 \end{cases} \tag{23}$$

Then we set $L = 30$, $\sigma = 0.2$, $\epsilon = 0.9$ and $\gamma = 0.05$ and can use it to take the predicted value after 15 days.

For all four models, we use the same evaluation method, i.e., we analyze the three indicators FAV, DR and MD.

The comparison of final asset value FAV and downside risk DR for different models is shown in the following table2:

Table 2: Comparison of FAV and DR

Model	FAV	DR
ES-based	8723.94	21960.32
LSTM-only	77659.51	4656.52
VaR-only	26373.68	14751.78
LSTM+VaR (ours)	116296.42	5598.99

From the table, we can conclude that our model is the best one under these two metrics. Although its DR value is not the lowest, it is close to the lowest value, and its FAV value is much higher than several other models. In addition, we note that the model based on the three-time exponential smoothing mechanism performs the worst on both FAV and DR. We believe this is since exponential smoothing is suitable for time-series data with seasonal fluctuations, while the price trends of gold and bitcoin do not have this feature.

For the max drawdown MD, we plotted box plots based on the values in the window time period to explore the overall volatility of the maximum retracement values for different decision models, as shown in the **Figure**7.

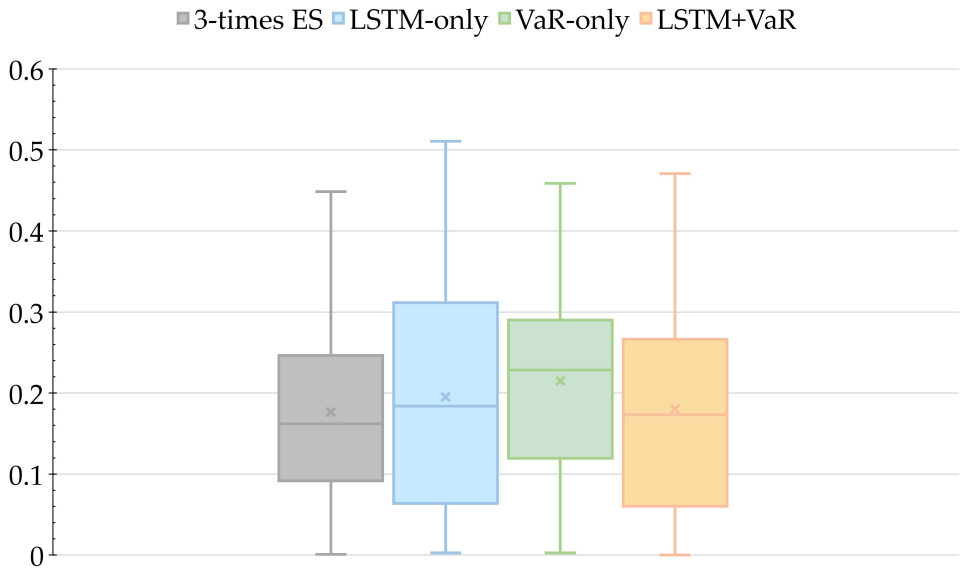


Figure 7: Dispersion of MD values

Observing the figure, we find that our model outperforms the LSTM-only model and the VaR-only model, which has lower first-quartile, median and third-quartile than the other two. In addition, based on the fact that the VaR-only model has a more concentrated and optimal value distribution than the LSTM-only one, we conjecture that the inclusion of the VaR makes the value distribution of our model better than that of the LSTM-only one.

Further, we find that the model based on cubic exponential smoothing possesses lower median, third-quartile, and maximum values than our model, but the difference is very slight. And ours has a lower first-quartile. We believe that this is because the curves using three times exponential smoothing are much slower and smoother in terms of movement, i.e. each time the increase or decrease is small, which coincides with the unsatisfactory data of exponential smoothing on FAV.

At this point, we can conclude that our model is optimal in that it has the best combined performance on the three metrics of FAV, DR, and MD.

5 TASK 3: Exploration of the Transaction Costs

In this section, we will analyze the sensitivity of our model to transaction costs by varying the value of transaction costs. And we will explore how transaction costs affect the decision based on three valid indicators: the down risk DR, the final asset value FAV and the max drawdown of the entire trading proccess MD_{avg}, where FAV and DR have been defined in section 4.1.2.

To facilitate the analysis, we averaged the max drawdown values within each sliding window to obtain MD_{avg}:

$$MD_{avg} = \frac{\Sigma_{i=1}^{n} MD_i}{N} \tag{24}$$

where N is the number of phases divided by the sliding window mechanism during the entire trading process.

5.1 Sensitivity of the Model to Transaction Costs

In our model, ρ in equation13 is a factor directly related to transaction cost α, so changing the value of transaction cost is equivalent to changing the value of ρ. We let the vector $[\rho_g, \rho_b]^T$ used in TASK1 be ρ_0. According to equation13, we calculate the ρ values corresponding to five different combinations of transaction costs $[\alpha_g, \alpha_b]$. The following **Table**3 shows their correspondence.

Table 3: Correspondence of ρ and α

ρ	$[\alpha_g, \alpha_b]$
ρ_0 - 0.02	$[0\%, 1\%]$
ρ_0 - 0.01	$[0.5\%, 1.5\%]$
ρ_0	$[1\%, 2\%]$
ρ_0 + 0.1	$[5.5\%, 6.4\%]$
ρ_0 + 0.2	$[9.5\%, 10.4\%]$

Then, we set the above ρ values in the model of TASK1 in turn, and then performed the output.

First, We compared the impact of different transaction costs on DR and plotted a bar chart, as shown below:

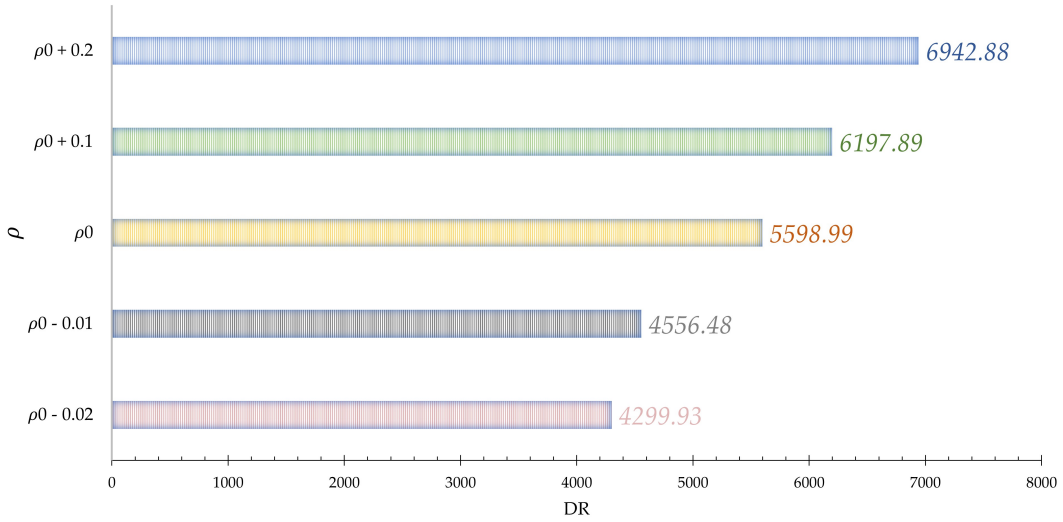


Figure 8: DR values obtained from different ρ values

From **Figure**8, we can conclude that DR increases as the value of ρ increases, and from the equation17 for calculating DR we know that a larger DR indicates that it is closer to the expected return. When ρ decreases, the transaction costs α will be smaller, which means the value of F is easier to reach the threshold, thus the frequency of transactions will increase, and then it will be more risky in the same period; conversely, when ρ becomes larger, F will be difficult to reach the threshold, and the number of transactions will be reduced accordingly, which leads to a more conservative trading strategy.

Then we plotted **Figure**9 based on the value of MD_{avg} and the value of FAV to compare the degree of volatility of the entire trading process and the final asset value at different transaction costs.

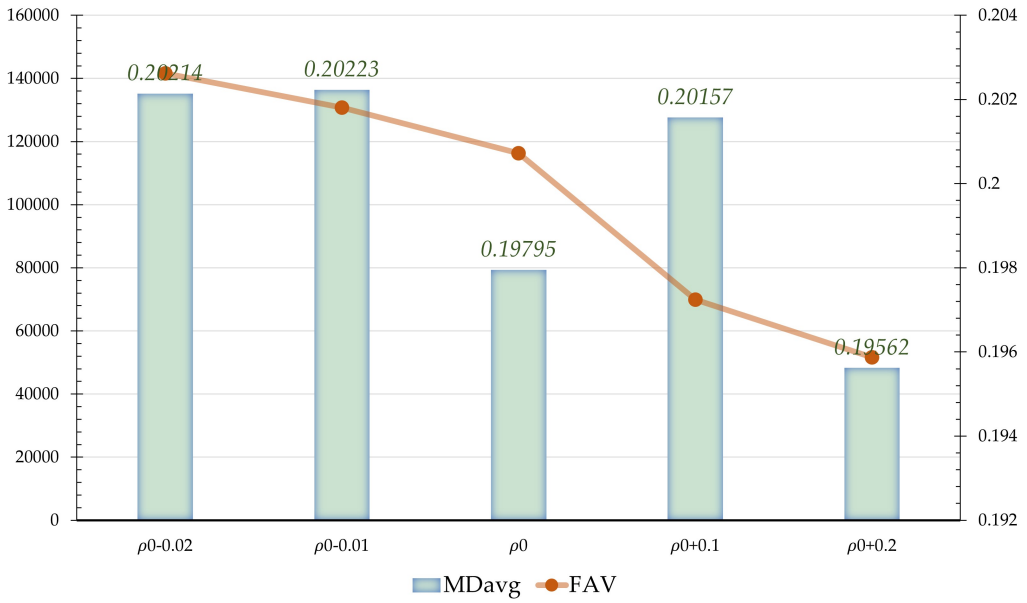


Figure 9: MD_{avg} values obtained from different ρ values

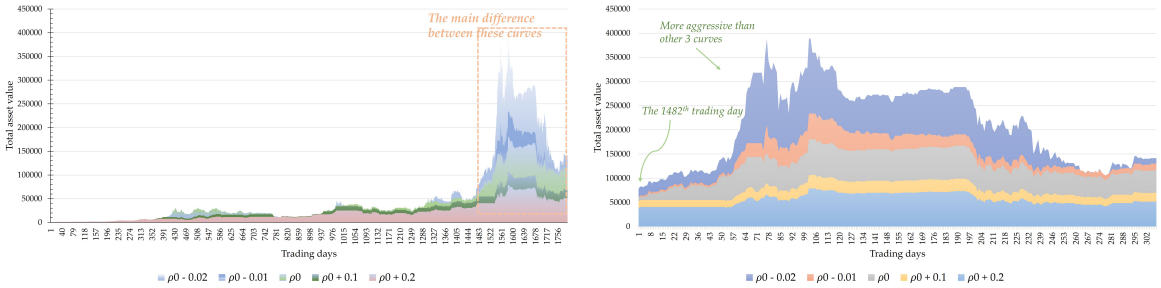
Observing the figure, we find that the value of FAV gradually decreases as the transaction cost increases. This result is consistent with common sense because higher transaction costs will lead to lower returns while ignoring other conditions. (More

analysis of the asset value curve is in section 5.2.) In addition, the value of MD_{avg} is higher when ρ is relatively low. We believe that this may be because transaction costs lead to changes in the decision threshold, and lower transaction costs will lead to more frequent transactions, and on the other hand, the risk increases, so the value of MD_{avg} will be larger. However, as ρ becomes larger, there is no intuitive trend in the value of $MDavg$, and we speculate that this is probably because the increase in the decision threshold causes the model to miss some opportunities that could be profitable.

Finally, we can conclude that our model has a degree of sensitivity to transaction costs and performs better when transaction costs are low. But although higher transaction costs can cause our model to underperform, our strategy is always guaranteed to be profitable.

5.2 Variations in Trading Strategies Due to Different Transaction Costs

To explore exactly how changes in transaction costs affect trading decisions, we compare the asset movement curves under the five transaction costs mentioned mentioned above, as shown in the figure below.



(a) Asset movement curves (b) Main difference between these curves

Figure 10: Impact of transaction costs on trading decisions

From **Figure**10(a), we can see that the five curves move similarly for the first large period, and the larger the transaction costs the smaller the profit. We believe that the small difference is because the larger the transaction cost is given the same decision, the smaller the profit per trade.

However, in the last few days (shown in **Figure**10(b)), the $\rho_0 - 0.02$ curve makes a larger difference in trading decisions than the other four, and it is significantly more aggressive in the period (framed in the figure). At that time, bitcoin's price rises rapidly, which is why the $\rho_0 - 0.02$ curve follows far ahead of the other four. We believe that the decrease in transaction costs caused the F value to be easier to reach the threshold, so more trades will be made. It is why the $\rho_0 - 0.02$ curve can gain more profit. In addition, we find that all curves declines in the last small period. We believe it is due to the dramatic drop in the price of bitcoin, so the more bitcoin is held the larger the loss.

In conclusion, the overall trend of these five curves is similar, which indicates that the decisions made by our model are less affected by transaction costs, which also implies that our model is sharp enough to make informed trading decisions even when transaction costs increase, and thus, it can be said that our model is highly robust.

6 Sensitivity Analysis

To explore the effect of the hyperparameter β on the results of our LSTM+VaR model, we choose different values of β_0 on the model of task1 to detect the impact of beta on the model. We define the volatility of downside risk volatility V_{dr} and the volatility of the final asset value V_{fav} as the sensitivity measure of the model.

For V_{dr}, we first choose a set of values of β_0 as $\{\beta_1, \beta_2, ..., \beta_n\}$. Thus, we can obtain a downside risk set $S_{dr} = \{DR(1), DR(2), ..., DR(n)\}$, where $DR(i)$ corresponds to the obtained downside risk value using β_i in equation15. Then for β_i, its downside risk volatility $V_{dr}(i)$ is:

$$V_{dr}(i) = \frac{DR(i)}{\min(S_{dr})} - 1 \tag{25}$$

where $\min(S_{dr})$ represents the minimum value in set S_{dr}.

Analogously, for V_{fav}, we can get a set $S_{fav} = \{FAV(1), FAV(2), ..., FAV(n)\}$. And $FAV(i)$ is the final assset value obtained by using β_i in equation15. Then the final asset value volatility $V_{fav}(i)$ for β_i can be calculated as:

$$V_{fav}(i) = 1 - \frac{FAV(i)}{\max(S_{fav})} \tag{26}$$

Then, we set six different β_0 values and plotted the curves of V_{dr} and V_{fav} in these six cases as shown in **Figure**11.

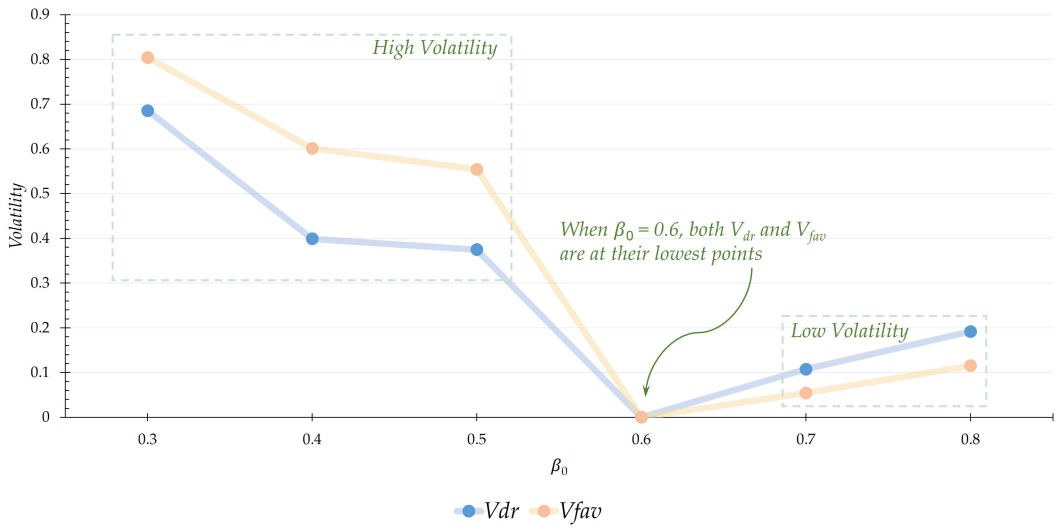


Figure 11: Sensitivity of the Model

We find that they have a substantial similarity: they both fall sharply followed by a slight rise, obtaining a minimum at $\beta_0 = 0.6$. Our explanation of it is as follows: according to equation12, when β is relatively low, the VaR model dominates, leading to larger volatility due to the instability of the VaR model; when $beta_0$ is high, the LSTM dominates, leading to a small increase in volatility because it is more aggressive; and when $\beta_0 = 0.6$, these two models reach a certain balance. Therefore, for the selection of the β_0 value, we suggest a range of values around $[0.55, 0.75]$ which gives our model a more stable performance.

7 Strengths and Weaknesses

7.1 Strengths

- We combine the advantages of LSTM and VaR to predict price movements and hedge risks. And we use reasonable metrics to make trading decisions. The dynamic weights also make our model more stable.
- In trading decision evaluation model, we utilize the sliding window mechanism to analyze the different stages of the entire trading process and consider three essential indicators to draw conclusions.
- We did sufficient visual analysis to facilitate the whole process and understanding of the model results.
- Our model is able to make sharp trading decisions and profit when trading costs are high, and is somewhat risk-averse when trading costs are low.

7.2 Weaknesses

- The asset value curves derived from our model still has a downward component, which indicates that some of the decisions made by our model result in a loss of property.
- Our model is sensitive to the value of β. In particular, a too small β may lead to a trading strategy overly concerned with risk.

References

[1] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.

[2] Jorion, Philippe. "Value at risk." (2000).

[3] Perrin, Dominique. "Finite automata." Formal Models and Semantics. Elsevier, 1990. 1-57.

[4] Tan H Q, Niu Q Q. *A Time Series Symbolization Method Based on Sliding Window and Local Features*[J]. Computer Application Research, 2013, 30(003):796-798.

[5] Gardner Jr, Everette S. "Exponential smoothing: The state of the artPart II." International journal of forecasting 22.4 (2006): 637-666.

[6] Ang, Andrew, Joseph Chen, and Yuhang Xing. "Downside risk." The review of financial studies 19.4 (2006): 1191-1239.

[7] Magdon-Ismail, Malik, and Amir F. Atiya. "Maximum drawdown." Risk Magazine 17.10 (2004): 99-102.

MEMO

To: Market Trader
From: Team 2215271
Date: February 21th, 2022
Subject: Analysis of Model and Results, Suggestions for Trading Decisions

Dear market trader:

At your request, we analyzed the historical data you provided on gold and bitcoin and consequently built models, investment strategies, and model rating metrics. We used a time-series forecasting model to predict potential price increases or decreases and an investment risk model to assist in obtaining risk profiles. We also used some indicators to evaluate the model. Based on the above we have obtained some meaningful results that will help you in the construction and evaluation of your future investment strategies.

First, our investment strategy is an all-buy, all-sell gold and bitcoin policy over a time period, meaning that we only hold gold or bitcoin or neither over a time period, because our model is robust enough to only make trading decisions when profits are high and risk is low. Specifically, our model gives us a profit indicator score and a risk indicator score for gold and bitcoin each day, and these two scores, along with their respective transaction costs, are passed through a function to get their respective scores F. We buy when the score F reaches a threshold, sell when it does not, and compare the two scores F if they both reach the threshold, and see which of the two scores is higher If they both reach the threshold, we compare their scores and buy whichever is higher.

To obtain the score F, we use a model based on a combination of LSTM (Long Short-Term Memory Network) and VaR (Value at Risk) models, i.e., we use LSTM for price prediction and VaR model to obtain the risk evaluation. With these two models, we can get two metrics and then weight the two metrics together to get our score F. With the above model and strategy, we invested from 2016/9/11 and ended 2021/9/10 with a final status of no gold or bitcoin holdings and a current principal of \$116,296.4, which is significantly higher than the original principal of \$1,000.

To evaluate our model, we compared the model to a cubic exponential smoothing model, an LSTM-only model, and a VaR-only model. We selected three metrics to evaluate the risk tolerance and profitability of the model, namely Final Asset Value (FAV), Down Risk (DR), and Max Drawdown (MD), and you can choose which metric to use according to your needs. Of course, you can also use all three metrics at the same time. By comparison, we found that LSTM+VaR is significantly better than the other three models, it not only has the highest profitability but also good risk resistance, which proves that our model and strategy are excellent, it combines the advantages of LSTM and VaR respectively.

Then, we consider the impact of changes in transaction costs on the model. In this session, we still consider the above three indicators and the selected transaction cost combinations are $\{[0\%, 1\%], [0.5\%, 1.5\%], [1\%, 2\%], [5.5\%, 6.4\%], [9.5\%, 10.4\%]\}$. By analyzing the output of the model, we obtained the following conclusions: When the transaction cost is low, the score F is more likely to reach the threshold, which implies an increase in the frequency of trading, since the real price is not under our control, so this also leads to higher risk; and since the model tends to trade more frequently, we will get more small profits. Additionally, the low transaction cost drives us to obtain

more profits. On the other hand, when the transaction cost is higher, the conditions for the score F to reach the threshold are more demanding, but our model is still shrape enough to make informed trading decisions and the final profit is also satisfactory.

Further, we conduct a sensitivity analysis of our model. We find that when $beta_0$ is high, the LSTM dominates the decision process, leading to a small increase in volatility because it is aggressive; when β_0 is relatively low, the VaR model dominates, causing larger volatility due to the instability of the VaR model; and when $\beta_0 = 0.6$, these two models reach a certain balance. Therefore, we suggest you to choose the value of β_0 within an interval $[0.55, 0.75]$, which enables our model to be more stable.

Finally, we summarize the advantages and disadvantages of our model: Its advantages are that it is profit-sensitive and risk-averse, and less affected by transaction costs; its disadvantage is that it is sensitive to β_0 values and needs to choose an appropriate β_0 value when using it.

Thank you for taking time out of your busy schedule to view this memo, and we hope our suggestions will be helpful.

Appendix: Code

strategy.py

```
from problem1_al import format_date, get_result, investment
import numpy as np
import  pandas as pd
import  os
from keras.models import Sequential, load_model
from sklearn.preprocessing import MinMaxScaler



time_step=30
predict_days=7
is_gold_buy,is_bit_buy=False,False
gold_p,bit_p=30,30
z_gold,z_bit=0.0204,0.0413
model_gold = load_model(os.path.join("DATA","LSTM_gold" + ".h5"))
model_bit = load_model(os.path.join("DATA","LSTM_bit" + ".h5"))


status=[1000,0,0]
buy_list=[]
money=[]
date=[]


df_bit = pd.read_csv('../BCHAIN-MKPRU.csv', engine='python',
                skipfooter=3)
df_gold = pd.read_csv('../LBMA-GOLD.csv', engine='python', skipfooter=3)
df_bit,df_gold=format_date(df_bit),format_date(df_gold)

data_LSTM_gold=df_gold.iloc[:,1].values
data_LSTM_bit=df_bit.iloc[:,1].values
data_LSTM_gold=np.reshape(data_LSTM_gold,(len(data_LSTM_gold),-1))
data_LSTM_bit=np.reshape(data_LSTM_bit,(len(data_LSTM_bit),-1))
data_LSTM_gold=data_LSTM_gold.astype('float32')
data_LSTM_bit=data_LSTM_bit.astype('float32')
scaler_bit = MinMaxScaler(feature_range=(0, 1))
scaler_gold=MinMaxScaler(feature_range=(0, 1))
data_LSTM_gold=scaler_gold.fit_transform(data_LSTM_gold)
data_LSTM_bit=scaler_bit.fit_transform(data_LSTM_bit)

data_VaR_gold=df_gold.iloc[:,1].values
data_VaR_bit=df_bit.iloc[:,1].values
data_VaR_gold=data_VaR_gold.astype('float32')
data_VaR_bit=data_VaR_bit.astype('float32')

alpha=0.6
temp=1000
while bit_p<df_bit.shape[0] and gold_p<df_gold.shape[0]:
    test_LSTM_bit=data_LSTM_bit[bit_p-30:bit_p]
    test_VaR_bit=data_VaR_bit[bit_p-30:bit_p]
    var1=gold_p if gold_p<31 else 31
    test_LSTM_gold=data_LSTM_gold[gold_p-var1:gold_p]
    test_VaR_gold=data_VaR_gold[gold_p-var1:gold_p]

    var1=(status[0]+0.99*status[1]*test_VaR_gold[len(test_VaR_gold)-1]+
```

```
        0.98*status[2]*test_VaR_bit[len(test_VaR_bit)-1])
    money.append(var1)
    if abs(var1-temp)>=10000:
        alpha+=-0.03 if var1>temp else 0.03
        temp=var1
    date.append(df_bit.iloc[bit_p,0])

    if is_gold_buy and  (not is_bit_buy):
        if df_bit.iloc[bit_p,0]!=df_gold.iloc[gold_p,0]:
            bit_p+=1
            continue
        f_bit=get_result(predict_days,test_LSTM_bit,
                         test_VaR_bit,model_bit,scaler_bit,z_bit,alpha)
        f_gold=get_result(predict_days,test_LSTM_gold,test_VaR_gold,
                         model_gold,scaler_gold,z_gold,alpha)

        if f_bit-f_gold>0 and f_bit>0:
            is_bit_buy,is_gold_buy,status=investment(1,buy_list,status,
                test_VaR_gold[len(test_VaR_gold)-1],
                df_gold.iloc[gold_p,0])
            is_bit_buy,is_gold_buy,status=investment(2,buy_list,status,
                test_VaR_bit[len(test_VaR_bit)-1],
                df_bit.iloc[bit_p,0])
        elif f_gold<0:
            is_bit_buy,is_gold_buy,status=investment(1,buy_list,status,
                test_VaR_gold[len(test_VaR_gold)-1],df_gold.iloc[gold_p,0])
        bit_p+=1
        gold_p+=1
        continue

    if (not is_gold_buy) and is_bit_buy:
        f_bit=get_result(predict_days,test_LSTM_bit,
                         test_VaR_bit,model_bit,scaler_bit,z_bit,alpha)
        if df_bit.iloc[bit_p,0]!=df_gold.iloc[gold_p,0]:
            if f_bit<0:
                is_bit_buy,is_gold_buy,status=investment(3,buy_list,status,
                    test_VaR_bit[len(test_VaR_bit)-1],
                    df_bit.iloc[bit_p,0])
            bit_p+=1
            continue
        f_gold=get_result(predict_days,test_LSTM_gold,
                         test_VaR_gold,model_gold,scaler_gold,z_gold,alpha)

        is_buy_gold=((status[0]+status[2]*test_VaR_bit[len(test_VaR_bit)-1]
                         *0.98)*0.99>test_VaR_gold[len(test_VaR_gold)-1])
        if f_gold-f_bit>0 and is_buy_gold and f_gold>0:
            is_bit_buy,is_gold_buy,status=investment(3,buy_list,status,
                test_VaR_bit[len(test_VaR_bit)-1],df_bit.iloc[bit_p,0])
            is_bit_buy,is_gold_buy,status=investment(0,buy_list,status,
                test_VaR_gold[len(test_VaR_gold)-1],df_gold.iloc[gold_p,0])
        elif f_bit<0:
            is_bit_buy,is_gold_buy,status=investment(3,buy_list,status,
                test_VaR_bit[len(test_VaR_bit)-1],df_bit.iloc[bit_p,0])
        bit_p+=1
        gold_p+=1
        continue

    if (not is_gold_buy) and (not is_bit_buy):
        f_bit=get_result(predict_days,test_LSTM_bit,
                         test_VaR_bit,model_bit,scaler_bit,z_bit,alpha)
```

```
        if df_bit.iloc[bit_p,0]!=df_gold.iloc[gold_p,0]:
            if f_bit>0:
                is_bit_buy,is_gold_buy,status=investment(2,buy_list,status,
                    test_VaR_bit[len(test_VaR_bit)-1],df_bit.iloc[bit_p,0])
            bit_p+=1
            continue
        f_gold=get_result(predict_days,test_LSTM_gold,
                        test_VaR_gold,model_gold,scaler_gold,z_gold,alpha)

        is_buy_gold=status[0]*0.99>test_VaR_gold[len(test_VaR_gold)-1]
        if f_bit<0 and f_gold<0:
            bit_p+=1
            gold_p+=1
            continue

        if f_gold>f_bit and is_buy_gold:
            is_bit_buy,is_gold_buy,status=investment(0,buy_list,status,
                test_VaR_gold[len(test_VaR_gold)-1],df_gold.iloc[gold_p,0])
        elif f_bit>f_gold:
            is_bit_buy,is_gold_buy,status=investment(2,buy_list,status,
                test_VaR_bit[len(test_VaR_bit)-1],df_bit.iloc[bit_p,0])
        bit_p+=1
        gold_p+=1
        continue
```