Problem Chosen	2022 MCM/ICM Summary Sheet	Team Control Number
C		2225151

Predict, Quantify, Trade, and Be Rich

Summary

Transaction of assets is always an important way to gain money and to protect the value of money we earned. In recent years, with the development of E-commerce, the Bitcoin has become a common trading asset. Combining traditional factors based upon quantitative trading strategies with the modern Deep Learning approach, we have successfully built a model that earned $179,580.55 from the initial $1000 dollar during 5-year period.

We firstly propose **Long Short Term Memory (LSTM) neural network** as our prediction model. LSTM takes the market price for the previous 5 trading days and predict the price for the next 5 trading days with all the history price information encoded in its cell state and hidden state. During later building of the planning model, we define a set of quantitative trading factors: Bias Ratio (BIAS), Sharp Ratio (SR), and Credit of Risk Return (RR) in order to quantify the market. We also compute Credit of Future Tendency (FT) calculated from the prediction given by LSTM. Based on these quantitative trading factors, we developed Credit to Action (A) as the signal to purchase or sell the asset by interpolating between the trading factors with linear weights.

Then, in order to determine the time for transactions, we propose and upgrade the programming model by taking more and more trading factors into consideration: **(i) Short and Long Term Moving Average (SLTMA) Model** is our baseline model, inspired by a simple concept of moving average, we are able to make about $45,000 with solely SLTMA. **(ii) SLTMA Model combined with the prediction data provided by LSTM**, resolves the problem of the delayed signal feedback and enables the model to make decision one step ahead, **(iii) The Deep Multi-factor Programming Model (DMPM)**, introduces Risk and uses a renewed quantification rule. Implemented with these predicted prices and the signal indicators, we are able to determine a best time to commit a transaction to find a maximal return during 5-year period. As a result, the maximal return the trader can gain is $179,580.55 using DMPM.

In order to prove DMPM is the best and find how sensitive DMPM is to the transaction cost, we first sampled 4 random variables from uniform distribution to replace the 4 hyper parameters utilized in Deep Multi-factor Programming Model. Next, we modified the transaction cost with fixed step length to perform sensitive analysis regarding to the transaction cost.

Based on the analysis above, we develop an end-to-end trading model that only takes the market price at previous days as input, and the model can be easily understood by the trader. Benefited by the accuracy of the prediction from deep learning model and the consideration for risks from quantitative factors, the DMPM gives careful considerations to possible opportunities.

Keywords: Deep Learning; LSTM neural network; Moving Average (MA), Bias Ratio (BIAS), Sharp Ratio (SR), Credit of Risk Return (RR); Deep Multi-factor Programming Model (DMPM).

Contents

1 Introduction

1.1 Background

Volatile assets, which has the property of volatility, are used in today's trade frequently. Volatility, the amount of uncertainty or risk related to the size of changes in a security's value, often influences people's choices of buying those assets. Today's most common volatile assets are gold and bitcoins. Thus, we are going to analyze the volatility of gold and bitcoin to help traders determine when they should commit a transaction of these assets to maximize their total return.

Figure 1: Background [1]

1.2 Restatement of the Problem

We have been asked by a trader to tell him when he should buy, hold, or sell his assets in his portfolio (Cash, Gold, Bitcoin). He only has $1000 cashes on 9/11/2016, which is the time we start. From his request, we need to do the following:

- **Developing a model** The model should run five-year period from the starting point 9/11/2016 to the ending point 9/10/2021 and use only the past stream of daily prices to date.
- **Determining the time for transaction** Based upon the model, we need to tell the trader when to buy, sell or hold the three assets (Cash, Gold, and Bitcoin).

1.3 Our Work

Based upon the requirements made by the trader, we have developed two kinds of models. First one is the prediction model for predicting price of two assets (gold and bitcoins). Second one is the planning model for determining the time to purchase, sale, or hold the three assets (gold, bitcoins, and cashes).

The Long Short Term Memory (LSTM) neural network is used in the prediction model, which will be discussed in detail in Section 4 and 5. The planning model, based upon machine learning and moving average, is used to determine when to commit a transaction. More details about the planning model will be discussed in Section 6.

Figure 2 has been included to illustrate our work.

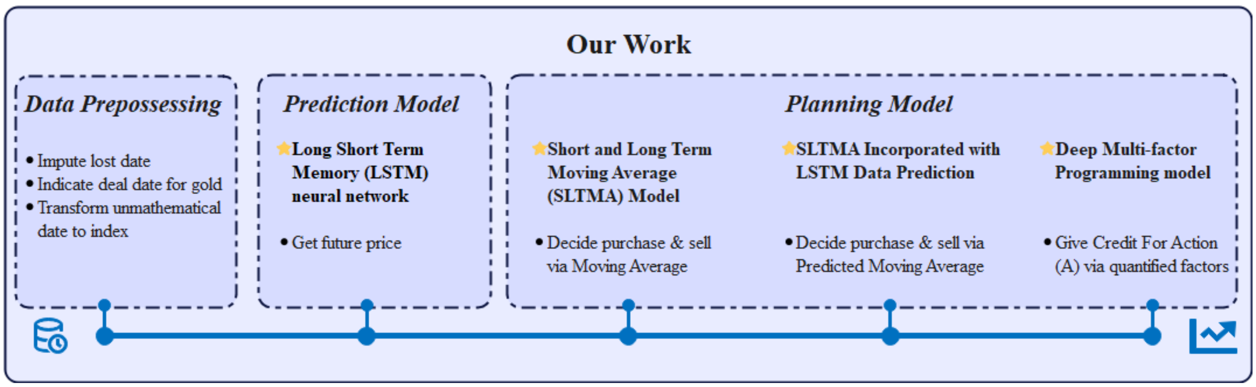

Figure 2: Our Work

2 Assumptions and Justification

To simplify the problem and make it convenient for us to simulate real-life conditions, we make the following basic assumptions, each of which is properly justified.

- **Transaction costs are constant for gold and bitcoin**.

  $\alpha_{gold} = 1\%$ and $\alpha_{bitcoin} = 2\%$
- **All data we have for the price of gold and bitcoin are correct and accurate**.
- **The trader has the ability to afford the risk of these two assets.**
- **The trader will follow our instruction, no matter how ridiculous it will be.**
- **Risk-free rate is zero ($R_f = 0$).** There are only 3 assets in the market and the risk-free option is to hold the cash.

3 Notations

Symbols	Description	Unit
l_i	Input Length, equals 5 in our model	–
l_o	Output length	–
l_p	Prediction length, equals 5 in our model	–
$P_{g,i}$	The actual gold price at day i	\$
$P_{b,i}$	The actual bitcoin price at day i	\$
MA_n	Moving average price for n days	\$

where we define the main parameters while specific value of those parameters will be given later.

4 Data Prepossessing

In order to make convenient to our calculation and simulation, we do following prepossessing in the given data:

1. **Impute the lost dates (weekends) in LBMA-GOLD.csv.** We impute price of Gold during Saturday and Sunday by the average price. Since we added the deal date indicator, this imputation will not affect the final assets value.
2. **Indicate the date for trade for Gold.** we assume 1 for work day and 0 for weekends. This would help us determine when we are able to sell the Gold.
3. **Transform date information to index.** e.g., 9/11/2021 is equivalent to index 0.

5 Model Overview

We have developed two models: (1) Prediction model, using **Long Short Term Memory (LSTM) networks** to predict the price of gold and bitcoins; (2) Planning model, utilizing machine learning to predict the future price of both assets.

For the planning model, we have developed two basic model, (i) **Short and Long Term Moving Average Model**; (ii) **Short and Long Term Moving Average Model Incorporated with LSTM Data Prediction**, to check the validity and sensuality of our indices; (iii) **Deep Multi-factor Programming Model** by renewal and quantification of the factors after receiving positive feedback in simple simulations of previous models.

6 Model I :Prediction Model

As mentioned in Section 4, we use LSTM networks to build the prediction model. First, we will introduce the basics of LSTM networks, which uses the past stream of data to predict the future data.

6.1 LSTM networks

LSTM networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems [2], which matches our problem here as we need to predict the price of gold and bitcoins to determine when to commit a transaction or not.

In LSTM Network, the internal state c_t and the hidden state h_t are updated with equations (1) and (2),

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{1}$$
$$h_t = o_t \odot \tanh(c_t) \tag{2}$$

where the i_t, o_t, and f_t represents the input gate , output gate, and forget gate, respectively. The formula for the three gates are presented by equations (3), (4), and (5):

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \tag{3}$$
$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \tag{4}$$
$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \tag{5}$$

The σ_g function is calculated with equation (6)

$$\sigma_g = \frac{1}{1 + e^{-x}} \tag{6}$$

The $\tilde{c}_t$ represents the activation vector, and is obtained in the equation (7):

$$\tilde{c}_t = \tanh W_c x_t + U_c h_{t-1} + b_c \tag{7}$$

6.2 Establishment of the Prediction Model

During the inference process, we are taking the market price of the previous l_i days as the input to the auto-regressive LSTM model to predict market price for the next day, denoted as P_0. Then we take P_0 as input to get P_1. We repeat this process until we get P_{l_p}. After the l_p trading days have passed, we collect the data for the l_p trading days and train the LSTM from the last checkpoint using teacher-forcing technique as illustrated in figure 3.

The MSELoss in the figure 3 are calculated in equation (8)

$$MSELoss = \frac{1}{n} \sum_{i}^{n} (P_{l_i} - y_i)^2 \tag{8}$$

, where y_i will be $P_{g,i}$ or $P_{b,i}$ depending upon the asset we use.

To prevent possible data leakage, we applied an adaptive way of normalizing the input data as illustrated in Algorithm 1. Let $S_p = P_i, P_{i+1}, ...P_{i+l_i}$ represent the market price of either gold or bitcoin from day i to day $i + l_i$. We let S_p denote the input data to the LSTM model. To perform the normalization of our input data, we let min_p denote $min(S_p)$ and max_p denote $max(S_p)$. We divide $S_p - min_p$ by $max_p - min_p$ to get the normalized data and use the normalized data for training. We repeat this process until all the data has been trained.

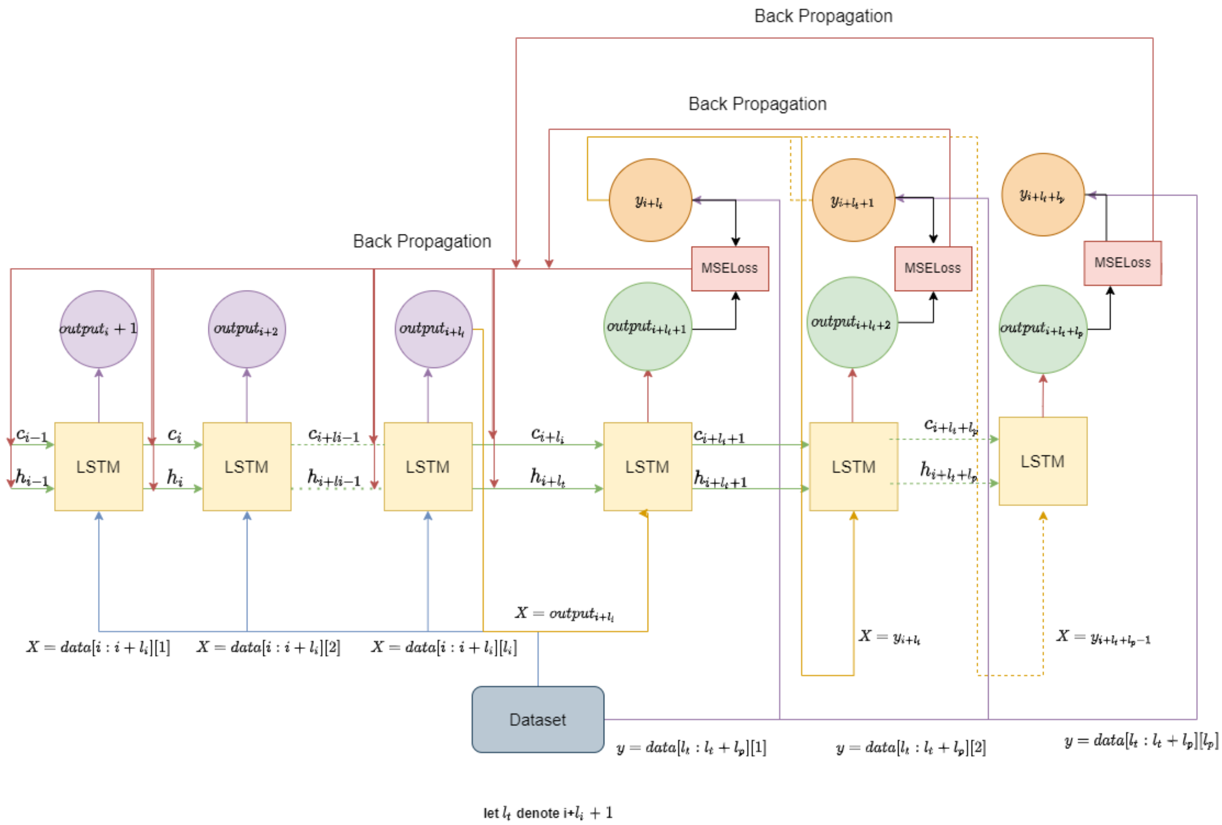

Figure 3: LSTM networks

Algorithm 1 Adaptive Normalization

Input $P_{\{i:i+l_p\}} = P_i, P_{i+1}, ..., P_{i+l_p}$
$min \leftarrow min(P_{\{i:i+l_p\}})$
$max \leftarrow max(P_{\{i:i+l_p\}})$
for $i \in P_{\{i:i+l_p\}}$ **do**
  $AdaptivelyNormalized(i) \leftarrow \frac{i-min}{max-min}$
end for
Return $AdaptivelyNormalized(P_{\{i:i+l_p\}})$

During inference time, we multiple the predicted result by the $max(S_p)$ to recover the predicted data.

6.3 Data visualization using Prediction Model

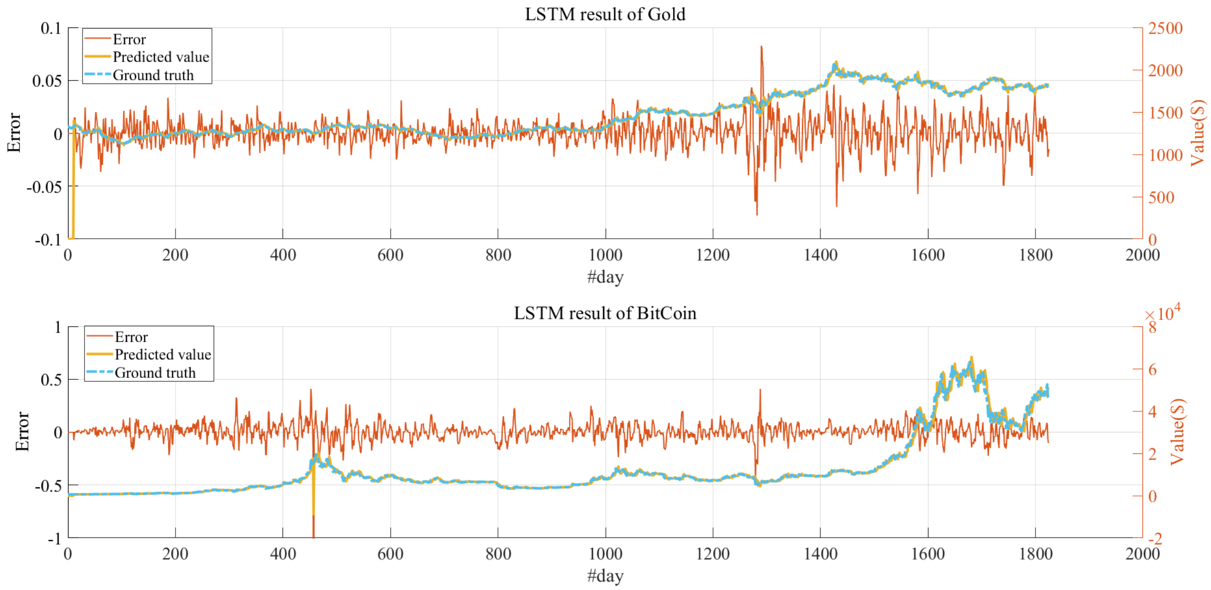

Figure 4: LSTM Prediction

As we saw in Figure 4, the error of LSTM between predicted and actual price is around 3% for Gold and 15% for Bitcoin.

7 Model II: Planning Model

This section intends to illustrate how we build the Deep Multi-factor Programming model by adding and testing different factors step by step with sub-models. We will first explain mechanics used during model building, then, 3 models will be given in later section to show:

1. How we use **Short and Long Term Moving Average** Model to test the sensuality of the moving average
2. How we incorporated Short and Long Term Moving Average Model with **LSTM predicted data** to test the validity of the prediction and improve the Model I
3. How we further take **Risk** into account and quantify all indices to give precise suggestions in purchase & sale in the final **Deep Multi-factor Programming model**.

The detailed dependency between indices and models could be found in Figure 5.

7.1 Mechanics of Planning Model

This subsection will introduce several indices used during model building and the operations we use to obtain them from the given data, Figure 5 is included to illustrate the relationship between indices and models.

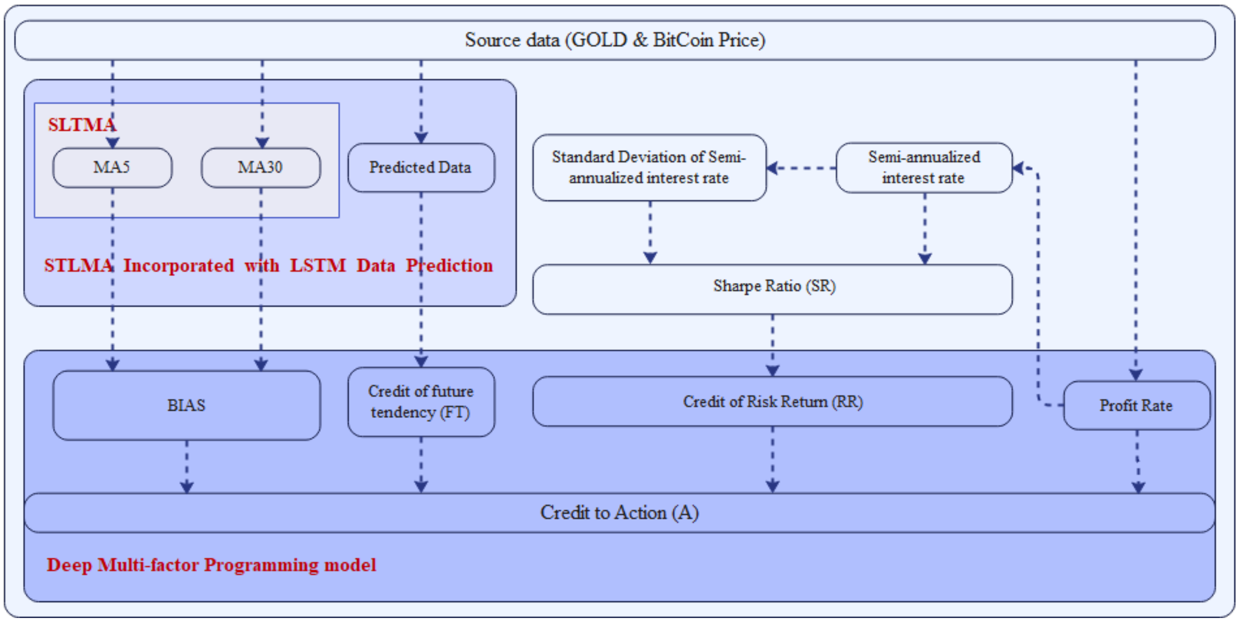

Figure 5: Variable and model inclusion

7.1.1 Moving Average

A moving average (MA) is a calculation to analyze data points by creating a series of averages of different subsets of the past dealing data set [3]. Here, we used it to calculate the mean price for the gold and bitcoins during 5 and 30 days, respectively.

$$MA_5(i) = \frac{1}{5}\sum_{i-4}^{i} P_i \tag{9}$$

$$MA_{30}(i) = \frac{1}{30}\sum_{i-29}^{i} P_i \tag{10}$$

, where P_i is the price for bitcoins and gold in day i, $MA_5(i)$ and $MA_{30}(i)$ are the meaning average for 5 days and 30 days, respectively.

7.1.2 Bias Ratio (BIAS)

BIAS is an indicator to measure the rate of difference between the price of the current day and the average historical price [4]. Generally, it is calculated by:

$$BIAS(i) = \frac{P_i - MA_n(i)}{MA_n(i)} \times 100\% \tag{11}$$

To prevent over-catching of the short-time fluctuation, we use a short-period moving average to replace the single-day price. The moving-average-BIAS, hence, should be calculated by:

$$BIAS_{avg}(i) = \frac{MA_{short}(i) - MA_{long}(i)}{MA_{long}(i)} \times 100\% \tag{12}$$

, where $MA_{short}(i)$ is a short-period moving average in day i and $MA_{long}(i)$ is a long-period moving average in day i. We choose to use 5 and 30 days, respectively.

When $abs(BIAS_{avg}(i))$ is very large (at least 8% empirically), with high probability that the market is either oversold ($BIAS_{avg}(i) < 0$) or overbought ($BIAS_{avg}(i) > 0$), and the later price is likely to move toward to the average price, which is a good signal for sale or purchase.

7.1.3 Sharp Ratio (SR) and Credit of Risk Return (RR)

Sharp Ratio (SR) is a factor that helps traders understand the return of an investment compared to its risk. The ratio is calculated in equation (13)

$$SR = \frac{R_p - R_f}{\sigma_p} \tag{13}$$

, where R_p is the return of the portfolio, R_f the return of the risk-free rate, and σ_p the standard deviation of of the portfolio's excess return in a given time period [5].

As shown in the equation (13), the sharp ratio is interpreted as how many unit of excess return the trader can gain for each additional unit of risk the trader takes. A SR value grater than 1 means that the return rate is higher than the volatility risk.

Based on our assumption, we set R_f = 0. In calculation, we use time period of 180 days to calculate R_p and σ_p. Additionally, to make it convenient for our calculation, we define another parameter *Risk Return Credit* (RR), which is

$$RR = SR - 1 \tag{14}$$

7.1.4 Credit of Future Tendency (FT)

Credit of future tendency (FT) is obtained from the data predicted by LSTM model, and is used to describe how future prices will change compared with current time. Short term FT (based on single day) could be calculated by:

$$FT_{i,short} = \frac{P_{i+1} - P_i}{P_i} \tag{15}$$

To eliminate the effect of fluctuations and make full use of the predicted n-day data, we use $MA_n(i)$ to replace P_i and $\sum(W_{predict} \times P_{[i+1:i+5]})$ to replace P_{i+1}, where $W_{predict}$ is the weight of each predicted days. The FT formula, hence, should be:

$$FT_i = \frac{1}{MA_5(i)} \sum_{j=i+1}^{i+5} (P_j - MA_5(i)) \cdot w_j = \frac{\sum(W_{predict} \cdot P_{[i+1:i+5]}) - MA_5(i)}{MA_5(i)} \tag{16}$$

In our model, we use $n = 5$. Based on the accuracy of the LSTM model and the importance rank of future prediction, we decide $W_{predict}$ as show in table 1.

Table 1: The weight for each days in 5-day period

Day 1	Day 2	Day 3	Day 4	Day 5
30%	20%	20%	20%	10%

For an asset in given date i with $FT_i > 0$, we decide that, with high probability, the price of the asset will keep elevating and vice versa.

7.1.5 Credit to Action (A)

The Credit to Action is an indicator that determines when is a good time to sell or purchase the asset. When the value is positive, it is a good time for sale; otherwise, it is time for purchase.

The way to calculate the credit to Action is shown in equation (17).

$$A = w_{Bias} \cdot Bias + w_{RR} \cdot RR + w_{PR} \cdot PR + w_{FT} \cdot FT \tag{17}$$

, where RR is the Credit of Risk Return, PR the Profit Rate, CF the Credit of Future Tendency, we further decide weight ,w, as show in table 2.

Table 2: The weight for each factor for Credit to Action

$Bias$	RR	PR	FT
0.214	0.392	0.023	0.359

7.2 Establishment of Planning Model

7.2.1 Sub-model I: Short and Long Term Moving Average (SLTMA) Model

In this sub-model I, we choose MA5 line as the short term line, MA30 line as the long term line, through their intersection we can deduce the future tendency.

- If the MA5 line crosses MA30 line from down to up, it indicates that the return for the average of five-day price supersedes that for past 30 days, which is a good time to purchase assets.
- If the MA5 line crosses MA30 line from up to down, it implies that the return for the average of five-day price is lower than that for past 30 days, which is a signal to sell assets.

We do a simple simulation with BChain data, as shown in figure 6, the red circle appears when MA5 line passes MA30 line from up to down; likewise, the black circle appears when MA5 line passes MA30 line from down to up.

However, the property of hysteresis of the moving averages limits the preference of such a model: when the intersection between short-term and long-term moving average occurs, the local lowest or highest price point must already be passed. Even one-day act ahead will result a large improvement of return in such model.

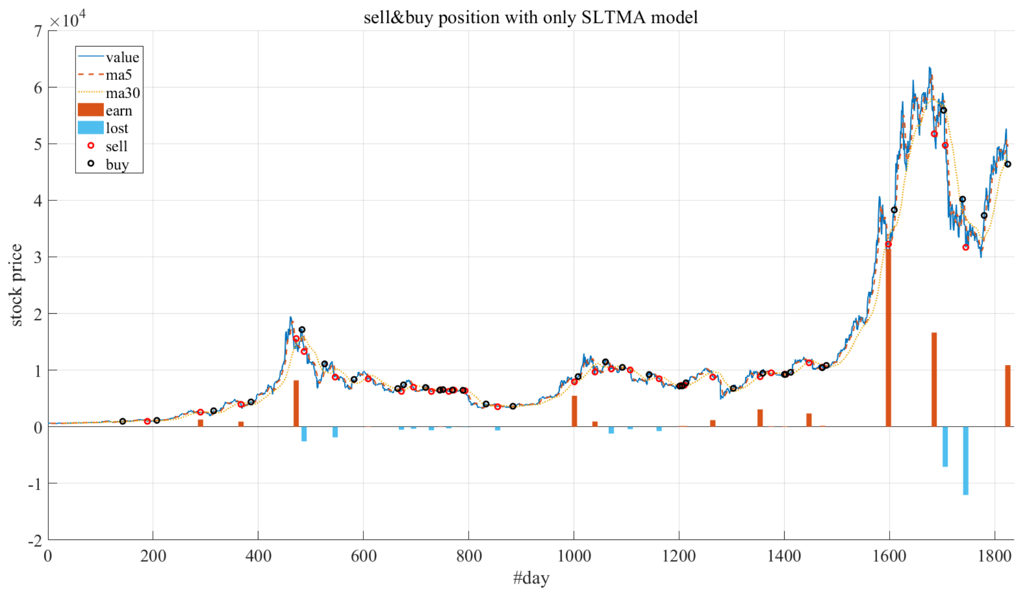

Figure 6: SLTMA model without Machine Learning

7.2.2 Sub-model II: Short and Long Term Moving Average Model Incorporated with LSTM Data Prediction

For this sub-model II, we use the same model as described above in Section 6.2.1; however, we apply LSTM data prediction to achieve early-action.

With LSTM, we get a prediction of future 5-day prices for a given date i and use such data to forecast $MA_5(i+5)$ and $MA_{30}(i+5)$. By determining possible intersections of moving averages in future 5 days, we are able to act ahead at most 5 days. The similar simulation with BChain data is showed in figure 7.

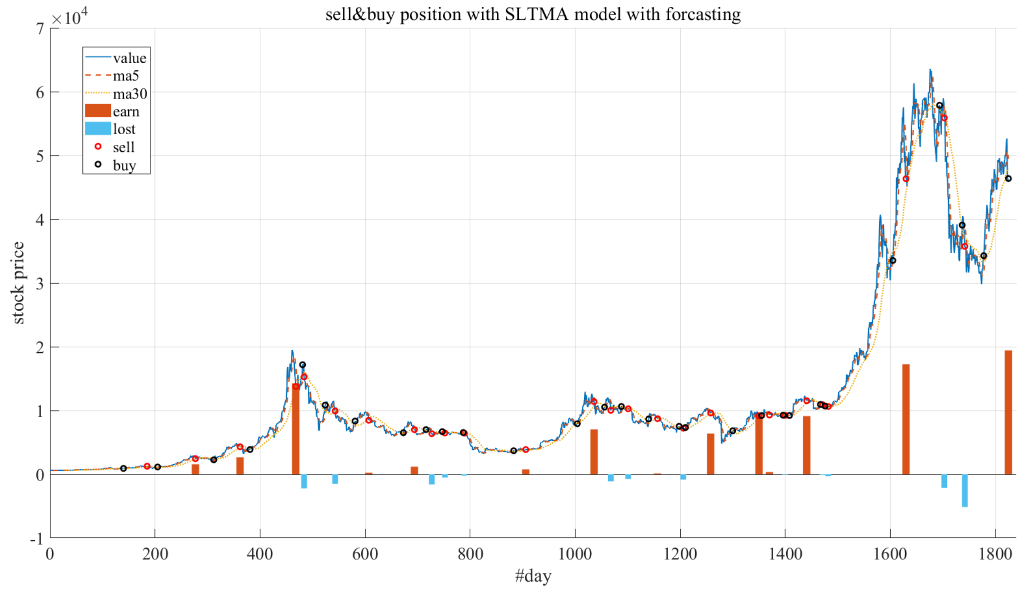

Figure 7: SLTMA model Incorporated with LSTM Data Prediction

Further analysis shows an out-compete of this improved model, figure 9 reflects how Model II ends up with nearly $\$20,000$, and figure 8 shows an example of how such an improved model acts ahead by predicting intersection.

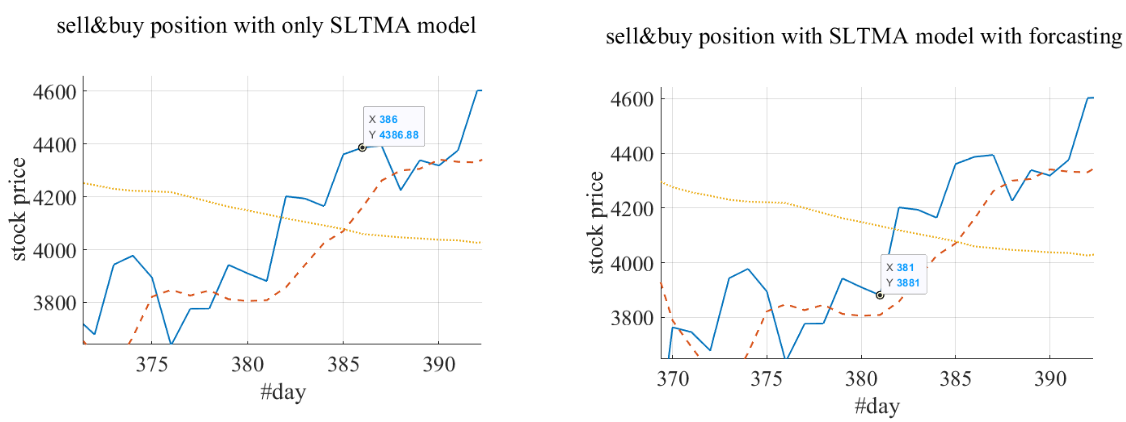

Figure 8: Act-ahead effect of SLTMA with LSTM data

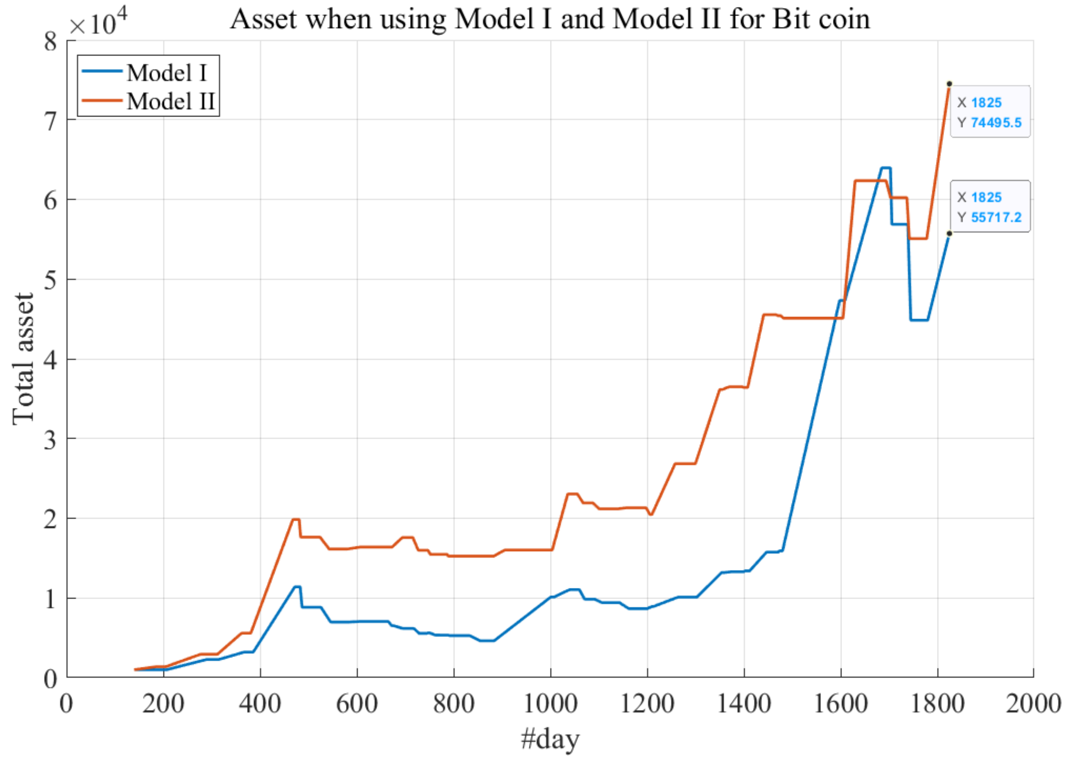

Figure 9: Total assets when using Model I and Model II in BChain

Shortcomings are still remained in such improved model: The model is overly dependent upon the intersection of moving averages and will perform extremely unstably during period with large fluctuations, and will miss many possible purchase & sale points without moving averages intersection.

7.2.3 Sub-model III: Deep Multi-factor Programming model

To overcome the shortcomings mentioned above and give precise purchase & sale suggestions, we must be able to take risk into account and quantify the importance of each factor.

- **Review the Abbreviations**

  Based on the shortcomings of previous models, we define several abbreviations for the model as indicated in table 3. The detail of definition of each abbreviation has already been described in the previous section and will not be stated redundantly here.

Table 3: Main Abbreviation used in model

Factors of Planning Model	**Abbreviation**	**Information**
Bias Ratio	BIAS	quantify the risk, derived from moving average
Profit Rate	PR	the rate of profit of day i, compared with day $i-1$
Credit of Risk Return	RR	quantify the relationship between BIAS and return rate
Credit of Future Tendency	FT	quantify the tendency of future price, derived from LSTM data
Credit to Action	A	quantify the extent of and time to purchase & sell an asset

- **Use the factors to decide purchase, hold, or sell**

  To plan when and the amount to purchase or sell assets at day j for the next 5 days, we compare the Credit to Action A with 4 different dynamically generated thresholds based on previous days' market prices.

  Let G_L, G_H, B_L, and B_H, M_C denote the lower bound for gold, higher bound for gold, lower bound for Bitcoin, higher bound for Bitcoin, and the current money we have respectively.

  First we obtain the normalized Credit to Action $A_{AN} \in [0, 1]$ using **Algorithm 1**. Let

$$G_{max,i} \equiv \mathbb{1}(P_{g,i} == max(P_{g,0}, P_{g,1}, ..., P_{g,i}) \tag{18}$$

  to represent if day i has exceeded the maximum Gold price in all previous days and

$$G_{min,i} \equiv \mathbb{1}(P_{g,i} == min(P_{g,0}, P_{g,1}, ..., P_{g,i}) \tag{19}$$

  to represent if day i is the minimum Gold price in all previous days, where $\mathbb{1}$ is the indicator function (i.e. $\mathbb{1}(True) == 1$ and $\mathbb{1}(False) == 0$). Then we define

$$G_H = \frac{\sum_{i=0}^{j} G_{max,i} A_{AN}[i]}{\sum_{i=0}^{j} G_{max,i}} \tag{20}$$

  and

$$G_L = \frac{\sum_{i=0}^{j} G_{min,i} A_{AN}[i]}{\sum_{i=0}^{j} G_{min,i}} \tag{21}$$

  . We can obtain the counterparts in Bitcoin,

$$B_H = \frac{\sum_{i=0}^{j} B_{max,i} A_{AN}[i]}{\sum_{i=0}^{j} B_{max,i}} \tag{22}$$

  and

$$B_L = \frac{\sum_{i=0}^{j} B_{min,i} A_{AN}[i]}{\sum_{i=0}^{j} B_{min,i}} \tag{23}$$

  .

  Following the way we defined the lower bound and upper bound for Gold and Bitcoin during planning time, these thresholds represent the average value of best selling prices and buying prices from historical trading data. For day i in the next five days, if $A_{AN}[i] < G_L$, then we buy $G_L M_C / P_{g,i}$ amount of gold; if $A_{AN}[i] > G_H$, then we sell $G_H M_C / P_{g,i}$ amount of gold. The way of deciding deals are the same for Bit Coin, where we buy $B_L M_C / P_{b,i}$ amount of gold if $A_{AN}[i] < B_L$ and sell $B_H M_C / P_{b,i}$ amount of gold if $A_{AN}[i] > B_H$

7.3 Data Visualization using Planning Model

With the Deep Multi-factor Programming model, we are able to decide the action for each day based on the Credit to Action. Figure 10 shows the way and the amount we should purchase & sell at each given date.

Figure 10: Actions of each day

Figure 11 shows the change of the number of different stocks we hold after following the instructions showed in Figure 10. Based upon all these figures shown above, We conclude that the model performs rationally, in agreement with human activities: purchase assets at low price and sell them at high price.

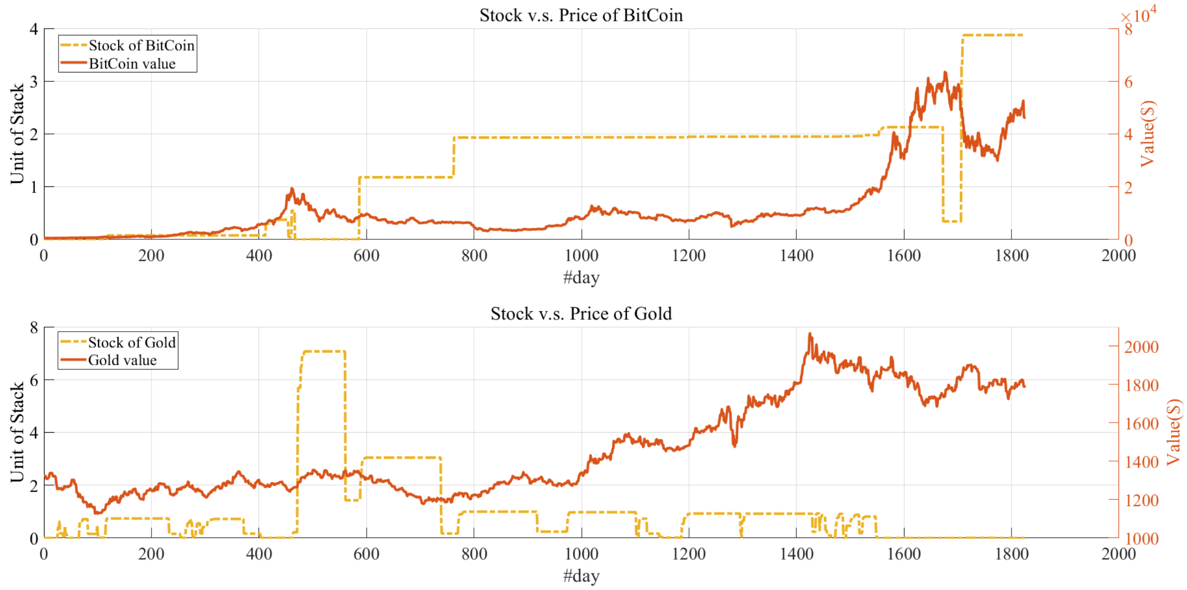

Figure 11: Stock v.s. Price

With such an operation, we plot Figure 12 to show the change of different combinations of the total assets during five-year period time:

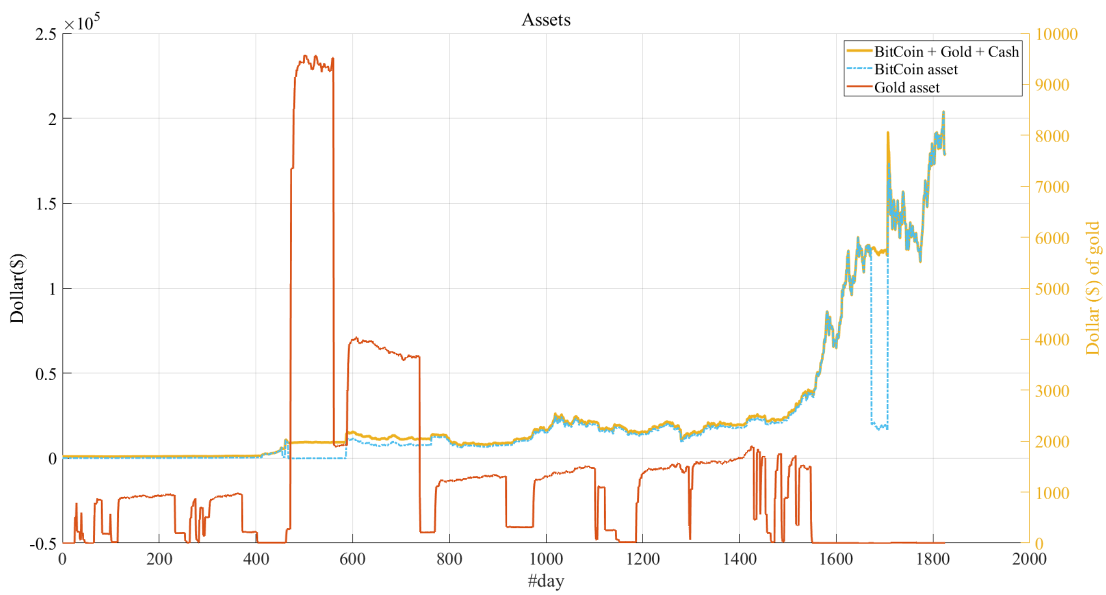

Figure 12: Assets when using Deep Multi-factor Programming model

8 Model Analysis and Sensitivity Analysis

To find how sensitive the Deep Multi-factor Programming model is to the 4 thresholds and the transaction cost α_G and α_B when we are making decisions, we conduct the following analysis

8.1 The proof of of the prediction model is optimal

For the prediction model, we used LSTM, which is a type of neural network. According to our observation, the training loss and evaluation loss have only around 10% of increment and decrement for either Gold or Bitcoin after 500 days and the difference between the predicted price and the real market price is close. Thus we believe this shows our model has settled around a regional minimum

In order to prove our Deep Multi-factor Programming mode is optimal, we have sampled 4 $X_1, X_2, X_3, X_4 \sim U(0, 1)$ thresholds from Uniform distribution to replace the four hyper parameters G_L, G_H, B_L, B_H defined in (21), (20), (23), and (22). We ran this process 200 times and generated the table with selected thresholds that are representative to the search space.

G_H	G_L	B_H	B_L	Result
0.9	0.6	0.9	0.6	$\$149,733.64$
0.8	0.5	0.8	0.5	$\$97,522.72$
0.7	0.4	0.7	0.4	$\$70,301.33$
0.6	0.3	0.6	0.3	$\$100,447.48$
0.5	0.3	0.6	0.3	$\$30,393.86$

Table 4: Generated Results With Swapped Hyper Parameters (inputs and results have been rounded)

To the best of our knowledge, our way of generating the 4 thresholds as the only hyper parameters in our Deep Multi-factor Programming model has surpassed all the samples in the 200 value pairs we generated from Gaussian distribution, Thus it has empirically shown our model is optimal.

8.2 Sensitivity of Deep Multi-factor Programming model to transaction costs

In order to find how sensitive the transaction costs are to the DMPM, we set the transaction costs at various values and generated the following results shown in table 5.

As shown in table 5, when α_G remain constant, the money returned is decreased by \$11,000 with the increasing of α_B in steps of 0.01; however, when α_B remains constant, the money returned is decreased by about \$50,000 with the increasing of α_G in steps of 0.01. This result shows that the returned money is mostly affected by the transaction cost of gold, α_G, with mild influence by that of Bitcoins, α_B.

This result matches our expectation. After we took risk into consideration, our model is trading more frequently in Gold compared with Bitcoin as shown in Figure 12, resulting to the Credit to Action signal of gold having a much lower standard deviation compared to that of the Bitcoin. Thus, the model is more sensitive to the transaction cost of gold because of the high trading frequency of gold.

Index	α_G	α_B	asset ($)
0	0.01	0.01	190,454.35
1	0.01	0.02	179,580.35
2	0.01	0.03	169,020.35
3	0.01	0.04	158,767.35
4	0.02	0.01	139,085.35
5	0.02	0.02	130,172.35
6	0.02	0.03	121,515.35
7	0.02	0.04	113,110.35
8	0.03	0.01	97,218.35
9	0.03	0.02	89,902.35
10	0.03	0.03	82,796.35
11	0.03	0.04	75,897.35
12	0.04	0.01	63,018.35
13	0.04	0.02	57,006.35
14	0.04	0.03	51,167.35
15	0.04	0.04	45,498.35

Table 5: Sensitivity analysis using 16 different pairs of transaction rate for gold and bitcoin, where α_G is the transaction cost of gold,and α_B is that of bitcoins.

9 Strength and Weakness

9.1 Strength

- The model is easy to understand. The model is developed step by step from a simple concept: **the Moving Average**.
- The model makes good quantification and hence is easy to follow by. The model quantifies several common factors mentioned in Section 7.1 and gives the final grade for each day. One only needs to compare the grade with a given threshold and simply to decide the action to take.
- The model actually makes money without Data Leakage

9.2 Weakness

- Some weights between the factors in the model are determined without much detailed thoughts. For example, the weights in table 1 and table 2 are determined with empirical results.
- The prediction model cannot distinguish the bear market from the bull market, which results in losses in our portfolio. We can further improve the prediction model by adding a new LSTM model to indicate the bear market or bull market, given previous days' of the market state.

- The model still has space for improvements. The model does not use anything except the past stream of daily prices to date to determine each day if the traders should buy, hold, or sell their assets in their portfolio.

References

[1] https://www.istockphoto.com/photo/three-experienced-stock-traders-talking-business-consulting-documents-and-argue-about-gm866662328-144146297. [Online; accessed 21-February-2022].

[2] Wikipedia contributors. *Long short-term memory — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Long_short-term_memory&oldid=1068745091. [Online; accessed 20-February-2022]. 2022.

[3] Wikipedia contributors. *Moving average — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Moving_average&oldid=1069105690. [Online; accessed 20-February-2022]. 2022.

[4] Wikipedia contributors. *Bias ratio — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Bias_ratio&oldid=1041891990. [Online; accessed 20-February-2022]. 2021.

[5] Wikipedia contributors. *Sharpe ratio — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Sharpe_ratio&oldid=1071968724. [Online; accessed 21-February-2022]. 2022.

Appendices

Appendix A First appendix

A.1 Glossary

Symbols	Description	Unit
l_i	Input Length, equals 5 in our model	–
l_o	Output length	–
l_p	Prediction length, equals 5 in our model	–
$P_{g,i}$	The actual gold price at day i	$
$P_{b,i}$	The actual bitcoin price at day i	$
MA_n	Moving average price for n days	$
c_t	Internal state	–
h_t	Hidden state	–
f_t	Forget gate	–
i_t	Input gate	–
o_t	Output gate	–
BIAS	Bias Ratio	–
SR	Sharp Ratio	–
RR	Risk Return	–
FT	Credit of Future Tendency	–
$W_{predict}$	weight of each predicted days	–
A	Credit to Action	–
G_{max}	Number of times the maximum price of gold has been surpassed	–
G_H	Higher price bound for gold	–
G_L	Lower price bound for gold	–
B_H	Higher price bound for Bitcoins	–
B_L	Lower price bound for Bitcoins	–
α_G	Transaction cost for gold	–
α_B	Transaction cost for Bitcoins	–

10 Memorandum

TO: Mr. Trader

FROM: TEAM #2225151

DATE: February 21, 2022

SUBJECT: **Trading Strategies to Gain the Maximal Return of Two Assets (Gold and Bitcoins).**

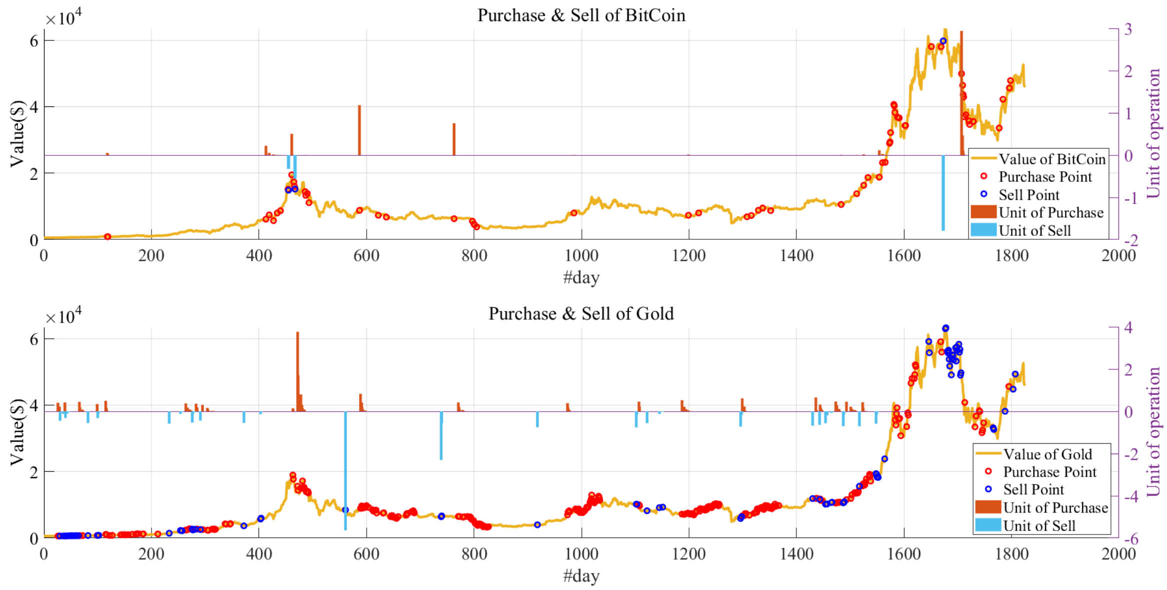

Figure 13: The time to purchase and sell assets

Based upon your request to earn the maximum return of two assets, we have proposed the e Deep Multi-factor Programming Model (DMPM), an end-to-end trading model that can predict the market price for the next 5 days and optimize your asset allocation from the predicted result, and all you have to do is The DMPM model consists mainly of 2 parts: the prediction model and the planning model.

Prediction model, as indicated by its name, is used to predict the future price of two assets, We used the LSTM Neural Network as the prediction model,which takes the market price for the previous 5 trading days and predict the price for the next 5 trading days with all the history price information encoded in its cell state and hidden state.

For the Planning model, we define a set of quantitative trading factors: Bias Ratio (BIAS), Sharp Ratio (SR), and Credit of Risk Return (RR) in order to quantify the market. We also compute Credit of Future Tendency (FT) calculated from the prediction given by LSTM. Based on these quantitative trading factors, we developed Credit to Action (A) as the signal to purchase or sell the asset by interpolating between the trading factors with linear weights. Then during inference time, we take the previous 5 days' market price as inputs to calculate the Credit to Action signal index.

Then we compare the signal index with the dynamically generated threshold based on historical data to determine whether to purchase or sell the assets and the amount of the transaction.

Following the combination of prediction model and planning model using the sub-model III, you need to commit purchase and sale shown by the figure 13, which can finally return with $ 179,580.55.

In conclusion, as mentioned above, after you perform these steps, you should earn about $179,580.55 if you start with $ 1000 cashes.

We hope this summary will prove helpful in meeting your needs.