Problem Chosen	2022	Team Control Number
C	MCM/ICM Summary Sheet	2225002

Mining the Past and Envisioning the Future

—— The "Optimal" Road to Fortune

Summary

Given the initial fund and the historical price information of gold and bitcoins, this article aims to explore the strategies that yields the maximal terminal value utilizing only the two given price curves, and only assessing the historical price data at each step.

Given such heavy dependence on the historical prices, the first step conducted by this article is to cast light on the general properties of historical data. In this part, **moving average** with window size 7 is applied for visualization and the **correlation coefficients** are calculated by aligning the prices of the bitcoins and the prices of the gold with padding and masking. Then, **macro environment** and **nature of bitcoin or gold** are incorporated to illustrate the behaviors of the price curves. Additionally, **AutoCorrelation function (ACF)** and **power band method** are applied to conclude that the cycle of bitcoin price curve is 6-8 days while the cycle of the gold price curve is 14-18 days.

After understanding the properties, **AutoRegressive Integrated Moving Average (ARIMA)**, **Long Short-Term Memory (LSTM) Neural Network**, and **Gradient Boosting Regression (GBR)** are applied to predict the curve based on the historical data. The subsequent analysis shows that ARIMA provides an accurate prediction but with a lag about 2-4 days, LSTM gives a dull and noise-sensitive prediction, and the GBR returns an accurate prediction. The performance of each time series prediction model is used to weigh three predictions for a consolidated price forecast.

With the given prediction models, three strategies, **Naive Strategy**, **Upheaval-Oscillation-based Strategy**, and **Deep Reinforcement Learning Strategy**, are developed respectively, combining the price data properties and the prediction models. Next, **sensitivity analysis on the transaction commission rates**, and the **ablation analysis on the prediction models** are conducted to verify our modeling choice.

Eventually, a rigorous proof is provided to prove the non-existence of an optimal strategy by showing that obtaining the maximal total terminal value is equivalent to obtaining the maximal total value in every intermediate step, and obtaining the optimal value at every step is equivalent to knowing the gold and bitcoins up and down trend and (and growth rates if they both increase). Even though, the results show that the strategies give a over 100 times return.

Keywords: ARIMA; LSTM; Gradient Boosting Regression; Upheaval-Oscillation-based Strategy; Deep Reinforcement Learning

Mining the Past and Envisioning the Future

—— The “Optimal” Road to Fortune

2225002

February 22, 2022

Summary

Given the initial fund and the historical price information of gold and bitcoins, this article aims to explore the strategies that yields the maximal terminal value utilizing only the two given price curves, and only assessing the historical price data at each step.

Given such heavy dependence on the historical prices, the first step conducted by this article is to cast light on the general properties of historical data. In this part, **moving average** with window size 7 is applied for visualization and the **correlation coefficients** are calculated by aligning the prices of the bitcoins and the prices of the gold with padding and masking. Then, **macro environment** and **nature of bitcoin or gold** are incorporated to illustrate the behaviors of the price curves. Additionally, **AutoCorrelation function (ACF)** and **power band method** are applied to conclude that the cycle of bitcoin price curve is 6-8 days while the cycle of the gold price curve is 14-18 days.

After understanding the properties, **AutoRegressive Integrated Moving Average (ARIMA)**, **Long Short-Term Memory (LSTM) Neural Network**, and **Gradient Boosting Regression (GBR)** are applied to predict the curve based on the historical data. The subsequent analysis shows that ARIMA provides an accurate prediction but with a lag about 2-4 days, LSTM gives a dull and noise-sensitive prediction, and the GBR returns an accurate prediction. The performance of each time series prediction model is used to weigh three predictions for a consolidated price forecast.

With the given prediction models, three strategies, **Naive Strategy**, **Upheaval-Oscillation-based Strategy**, and **Deep Reinforcement Learning Strategy**, are developed respectively, combining the price data properties and the prediction models. Next, **sensitivity analysis on the transaction commission rates**, and the **ablation analysis on the prediction models** are conducted to verify our modeling choice.

Eventually, a rigorous proof is provided to prove the non-existence of an optimal strategy by showing that obtaining the maximal total terminal value is equivalent to obtaining the maximal total value in every intermediate step, and obtaining the optimal value at every step is equivalent to knowing the gold and bitcoins up and down trend and (and growth rates if they both increase). Even though, the results show that the strategies give a over 100 times return.

Keywords: ARIMA; LSTM; Gradient Boosting Regression; Upheaval-Oscillation-based Strategy; Deep Reinforcement Learning

Contents

Memorandum

To: Traders Interested in Bitcoin and Gold Trading

From: ****, ****, and ****

Subject: Novel Trading Strategies Based on Time Series Prediction

Date: February 22, 2022

We are writing to inform you that some novel strategies in the field of gold investment and bitcoin trading have been developed by our team. Primary tests using these strategies on historical data have yielded an astonishing performance of making a profit of a hundred times compared with the initial investment over a period of 5 years.

We are confident about the performance and universality of our strategies, since they are purely data-driven. Our models and strategies combine advanced techniques from both econometrics and computer science, such as time series decomposition and deep reinforcement learning. As potential investors in gold market or bitcoin market, you may find our models and strategies both fancy and reliable.

More specifically, we provide you with two categories of strategies: time series forecast based strategy, and reinforcement learning based strategy. The first strategy takes the past several dozen days' price as input, forecasts one one day ahead using a model of your choice (autoregressive integrated moving average, long short-term memory, or gradient boosting regression), and predicts the best investing strategy for the current day based on an upheaval-oscillation analysis. The second strategy is more straightforward: it takes your current portfolio structure and the day's market price as input, and directly returns an investment recommendation.

The models and strategies are currently under further training and polishing. Both time series forecast strategy and reinforcement learning strategy have proven themselves to be able to make more than a hundred times' worth of profit, but we believe given enough computation power, the later is capable of performing much better. We expect a more mature model with robust strategies that can be applied to various investment scenarios to be released fairly soon.

If you are interested in our current work, or have any further question about our models and strategies, please feel free to contact us through e-mail at xxxxxxxx@gmail.com.

1 Introduction

1.1 Background

Gold is arguably the most traditional and most popular investment. As one of the most precious metals, it has long been regarded as an equivalence for currency, and has been able to maintain a relatively stable value across history, especially during inflation.

Meanwhile, since World War II and the establishment of United State as the world's single most powerful economy with its Bretton Woods system, US dollars have come to be the new equivalence for gold. However, its exchange rate still varies in response to factors including but not limited to wars, pandemic, and the rise of other economical bodies.

Bitcoin, on the other hand, is one of the newest form of investment and has just made its way into the public view in the 2010s. It is a kind of peer-to-peer digital currency with a value that varied dramatically in the past decade, establishing itself as one of the speculators' new favorites in the process.

Many investors treat gold and bitcoin as two distinct investment methods. But they are all traded in a market-based way, which leads to a common scenario of trading both assets concurrently.

1.2 Problem Statement

In this problem, we are given the daily closing prices of gold and bitcoins from 9/11/2016 to 9/10/2021. It is our task to put forward a model that, given 1000 dollars at the starting date of the data, maximizes the total return in the end by yielding the best trading strategy for each day using only the price information up to that particular day.

The model should be subject to the restrictions that gold and bitcoins have a commission rate of 1% and 2% respectively, and that gold may only be traded on workday.

1.3 Problem Analysis

Investment strategy is a classical application of time-series prediction. With a perfect knowledge of the future pricing trends of both gold and bitcoin, it would be quite easy to determine a best investment strategy using dynamic programming, or a nearly as good one using greedy strategy. Lacking that foresight, however, we have to make compromises and predict the future based on the past.

To achieve that, we first perform a straightforward trend-cycle decomposition on the raw data to observe any latent seasonality and cycle information, and then apply three different models - autoregressive integrated moving average, long short-term memory, and gradient boosting regression - to execute one-step forecast based on the prices in the past days. Besides, we also propose a intrinsically different strategy - instead of predicting the prices of bitcoin and gold and making decisions accordingly, we apply reinforcement learning to directly predict investment strategies from the past information.

After developing and evaluating our models, we also prove in section 4.3 that there is no optimal strategy to this problem.

1.4 Assumptions

In order to simplify the problem and build an effective, pragmatic mathematical model, we make the following assumptions on the problem:

- Trading is done near the end of each day, so that all transaction prices are the given closing price of that day.
- The minimum trading amount for gold is one ounce, while that for bitcoin is 0.01 piece [1]. At some circumstances, the minimum trading unit may be ignored. Also, there is no maximum trading amount.
- Risk preference assumption: With the problem specifying the goal of pursuing the best return, we assume the trader can ignore all risks in order to find the best strategy.
- Traders' behavior will not influence the trend of gold price and bitcoin price.

2 Price Analysis and Prediction

2.1 General Price Analysis

2.1.1 Price Series Visualization

The prices of the bitcoin and the gold is plotted as scatters below. Each scatter plot is accompanied by the line plot obtained by applying a moving average window size of 7 (see Fig. 2.1.1, 2.1.1).

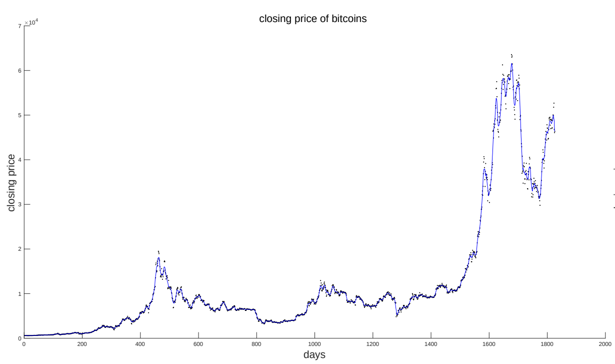

Figure 1: The closing price of bitcoins, with a moving average curve

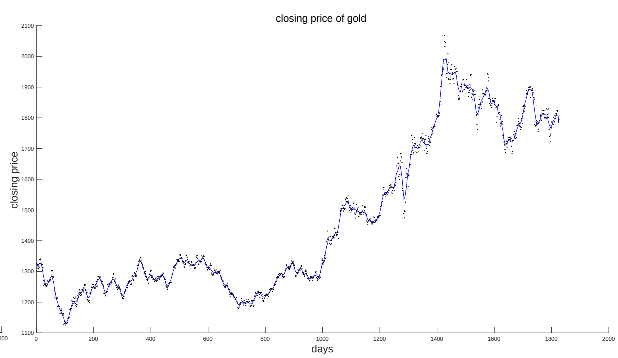

Figure 2: The closing price of gold, with a moving average curve

2.1.2 Correlation between Bitcoin Price and Gold Price

The Pearson correlation coefficient of two random variables X and Y is given by $\varrho_{X,Y} = \frac{Cov(X,Y)}{\sigma_X \sigma_Y}$, with $Cov()$ being the covariance and σ beging the standard deviation [10]. Since gold is traded only in certain days. One way to align the two price series is pad the gold price series with the rules as: 1. pad Saturday and holidays with the former day' value; 2. pad Sunday with the following day's value, which yields the correlation as 0.64936. Another way to align the two price series is to neglect the bitcoins' price in no-trading days, which yields the correlation coefficient as 0.64947.

2.1.3 Behaviour Analysis on Price Series

For missing values of gold prices, it can be seen that the days when no trading occurs are Saturdays and Sundays and one holiday.

For the macro trend of the gold prices, it can be seen that starting from around July 2019 to Sep. 2020, the gold price climb rapidly, which may be due to the depreciation of U.S. dollars since wealth preservation of the gold.

For the macro trend of the bitcoin prices, it can be seen that starting from around Jan. 2021 to May 2021, there is a gigantic price climb, and after that there is a great jump followed by a large climb. This may be related with the covid-19 pandemic which disturbs the world since bitcoins' prices are largely related with the stability of the world.

2.2 Extracting the Cycles

First, we estimate the cyclic patterns in gold and bitcoin pricing. We apply a moving average with window size 7 to both data to obtain the rough trends (Fig 2.1.1, 2.1.1). Then, after subtracting the trends from raw data, we apply autocorrelation function to the detrended data, the results of which are shown in Fig 2.2, 2.2.

In the AFC plot of bitcoin pricing, significant negtive values can be found at lags 2 and 3, suggesting a fluctuating cycle of 5 to 6 days. In the gold pricing data, on the other hand, a much stronger fortnight-long seasonal pattern can found.

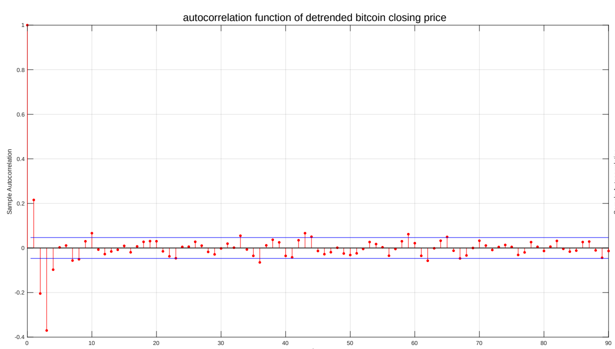

Figure 3: The autocorrelation function of detrended bitcoin closing price

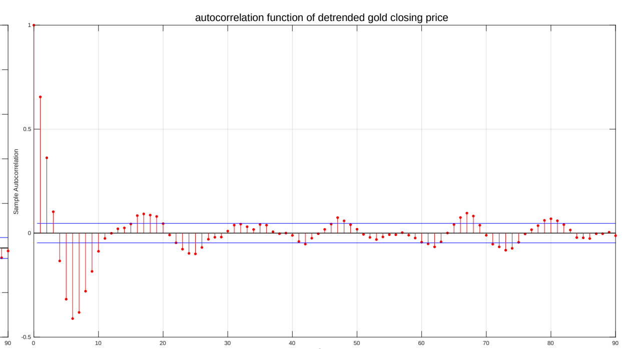

Figure 4: The autocorrelation function of detrended gold closing price

To confirm the patterns we discovered above, we use bandpower method to verify the seasonality of raw data. By calculating the average power for a input signal in the frequency domain, one will be able to verify whether the expected frequency is indeed the reciprocal of the signal's period.

According to the result derived from Fig 2.2, we set the test interval to be $[\frac{1}{18}, \frac{1}{14}]$. The calculating result shows that 31.8% of the total power accumulated in this domain. Compared with an average of 20% or so in other domains like $[\frac{1}{8}, \frac{1}{4}]$, we can verify that the period of gold price may have larger possibility lies in $14 - 18$ days.

Similarly, we apply the same method to bitcoin on interval $[\frac{1}{6}, \frac{1}{4}]$ and get a percentage of 20.6% as result. Compared with other domains with average percentage of 17%, we can conclude that the bitcoin price may have a period of $4-6$ days.

The bandpower method partially verifies the result derived using AFC plots. In addition, the bandpower verification shows relatively larger bandpower accumulation in gold price than bitcoin price. This indicates that the oscillation of gold price has a more significant periodicity than that of bitcoin.

2.3 Price Prediction

2.3.1 ARIMA Model

Based on the simple time series decomposition given in section 2.2, we now apply a more advanced time series model - ARIMA, or AutoRegressive Integrated Moving Average, to try dynamically predicting the prices of gold and bitcoin.

ARIMA combines the techniques of autoregression and moving average to model a time series. Autoregression forecasts the variable of interest using a combination of its past values, while moving average uses past forecast errors to perform this task. [6]

The basic mathematical formula for ARIMA is

$$\hat{y}_t = \mu + \phi_1 y_{t-1} + ... + \phi_p y_{t-p} + \theta_1 e_{t-1} + \theta_q e_{t-q}$$

where μ is a constant. ϕ is the coefficients for AR. θ is the coefficients for MA. p, q are the lags values. By learning these coefficients from historical data, this equation will give the prediction for the value $\hat{y}_t$ for point t.

According to literature published by Taieb, the Multi-Step-ahead Time Series Forecasting methods (1 step for our ARIMA model in specific) can be divided into several main streams:[13]

- **Recursive method.** In this method, one just look one step ahead based on model f constructed according to the historical value. If we set the moving average coefficient to be d, then the prediction model can be formalized by the equation

  $$\hat{y}_{t+1} = f(y_t, y_{t-1}, ..., y_{t-d+1})$$

  with

  $$t \in \{d, ..., N-1\}$$

  To be vivid, we can view this model as moving a window with length d over an array with length $N-1$. During each step, one takes the outcome of the model exerted on the historical data in the window as the predicted data for the next day. As time goes, new data becomes available and is put into the end of the window, while the oldest data is popped out of the window in the front.

- **Direct method.** Similar to the techniques used in Recursive method. The directed method also uses a sliding window. The major difference is the Direct method predicts each horizon independent of others. In other words, new models are constructed for each step as the sliding window moves forward. The equation for this method can be modified as

$$\hat{y}_{t+1} = \hat{f}(y_t, y_{t-1}...., y_{t-d+1})$$

where

$$t \in \{d, ..., N-1\}$$

and $\hat{f}$ is the model learned from the data in each step's sliding window.

One of the main shortcoming for this method is the time complexity. Learning a $\hat{f}$ from sliding window each step requires additional iterations on the data and may introduce a huge amount of extra running time to each iteration. Thus, the Direct method was not considered as an efficient method.

- **DirRec method.** The DirRec method improves the Direct method by using different horizon size for model training, which we will not introduce in great detail.
- **MIMO/DIRMO method.** These two methods (MIMO for Multi-Input Multi-output and DIRMO for Direct MIMO) have multiple outputs and thus be able to predict several days ahead. It is more suitable for Neural Networks and we will omit them here for ARIMA.

Inspired by these methods, we tried to construct multiple models with ARIMA, namely, the Recursive Method with one step looking ahead with different window sizes and the Direct Method with one step looking ahead. We also constructed a multi-step looking ahead model. The result is presented as follows:

1. **Recursive Method with One Step Looking Ahead (Window size = 30), referred as 30-day ARIMA later.** In this model, we select the historical gold price from day 0 to day 69, then moving the window ahead, trying to predict the gold price from day 70 to day 200. The moving average window size is set to be 30 days. Correspondingly, the result is shown in Fig 1.

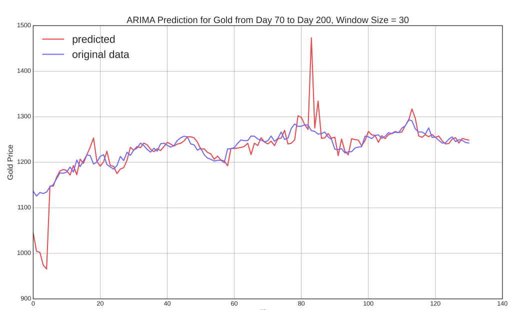

Figure 5: ARIMA Prediction for Gold Price from Day 70 to Day 200 with Moving Average Window Size 30

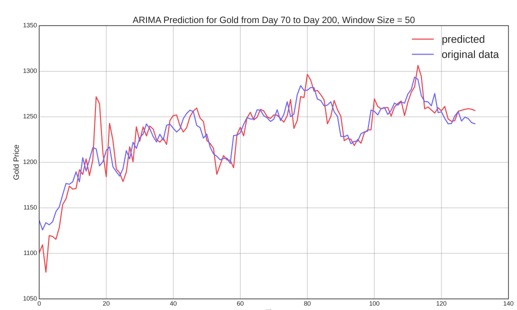

Figure 6: ARIMA Prediction for Gold Price from Day 70 to Day 200 with Moving Average Window Size 50

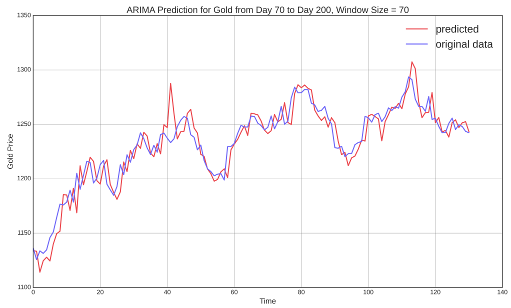

Figure 7: ARIMA Prediction for Gold Price from Day 70 to Day 200 with Moving Average Window Size 70

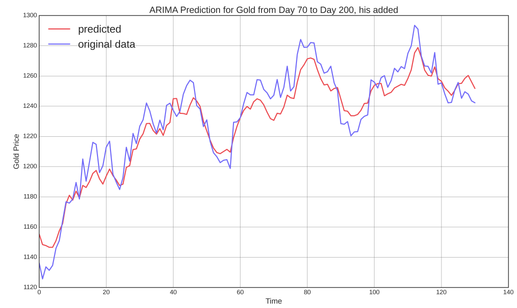

Figure 8: ARIMA Prediction for Gold Price from day 70 to Day 200, Balanced by Historical Value

We can spot that 30-day ARIMA successfully sketches the basic trend of the gold price. However, some uncommon noise presents in the first 5 days and between day 75 and day 85. This may be caused by the intense oscillation happened beforehand. The overall MSE for this model is 1271.40, which is acceptable compared with the absolute price of 1200.

2. **Recursive Method with One Step Looking Ahead (Window size = 50, referred as 50-day ARIMA later.** Compared with the 30-day ARIMA, we extend the moving average window size to 50 days, and try to predict the gold price in the same time interval, namely, day 70 to day 200. The result is shown in Fig 1.

   This model has significantly smaller noise scale compared with the 30-day ARIMA. The MSE is calculated to be 253.33, which is more feasible than the 30-day ARIMA.

3. **Recursive Method with One Step Looking Ahead (Window size = 70, referred as 70-day ARIMA later.** Again, we extend the moving average window size to 70 days, in order to gain a more precise prediction for day 70 to day 200. The result is shown in Fig 1.

   Comparing the Fig 1 and 1, we can conclude that the larger moving window size further reduces the noise ratio and almost sketches out the detailed moving pattern of the original value. The MSE for ARIMA-70 is further reduced to 159.80, which is quite satisfying.

Despite the fact that the prediction accuracy is improved by extending the moving window size, however, another shortcoming of these prediction models emerges. If one observes the models carefully, a significant lag between predicted data and original data can be seen. In other words, the trend of predicted data changes days after the real data made his change. Peaks and valleys in predicted data shows later than those in the original data.

With this problem discovered, we conduct literature reviews on relevant articles, looking for appropriate solutions. The reviews end up in the conclusion that this lag remains a critical issue in time series prediction problem and is inevitable due to following reasons:[14]

1. The lag issue has an intuitive cause, which can be illustrated by a simple example: One may want to use ARIMA to predict the value of day 10 $\hat{x}_{10}$ based on the historical values known

as $\{y_7, y_8, y_9\}$. Now, assume that the values of $\{y_7, y_8, y_9\}$ are strictly increasing and the real value of y_{10} is less than all of the previous values. Then it becomes obvious that the descent will only be shown in the prediction data later than $\hat{x}_{11}$, for they take y_{10} into consideration. While $\hat{x}_{10}$ does not aware of potential descent.

2. For the specific models of ARMIA, it is assumes the historical values are stationary. And all the predictions it makes are based on historical values. However, the gold market and Bitcoin market is ultra-dynamic with sharp changes in their prices. This feature makes ARIMA not as precise as expected.

One potential solution, as suggested by Zhang, the author of [14], is considering the hidden period lies in the historical values. To be specific, we need to find the potential oscillation period of gold price in our data set. If there is indeed a period pattern, then the historical data may be added to the moving average window or be used to balance the predicted data with certain weight distribution. In our model, we choose to balance the predicted data with historical value 16, 48, 68 days ago based on the result shown in Fig 2.2, with weight distribution 0.8, 0.1, 0.1 respectively. The result is shown in Fig 1.

Compared with the previous versions without historical values, the lag is slightly reduced. Furthermore, the prediction MSE is reduced to 108.65.

Based on this historical-considered ARIMA model, we can offer a relatively precise prediction for future gold price (MSE = 988.10, shown in Fig 2.3.1) and bitcoin price (MSE = 1508417.31, shown in Fig 2.3.1).

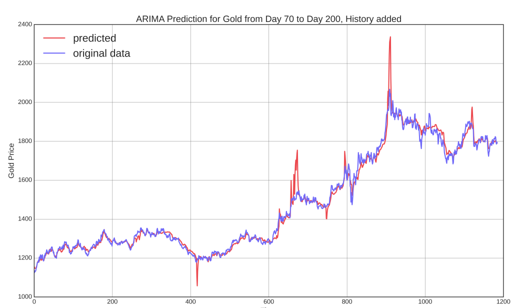

Figure 9: ARIMA Prediction for Overall Gold Price, Balanced by Historical Value

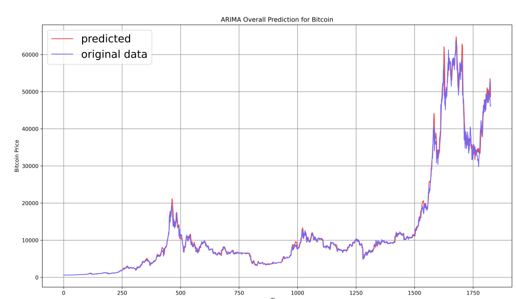

Figure 10: ARIMA Prediction for Overall Bitcoin Price, Balanced by Historical Value

One potential hidden problem for ARIMA is that, since matrix inversion is used in ARIMA algorithm, some values may not have invertible, which means values on these days cannot be predicted by ARIMA.

2.3.2 Long Short-Term Memory Model

Recurrent Neural Network, abbreviated as RNN, is a variety of neural network that specializes in dealing with sequential data. The fundamental mechanism of a RNN is to take the output of itself at one time step as its input for the next time step, along with the next sequential input data. Long Short-Term Memory, or LSTM, is an RNN variant designed to deal with the vanishing gradient problem. It introduces several gates - forget gate, update gate, and output gate - to allow useful information to be kept in the network for a long time, and thus being able to capture long-stretched dependencies in the data [11].

For each element at sequential time step t, each layer computes the following function:

$$
\begin{aligned}
i_t &= \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \\
f_t &= \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \\
g_t &= \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg} \\
o_t &= \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
$$

, where h_t is the hidden state at time t. c_t is the cell state at time t, x_t is the input at time t, i_t, f_t, g_t, o_t are the input, forget, cell, and output gates, respectively. σ is the sigmoid function and $\odot$ is the Hadamard product.

In a multilayer LSTM, the input $x_t^{(l)}$ of the l-th layer is the hidden state h_t^{l-1} of the previous layer multiplied by dropout δ_t^{l-1} where each $\delta_t^{(l-1)}$ is a Bernoulli random variable with probability predefined. All the other parameters are trainable.

In the following usage of prediction, the sequence length for predicting bitcoin prices set as 6 and 16 for predicting gold prices. The hidden vectors are of 50 dimensions, the number of layers is 3, and the dropout rate is chosen as 0.2. The result shows that the prediction is not optimistic (see Fig. 2.3.2 and 2.3.2). It may be because LSTM have a quite good memory on the short-term memory, which makes it to be sensitive to the noise in the price curve.

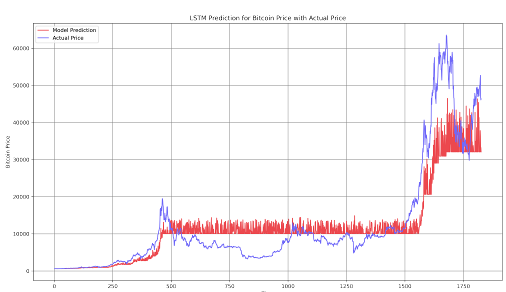

Figure 11: LSTM Prediction for Bitcoin Price with Actual Price

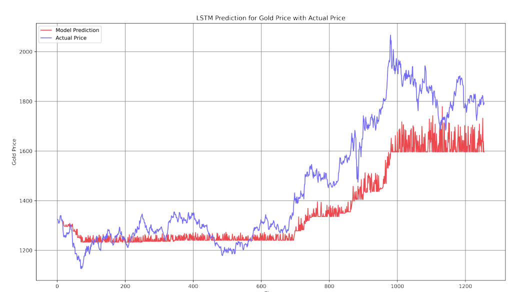

Figure 12: LSTM Prediction for Gold Price with Actual Price

2.3.3 Gradient Boosting Regression Model

Gradient boosting is a machine learning ensemble technique. It is introduced as an iterative functional gradient descent algorithm which optimizes a cost function over function space by iteratively choosing a function (weak hypothesis, especially decision trees) that points in the negative gradient direction [8, 9]. See Algorithm 1 for the fundamental training process [3, 4].

Algorithm 1: Generic Gradient Boosting Algorithm

Input: training set $(x_i, y_i)_{i=1}^n$, a differentiable loss function L(y, F(x)), number of iterations M

Output: $F_M(x)$

Initialize model with a constant value: $F_0(x) = arg\min_\gamma \sum_{i=1}^n L(y_i, \gamma)$.

for *m = 1 to M:* **do**

1. compute so-called pseudo-residuals:

$$r_{im} = -[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}]_{F(x)=F_{m-1}(x)} \text{ for i = 1, \ldots, n}$$

2. Fit a base learner (or weak learner, e.g. tree) closed under scaling $h_m(x)$ to pseudo-residuals, i.e. train it using the training set $(x_i, r_{im})_{i=1}^n$;

3. Compute multiple γ_m by solving the following one-dimensional optimization problem:

$$\gamma_m = arg\min_\gamma \sum_{i=1}^n L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i))$$

4. Update the model:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$$

return $F_M(x)$;

The loss function here is set as huber loss function, the learning rate is set as 0.3, and the weak learners as 500 decision trees with maximal depth as 3 [5]. The predicted results below show that the fitness is quite ideal which capture most of the trends and values.

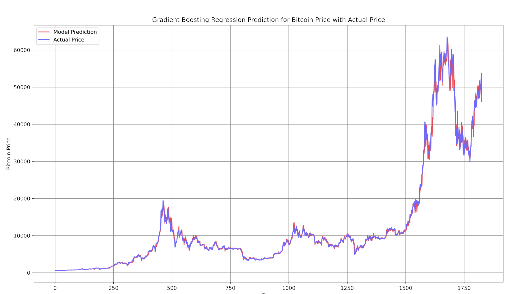

Figure 13: GBR Prediction for Bitcoin Price with Actual Price

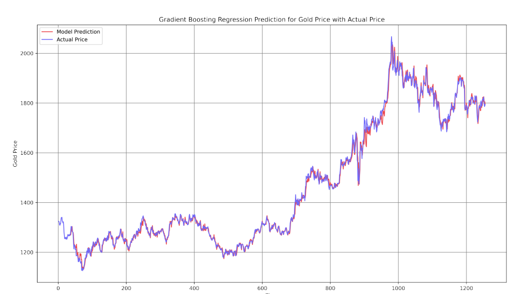

Figure 14: GBR Prediction for Gold Price with Actual Price

3 Investing Strategies

3.1 Preinformed Investment

With the benefit of hindsight, we first calculate a greedy solution using the future information on each day, as a reference for the strategies we are going to propose. As the price fluctuation of bitcoin is much more drastic than gold, we consider only investing in bitcoin.

The method is simple: if bitcoin value is going to rise more than 2% (which is the commission rate) the next day, use all cash to buy bitcoin; if it's going to depreciate more than 2%, sell all bitcoins in possession. This results in an astonishing amount of 3.5 billion dollars on September 10, 2021.

3.2 Naive Strategy

Utilizing the cycle information we uncovered in section 2.2, it is intuitive to develop a simple model: sell bitcoin and gold when their prices have been rising for several consecutive days, and buy them when their prices have been falling. More specifically, we stipulate that all the cash be used to buy bitcoin when it has been appreciating for six consecutive days, and sell all the possessed bitcoin in the reverse condition. When not trading for bitcoins, we buy gold if its price has been rising for four days in the past five days, and reversely sell it.

The rationale behind this strategy is that gold can only be traded on workdays, and may undergo major price fluctuation on weekends, so we only look at the past five days. Bitcoins, on the other hand, do not have trading gaps, and its cycle information may thus be fully utilized to develop the model.

Table 1 shows the deterministic result of this method. During the process a total of 9 transactions took place, including 4 buyings and 3 sellings of bitcoin, and 1 buying and 1 selling of gold. The transaction frequency is quite low, but we ended up with more then 13 hundred thousand dollars, more than a hundred times the principal.

Table 1: A demonstration of the simple trading strategy

Day	Asset ([C, G, B])	Total Worth
1	[1000, 0.00, 0.00]	1000
200	[6.85, 0.00, 1.30]	1356.13
400	[6.85, 0.00, 1.30]	7431.42
600	[41.68, 0.00, 2.27]	22181.79
800	[25.35, 0.00, 2.14]	10023.36
1000	[25.35, 0.00, 2.14]	17141.69
1200	[573.15, 14.00, 0.00]	21322.55
1400	[39.81, 0.00, 2.84]	26269.94
1600	[39.81, 0.00, 2.84]	86430.25
1800	[39.81, 0.00, 2.84]	133680.01
1826	[39.81, 0.00, 2.84]	131726.89

3.3 Upheaval-Oscillation-based Strategy

3.3.1 Strategy Motivation

One of the essential shortcoming of the naive strategy discussed above is the lack of consideration about the oscillations in bitcoin price and gold price. Indeed, considerable profit can be made during the consecutive rising period, while considerable loss can be avoided by escaping from the consecutive falling period. However, these consecutive changing periods are **rare**. Note that the naive strategy yields a total of 9 transactions are made, adding up to 54 days of consideration, leaving the rest 1800 days unmoved.

Observing this fact reminds us of the necessity of treating different periods with various strategies. And thus dividing the whole time domain into different periods becomes the critical task. We come up with a strategy and name it "Upheaval-Oscillation-based Strategy" (UO strategy).

3.3.2 Strategy Definition

The basic idea is analyzing the features of each point in its neighbor (e.g. in a 7 days period from started one week ago), and then divide the points into two classes, namely, the Upheaval Class and the Oscillation Class. The inclusion/exclusion criteria is the standard variance of the points within one period ago. The algorithm is shown in algo 2

Algorithm 2: Determine the Class for a Certain Point

```
Input: The Historical Value Array of Gold/Bitcoin Arr, a time t
Output: The Class of the Point at Time t
if We analysis gold then
    T = 6;
else
    T= 16;
σ = Standard Deviation of Arr[t − T : t];
μ = Average of Arr[t − T : t];
if Arr[t] > μ + σ Or Arr[t] < μ − σ then
    return Upheaval Class;
else
    return Oscillation Class;
```

With this definition, we can process the data and result in the following Figures:

For points in Upheaval class, we further divide them into increasing subclass and decreasing subclass. The strategy targets at selling the whole possessed gold/bitcoin at a relatively high price. Thus, we treat these points (marked as x_t) as follows:

1. For points in decreasing subclass, if we still have investment in that assets (gold/bitcoin), sell all of them and escape as soon as possible (at x_t if it is in trade time, x_{t+1} or x_{t+2} if it is not in trade time).

2. For points in increasing subclass, we put a sliding window in the points one period (6 days

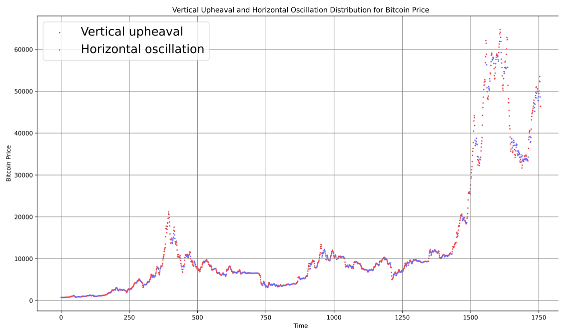

Figure 15: UO Analysis for Bitcoin Price

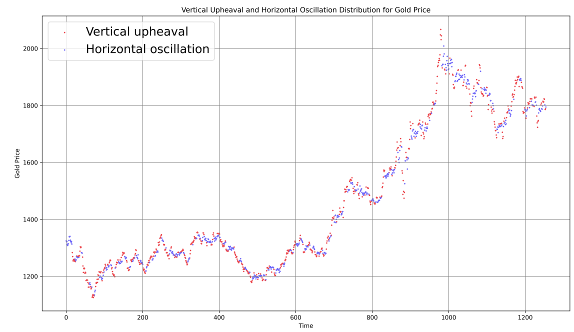

Figure 16: UO Analysis for Gold Price

for bitcoin and 16 days for gold) before, namely, from x_{t-T} to x_{t-1}. Then we choose two smallest value point in the sliding window, constructed a straight line through these two points. If we have predicted x_{t+1} to be lower than the line, we expect a decrease in the future, and we sell all the investment in that field at x_t. Else we continue to hold the investment. (see the following figures)

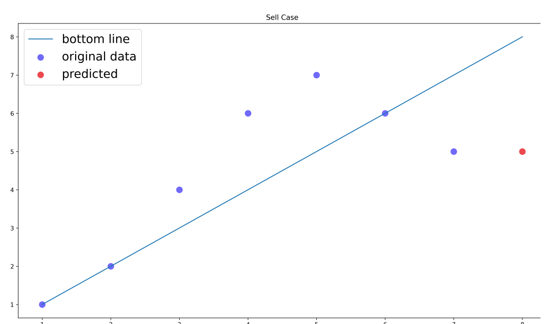

Figure 17: Sell Case

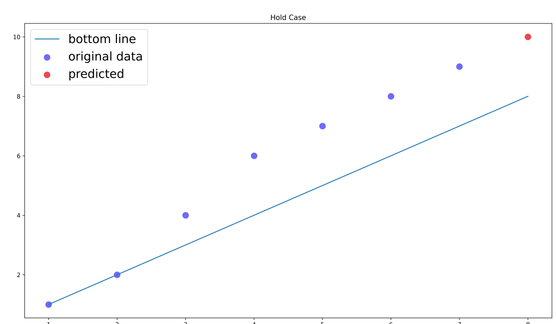

Figure 18: Hold Case

For points in Oscillation Class, we aim to buy at the valley points and sell at the top points. Define the price one day ago as x_{t-1} and the price predicted one day later as x_{t+1}. Also define the commission fee rate is α. If we spot one kind of assets is currently experiencing relations $x_{t-1} > x_t$ and $x_{t+1} > (1+\alpha)x_t$, which shows a signal of valley, we buy that assets. In the contrary, if we spot one kind of assets is current experiencing relations $x_{t-1} < x_t$ and $x_{t+1} < (1-\alpha)x_t$, which shows a signal of top, we sell that assets. Otherwise we hold the assets.

Now we further take both gold, bitcoin into consideration at the same time. The conflict takes place when we expect both of them are in valley. In this particular scenario, we calculate the absolute gain in the following day, choosing the higher scheme.

3.3.3 Strategy Result

We have constructed the Upheaval-Oscillation-based Strategy and used a sliding window size of 70 days. This means the first 70 days should be defined as observation period, when all three models (ARIMA, LSTM, GBR) learn the coefficients for prediction. After the observation period, models will give prediction for day 71 till the end based on historical data independently. We collect a weighted prediction value as the final prediction value for each day. Namely

$$\hat{x}_t = \alpha \hat{x}_{ARIMA,t} + \beta \hat{x}_{LSTM,t} + \gamma \hat{x}_{GBR,t}$$

where $\alpha + \beta + \gamma = 1$. We have tested a bunch of coefficients based on the performance results discussed before, with $\alpha = 0.9, \beta = 0.09, \gamma = 0.01$ yielding the best result, which is shown in the following Fig. The detailed trading strategy can also been found in that picture. The final value of

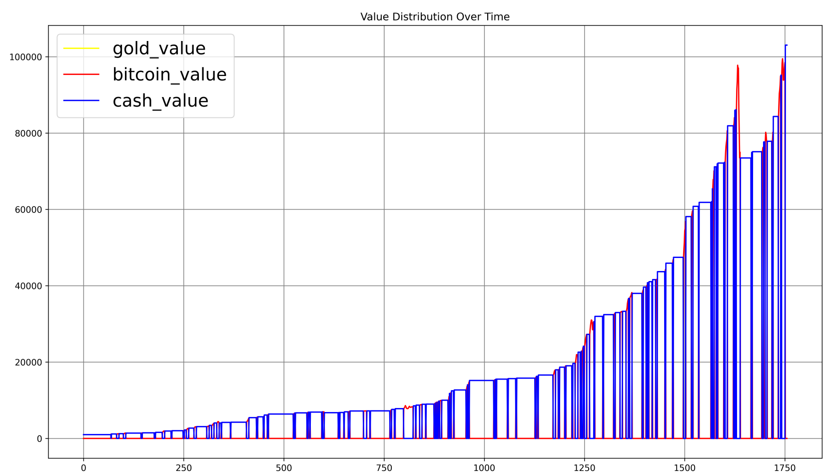

Figure 19: Value Distribution Over Time

this model is 114030.00 USD.

Indeed, the maximum return is relatively smaller than that of naive strategy, but the naive strategy has high randomness. One may not ensure the naive strategy can always produce a high quality policy. In the comparison, the robustness of this model guarantees a relatively high return expectation $E(X)$, which may bemoreessential for traders.

3.4 Reinforcement Learning Strategy

Lastly, we propose a intrinsically different model for investment strategy - using reinforcement learning (RL) to directly map from the current observation to an investing action. As can be seen from sections 3.1 and 3.2, gold does not play much a role in the final return compared to the theatrical fluctuation of bitcoin's price, so in this section we only consider investing in bitcoin.

As one of the three major machine learning paradigms, reinforcement learning is somewhat a combination of supervised learning and unsupervised learning - it does not have a true label corresponding to its output for every input example, but receives a possibly delayed reward from the environment for its action. A major difference in RL from the other two machine learning paradigms is that it does not assume identical and independent distribution between data examples, which makes it particularly suitable for problems involving time series.

The framework of our RL algorithm is shown in algorithm 3. We define the state space as a three-dimensional space, with the first two dimensions being amount of cash and number of bitcoins we possess, and the third one the bitcoin's price of a particular day. The action space is real number between 1 and -1. A positive action stands for using a certain percentage of cash to buy bitcoin, while a negative action means selling a percentage of bitcoin.

As for the transition probability, the first two dimensions of our state space is completely deterministic given an action, while the third space is completely independent of actions. Given this characteristic, we use naive Bayes method to model an approximate transition probability distribution of bitcoin prices, and feed it to the reinforcement learning algorithm as input.

Algorithm 3: Fitted Value Iteration

Input: state space $\mathbf{S}$, action space $\mathbf{A}$, transition probability $\mathbf{P}$, $EPOCH$, m, k
Output: θ

Initialize θ randomly;
for *epoch in range*($EPOCH$) **do**
    sample $\mathbf{S} = \{S^{(1)}, S^{(2)}, \cdots, S^{(m)}\}$ randomly;
    for *each* $S^{(i)}$ **do**
        for *each* $a \in \mathbf{A}$ **do**
            sample $\{S'_1, S'_2, \cdots, S'_k\} \sim P_{S^{(i)}}$;
            $q(a) = \frac{1}{k}\sum_{j=1}^{k}\left[R(S^{(i)}) + \gamma\theta^T\phi(S'_j)\right]$;
        $y^{(i)} = \max_a q(a)$;
    $\theta = \arg\min_\theta \frac{1}{2}\sum_{i=1}^{k}\left[\theta^T\phi(S^{(i)} - y(i))^2\right]$;
return θ;

As this algorithm involves random sampling when modeling the transition probability, it does not produce a deterministic result. We ran it for multiple times, and it is able to yield an average total value of about 55 thousand dollars at the end of time horizon. This is probably due to the lack of expressivity of this model - a simple linear transformation can be hardly expected to capture the inscrutable variation of stock market.

To address this shortcoming, we also tried deep reinforcement learning, where the simple linear transformation is replaced with a small neural network with one hidden layer. The framework of algorithm 3 remains the same, with the only difference being that instead of solving for θ algebraically, we leave the heavy work to back propagation, and gradient descent. Unfortunately, due to the lack of computation power, we were unable to obtain a converged model, and the one we did get after several dozens of epochs of training only performs a bit better than the simple RL version - about 96 thousand dollars in the end. A demonstrative investment is shown in Fig 20.

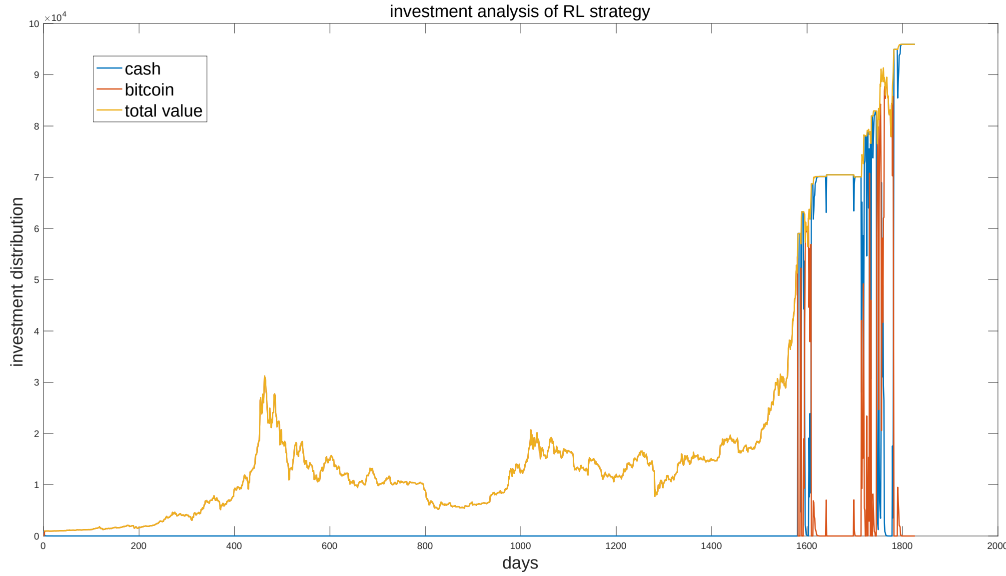

Figure 20: An investment strategy produced by the deep reinforcement learning model

4 Strategy Verification

4.1 Sensitivity Analysis on Transaction Commission

4.1.1 Naive Strategy

As the naive strategy does not take transaction costs into consideration when making decisions, it is fairly robust to changes of commission rates. This is vindicated by our experiments: when there is no transaction cost, the model still produces the same 9 transactions, only this time with a total return of 15.5 hundred thousand dollars; when the commission rates for both bitcoin and gold are set to 5%, it's still the same transactions, but with only 99 thousand dollars worth of return in the end. The influence of commission rates over naive strategy's total return is shown intuitively in 4.1.1. It is apparent that the actions produced by this strategy is not affected by commission rates.

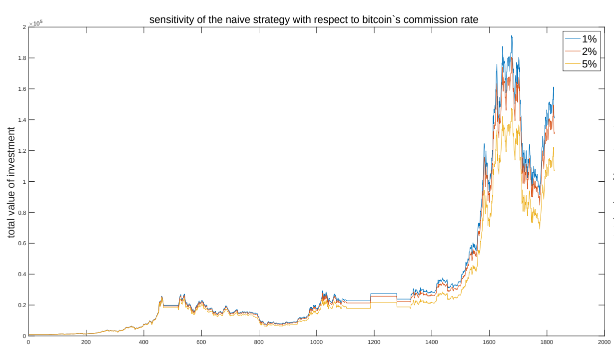

Figure 21: Naive strategy's sensitivity to bitcoin's commission rate

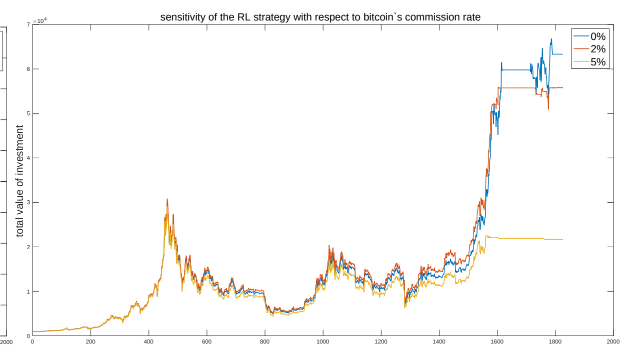

Figure 22: Reinforcement learning strategy's sensitivity to bitcoin's commission rate

4.1.2 Upheaval-Oscillation-based Strategy

To test the sensitive of the Upheaval-Oscillation-based Strategy, we set three different transaction fees to be 1%, 5% and 20%. Following Figs presents the results correspondingly

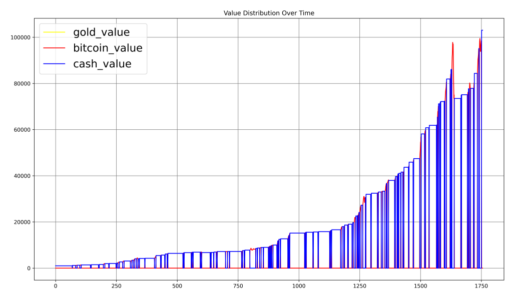

Figure 23: The Sensitivity Test Result for Transaction Fee Rate $\alpha = 1\%$

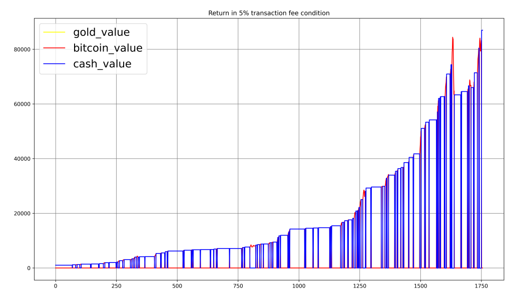

Figure 24: The Sensitivity Test Result for Transaction Fee Rate $\alpha = 5\%$

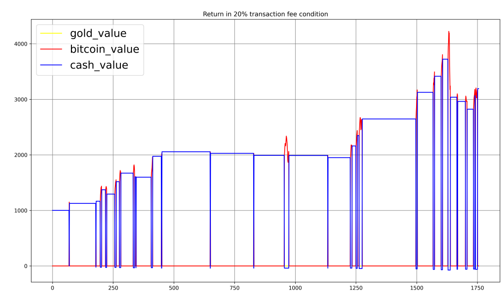

Figure 25: The Sensitivity Test Result for Transaction Fee Rate $\alpha = 20\%$

4.1.3 Reinforcement Learning Strategy

We mentioned in section 3.4 that the RL strategy is not entirely deterministic. However, by averaging multiple sample runs, we conducted sensitivity test on this strategy as well, with the result shown in Fig 4.1.1. It can be seen that this strategy is still quite robust to commission rate changes, although inconsistencies may occur when the bitcoin price is changing too rapidly.

As for the deep reinforcement strategy, as is mentioned earlier, we do not have the time to train it with different commission rates due to limited computation power, so its sensitivity test is ommited.

4.2 Ablation Analysis on Prediction Mechanism

Since in our Upheaval-Oscillation-based strategy, the predicted price is assumed to be the actual price in the next time step for decision. So it's vital to conduct a ablation analysis to show the necessity of the voting method.

4.2.1 Removal of ARIMA Prediction

The result after removing the ARIMA can be seen in Fig. We can clearly spot the result of moving ARIMA from the weighted sum of prediction is disastrous. The return is reduced by about 10 folds to about 7000 USD. This means the ARIMA plays a major role in prediction.

4.2.2 Removal of LSTM Prediction

The result after moving LSTM from the the combined prediction can be seen in Fig. We can rarely see difference in the return value before and after the removal of LSTM. This is due to the poor prediction precision discussed before.

4.2.3 Removal of Gradient Boosting Regressor Prediction

The result of removing GBR from prediction model is shown in Fig The removal caused the return value cut to one-half. Although the difference is not as huge as the removal of ARIMA, it is still significant. Thus, the prediction precision of GBR is also feasible.

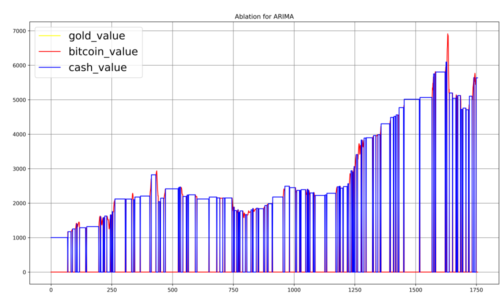

Figure 26: Ablation for ARIMA

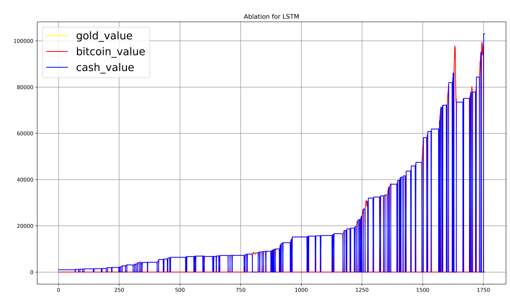

Figure 27: Ablation for LSTM

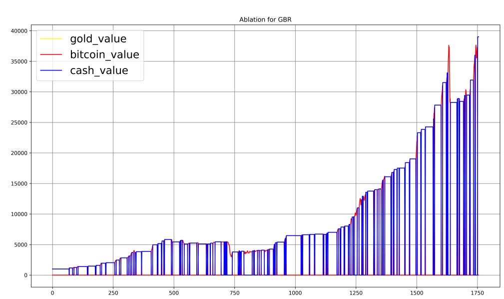

Figure 28: Ablation for GBR

4.3 Optimal Strategy Analysis

Below, the "best strategy" is defined as a strategy that gives the maximal amount of value (evaluated in U.S. dollars) corresponding to the problem's requirement.

First of all, we prove that obtaining the maximal total value at the end of the sequence is equivalent to obtaining the maximal total value at every day in that sequence. For necessity, it's trivial. For sufficiency, suppose that in one intermediate day of a strategy S, the total value is less than that of the corresponding day in the optimal strategy, and the total value of such a strategy in the end is no less than the total value in the optimal strategy. Then we can adjust the optimal strategy in the way that in that day, the same value as strategy S is operated in the same way as strategy S in the following days, while keep the extra value in cash. Then in the end, we will have more value than the optimal strategy, which contradicts with the definition of "optimal" and hence completes the first proof.

Secondly, we prove that obtaining the maximal value every day is equivalent to knowing the gold and bitcoin's up and down trend and (and growth rates if they both increase). For necessity, if known those information, we can just choose the asset portfolio that will bring the maximal growth rate, which yields the maximal value every day. For sufficiency, suppose we don't know the gold and bitcoin's up and down trend and (and growth rates if they both increase), then tomorrow's situation is uncertain, which may cause any certain strategy to fail to achieve the maximal value in the following day.

In conclusion, by transitivity, we have proven that the optimal strategy in our definition is equivalent to knowing the gold and bitcoin's up and down trend and (and growth rates if they both increase). However, with references to the current findings, we argue that these information is not completely predictable.

The market price of the bitcoin and the gold is partially unpredictable [12, 2, 7]. The market price is the net present value from all the investors' evaluations on the assets' future utility. However, the future utility of the good is affected by a variety of factors, including macro factors such as supply and demand relation, the world economy situation, and the stability of the world, as well as micro factors such as investors' irrational behaviors, both of these two factors contains a great deal of randomness [12, 2, 7]. The price of the assets is partially unpredictable due to the difficulty of considering comprehensively or make an accurate prediction on all the influencing factors and their randomness. So, given only the two price curves, which carry limited information about the future and thus leads to a bad prediction. Therefore, we conclude the unpredictability of the gold and bitcoin's up and down trend and (and growth rates if they both increase), and further conclude that any strategy heavily based on the predictions on the future prices is not optimal and not even robust.

Additionally, in reality, the effect of a strategy is not stationary and is related with other players' strategies. As discussed above, the price is affected by the supply and demand relation, which is in turn defined by all the players in the market. However, every player competes with his/her own strategy. So the performance of your strategy is largely determined by the dynamic environment defined by all the other strategies.

As a supplement to the whole problem setting, if loosening the definition of the "optimal

strategy", the strategy space is larger and the complexities will be multipled. In the current problem setting, we assume the investors only care about the maximal return when designing strategies, which is similar to maximizing the expectation. However, if considering the risk of strategies, we need also to approximate investors' risk utility function, which leads to more complicated and uncertain strategy space.

Below we consider our upheaval-oscillation-based strategy as the suboptimal strategy and here is our evidence. First, in terms of strength, it gives higher returns than the reinforcement learning strategy and it's as data-driven as the reinforcement learning algorithm in contrast to the naive strategy's luckiness-dependence. So it has better generalization ability than the naive strategy. In terms of weakness, it is only largely affected by extremely high transaction commissions which outperforms the deep reinforcement learning algorithm. So to sum up, the upheaval-oscillation-based strategy is the "best" strategy we provide even though the other strategies are acceptable.

5 Conclusions

In this article, we apply general data analysis on the price curves of the bitcoins and gold. And three strategies, Naive Strategy, Upheaval-Oscillation-based Strategty, and Deep Reinforcement Learning Strategy are developed based on the general data properties and three prediction models, ARIMA model, LSTM model, and Gradient Boosting Model. In the prediction, ARIMA and GBR are shown to give the best prediction accuracy. With the sensitivity analysis on the transaction commission rate, we have shown that deep RL strategy is sensitive to the transaction commission rate when the price of bitcoins increases dramatically. Also, uphaval-oscillation strategy will be greatly affected by transaction commission rates with large values, while the naive strategy sacrifies generalization ability for the robustness on the change of transaction commission rates. Additionally, the ablation analysis is conducted, leading to the fact that ARIMA and GBR models are more important than the LSTM model. In terms of the return, naive strategy gives the highest return while the reinforcement learning gives the worst return.

References

[1] ICE Benchmark Administration. Committee role, procedures and terms of reference, 2021.

[2] Eugene F Fama. Efficient capital markets: A review of theory and empirical work. *The journal of Finance*, 25(2):383–417, 1970.

[3] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.

[4] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[5] Peter J. Huber. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73 – 101, 1964.

[6] Robin John Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. OTexts, Australia, 2nd edition, 2018.

[7] Benoit B Mandelbrot. The variation of certain speculative prices. In *Fractals and scaling in finance*, pages 371–418. Springer, 1997.

[8] Llew Mason, Jonathan Baxter, Peter Bartlett, and Marcus Frean. Boosting algorithms as gradient descent. *Advances in neural information processing systems*, 12, 1999.

[9] Llew Mason, Jonathan Baxter, Peter Bartlett, and Marcus Frean. Boosting algorithms as gradient descent in function space, 1999.

[10] K Pearson. Notes on regression and inheritance in the case of two parents proceedings of the royal society of london, 58, 240-242, 1895.

[11] Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *arXiv preprint arXiv:1402.1128*, 2014.

[12] Paul A Samuelson. Proof that properly anticipated prices fluctuate randomly. In *The world scientific handbook of futures markets*, pages 25–38. World Scientific, 2016.

[13] Souhaib Ben Taieb, Gianluca Bontempi, Amir F Atiya, and Antti Sorjamaa. A review and comparison of strategies for multi-step ahead time series forecasting based on the nn5 forecasting competition. *Expert systems with applications*, 39(8):7067–7083, 2012.

[14] Zhengchao Zhang, Meng Li, Xi Lin, Yinhai Wang, and Fang He. Multistep speed prediction on traffic networks: A deep learning approach considering spatio-temporal dependencies. *Transportation Research Part C: Emerging Technologies*, 105:297–322, 2019.