Problem Chosen	2022	Team Control Number
C	MCM/ICM Summary Sheet	2221805

Quantitative Trading: Stirring up the Storm

Summary

Volatility and unpredictability have long been two of the most enthralling aspects of financial markets. A rational trading approach can assist traders in generating consistent profits with a low risk. In this competition, we have been asked to develop just this form of trading strategy. The specific tasks include three questions.

As for question 1, we divide the assignment into two submodules: future price predicting and investment decision. The price forecasting section begins by pre-processing the historical price stream. It consists of the following four steps: **removing nulls**, **first-order differencing**, **reconstructing dimensions**, and **normalization**. Then, we anticipate future prices using LSTM, SVM, RandomForest, XGBoost, LightGBM, and LinearRegression, and evaluate their error levels using MSE, etc. Based on a thorough evaluation, we determine that the **LSTM** performs better and hence choose it as a predictor for further decisions.

On the investment decision part, our core notion is to simplify the investment decision as a **optimization issue**. First, we must evaluate the best time to enter the market. **E-ratio** is a sensitive indication we use to gauge market entry. When it exceeds our criteria, the current time is regarded as eligible for investment. Then we create an **investing factor** by comparing the current and prospective portfolio returns and Sharpe ratios. It is positively correlated with the rate of return and negatively correlated with risk. That is, to construct a portfolio for tomorrow that maximizes the investment component while guaranteeing total asset growth. We use Particle Swarm Optimization (PSO) to generate the daily optimal portfolio. After a 5-year investment period, we can achieve a 129.6% average yearly return and a total income of $108387.309.

To answer question 2, we evaluate our strategy's performance using three metrics: **annualized rate of return**, **Sharpe Ratio**, and **maximum retracement**. Our investing method outperforms many established quantitative trading models in terms of these metrics, making it a viable optimal option.

For question 3, we undertake a **sensitivity analysis** by altering the gold and bitcoin commission ratios and observing the trend in total assets and the respective percentages of cash, gold, and bitcoin in daily portfolio holdings. We find that the percentage of holdings increases when the asset's commission decreases, and vice versa. This follows basic market principles. In addition, adjusting the gold and bitcoin commission ratios has no significant effect on the overall asset trend or value, indicating that our model is not sensitive to commission changes.

Lastly, we summarize our models, strategies and results and present them in a memo to traders with the aim that this quantitative trading system will deliver stable and sustainable returns.

Keywords: Quantitative Trading; LSTM; Trend-following strategy

Contents

1 Introduction

1.1 Background

The financial business exists to maximize revenues by distributing resources efficiently. Market traders routinely acquire and sell volatile assets to secure investments with the potential for profit maximization. Typically, traders made decisions about each transaction based on their opinions of the market. However, complexity and dynamics are increasing drastically in today's financial markets, where traditional assets (like gold) coexist with digital ones (like Bitcoin). The traditional trading strategy is often inefficient and prone to big losses due to irrational conduct[1].

Unlike traditional approaches, quantitative trading is an automated method of profit-mining from historical market data that follows a predetermined trading strategy or logic. It outperforms human traders in terms of stability, speed, and reliability[2]. Thus, quantitative trading algorithms can aid investors in achieving desired returns across a variety of portfolios.

1.2 Problem Restatement

At the request of the trader mentioned in the problem, We need to develop a model that uses only the past stream of daily prices to date to determine daily transactions. Specifically, the investment is confined to a gold and bitcoin portfolio with a \$1000 beginning capital for a five-year commitment. Each time a transaction is made, a commission must be paid. Our goal is to reap the greatest possible profit at the cut-off day.

In order to accomplish the requirement, our specific works are as follows:

- Based on the previous price data, predict the price of gold or bitcoin for the day.
- Create a trading strategy that maximizes the 5-year benefit based on the anticipated prices determined in the previous step and the assets currently own.
- Confirm the quality and superiority of our model by comparing our results to those of other methods.
- Perform model sensitivity analysis by varying the commission amount, and investigate the influence on trading strategies and final returns of this adjustment.
- Write a memo to the trader summarizing our approach, model, and outcomes.

1.3 Our Work

The overview of our work is presented in Fig.1 , including price forecast,trading strategy,result analysis.

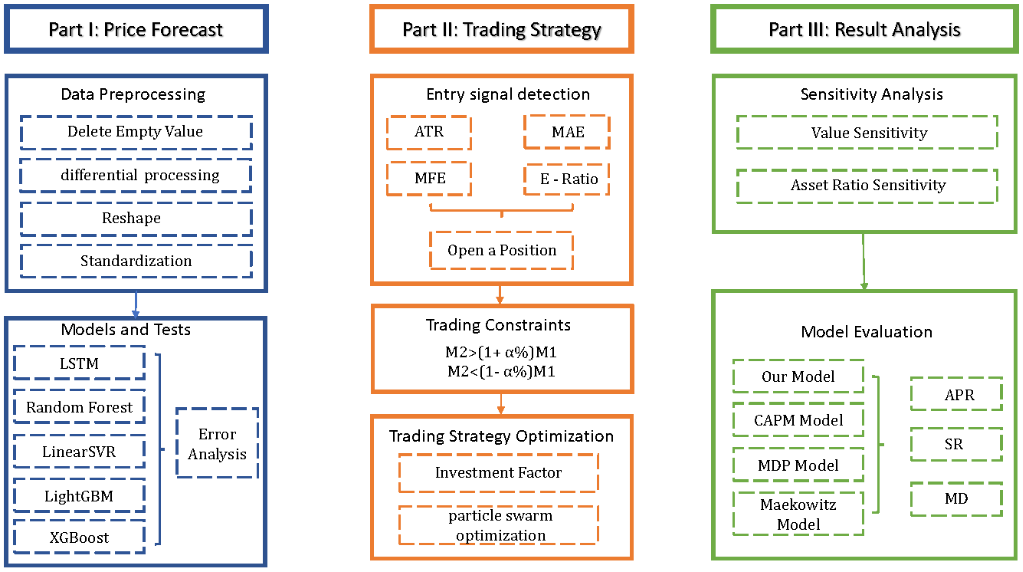


Figure 1: Overview of ourworks

2 Assumptions and Justifications

To simplify the problem, we make the following assumptions:

1. **Assume that the gold and bitcoin price series are fully determined by a time-series connection, disregarding influences such as public opinion and government policy.** Whereas the real price series exhibits a time series relationship, it is not entirely dependent on it and is modified by a variety of factors. However, as stated in the question, we can only forecast future pricing using historical data. That is, in this case, a forecasting model that ignores other variables is acceptable.

2. **Prudent investor assumption: when making a deal, avoid spending all available cash or selling an asset (e.g. gold, bitcoin) completely short.** In this case, because both buying and selling entail fees, frequent trading of big quantities will extract a significant amount of fees, masking a decline in investment returns. A limit-based trading strategy enables investors to mitigate the negative impact of fees on the growth of the total assets.

3. **The optimal strategy indicates the best strategy after evaluating the return and risk, not the strategy that takes the extreme value in one aspect.** In order to make profit in a sustainable way, the strategy must have a particular amount of risk resistance while maintaining a certain rate of return

3 Model 1: Price Forecast

3.1 Long and Short-Term Memory Model

Accurate price forecasting is the first step to developing a successful trading strategy. With the help of price forecasting, we can see where prices are headed in the near future and make adjustments to our trading strategy correspondingly. In order to fulfill the accuracy standards, this forecasting algorithm needs to mix short-term historical data with long-term patterns in the market. For this reason, we rely on Long and Short-Term Memory (LSTM) to make our predictions[3].

The LSTM algorithm is a variant of recurrent neural network (RNN). Due to its capacity to learn long-term dependant information, it is commonly employed in time series analysis such as pricing. For the single-day price prediction challenge, our core idea is to use the price series from a prior period (but not all of them) as the training dataset. Prior to running the LSTM model, the original data must be pre-processed. Thus, our entire model may be separated into three distinct components: pre-processing, LSTM, Output. Fig.2 is a schematic showing the exact structure.

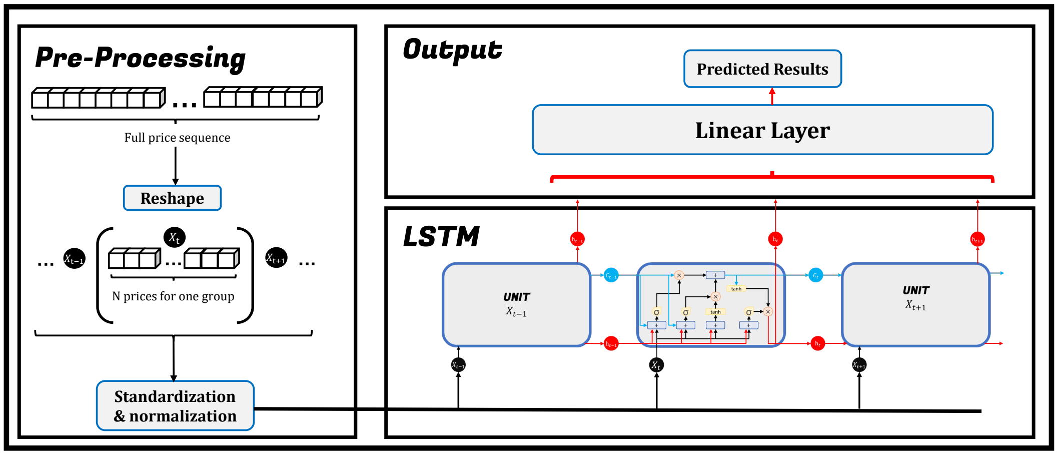


Figure 2: Process of price forecasting

1. **Data Pre-processing**

When generating the data needed to predict the price of a particular day, one point to keep in mind is that the dataset we utilize is obtained only from prices before that day; this portion of the price series is referred to as the original price series. Before the preprocessing operation, we remove the null values from the original price sequence. In the preprocessing section, we first reshape the original price series to conform to the requirements for training data format. The initial one-dimensional price series is restructured into a dataset with an N-dimensional vector X serving as the feature and a scalar Y serving as the label. X indicates the price data we selected for the previous N days, and Y represents the actual price of the day as a label.

Following that, we apply the necessary normalization to the data to keep the various variables within a comparable range and to accelerate the model's convergence:

$$x_i = \frac{x_i - \mu_i}{\sigma_i} \tag{1}$$

Table 1: **Symbols used in the pre-processing process**

Notation	Definition
N	The length of the price series used for forecasting
X	The price sequence of the previous N days, used as a N-dimensional feature
Y	The real price on a particular day, used as a label in the dataset
x_i	An element in the i^{th} N-dimensional feature
μ_i	The mean of the i^{th} feature
σ_i	The standard deviation of the i^{th} feature

2. **LSTM Model**

Fig.3 depicts the workflow of an LSTM unit.

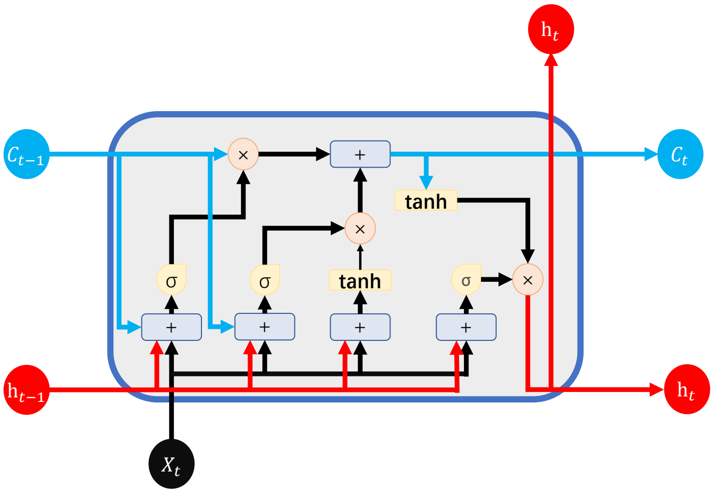


Figure 3: Workflow of a single LSTM unit

A series of actions will be carried out in this cell in order to produce the ultimate result:

$$\begin{cases} i_t = \sigma\left(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}\right) \ , \\ f_t = \sigma\left(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}\right) \ , \\ g_t = \tanh\left(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}\right), \\ o_t = \sigma\left(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}\right) \ , \\ c_t = f_t * c_{t-1} + i_t * g_t, \\ h_t = o_t * \tanh\left(c_t\right), \end{cases} \tag{2}$$

The notations for equations above is shown in Table 2:

Table 2: **Symbols for a single LSTM cell**

Notation	Definition
X_t	The price series for the N days preceding the time t, used as the input vector for LSTM
c	Memory cell, acting as an accumulator of the state information
h	Hidden state, containing encoded information for the sequence flow
i	Input gate, controling whether the cell can acquire new input information
f	Forget gate, controlling whether the past cell status could be forgotten in this process
o	Output gate, controlling whether the latest cell output will be propagated to the final state
W	The weight matrix of each node
b	The bias of each node
σ	The sigmoid function
tanh	The hyperbolic tangent function

3. **Output**

The output module is implemented here using a single linear layer. It accepts the LSTM's hidden layer state as input and produces the associated forecasted price.

3.2 Prediction Results

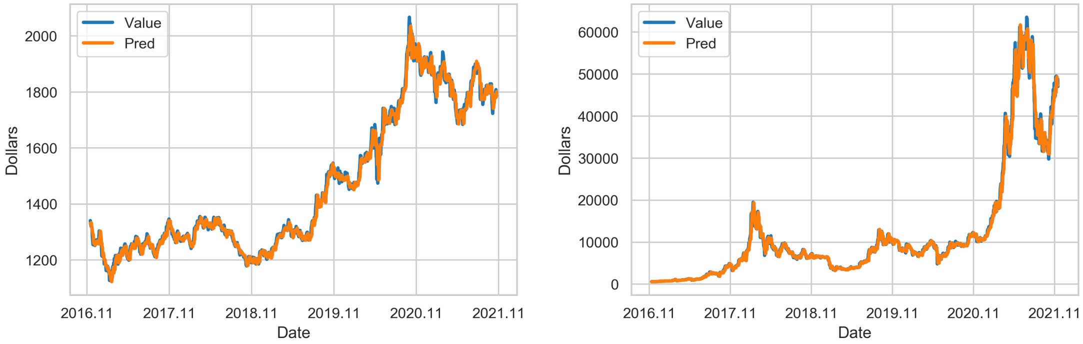
1. **Experiment setup**

During the model's training process, the following settings are used

- N, the length of the input price sequence (input feature vector), is set to 5.
- The experiments are run for 30 epochs with the learning rate 1e-2 and Adam optimizer.
- Each minibatch contains 10 5-dimensional sequences.
- Every 30 days, the model is retrained to ensure it remains up to date.

2. **Result**

After training, we fed the complete 5-year price series (gold and bitcoin) into the model for prediction and obtained daily price predictions, which are compared to the real price series in Fig.4



(a) Gold price forecasting by LSTM model (b) Bitcoin price forecasting by LSTM model

Figure 4: LSTM prediction results

As illustrated in Fig.4, forecasted prices can match and fit both gold and bitcoin's genuine values in terms of long-term trends and short-term variations. But there is a certain drawback to using this method for predicting early pricing (e.g., the first 30 days). It is worth noting that the sparse training data used to predict upfront prices results in somewhat unstable upfront prediction results. As a result, when developing the subsequent trading strategy, we will account for this effect and adopt a more cautious investment strategy in the early stages to mitigate its influence.

3.3 Price Forecasting Model Evaluation

In addition to the LSTM model, we applied several machine learning regression algorithms, including RanomForestRegressor, and eXtreme Gradient Boosting(xgboost) to forecast single-day price movements. We use metrics including Mean Square Error(MSE), Mean Absolute Error(RMSE), Root Mean Square Error(MAE), Mean Absolute Percentage Error (MAPE), coefficient of determination(R^2)to measure the accuracy of these forecasting models, defined as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2 \tag{3}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2} \tag{4}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i - y_i\right| \tag{5}$$

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \tag{6}$$

$$R^2 = 1 - \frac{\sum_i \left(\hat{y}_i - y_i\right)^2}{\sum_i \left(\overline{y_i} - y_i\right)^2} \tag{7}$$

where $\hat{y}_i$ represents the predicted value, y_i represents the true value, n represents the number of samples. These indicators measure the current error size, i.e. the smaller the value of the indicator, the higher the accuracy of the prediction model and the more reliable the prediction results. By substituting the anticipated values from the previously mentioned models for the genuine values, the following summary table of findings is generated:

Table 3: **Error indicators for each model in predicting gold prices**

Model	MSE	MAE	R^2	RMSE	MAPE
LSTM	**1534106.364**	**644.153**	0.992	**1238.590**	5.163
LinearSVR	1896091.426	820.490	**0.994**	1376.986	**2.770**
RanomForestRegressor	357287162.316	12441.370	-0.113	18902.041	28.602
LightGBMRegressor	399364281.003	13310.400	-0.244	19984.100	30.865
LinearRegression	1960022.9	853.448	0.993	1400.008	2.897
XGBoost	400260220.0	13338.557	-0.246	20006.504	31.107

Table 4: **Error indicators for each model in predicting bitcoin prices**

Model	MSE	MAE	R^2	RMSE	MAPE
LSTM	**407.601**	**14.450**	**0.952**	**20.189**	**0.800**
LinearSVR	6877.521	17.101	0.889	82.930	1.107
RanomForestRegressor	45544.525	195.059	-4.293	213.411	10.555
LightGBMRegressor	39998.423	180.965	-3.649	199.99	9.780
LinearRegression	409.262	14.622	0.952	20.230	0.809
XGBoost	61526.793	230.843	-6.151	248.045	12.522

The bolded sections of the table represent the higher performing metrics, and you can see that the LSTM outperforms both gold and bitcoin prediction. So the aforesaid model will be used with the LSTM model in the subsequent trading strategy building. Due to the image scale's influence, the actual performance difference between different models is intuitively deduced from the graph, and we will evaluate these models using quantitative indicators throughout the subsequent actual trading process.

4 Model 2: Quantitative Trading

4.1 Model Overview

Quantitative trading is a type of trading approach in which a large amount of investment-related data is used as a sample to develop quantitative mathematical models and formulas, estimate the probability of various market movements, and programmatically execute buy and sell orders to achieve investment[4].

Generally, quantitative trading strategies can be classified as the trending strategy and the market neutral strategy. The trend-following strategy is a very systematic approach with very specific rules: **purchase when an entry signal occurs and sell when an exit signal appears**. Price is the most important factor in a trend following strategy, while asset management and risk control are also very important components. Thus, a comprehensive trading strategy will often comprise the underlying's selection, entry and exit timing, position and capital management, and the risk control solution. Based on the trend-following strategy, we develop a quantitative trading method that is sensitive to market fluctuations and stable to risk.

Among the components of a trading strategy, the underlying asset for the trade has been specified in this question, which includes a more stable traditional asset (gold) and a riskier digital asset (bitcoin). As such, we will focus on creating our trading strategy on the following points:

- Establishing the proper **entry and exit signals**
- Optimizing the **asset-to-asset allocation ratio**
- Constructing a tool for **precise positioning adjustment**
- Creating a workable **risk management systems**

In the post, we begin by introducing the key indicators that will be used and then describe the entire strategy in modules.

4.2 Description of the Indicator

1. **Average True Range (ATR)**

ATR is a volatility indicator that measures market volatility by decomposing the entire range of an asset price for that period[5]. There are two steps in the calculation of the ATR: first, the true range (TR) of the asset is determined; and then a moving average (MA) is used to smooth the ATR:

$$TR = \mathrm{Max}\left[(H-L), \mathrm{Abs}\left(H-C_P\right), \mathrm{Abs}\left(L-C_P\right)\right]$$

$$ATR = \left(\frac{1}{n}\right)\sum_{(i=1)}^{(n)} TR_i$$

where H denotes the day's highest price, L denotes the day's lowest price, and P_{t-1} denotes the previous day's price, and n denotes the time period used.

Due to the fact that the daily price in the data provided in this question is a constant amount, in other words, H equals L, refer as the day's price P_t. Therefore, the computation process can be streamlined as follows:

$$ATR = \left(\frac{1}{n}\right)\sum_{i=1}^{n}(P_t - P_{t-1}) \tag{8}$$

2. **Maximum Adverse Excursion(MAE) and Maximum Favorable Excursion(MFE)**

MAE indicates the low point of a long trade and the high point of a short trade. It assists in determining the trade's maximum loss. This is often referred to as the position's maximum drawdown. MFE indicates the high point of a long trade and the low point of a short trade. It shows the profit earned during a trade. The calculation steps for these two indicators are as follows:

$$\mathrm{MFE} = \left(\frac{1}{n}\right)\left(\sum_{1}^{n} \mathrm{Abs}(P_i - P_{iL})/ATR\right) \tag{9}$$

$$\mathrm{MAE} = \left(\frac{1}{n}\right)\left(\sum_{1}^{n} \mathrm{Abs}(P_i - P_{iH})/ATR\right) \tag{10}$$

where n represents the total times of market entry signals, P_i represents the price of the i^{th} transaction, and P_{iH} and P_{iL} represent the maximum and minimum price in the calculation period.

Based on the MAE and MFE, we can get a plausible signal for market entry: E-ratio. The following is the definition:

$$\mathrm{E}_{\text{ratio}} = \frac{MFE}{MAE} \tag{11}$$

3. **Sharpe Ratio**

The Sharpe ratio is used to help investors understand the return of an investment compared to its risk. When comparing two assets, the one with a higher Sharpe ratio appears to provide better return for the same risk, which is usually attractive to traders[6]. It is a crucial factor for risk evaluation in our trading techniques and is defined as follows:

$$S_a = \frac{E\left[R_a - R_b\right]}{\sigma_a} = \frac{E\left[R_a - R_b\right]}{\sqrt{\operatorname{var}\left[R_a - R_b\right]}} \tag{12}$$

where R_a is the asset return, R_b is the risk-free return. $E[R_a - R_b]$ is the expected value of the excess of the asset return over the benchmark return, and σ_a is the standard deviation of the asset excess return.

4.3 Model Structure

The basic flow of our trading strategy is shown in Fig.5

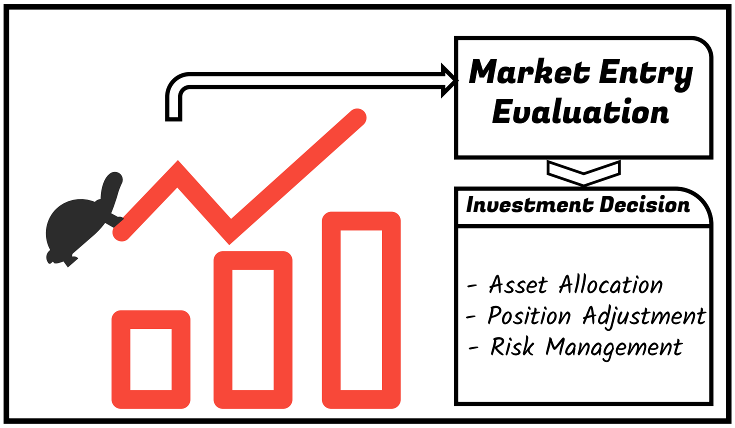


Figure 5: Structure of the trading model

The initial step is to evaluate current market conditions and then enter the market for investment after those conditions are met. The following phase will be to establish the fairest investment ratio and allocation ratio for the two assets using a series of judgment procedures and optimization techniques to ensure that return and risk are balanced. We will outline the approach and important judgment indicators in chronological order, beginning with market entry and ending with investment.

1. **Market Entry Evaluation**

In the forecasting section we have mentioned that due to the small number of price datasets corresponding to the first 30 days, there are large fluctuations and deviations in the forecast values. As a result, entering the market during the early stages is a dangerous and ill-advised action. We choose to evaluate the entry signal after 30 days.

The E-ratio is a frequently utilized market entrance indicator. We update it on a 10-day cycle for gold and bitcoin. When the E-ratio exceeds 1.4, we consider it a strong entry signal and choose to enter the market by withdrawing a certain percentage of cash in hand to purchase bitcoin/gold to begin the investing adventure; the percentage of money used for the entry is an arbitrarily chosen hyperparameter.

2. **Basic conditions and guidelines**

The basic concept of investing is to **buy today for future gains when the price is projected to climb tomorrow** and to **sell today for a stop loss when the price is predicted to fall tomorrow**. Specifically for purchase operations, buying is only authorized if the buy return is greater than the hold return. This condition can be represented as follows:

$$\left\{\begin{array}{l} x+n_{1g}m_{2g}<x-1.01(n_{2g}-n_{1g})m_{1g}+m_{2g}n_{2g} \\ x+n_{1b}m_{2b}<x-1.02(n_{2b}-n_{1b})m_{1b}+m_{2b}n_{2b} \end{array}\right. \Rightarrow \left\{\begin{array}{l} m_{2g}>1.01m_{1g} \\ m_{2b}>1.02m_{1b} \end{array}\right. \tag{13}$$

Selling is the same , We only sell if the sell loss is less than hold loss .The conditions are as follows:

$$\left\{\begin{array}{l} x+n_{1g}m_{2g}>x+0.99(n_{2g}-n_{1g})m_{1g}+m_{2g}n_{2g} \\ x+n_{1b}m_{2b}>x+0.98(n_{2b}-n_{1b})m_{1b}+m_{2b}n_{2b} \end{array}\right. \Rightarrow \left\{\begin{array}{l} m_{2g}<0.99m_{1g} \\ m_{2b}<0.98m_{1b} \end{array}\right. \tag{14}$$

where x represents cash held this day, n_1 represents the number of assets held this day, n_2 represents the number of assets held tomorrow, m_1 represents the unit price of the asset today, and m_2 represents the unit price of the asset tomorrow. The subscript b means bitcoin and the subscript g means gold, with different trading commissions. These two inequalities, which apply to the acquisition of gold and bitcoin, will be referred to as $Condition1$ and $Condition2$, respectively.

3. **Investment Decision**

Table 5 summarizes the parameters utilized in the investment decision-making process.

Table 5: **Symbols for the trading strategy**

Notation	Definition
C	The amount of cash held on the day (in U.S. dollars)
G	The amount of gold held on the day (in ounces)
B	The amount of bitcoin held on the day
P_g	The price of gold per ounce on the day
P_b	The price per bitcoin on the day
ν	The total value of the current portfolio
τ	The risk volatility of a portfolio
γ	The daily rate of return on a portfolio
s	The Sharpe Ratio
I_f	The Investment factor

Given the fact that gold has a trading day limit, we need to verify the current day before trading, and the results fall into two categories:

i. If the day is not a gold trading day, the trade will be limited to bitcoin. Purchase bitcoin when the estimated bitcoin price for tomorrow meets $\boldsymbol{C}ondition1$ from equation(14), the exact percentage of this purchase is calculated below.

ii. While a gold trading day occurs, take into account both the rise and fall of gold and bitcoin when determining the ideal investment configuration.

Following that, we ascertain the precise quantity and percentage of investment. Take note that we are addressing a hypothetical situation here; the scenario in which only bitcoins can be traded is a simplification of this scenario. To begin, we assume that the portfolio holdings for today are $[C1, G1, B1]$, those for tomorrow are $[C2, G2, B2]$, and the total value of the current portfolio are:

$$\nu = C + G \times P_g + B \times P_b \tag{15}$$

From the above formula, the total value of assets today and total value of assets tomorrow are indicated as ν_1 and ν_2 correspondingly.

Then the rate of return of this investment is:

$$\gamma = (\nu_2 - \nu_1)/\nu_1 \tag{16}$$

Presuming no extra return on cash holdings, we define the risk volatility of the portfolio for tomorrow as follows:

$$\tau = \sqrt{(G \times P_{g-total} \times S_{td_1})^2 + (B \times P_{b-total} \times S_{td_2})^2} \tag{17}$$

where $P_{g-total}$ refers to the total value of gold held on tomorrow, $P_{b-total}$ refers to the total price of bitcoin held on tomorrow, S_{td_1} refers to the standard deviation of the price of gold in the last month, and S_{td_2} refers to the standard deviation of the price of gold in the last month.

The Sharpe ratio for this portfolio is:

$$s = \frac{\gamma - R_f}{\tau} \tag{18}$$

where R_f is the risk-free return, which is assumed to be 0 here.

By combining the aforementioned parameters, we create a metric called the **Investment Factor** that reflects a composite representation of return and risk.It is defined as follows:

$$I_f = \alpha \cdot \gamma + \beta \cdot S \tag{19}$$

where α is the ratio of daily return rate to investment factor, and β is the ratio of daily Sharpe Ratio to investment factor. They are two hyperparameters that have been artificially established. We can infer that investment factor I_f possesses the following two properties:

- Directly proportional to the rate of return, i.e. the higher the rate of return, the greater the investment factor I_f
- Inversely proportional to the risk, i.e. the smaller the risk, the greater the investment factor I_f

From the above, we can maximize the investment factor I_f by optimizing the portfolio $[C2, G2, B2]$, thereby obtaining the optimal portfolio with a relatively high rate of return and a low rate of risk.

By changing the size of α and β, investors with different degrees of aggressiveness can be simulated: if one is an aggressive investor who only values returns and does not count risks, one can set α to be larger and β to be smaller, while if one is a conservative investor who favors less risky investments, one can set β to be larger.

4.4 Model Optimization and Solution

According to the above description of the investment factor, we can deduce that the investment factor I_f is only related to the portfolio $[C2, G2, B2]$, and that portfolio satisfies the following constraint relationship:

$$C_1 + G_1 \times P_{g_1} + B_1 \times P_{b_1} = C_2 + G_2 \times P_{g_1} + B_2 \times P_{b_1} - M \tag{20}$$

Where M is the fee required for this transaction, which is calculated as:

$$M = 0.01 \times (G_2 - G_1) \times P_{g_1} + 0.02 \times (B_2 - B_1) \times P_{b_1} \tag{21}$$

So we can use the PSO(Particle swarm optimization), change different portfolios, and perform fast iterative optimization to obtain the maximum investment factor, the specific algorithm process of the pseudo code is as follows:

Algorithm 1 Maximize I_f

```
for each particle $i$ do
    Initialize velocity $V_i$ and position $X_i$ for particle $i$
    Evaluate particle $i$ and set $pBesti = X_i$
end for
$IBest = max(pBesti)$
while not stop do
    for $i = 1toN$ do
        Update the velocity and position of particle i
        Evaluate particle i
        if $fit(X_i) < fit(pBesti)$ then
            $pBesti = X_i$
        end if
        if $fit(pBesti) < fit(IBest)$ then
            $IBest = pBesti$
        end if
    end for
end while
$I_f = IBest$
```

Finally, using our model and strategy, the initial **1,000 $** becomes **108,387.309 $** after five years

5 Model Evaluation

To prove that our model gives the best trading strategy, we must assess a variety of investment models and objectively compare their strengths and flaws to show that ours is the best. To that purpose, we go through three metrics that are frequently used to assess quantitative trading: APR(Annual Percentage Rate),SharpeRatio, MD(Maximum drawdown).

Based on these quantitative metrics, we compare our model with several traditional quantitative investment models to demonstrate that our model offers the best trading strategy.The results obtained are shown in the following table:

Table 6: **Model evaluation results**

Model	APR	Sharpe Ratio	MD
Our Model	1.296	1.28	0.38
CAPM Model	1.15	1.07	0.51
MDP Model	1.05	1.21	0.31
Maekowitz Model	1.37	0.78	0.63

The above table is created in the style of a radar chart to help you see the benefits of our model more clearly.

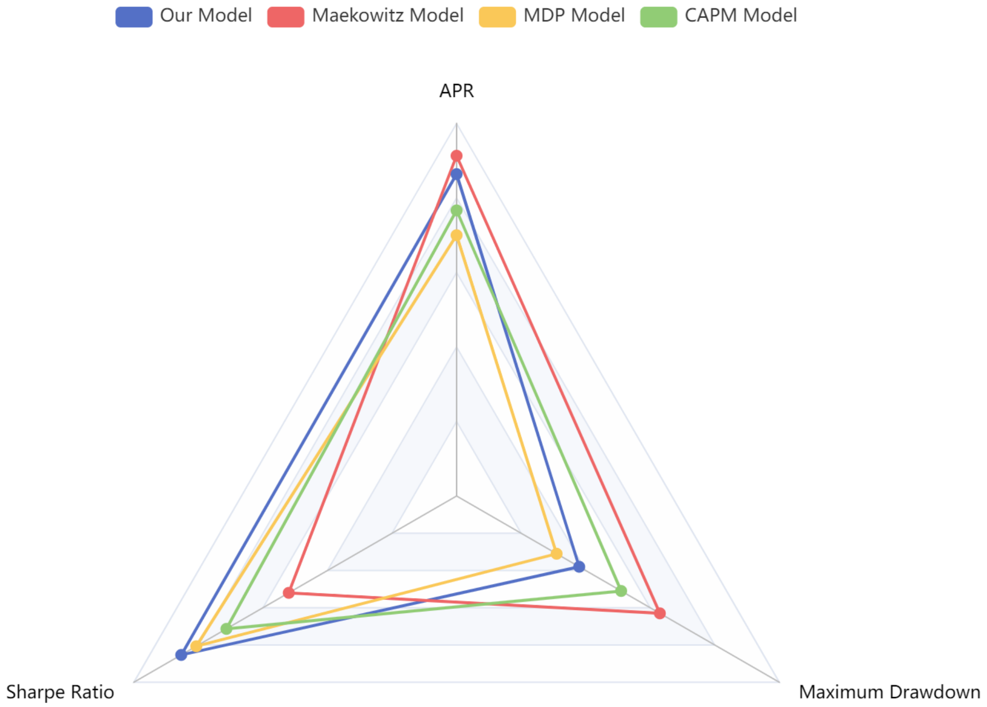


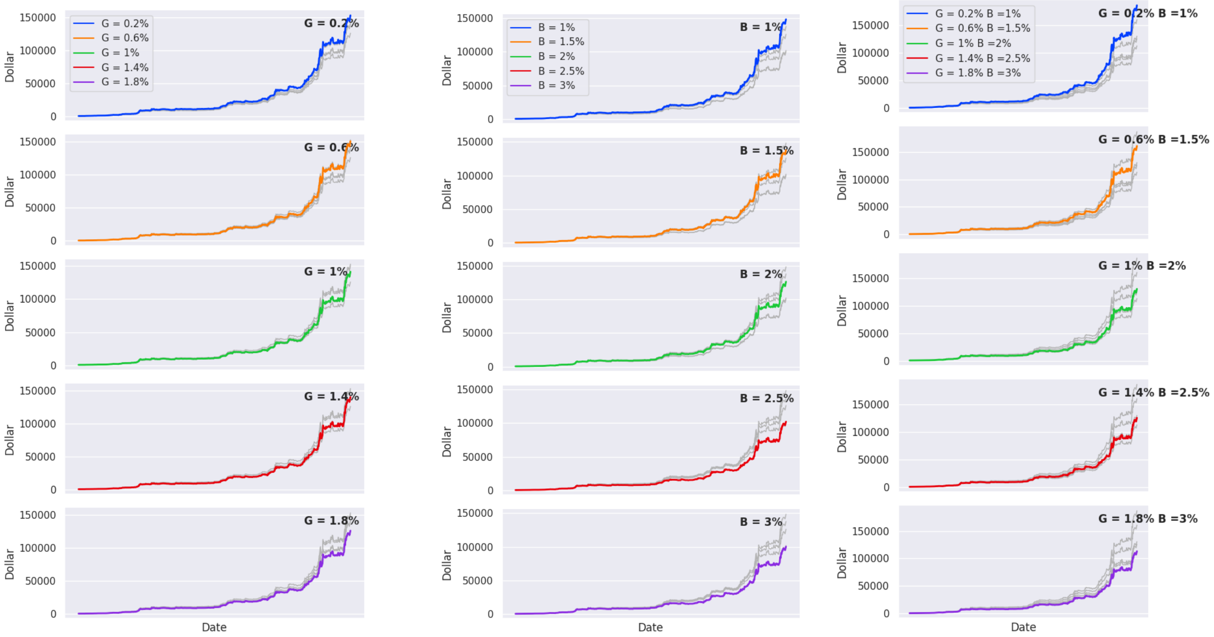
Figure 6: Radar plot of model evaluation results

Compared with other models , our decision model has the highest APR and Sharpe ratio, and the 30-day maximum retracement is below 40%, as shown in the Figure 5, indicating that our model has the highest return with managed risk and can prove that our model provides the best strategy.

6 Sensitivity Analysis

6.1 Value Sensitivity

Two main parameters are subjected to sensitivity analysis: α_{gold} (Commissions on gold transactions) and $\alpha_{bitcoin}$ (Commissions on bitcoin transactions). First, we change the value of α_{gold} from 0.2% to 1.8%, with each step increasing by 0.4%. The results, as shown in Figure 5(a), show that the trend in the overall value of the portfolio only varies significantly. Then, similarly, we change the value of $\alpha_{bitcoin}$ from 1% to 3%, with each step increasing by 0.5%, and evaluate its sensitivity. It does not appear to be notably sensitive,except after 2015, as illustrated in Figure 5(b). Finally, we modify the values of α_{gold} and $\alpha_{bitcoin}$ at the same time to see how the trend changes. It can be found in Figure 5(c) that even when changing the commission ratio for both gold and bitcoin, the total value of the portfolio generated by our model remains within a tolerable range, demonstrating good robustness and insensitivity to commission changes.



(a) Only α_{gold}

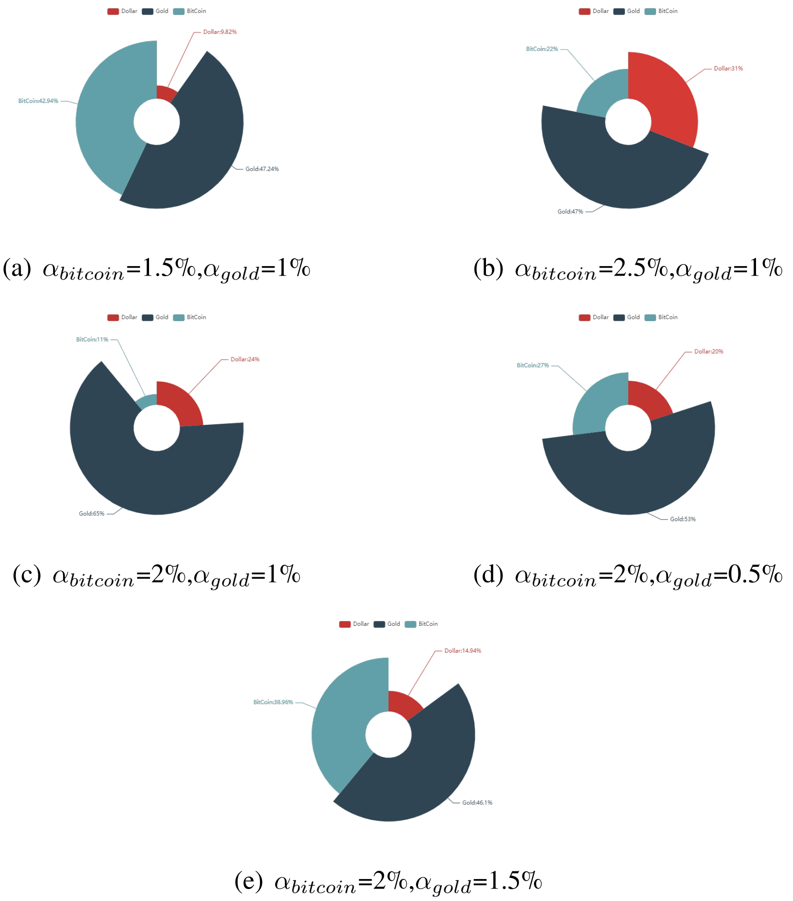
(b) Only $\alpha_{bitcoin}$

(c) α_{gold} and $\alpha_{bitcoin}$

Figure 7: Value Sensitivity

6.2 Asset Ratio Sensitivity

We examine the influence of different commission percentages on asset ratios by adjusting the values of a and b and examining the respective percentages of cash, gold, and bitcoin in the final portfolio



(a) $\alpha_{bitcoin}$=1.5%,α_{gold}=1%

(b) $\alpha_{bitcoin}$=2.5%,α_{gold}=1%

(c) $\alpha_{bitcoin}$=2%,α_{gold}=1%

(d) $\alpha_{bitcoin}$=2%,α_{gold}=0.5%

(e) $\alpha_{bitcoin}$=2%,α_{gold}=1.5%

Figure 8: Value Sensitivity

As you can see from these asset share distribution charts, the percentage of bitcoin holdings increases slightly when bitcoin commissions fall and decreases slightly when bitcoin commissions rise, ditto for gold. This is in line with the basic market rule

7 Strengths and Weaknesses

7.1 Strengths

1. In the price forecast section, we combine short-term price fluctuations and long-term pricing patterns and anticipate future prices using a variety of models, including LSTM, random forest, and others. To assure the process's integrity and reliability, these forecasting approaches are reviewed using a range of quantitative indicators and graphical analyses.

2. In the investment decision section, we develop a comprehensive market entry strategy based

on the ATR, MAE, MFE, and E-ratio, and designed the investment factor as an indicator to evaluate the trading upsides and downsides by integrating the price trend, return rate, and risk appetite. Thus, determining trading strategies boils down to an optimization problem in which alternate hyperparameters can modify the model's long-term tendency, enhancing the model's ultimate outcome while increasing the strategy's customizability and interpretability.

3. In the strategy evaluation section, we evaluate and compare different strategies in multiple dimensions such as return, stability, risk, maximum retracement, etc. We provide a comprehensive description of the performance differences between the models horizontally and systematically explain the model's unique benefits and justifications for these advantages.

4. In the sensitivity analysis section, we examine the impact of varying commission ratios on the model's results on total assets. Additionally, we monitor daily fluctuations in the trading volume of each asset class to demonstrate the impact of commission changes on the strategy preferences and allocation ratio of the two assets, as well as to demonstrate the complete strategy modifications in each asset.

7.2 Weaknesses

1. The accuracy of the price prediction findings is limited because the data given in the question does not include additional information, such as the opening price, closing price, trading volume, etc. This has an impact on the assessment of price trends when making investment decisions.

2. The investing approach is primarily concerned with single-day benefits, resembling a short-term trading model with a relatively short asset holding period and little attention for long-term holding benefits.

3. As the value of both assets rose during the trading period, no liquidation signals were constructed to counteract the impact of force majeure factors such as politics and war on market sentiments.

8 Conclusion

Today, quantitative trading systems are a crucial component of investing in a variety of financial goods. The extensive use of quantitative trading algorithms enables traders to earn substantial profits while maintaining a constant and balanced return-to-risk ratio.

In this study, we rationalize the organization of a particular pricing dataset and investigate price change trends using a variety of models including LSTM. Following that, we create an investment factor to optimize our method by following the trend strategy and integrating numerous indicators with price forecast data. We also depict the cross-dimensional analysis of different methods in a variety of ways to demonstrate our models efficacy and stability, as well as to give traders with dependable assistance.

Memorandum

To: Trader

From: Team 2221805

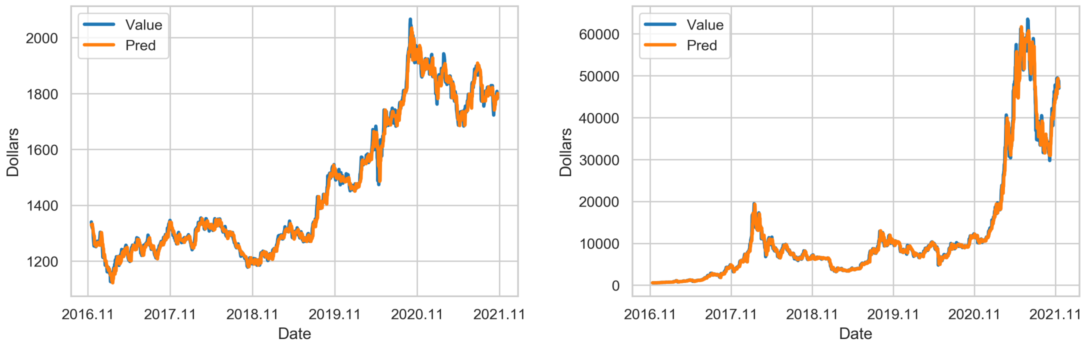
Subject: Quantitative Trading Strategies: Making a splash in the market

Date: February 21, 2022

At your request, we established an effective, stable, and strong trading technique in the gold and bitcoin markets. This strategy can generate a return of $108,387.309 over a five-year period with a starting capital of $1,000. Its major advantages are as follows:

- Combine the demand for large returns and the need for relativelyălow risk
- Maximum loss is less than similar strategies
- Long-term growth pattern of total assets is not sensitive to changes in commission rates
- Investment strategies are less susceptible to commission rates and are stable over time

Our approach is categorized into two main components: future price forecasts and investment decisions. The price prediction module is primarily comprised of a set of deep learning and machine learning algorithms that forecast the price of the day based on the previous N days and update our model at regular intervals to assure its accuracy. Fig.9 shows how our anticipated value compares to the actual value.



(a) Gold price forecasting by LSTM model (b) Bitcoin price forecasting by LSTM model

Figure 9: LSTM prediction results

Our basic investing framework is to buy assets when the expected future price is greater than the commission rate and sell assets when the predicted future price is lower, ensuring a rising trend in total assets. We use E-Ratio to find the best moment to open a position. And then weădesign a proprietary metric investment factor to measure a portfolio's gold-bitcoin allocation, asset purchases, and risk management. Daily return can be guaranteed at a low risk by optimizing the portfolio according to this indicator. After a 5-year investment period, we can achieve a 129.6% average yearly return. The trends in total assets over the five-year period are shown below

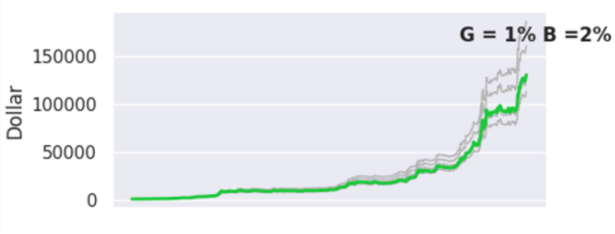


Figure 10: Change in total assets (in USD)

Additionally, our model outperforms other trading strategies. The radar chart below is constructed using the three indicators APR, Sharpe Ratio, and maximum drawdown.

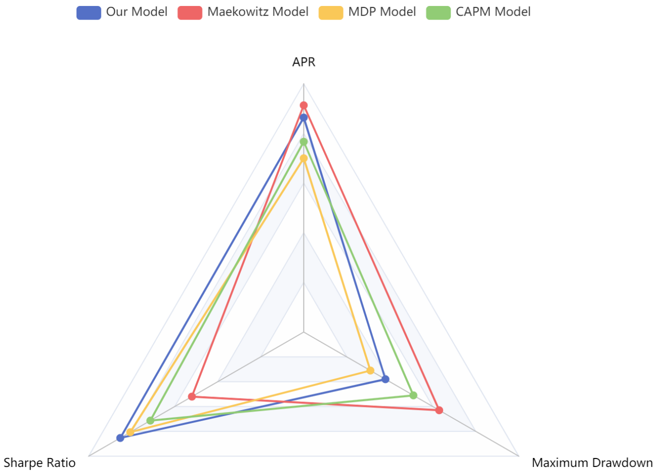


Figure 11: Radar plot of model evaluation results

From Fig.11, we can know that our model has the highest APR and Sharpe ratio, and the maximum 30-day retracement is less than 40%, proving that it offers the best return with the least risk among these strategies.

We have also evaluated the influence of commission changes on model returns and strategies, and our experiments show that commission changes have no impact on long-term returns. The total asset trend remains unchanged, but the allocation ratio of the two assets may be altered. Because of this, our models can reliably create earnings for you regardless of commission adjustments.

Finally, we genuinely hope that our models will be beneficial to you and generate sustained value for you.

Appendices

Here are some programmes we used in our model as follow.

Input python source:

```
N = 1
M = 1
INPUT_FEATURES_NUM = N
HIDDEN_SIZE = 100
OUTPUT_FEATURES_NUM = M
EPOCHS = 200



TRAIN_STEP = 2
MONTH = 5
name = ’bc’
# mean = 0
# std = 0
max = 0
min = 0

def normalize(data, min, max):
    return (data - min) / (max - min)

def step_train(dataset, df, path):
    pred_value = [0 for i in range(len(df))]


    for i in range(TRAIN_STEP, len(dataset)):

        lstm_model = LSTM(INPUT_FEATURES_NUM, HIDDEN_SIZE, \
            OUTPUT_FEATURES_NUM, num_layers=1) # 16 hidden units

        if i % TRAIN_STEP == 0:
            if i < MONTH:
                X = dataset[0:i, 0]
                y = dataset[0:i, 1]
                # X = dataset[:i, 0:N]
                # y = dataset[:i, N:N+M]
            else:
                X = dataset[i-MONTH:i, 0]
                y = dataset[i-MONTH:i, 1]
                # X = dataset[:i, 0:N]
                # y = dataset[:i, N:N+M]
            # mean = np.mean(y[0:i])
            # std = np.std(y[0:i])
            # X = (X - mean) / (std)
            # y = (y - mean) / (std)
            # print(X, mean, std)
            max = np.max(y)
            min = np.min(y)
            # print(np.ravel(y), min, max)
            X = normalize(X, min, max)
            y = normalize(y, min, max)
            # print(np.ravel(y), min, max)
            train(X, y, name)

        lstm_model.load_state_dict(torch.load(name+’.pkl’))
        # load model parameters from files
        lstm_model = lstm_model.eval() # switch to testing model
        # prediction on test dataset
        X = normalize(dataset[i, 0], min, max)
        pred_x_tensor = X.reshape(-1, 1, INPUT_FEATURES_NUM)
        # set batch size to 5, the same value with the training set
        pred_x_tensor = torch.from_numpy(pred_x_tensor)
        pred = lstm_model(pred_x_tensor)
        pred = pred.view(-1, OUTPUT_FEATURES_NUM).data.numpy()

        # print(i, pred[0][0] * (max - min) + min + df.iloc[i, 1])
        print(i, pred[0][0] * (max - min) + min)
        pred_value[i] = pred[0][0] * (max - min) + min
    # df[’pred’] = pred_value + df[’Value’]
    df[’pred’] = pred_value
    df.to_csv(path, index=False)

bc_df = pd.read_csv(bc_path)
gold_df = pd.read_csv(gold_path)

bc_diff = np.array(pd.read_csv(bc_diff_path))
gold_diff = np.array(pd.read_csv(gold_diff_path))

step_train(bc_data, bc_df, bc_pred)
step_train(gold_data, gold_df, gold_pred)
print(’step train finished.’)
```

```
    ATR = df_gold.loc[day - 10:day, "Increase"].mean() #ATR
    MAE = abs(df_gold.loc[day, "USD"] - df_gold.loc[day - 10:day, "USD"].min())
    MFE = abs(df_gold.loc[day, "USD"] - df_gold.loc[day - 10:day, "USD"].max())
    MAE = MAE/ATR
    MFE = MFE/ATR
    E = MFE/MAE

def buy_bc(day,day_gold):
    real_cash_percent = cash_percent
    if (real_cash_percent > 1):
        real_cash_percent = 0.99
    deal_ratio = []
    for a in range(1,11):
        a = a/10
        proper_temp = proper.copy()
        proper_value_before = proper_temp[0] + \
            proper_temp[1] * proper_temp[3] + proper_temp[2] * df_bc.loc[day, "USD"]
        proper_temp[2] = proper_temp[2] + \
            proper_temp[0] * real_cash_percent * a / df_bc.loc[day, "USD"]
        proper_temp[0] = proper_temp[0] * \
            (1 - (1+bc_ratio) * real_cash_percent * a)
        proper_value_after = proper_temp[0]  + \
            proper_temp[1] * df_gold.loc[day_gold+1, "pred"] + \
                proper_temp[2] * df_bc.loc[day+1, "pred"]
        yield_ = (proper_value_after - proper_value_before)\
            /proper_value_before
        bc_A = proper_temp[2]*df_bc.loc[day, "USD"]/(proper_temp[2]*df_bc.loc[day, "USD"] \
            + proper_temp[1]*proper_temp[3]+proper_temp[0])
        gold_A = proper_temp[1]*proper_temp[3]/(proper_temp[2]*df_bc.loc[day, "USD"] \
            + proper_temp[1]*proper_temp[3]+proper_temp[0])
        risk = math.sqrt(math.pow(bc_A*df_bc.loc[day - 30:day, "USD"].std(),2)\
            +math.pow(gold_A*df_gold.loc[day_gold - 30:day_gold, "USD"].std(),2))
        deal_ratio_temp = yield_ratio*yield_ + sharp_ratio*yield_ / risk
        deal_ratio.append(deal_ratio_temp)
    real_a = (deal_ratio.index(max(deal_ratio))+1)/10
    proper[2] = proper[2] + proper[0] * real_cash_percent * real_a / df_bc.loc[day, "USD"]
    proper[0] = proper[0] * (1 - (1+bc_ratio) * real_cash_percent * real_a)
```

References

[1] Xing Wu, Haolei Chen, Jianjia Wang, Luigi Troiano, Vincenzo Loia, and Hamido Fujita. Adaptive stock trading strategies with deep reinforcement learning methods. *Information Sciences*, 538:142–158, 2020.

[2] Cong Ma, Jiangshe Zhang, Junmin Liu, Lizhen Ji, and Fei Gao. A parallel multi-module deep reinforcement learning algorithm for stock trading. *Neurocomputing*, 449:290–302, 2021.

[3] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997. Conference Name: Neural Computation.

[4] Lijun Liu, Guopeng Liang, and Haitao Wang. Quantitative trading strategy system model research. *Shanghai Business*, (09):158–159, 2021.

[5] Tanishq Salkar, Aditya Shinde, Neelaya Tamhankar, and Narendra Bhagat. Algorithmic Trading using Technical Indicators. In *2021 International Conference on Communication information and Computing Technology (ICCICT)*, pages 1–6, June 2021.

[6] Hayette Gatfaoui. Sharpe Ratios and Their Fundamental Components: An Empirical Study. SSRN Scholarly Paper ID 1486073, Social Science Research Network, Rochester, NY, August 2009.