

A Two-stage Model: Investment Market Trading Model Based on Prediction and Adaptive Strategies

Shancheng Li^a, Kefei Hu^b, Yuyang Wu^c

Business School, Central South University, Changsha, China

^alisc_dsba@163.com, ^b8210191915@csu.edu.cn, ^c1186306554@qq.com

Abstract. Market traders buy and sell volatile assets frequently, with a goal to maximize their total return. Two such assets are gold and bitcoin. This paper constructs a two-stage model for price prediction and trading strategy formulation. Firstly, we build a dynamic ARIMA-LSTM hybrid model. And before applying it, we use CEEMDAN method to decompose the non-stationary time series first and then reconstruct the final result by predicting each IMF and summing up weighted. And then the model can update the training set dynamically when new price data is released. After getting prices predicted, we calculated several quantitative trading indicators so that we can make decisions more comprehensively instead of only focusing on the predicted price. And we use semi-supervised SVM to develop an adaptive strategy to maximize the total return. Finally, we demonstrate the superiority of our strategy from two perspectives. In actual investment transactions, the two-stage model can be used as a guide for the formulation of trading strategies, thereby avoiding risks and increasing returns.

Keywords: Two-stage model; ARIMA-LSTM; CEEMDAN; Quantitative trading indicators; Semi-supervised SVM; Adaptive strategy.

1. Introduction

1.1 Background

Bitcoin and gold are currently highly sought-after assets that can be traded. Gold is a safe-haven asset, a precious metal that is widely considered to be valuable. Bitcoin, on the other hand, differs from traditional assets as it is blockchain-based and has decentralized nature. Its price increases tend to be more variable.

Financial time series prediction uses historical data generated by financial markets to establish prediction models to mine inherent fluctuation trends [1]. Therefore, how to accurately reveal the changing trend of financial time series and reasonably predict financial time series is a key concern of academia and practitioners.

After getting the prediction results, how to formulate the optimal quantitative investment strategy is another key issue in the financial investment market. The multi-factor model is the basis of all kinds of complex models. It is suitable for various capital market environments, and assets selected through this model have stable performance and are widely used in China. Another type of model in quantitative investment is mainly based on machine learning-related algorithms. These models are completely based on statistics. Through continuous training and simulation of large samples, a prediction close to the real situation is obtained. The mainstream models include support vector machines, decision trees, neural networks, etc.

1.2 Related Literature

1.2.1 Assets Price Prediction

Research methods for financial time series prediction have undergone extensive evolution, from statistical models to machine learning models, from single models to combined models. For example, the combination of AMIMA and MLPs is used to predict the Standard & Poor's 500 Index, Shenzhen Stock Exchange Index and Dow Jones Index [2]. Using machine learning methods to predict financial time series is also a research hotspot in the field of financial data analysis in recent years. Bao et al. empirically proved that the prediction performance of LSTM for financial time series is better than

that of traditional RNN [3]. Ravi extracted indicators such as double exponential moving average, simple moving average, and moving average convergence divergence from Bitcoin price data, used Deep-LSTM for prediction, and conducts experiments on public data sets [4]. Both statistical models and neural network models have their own advantages and limitations. Therefore, some scholars also constructed a combination model for financial time series made predictions and confirmed the advantages of the combined model over the single model [5].

The investment market is a complex system that responds to changes in the external environment in many ways, with strong randomness and complex nonlinear intrinsic relationships among various phenomena. Therefore, some scholars identify different patterns of data through their different sub-modules, and then summarize and obtain the complete change laws contained in them, so as to achieve high-precision prediction of financial time series. Torres et al. proposed the CEEMDAN method. The IMFs obtained by decomposition are relatively simple and independent of each other, which provides favorable conditions for fully extracting the fluctuation characteristics of IMF subsequences [6]. He et al. used CEEMDAN to decompose and reconstruct five stock indices including CSI 300, built LSTM models for low-frequency components and prediction, optimized the combination of high-frequency components [7].

1.2.2 Investment Trading Strategy

The application of machine learning methods in quantitative investment strategy formulation in financial markets is becoming more and more extensive. The earliest machine learning methods used abroad are SVM [8] and Decision Tree [9]. Harris used the cluster support vector machine (CSVM) method to conduct research in the application scenario of Internet financial customer rating. Through analysis, it can be found that the model can obtain better rating results compared with the commonly used SVM [10]. Li et al. applied the semi-supervised K-means kernel function clustering method to the multi-factor stock selection model. The model has stronger generalization ability, and can select the best model even when the sample has obvious nonlinear and complex features [11]. Lv et al. used the multi-factor scoring model and the two-stage model of the support vector machine classification algorithm [12] to select the CSI 300 constituent stocks from the long-term and short-term advantages of the stock, and obtained a strategy with higher expected returns.

Most of the existing machine learning methods are based on existing experience or known data to solve strategies, and do not fully consider market fluctuations. Therefore, we develop a support vector machine based on semi-supervised learning for the self-learning and decision making of strategy thresholds.

1.3 The research content and structure of this paper

This paper used the daily prices of gold and Bitcoin for the five-year period from September 9, 2016 to September 10, 2021, as provided by the London Bullion Market Association and the NASDAQ. We started with \$1000 on 9/11/2016. On each trading day, we have a portfolio consisting of cash, gold, and bitcoin [C, G, B] in U.S. dollars, troy ounces, and bitcoins, respectively. The initial state is [1000, 0, 0]. The commission for each transaction (purchase or sale) costs $\alpha\%$ of the amount traded.

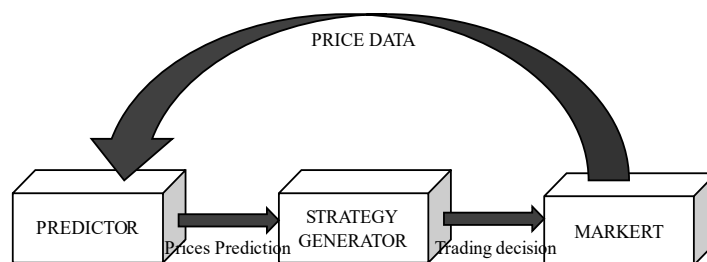


Fig. 1. The operation process of the two-stage investment decision model.

We developed a two-stage investment decision model to form long-short portfolios. In the first stage, we use the previous days' data to predict the next trading day's price. In the second stage, we

use the predicted prices to form a strategy that interacts with the market. After the interaction with the market is complete, we obtain new data to continue the prediction, strategy formation which forms a circle. The two-stage model is schematically illustrated as Fig. 1.

2. Assumptions and Notations

2.1 Model Assumptions

- If the total value of assets becomes 0 during the 5-year trading process, then the transaction cannot continue.
- Both assets are assumed to be infinitely divisible and can be purchased in any quantity.
- Bitcoin can be traded every day, but gold is only traded on days the market is open.
- $\alpha_{\text{gold}} = 1\%$ and $\alpha_{\text{bitcoin}} = 2\%$, and there is no cost to hold an asset.

2.2 Notation Definition

Table 1. Summary of Notations

| Symbol | Definition |
|-----------------------|---------------------------------------|
| P_i | The price on day i |
| MTM | Momentum Index |
| $Bias$ | Bias Ratio |
| \bar{E} | Mean price of previous days |
| $Index_n$ | market index on day n |
| $Risk_n$ | market risk on day n |
| A_n | Predicted rise on day n |
| $V_{i\text{-signal}}$ | Threshold of asset i of buy or sell |
| Pg_n / Pb_n | Proportion of assets on day n |
| Hb_n / Hg_n | Number of assets held on day n |
| M_n | Number of cash on day n |
| ω | adjustment factor |

3. Dynamically Updated Price Prediction Model

In this section, we try to predict the price of gold and bitcoin. Since only all price data up to that day can be used when trading, the prediction model should have the ability to dynamically update the training data set. Firstly, we collect real price data instead of predicting any data on the first 50 days. We weighted and summed the results of the traditional time series analysis ARIMA model as well as the deep learning LSTM model to get the final prediction results of day 51. After the closing price is released on day 51, it is added to the training dataset and the prediction results on day 52 are obtained using the ARIMA-LSTM model. The weights of the two models in this study change dynamically over time, showing a reciprocal relationship. By analogy, until September 10, 2021. It is worth noting that we firstly decompose the unsteady price series into a number of component series with different frequencies using the CEEMDAN method before using LSTM for prediction. The LSTM model is then used to predict each component and then summed to aggregate the final prediction results. This can better handle non-stationary price series. The work flow of the prediction model is shown in Fig.2.

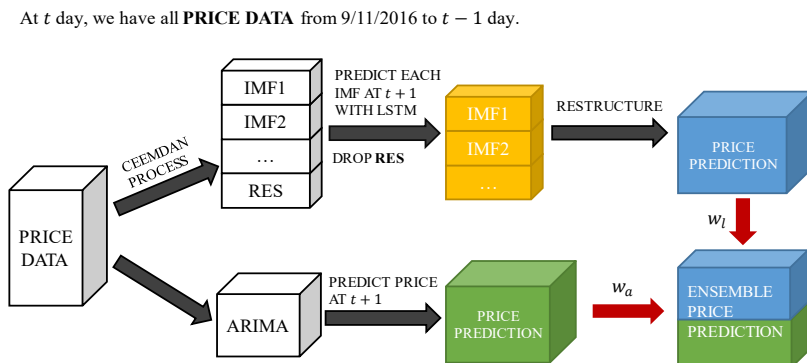


Fig. 2 Work flow of prediction model.

3.1 The Development of ARIMA Model

We collected the first 50 data and then tried to build an ARIMA model. Before building the model, it is necessary to differential the series and do the stationarity test, and then determine the model parameters.

3.1.1 Differential and Stationarity Test

Firstly, we plotted the change of autocorrelation coefficient for the first 50 days of data to initially determine whether the series is smooth, and the results are shown in Fig. 3.

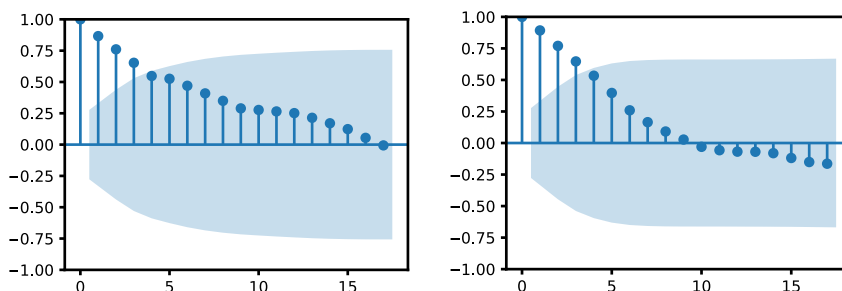


Fig. 3 Autocorrelation: gold on the left and bitcoin on the right.

From the result graph we can see that the autocorrelation coefficients of both assets fall to zero slowly, so we can initially judge the price series as non-stationary. Of course, we also need to do a unit root test to get a more precise and reliable conclusion. The ADF test is used here, and the original hypothesis is that there is a unit root in the series. The test results show that the p-values for gold and bitcoin prices are 0.46 and 0.99 respectively, which are both greater than 0.05. Therefore, we do not reject the original hypothesis and consider that the original series has a unit root. Thus, the price series is a non-stationary series.

Then we performed a first order difference on the original series and plotted the autocorrelation coefficient ACF with the partial autocorrelation coefficient PACF. The gold price is shown in Fig. 4 and the bitcoin price is shown in Fig. 5.

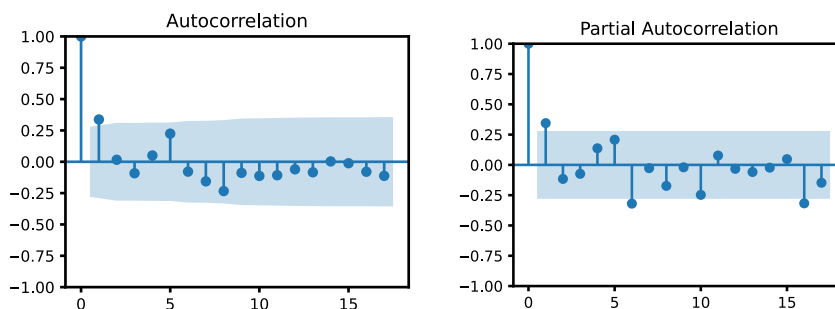


Fig. 4 ACF and PACF changes after first order difference of gold price.

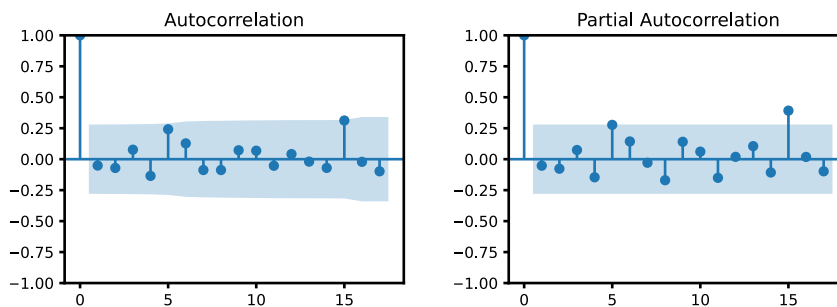


Fig. 5 ACF and PACF changes after first order difference of bitcoin price.

It can be seen that both ACF and PACF coefficients show a significant downward trend after the first-order difference between the prices of the two assets. Then we performed a unit root test with the ADF indicator again. Results show that the p-values are zero for both assets. Therefore, we can determine that both asset price series can be transformed to a smooth series after the first order difference.

3.1.2 The Construction of ARIMA Model

The ARIMA(p,d,q) model has three parameters, where d equals one, and then the optimal values of p and q are determined by applying the criterion of minimum AIC index. The final results show that the optimal model for gold differential series is ARIMA(2,1,2) and that of bitcoin is ARIMA(0,1,0). The left figure shows the prediction results after differencing and the right figure shows the prediction results after differencing reduction.

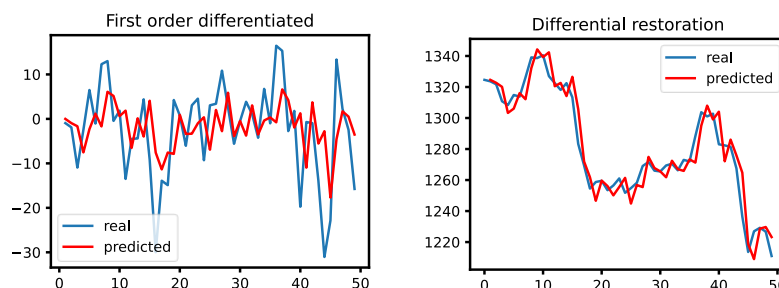


Fig. 6 Prediction result of gold price ARIMA model.

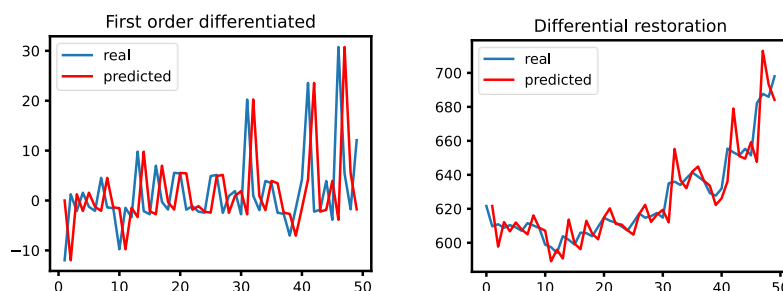


Fig. 7 Prediction result of bitcoin price ARIMA model.

From the back-test results we can see that the ARIMA model predicts well to certain extent, though there is some lag.

3.2 LSTM Model Based on CEEMDAN Method

In this section, we firstly used the CEEMDAN method to decompose the signal for the first 50 data, then we built an LSTM model for each individual IMF for prediction, and then summed up the prediction results of all component sequences to get the final prediction results.

3.2.1 CEEMDAN Model for Signal Decomposition

We have analyzed that the price series is nonlinear and nonstationary, so we can use the EMD algorithm to decompose it into a sequence of components with different frequencies, i.e., several IMFs. However, the residuals after EMD decomposition are large and may have an impact on the model prediction, so we tend to use the more accurate CEEMDAN algorithm to construct the signal decomposition. The residuals obtained when decomposing by this method are small, which means that the residuals can be ignored when reconstructing and predicting.

We used Python to do sequence decomposition on the first 50 data, and the result is shown in Fig.8.

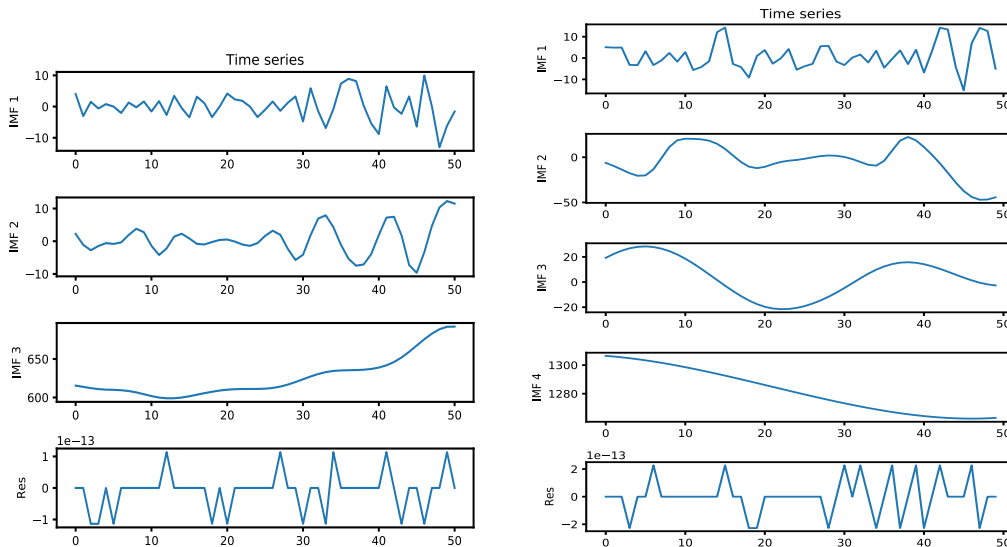


Fig. 8 Decomposition result: gold on the left and bitcoin on the right.

We can see that the residuals obtained after the decomposition of this method are so small that they can be completely ignored. At this point the gold price decomposes out to 3 IMFs and the bitcoin price decomposes out to 4 IMFs.

3.2.2 LSTM Model for Prediction

Based on the cyclic neural network, LSTM replaces the neurons in the hidden layer with hidden states and three gate structures to control the state of each memory storage unit and realize the update of control information on the hidden states.

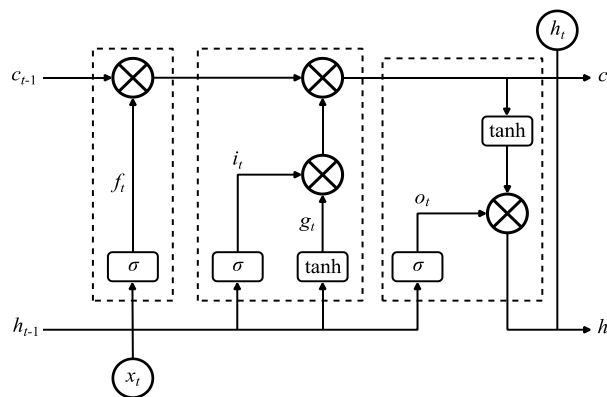


Fig. 9 The general structure of LSTM.

We divided these 50 data into training and validation sets in a ratio of 4: 1. Then, we blend the ARIMA model with the LSTM model to predict and observe whether the effect is better than that of one model alone on the data sets covering five years. It is worth noting that though we use all the

given data, it does not mean that there is a data leakage problem. Our model is dynamically updated, and the t th day predict uses only the price data before that day, which are $(p_1, p_2, \dots, p_{t-1})$, and then gets the result \hat{p}_t . When the t th day ends, our model calculates the MAPE between \hat{p}_t and the real closing price p_t on that day. After that, p_t is added to the training set, and then the model is retrained to make the prediction on the $(t+1)$ th day.

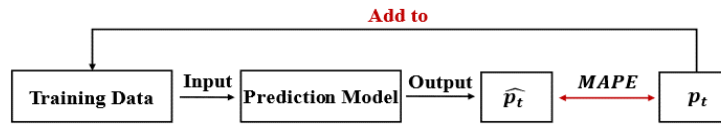


Fig. 10 The circle of dynamic prediction.

The calculation of MAPE is as follows:

$$MAPE_{all} = \frac{1}{N} \sum_{t=1}^N \frac{|\hat{p}_t - p_t|}{p_t} \times 100\% \tag{1}$$

$$\hat{p}_t = Model_{prediction}(p_1, p_2, \dots, p_{t-1}) \tag{2}$$

3.3 Ensembled Price Prediction

3.3.1 Experiments for the choice of the dynamic weights

Since we try to construct an ensembled model of ARIMA and LSTM, we need to determine the weights corresponding to the models separately. Since the ARIMA model works better in the early stage and the LSTM model works better in the later stage, we let the weights be dynamic throughout the prediction process, as long as w_a becomes smaller while w_l becomes larger and the sum remains one as time passing by. Therefore, we divide the five-year period into different time intervals, then give uniform weight changes for each time interval. Finally, we choose the combination of weights that minimizes the MAPE index as the weights used in the final prediction.

We tried to do experiments with six months, one year, and two and a half years as the time interval. The results show that the error of assigning weights with a one-year time interval is the smallest. The change of weights at this time is shown in Table 2.

Table 2. Dynamic weights for ARIMA-LSTM model

| Date | 9/11/2016-9/11/2017 | 9/12/2017-9/11/2018 | 9/12/2018-9/11/2019 | 9/12/2019-9/11/2020 | 9/12/2020-9/10/2021 |
|--------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| $[w_a, w_l]$ | [1,0] | [0.75,0.25] | [0.5,0.5] | [0.25,0.75] | [0,1] |

After determining the weights, we then experimentally demonstrate that ensembled model predicts better than the single model. We use ARIMA model, LSTM model and ARIMA-LSTM model to predict the five-year data respectively. The final best prediction results are shown in Fig. 11 and the error results are shown in Table 3.

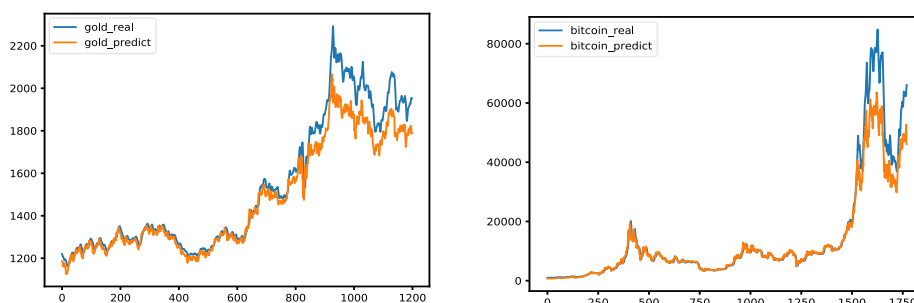


Fig. 11 ARIMA-LSTM model prediction result.

Table 3. Model Comparison

| Model | Gold_MAPE | Bitcoin_MAPE |
|------------|-----------|--------------|
| ARIMA | 13.78% | 15.18% |
| LSTM | 10.23% | 13.44% |
| ARIMA-LSTM | 9.15% | 11.25% |

We can see that the ensembled model has a stronger predictive effect than the single model. The biggest role of price prediction is to predict the trend of change, like up or down. How to decide whether to hold, which kind of assets to buy, as well as what amount to buy according to the existing data and the predicted data, is the problem we need to solve in next section.

4. The Development of Trading Strategy

In order to make a more comprehensive and scientific plan for the daily trading strategies of gold and bitcoin, we added several quantitative trading indicators to support the trading strategy formulation. The work flow of the development of the strategy is shown in Fig. 12.

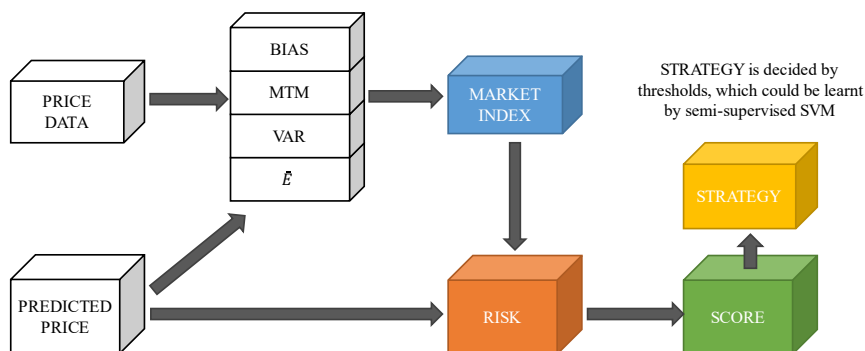


Fig. 12 The work flow for strategy generation.

4.1 The construction of quantitative trading indicators

According to the trading rules of gold and bitcoin, we mark the gold trading day first, and supplement the price of the previous trading day above the vacancy price. After that, calculate the ups and downs of gold and bitcoin cycles, and observe the trend of price fluctuations. Since the price of gold is less volatile, consider using the 15-day price fluctuation to calculate its phase average. The price of bitcoin is more volatile, so the phase average is calculated based on the 5-day price fluctuation. The cyclical fluctuations are shown in Fig. 13.

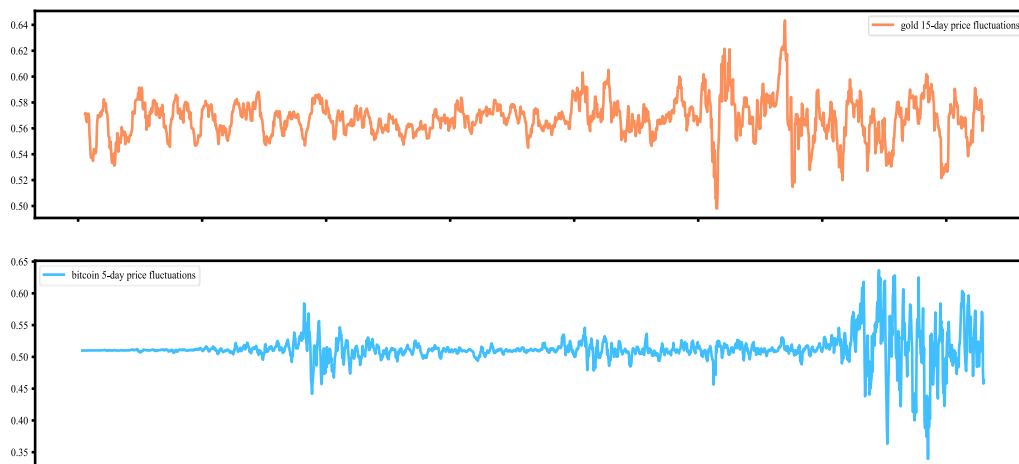


Fig. 13 Price fluctuation.

According to the moving average trading rule, the 5-day Bias can be calculated for the gold price and the 15-day Bias can be calculated for the bitcoin price, which can be added to the market environment model as volatility indicators. The formula for calculating the n-day Bias is as follows:

$$Bias(n) = (P_{today} - \bar{P}_n) / \bar{P}_n \quad (3)$$

The Momentum Index (MTM) is also an important factor to consider in a portfolio strategy for short-term analysis of price fluctuations in the investment market. In general, when MTM falls from the top to the bottom of the mean line, it is the time to sell, and when the MTM exceeds the mean line from bottom to top, it is the time to buy. The formula for calculating the n-day momentum index is as follows:

$$MTM(n) = P_{today} - P_{n-days-ago} \quad (4)$$

The investment market can be divided into bull market and bear market according to market conditions. We set the gold market quotation phase to 90 days and the bitcoin market quotation phase to 30 days. Define the market index as follows.

$$Index_n = \omega_1 \times \overline{Bias} + \omega_2 \times \bar{E} + \omega_3 \times MTM + \omega_4 \times Var_n \quad (5)$$

Since market conditions is cyclical but market fluctuations are inevitable within the same cycle, to ensure the accuracy of market division, we use the voting method to determine the market sentiment of both markets for each phase. If more than half of the days market index is greater than the threshold then market sentiment is high, otherwise it is low. The results are shown in the Fig. 14.

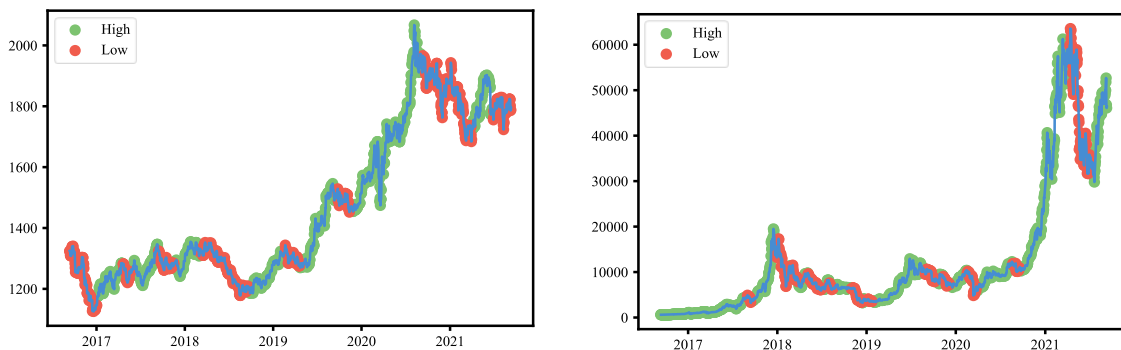


Fig. 14 Market sentiment indicator: gold on the left and bitcoin on the right.

According to the market division, we can calculate the market risk. We define that market risk is related to the market index and the proportion of investment assets :

▪ **As for gold:**

$$Risk_n = \omega_1 \times (1 - Index_n) + \omega_2 \times Pg_n + 0.5 \quad (6)$$

▪ **As for bitcoin:**

$$Risk_n = \omega_1 \times (1 - Index_n) + \omega_2 \times Pb_n + 0.5 \quad (7)$$

Among them, ω_1 and ω_2 are the adjustment weights, which are set according to the actual data characteristics. Take 0.5 as the initial market risk. The final market risk of gold and bitcoin are obtained as shown in Fig. 15.

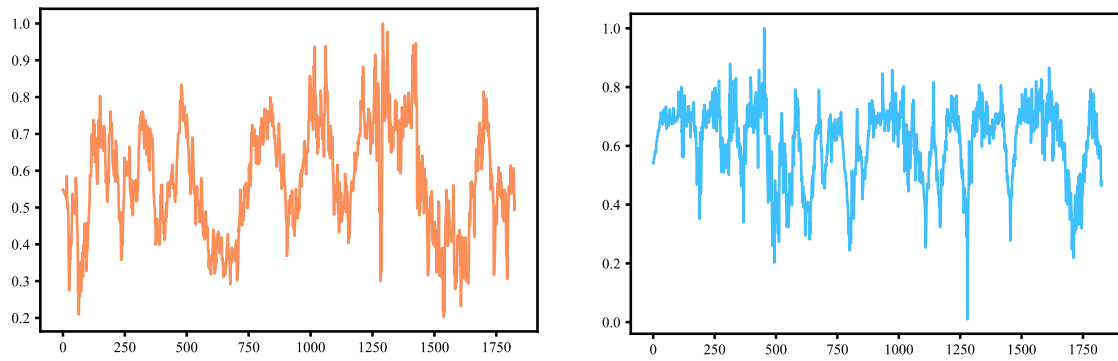


Fig. 15 Market risk: gold on the left and bitcoin on the right.

Based on the predicted fluctuation range and market risk, we can define the trade score. Then we can make a dynamic evaluation of the gold market and bitcoin market every day, and formulate trading strategies from the score. Buy when the trade score is higher and sell when it is lower. The n th day trade score is defined as follows.

$$Score_n = \omega_1 * A_n + \omega_2 / Risk_n \quad (8)$$

Regarding the buy and sell thresholds for transaction scoring, we build a semi-supervised support vector machine model for learning and updating. This modeling process is given in the next section. We suggest to follow the below trading rule:

- when the trade score is greater than the buy threshold, then we should buy;
- when it is less than the sell threshold, we should sell;
- when it lies between the buy and sell thresholds, we should hold.

Based on the trading rules of gold and bitcoin, when gold is on a non-trading day, the decision to buy, sell or hold it is made by considering only the trading score of bitcoins. When gold can be traded, then the value of gold and bitcoin needs to be judged. When

$$Score_{gold} - V_{gold-buy} > \omega * (Score_{bit} - V_{bit-buy}),$$

It means that the value of gold is higher than bitcoin on that day so only gold needs to be bought. Otherwise, only bitcoin needs to be bought.

Finally, we plan the buy and sell amount, while considering the commission as the transaction cost. The strategy is arranged as follows:

- **As for gold:**

$$Buy - quota_n = M_n \times Score_{n-gold} \times (1 - \alpha_{gold}) / P_{n-gold} \quad (9)$$

$$Sell - quota_n = Hg_n \times (1 - Score_{n-gold} + V_{gold-sell}) \quad (10)$$

- **As for bitcoin:**

$$Buy - quota_n = M_n \times Score_{n-bitcoin} \times (1 - \alpha_{bitcoin}) / P_{n-bitcoin} \quad (11)$$

$$Sell - quota_n = Hb_n \times (1 - Score_{n-bitcoin} + V_{gold-bitcoin}) \quad (12)$$

The higher the score is, the larger share the trader should buy. When selling, it is necessary to sell as many holdings below the scoring threshold as possible. In this way, the strategy will increase the share of assets traded and increase the amount of cash held.

4.2 The Strategy for Updating Threshold

Using fixed empirical thresholds to trade will not be able to adapt to changes in the market environment. Therefore, we develop a support vector machine based on semi-supervised learning for the self-learning and decision making of strategy thresholds. The learning process for updating is as follows:

- When we are about to make a decision on a particular day, we use the price data from the previous n days to find all the non-declining continuous subsequences.
- Then we consider the valley of the subseries as a buy signal and the peak as a sell signal. At the same time, we get the trade scores for these points.
- The current transaction scores are then added to the dataset along with the unconfirmed signals.
- The signals are then screened again using simulated trades in bitcoin and gold respectively.
- If those signals cannot indicate higher returns due to the presence of commission, then this set of buy and sell signals will be deleted.

The work flow for the learning process of threshold updating is shown in Fig. 16.

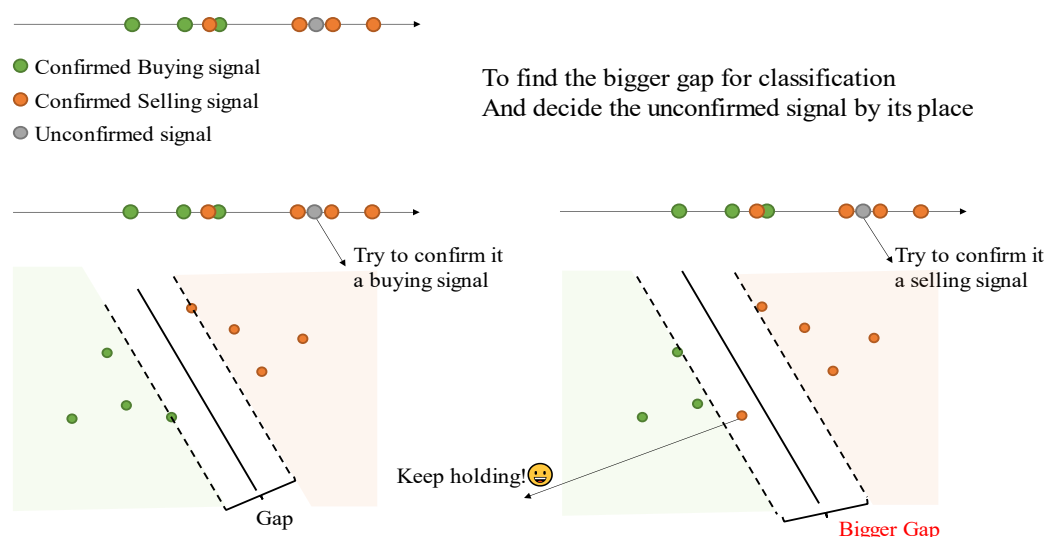


Fig. 16 The illustration of semi-supervised model.

It represents the mechanism of semi-supervised SVM model for our strategy:

- Firstly, we identify the unconfirmed signals as buy and sell signals separately to learn using a soft interval support vector machine with kernel functions.
- The kernel function is used to map the data to a higher dimension in order to improve the distinguishability of the signal and the nonlinear classification ability of the original dimension (Mapping to 2 dimensions is used as an example in the figure for illustration. RBF kernel is used for higher dimensional mapping in the model).
- The final decision boundary obtained is based on the signal confirmation process with the maximum gap.
- After that, if the unconfirmed data point lies in the gap, the point is considered neither a buy signal nor a sell signal. We decide to keep the asset in this case. If the unconfirmed data point is not in the gap, it is considered as a buy signal and a sell signal based on its position related to the gap.
- In addition to using transaction scores, data obtained from other calculations can also be used to classify them.

Based on the above trading strategy, we calculated the amount of total assets that can be traded from September 11, 2016 with an initial capital of \$1,000 until September 10, 2021, which can reach about \$97182.58 in total after 5 years.

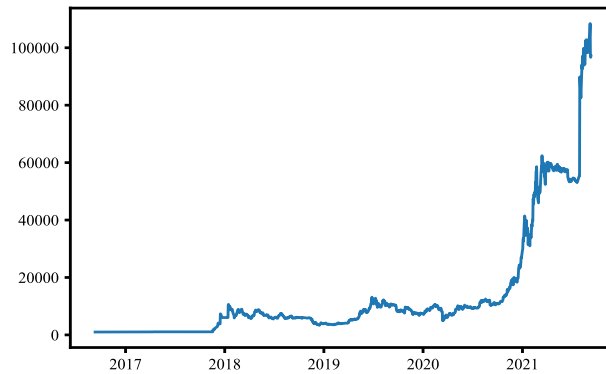


Fig. 17 Daily profit curve.

5. The evidence of the good strategy

5.1 Comparison between different strategies

We tested 3 strategies to face the market and got the profit result as Fig. 18 shown below:

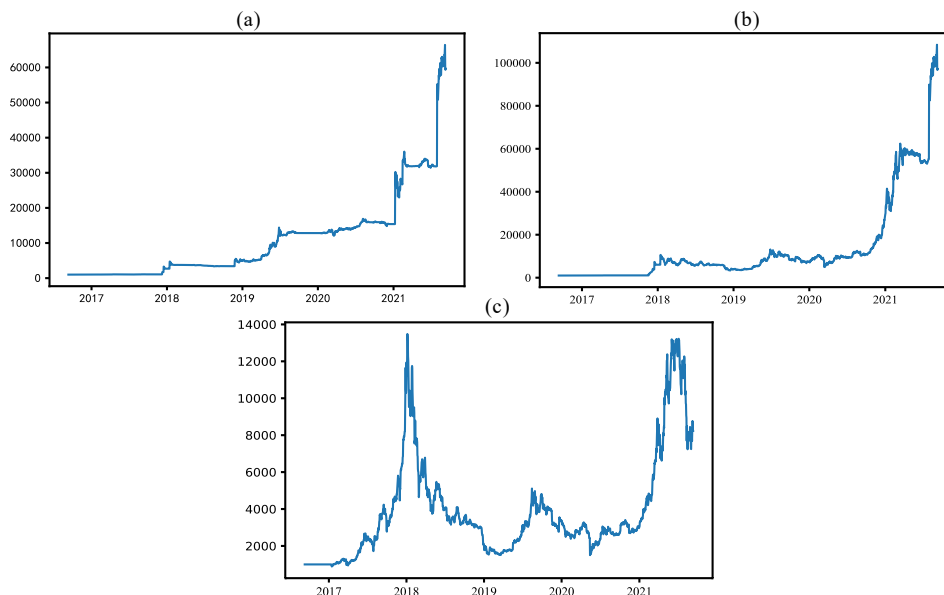


Fig. 18 The profit result of three different strategies.

(a) represents the return curve obtained using a fixed empirically set buy-sell threshold, (b) represents the return curve obtained using a semi-supervised learning method, it is our best strategy, and (c) is the return curve obtained by simply setting different periods of fixed buy-sell.

Looking at the curves in strategy (a), we can see that its main gains come after 2021, but it does not accumulate more money in the first 4 years. Strategy (b), on the other hand, not only better captures the market sentiment in 2021, but also manages to earn more money in the first 4 years than (a). Strategy (c) does not make much sense because it does not take advantage of the information related to the market, and we can also find that even if the model makes money in the first 4 years, it will eventually lose almost all of it, but it is able to make money after 2021. The comparison is shown in Table 4.

Table 4. The Result Comparison of Three Different Strategies

| Strategy | rate of return | asset value |
|----------|----------------|-------------|
| (b) | 9718.26% | 97182.58 |
| (a) | 6532.73% | 65327.29 |
| (c) | 823.91% | 8239.11 |

Compared to (a) and (c), (b) is a self-learning model that constantly updates its knowledge of the market, while (c) is a model that lacks sufficient knowledge of the market and will face large retracements and lost opportunities to make big money, which proves that (b) is a relatively optimal model.

5.2 The relationship between daily price and trade score

Another piece of evidence is that our model yields a trade score that is relatively consistent with what we expect to achieve. The higher the indicator, the greater the probability that it is greater than the threshold we have learned, as we should be more inclined to buy and conversely more inclined to sell.

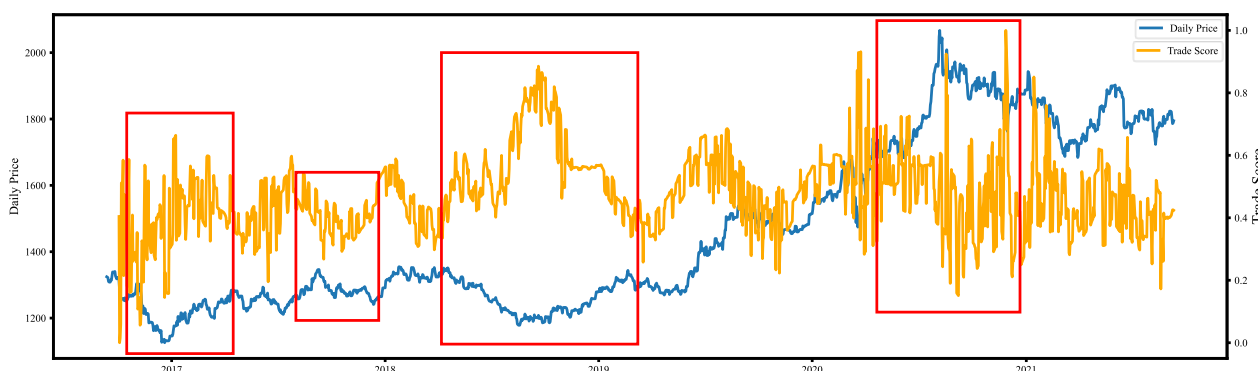


Fig. 19 Two-axis chart of daily price and trade score.

As we can see in Fig. 19, in the marked section, when the daily price is low, the higher our trade score is, predicting a buy, while when the overall price area is high, the lower trade score is, predicting a sell. And a more precise strategy is still determined by the thresholds we learn dynamically.

6. Conclusion

In order to provide a trading strategy for volatile assets, our model will first predict the asset price for the next day. Here we use ARIMA and an LSTM model with the CEEMDAN method for prediction. We have decomposed the price signal and performed noise processing as well as prediction. For trading strategies, we need to use the predicted values to calculate a series of indicators such as Momentum Index, Bias Ratio and Mean price of previous days. The semi-supervised support vector machine method in our model can help us to better determine our position strategy based on the value of the signal. Our model has been run on data from September 11, 2016 and eventually we made over \$97,000 on September 11, 2021 from an initial amount of \$1,000\$ using our forecasting and position adjustment strategies. By comparing with other strategies, we can regard that our trading signal indicators are good indicators of the right moment to buy and sell.

Since our model was tested on real transactions and the data source is relatively old, further improvements may be needed. In addition, there is a certain subjectivity in the selection of indicators. For different types of people, the setting or weight of indicators may be different. In the future, we will consider introducing the characteristics of investment preferences of different groups to improve the model.

Acknowledgments

The work was partially supported by National Natural Science Foundation of China under Grant No. 71971219 and Entrepreneurship Training Program for University Students No. 202321012.

References

- [1] He Y, Li J M, Ruan S M and Zhao S P. A Hybrid Model for Financial Time Series Forecasting—Integration of EWT, ARIMA With the Improved ABC Optimized ELM [J]. *IEEE Access*, 2020, (8).
- [2] Rahimi Z H and Khashei M. A Least Squares-based Parallel Hybridization of Statistical and Intelligent Models for Time Series Forecasting [J]. *Computers & Industrial Engineering*, 2018, (11).
- [3] Bao W, Yue J, Rao Y and Boris P. A deep learning framework for financial time series using stacked autoencoders and long-short term memory [J]. *PLOS ONE*, 2017, 12(7): e0180944.
- [4] Ravi C. Fuzzy Crow Search Algorithm-Based Deep LSTM for Bitcoin Prediction [J]. *International Journal of Distributed Systems and Technologies (IJ DST)*, 2020, 11(4): 53-71.
- [5] Sun Y, Zhao Z J, Ma X B and Du Z H. Short-time Scale Gravitational Microlensing Events Prediction With ARIMA-LSTM and ARIMA-GRU Hybrid Model [J]. *Lecture Notes in Computer science*, 2019, (11473).
- [6] Torres M E, Colominas M, Schlotthuer G and Flandrin P. A Complete Ensemble Empirical Mode Decomposition with Adaptive Noise [J]. *Brain Research Bulletin*, 2011, 125(3): 4144-4147.
- [7] He Y Y, Li P and Han J B. Research on Predictive Modeling on Stock Market Index Based on CEEMDAN-LSTM [J]. *Journal of Statistics and Information*, 2020, 35(06): 34-45.
- [8] Kim K J. Financial time series forecasting using support vector machines [J]. *Neurocomputing*, 2003, 55 (1): 307-319.
- [9] Sakthivel N R, Sugumaran V and Nair B B. Comparison of decision tree-fuzzy and rough set fuzzy methods for fault categorization of mono-block centrifugal pump [J]. *Mechanical Systems & Signal Processing*, 2010, 24 (6): 1887-1906.
- [10] Harris T. Credit scoring using the clustered support vector machine [J]. *Expert Systems with Applications*, 2015, 42(2): 741 -750.
- [11] Li W X, Li J Q, Economics S O and University J. Improvement of Semi-supervised Kernel Clustering Algorithm Based on Multi-factor Stock Selection [J]. *Journal of Statistics and Information*, 2018, 33(03): 30-36.
- [12] Lv K C, Yan H F and Chen C. Quantitative Investment Strategy Based on CSI 300 [J]. *Journal of Guangxi Normal University (Natural Science Edition)*, 2019, 37(01): 1-12.