BIRP: Bitcoin Information Retrieval Prediction Model Based on Multimodal Pattern Matching

Anonymous Author(s)*

ABSTRACT

Financial time series have historically been assumed to be a martingale process under the Random Walk hypothesis. Instead of making investment decisions using the raw prices alone, various multimodal pattern matching algorithms have been developed to help detect subtly hidden repeatable patterns within the financial market. Many of the chart-based pattern matching tools only retrieve similar past chart (PC) patterns given the current chart (CC) pattern, and leaves the entire interpretive and predictive analysis, thus ultimately the final investment decision, to the investors. In this paper, we propose an approach of ranking similar PC movements given the CC information and show that exploiting this as additional features improves the directional prediction capacity of our model. We apply our ranking and directional prediction modeling methodologies on Bitcoin due to its highly volatile prices that make it challenging to predict its future movements.

CCS CONCEPTS

• **Information systems → Retrieval models and ranking**.

KEYWORDS

Pattern Matching, Information Retrieval, Multi-modality, Bitcoin, Trading system, Financial Machine Learning, Feature Engineering

ACM Reference Format:

Anonymous Author(s). 2018. BIRP: Bitcoin Information Retrieval Prediction Model Based on Multimodal Pattern Matching. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX).* ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Since the introduction of Bitcoin (BTC) in 2008 [10], the cryptocurrency (crypto) market has grown significantly. As a result, it attracted many investors and researchers attempting to forecast the movement of these crypto assets in search of profits.

Technical analysis is a discipline that analyzes the statistical transformation of the undelying financial price and volume time series. In this paper, we apply Balance of Power (BOP), Even Better Sinewave (EBSW), Chaikin Money Flow (CMF), and Differencing, Inter Ratios (IR) technical analysis on the historical Bitcoin prices and volume to help rank past chart (PC) movements given the current chart (CC) movements. In this work, we attempt to compare predictive capabilities of various ranking methodologies between PC patterns given CC patterns. Our contribution is mainly two folds:

- We propose four different ranking methodologies (Random Sampling, Euclidean Distance, TS2Vec Embedding Similarity, Multimodal Financial News Incorporated Embedding Similarity) for chart pattern matching and rank similar PC segments based on these metrics.
- We propose a BTC directional forecasting model for trading BTCUSDT Perpetual and show that using voting information from pattern matched chart segments in the past improve the performance of our forecasting model.

2 RELATED WORK

There are numerous researches on how to detect chart patterns such as [6] and [11]. However it was difficult to find literature that uses detected chart patterns for further modeling or using such information to devise a trading strategy. Works such as [11] and [1] uses similar patterns for modeling but they do not apply these techniques specifically in the domain of finance, much less crypto or BTC. Moreover, there are chart pattern detecting applications available for the traders such as that provided by TrendSpider [1] and BTC pattern calculator [2], but again they simply detect the patterns and do not take it a step further. Our research is also closely related to CBITS [8] as we incorporate our Information Retrieval (IR) based feature engineering (FE) technique into the CBITS framework. Furthermore, we use one of the crypto language models (LMs) introduced in CBITS, Crypto DeBERTa, a transformer based LM that improves upon BERT [4] by using disentangled attention and enhanced mask decoder [7] for our multimodal embedding based ranking method.

3 PROBLEM DEFINITION

Our BTC directional modeling framework is very similar to [8]. We approach BTC price directional forecasting task as a three class classification problem. The labels are defined as follows:

$$u_{t+1} = \frac{\text{high}_{t+1} - \text{close}_t}{\text{close}_t}, v_{t+1} = \frac{\text{low}_{t+1} - \text{close}_t}{\text{close}_t}$$

- c_0 : BTC price rises by at least 0.75% within the next 4 hours i.e. $u_{t+1} >= 0.0075$.
- c_1 : BTC price drops by at least 0.75% within the next 4 hours i.e. $v_{t+1} <= -0.0075$.
- c_2 : BTC price change within the next 4 hours is less than 0.75% i.e $u_{t+1} < 0.0075$ and $v_{t+1} > -0.0075$.

The label c_0 translates to long position, the label c_1 translates to short position and c_2 translates to holding (taking no action).

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06...$15.00
https://doi.org/XXXXXXX.XXXXXXX

[1] http://trendspider.com/
[2] https://miningcalc.kr/chart/btc

If both u_{t+1} >= 0.0075 and v_{t+1} <= −0.0075 occur for the next 4 hours, then we gave the labeling priority to c_0 (i.e. when both taking long or short results in at least 0.75% profit then we simply label that timestep as long).

3.1 Dataset Description

We collected 4 hourly BTC/USDT data from Binance [3], one of the largest crypto exchanges in the world, and we ended up with a total of **11,812** data points ranging from **2017-08-23 16:00:00** to **2023-01-15 20:00:00**. After labeling the data we end up with a label distribution of approximately 48.11% for c_0, 29.40% for c_1 and 22.49% for c_2. Out of the 11,812 data points we use the first 80% of the data as candidates (**9,449** data points) for pattern matching and the rest for experimentation. **2,363** data points ranging from **2021-12-18 04:00** to **2023-01-15 20:00:00** were split into train/validation/test data set in 8:1:1 ratio and each of these 2,363 data points were compared with the candidates to find its similar counterparts in the past. For each of the 2,363 data points we collected the **top 30** similar chart patterns for each of the four different similarity calculation methods.

3.2 Modeling

We employ XGBoost[2] as our directional forecasting model as it is fast to train and is robust for tabular data based classification tasks. To highlight some important hyper-parameters, we used 200 for the number of boosting rounds with a learning rate of 0.3. The maximum tree depth for base learners was set to 6 and the tree method was set to "gpu_hist". We also considered the class weights of the train dataset when training XGBoost. Essentially we compare the performance of when we use similar past chart information and when we do not. We will denote these two cases as a system with information retrieval (IR) based feature engineering (FE) and a system without IR-based FE. The overall approach is illustrated in Figure 1. We compare the performances of each methods by calculating accuracy and weighted F1 scores.

3.2.1 Without IR-based FE. When we do not use IR-based FE we simply use chart based features only as inputs to XGBoost for training. Most of the chart features that were used are features focused on calculating volatility or simply ratios of the open, high, low, close and volume features. Although XGBoost does not require feature scaling/normalization, in order to make the training more stable we purposely chose features with similar value ranges.

- **Balance of Power**: The balance of power (BOP) is an oscillator that measures the strength of the buy and sell pressures. When BOP is positive it suggests that the market is bullish and vice-versa. BOP that is close to zero indicates a balance between the two powers and it may signify a trend reversal.
- **Even Better Sinewave**: The even better sinewave (EBSW) is a variation of the Hilbert sine wave, and it is an indicator that can inform the model about the bullish and bearish cycle of prices. The pandas-ta library[4] was used for EBSW calculation.

[3]www.binance.com/en
[4]https://github.com/twopirllc/pandas-ta

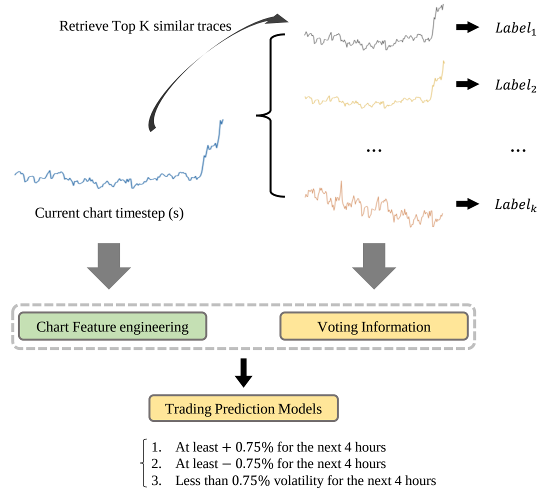

Figure 1: IR-based FE System for modeling BTC directional prediction.

- **Chaikin Money Flow**: The Chaikin money flow (CMF) is an indicator used to monitor both the accumulation and distribution of an asset over a specified period. The default period of 20 is used for CMF.
- **Differencing**: This is identical to the differencing features presented in [8]. It is the ratio of raw chart features across different periods. The first differencing of close prices would be calculated as

$$\text{First Difference of Close} = \frac{\text{close}_t}{\text{close}_{t-1}}$$

and in general the Kth differencing of the close prices is simply

$$\text{Kth Difference of Close} = \frac{\text{close}_t}{\text{close}_{t-K}}$$

Similarly we carry out this differencing procedure for all open, high, low, close and volume and used K = 1,2,...,12 for differencing.
- **Inter Ratios**: This feature is a ratio of different features in the current timestep. Specifically we used $\frac{\text{high}_t}{\text{low}_t}$, $\frac{\text{high}_t}{\text{open}_t}$, $\frac{\text{low}_t}{\text{open}_t}$, $\frac{\text{close}_t}{\text{open}_t}$, $\frac{\text{high}_t}{\text{close}_t}$, $\frac{\text{low}_t}{\text{close}_t}$.

after pre-processing the chart features we end up with a total of 68 features for training.

3.2.2 With IR-based FE. We use the same chart features presented in section 3.2.1 in addition to the vote information from past similar ranked chart features. Given a timestep t, we retrieve at most top 30 timesteps t' that is the most similar to the timestep t under one of the four ranking methods we propose in section 4. Then we separately calculate the performance of the model's directional forecast when we use voting information from the top 5, top 10, top 15, top 20, top 25, top 30 most similar past patterns. Given top K $K \in \{5, 10, 15, 20, 25, 30\}$ similar t', we count the number of cases when the labels were c_0, c_1 or c_2. For example, if $K = 10$ and out of those top K similar instances if 6 of them were c_0 and 3 of them were c_1 and 1 was c_2, then the voting vector can be formed as $(6, 3, 1)$. Before we use this as additional input features to the XGBoost model we softmax normalize these scores. For example, the count for c_0 will be normalized as follows:

$$\text{cnt}(c_0) \rightarrow \frac{e^{\text{cnt}(c_0)}}{e^{\text{cnt}(c_0)} + e^{\text{cnt}(c_1)} + e^{\text{cnt}(c_2)}}$$

Similar transformation is applied to the count for c_1 and c_2. After repeating this procedure for all $K \in \{5, 10, 15, 20, 25, 30\}$ we calculate the average accuracy and weighted F1 for final comparison with the model that does not use IR-based FE.

4 RANKING METHODS

In this section we describe our ranking approaches in detail. For all the ranking methods we ranked the top 30 most similar past timesteps given the query timestep.

4.1 Euclidean Distance

Our data is represented as follows: for timestep t.

$$p_t = [f_{t,1}, f_{t,2}, ..., f_{t,m}]$$

where f denotes a feature and m is the total number of features (in our case 68 as explained in section 3.2.1). Given some timestep $t' < t$ the euclidean distance (L2 norm) is calculated via the following formula:

$$SIM(p_t, p_{t'}) = \sqrt{(f_{t,1} - f_{t',1})^2 + ... + (f_{t,m} - f_{t',m})^2}$$

we retrieve top 30 such p_t' that has the smallest euclidean distance to p_t.

4.2 TS2Vec

The Timeseries to vector (TS2Vec) embedding method was first proposed in [12] and it has proved to effectively extract time series representations that are task agnostic. There are two major components to the TS2Vec architecture:

- The TS2Vec encoder consists of the input projection layer, the timestamp masking layer and the dilated convolutions layer in this order. The timestamp masking layer randomly binary masks the latent vector and this idea was motivated by [3] and [5] to create augmented context views. The encoder is then optimized via the temporal contrast loss and instance-wise contrastive loss.
- The input time series is randomly cropped into two different time series with overlapping timesteps for positive pair creation in an unsupervised setting.

Due to its design the TS2Vec requires a time series of length $l > 1$ and we set $l = 6$, or 24 hours worth of time frame since we are dealing with 4 hourly chart data. We first train the TS2Vec encoder for 100 epochs, batch size 16, learning rate 0.001, hidden dimension size 64 and output dimension size 128 on NVIDIA A100-80GB GPU. The TS2Vec encoder is trained only on the candidate pool and not on the train/validation/test data. Afterwards, we calculate the TS2Vec embeddings for the train/val/test data and all the embeddings for the candidate pool and calculate the cosine distance between the embeddings. We obtain the top 30 embeddings based on the closeness of the cosine distances. Given that the query is $X_t = [x_{t-5}, ..., x_t]$ and the candidate is $X_{t'} = [x_{t'-5}, ..., x_{t'}]$

$$Q_{\text{chart\_emb}} = \text{TS2Vec}(X_t), C_{\text{chart\_emb}} = \text{TS2Vec}(X_{t'})$$

$$\text{SIM}(Q_{\text{chart\_emb}}, C_{\text{chart\_emb}}) = 1 - \frac{Q_{\text{chart\_emb}} \cdot C_{\text{chart\_emb}}}{||Q_{\text{chart\_emb}}||||C_{\text{chart\_emb}}||}$$

4.3 Multimodal

The multimodal method of ranking involves the use of both chart and news data to generate the embeddings for the time series. It uses the same length $l = 6$ like method 4.2 and calculates the TS2Vec embedding of that time series first, then additionally computes the average news embedding between times $t-1$ and t by using Crypto DeBERTa[8]. If the current timestep is t then we use information from timesteps $[t-5, ..., t]$ to calculate the TS2Vec embedding and gather all the news that were released in $[t-1, t]$ (not the entire 24 hours but just the past 4 hours) to calculate the average news embedding in this timeframe. We simply extract the CLS embedding of DeBERTa's output for each news and average these CLS token representations. Then the average news embedding and the TS2Vec embedding are summed to create a multimodal embedding and similar to 4.2, we use the cosine distances of these embeddings to get the top 30 most similar past timesteps. Before summing the average news embedding and the TS2Vec embedding, we shrink the news embedding dimension from $768 \rightarrow 128$ by applying uniform manifold approximation and projection[9]. For the news data we use Coinness Korea[5], which was also mentioned in [8]. Multimodal embeddings should allow us to capture both the news sentiments and chart dynamics when searching for past patterns.

$$\bar{Q}_{\text{news\_emb}} = \frac{1}{N}\sum_{i=1}^{N} \text{LM}(\text{news}_i), Q_{\text{chart\_emb}} = \text{TS2Vec}(p_t)$$

$$\bar{C}_{\text{news\_emb}} = \frac{1}{M}\sum_{i=1}^{M} \text{LM}(\text{news}_i), C_{\text{chart\_emb}} = \text{TS2Vec}(p_{t'})$$

$$Q_{\text{multimodal}} = \text{UMAP}(\bar{Q}_{\text{news\_emb}}) + Q_{\text{chart\_emb}}$$

$$C_{\text{multimodal}} = \text{UMAP}(\bar{C}_{\text{news\_emb}}) + C_{\text{chart\_emb}}$$

$$SIM(Q_{\text{multimodal}}, C_{\text{multimodal}}) = 1 - \frac{Q_{\text{multimodal}} \cdot C_{\text{multimodal}}}{||Q_{\text{multimodal}}||||C_{\text{multimodal}}||}$$

4.4 Random Sampling

Given some query timestep t, the random sampling method samples from all t' such that $t' < t$ randomly. We use random sampling mainly to observe the differences in performance boost when using random sampling versus some other ranking method, thus verifying that our ranking methods do indeed catch patterns from the past that in turn help model BTC price movement. For each query, we randomly sample top 30 past patterns 100 times and calculate the performance of random sampling 100 times with these 100 sampling cases to get a better idea of how random sampling performs.

[5] https://coinness.com/

	No FE	Top 5	Top 10	Top 15	Top 20	Top 25	Top 30	Average
Ranking Strategy 1. Random Sampling								
Accuracy(%)	51.899	54.084	53.873	53.608	53.485	53.840	53.658	53.758
F1 score	0.580	0.598	0.595	0.593	0.592	0.595	0.593	0.595
Ranking Strategy 2. Euclidean Distance								
Accuracy(%)	51.899	51.477	55.274	56.118	52.321	56.540	56.118	54.641
F1 score	0.580	0.579	0.611	0.614	0.583	0.622	0.616	0.604
Ranking Strategy 3. TS2Vec Embedding								
Accuracy(%)	51.899	56.540	58.650	50.211	53.586	51.899	52.743	53.938
F1 score	0.580	0.616	0.635	0.563	0.593	0.578	0.582	0.595
Ranking Strategy 4. Multimodal Embedding								
Accuracy(%)	51.899	57.806	51.899	55.274	56.540	57.384	55.696	55.767
F1 score	0.580	0.628	0.578	0.610	0.615	0.624	0.609	0.610

Table 2: Experiment results of the four ranking strategies.

5 EXPERIMENT RESULTS

5.1 Performance Comparison

We can make some interesting observations from the results in Table 2.

- On average all of our IR based FE improves over the baseline of no FE, with the multimodal strategy performing the best. Using the top 5 most similar multimodal embeddings has the best F1 score of **0.628** and we will be using this model for backtesting in section 5.2. The TS2Vec and multimodal strategies may have improved performance had we used a longer l, but we leave this investigation for future research.
- For our multimodal strategy, we separately calculated the accuracy of the model for the cases when it predicts c_0 or c_1 and when the ground truth action is also c_0 or c_1. Considering this case is important because when the model chooses c_2 it does nothing so it does not incur any profit or loss. If the model chooses c_0 or c_1 but the ground truth is c_2 then even if the model's decision is wrong, it would not result in a huge profit or loss since the volatility for that time frame would be small. Significant gains or losses happen when the model predicts c_0 or c_1 and when the ground truth action is also c_0 or c_1. In this case the multimodal strategy outperforms the no IR based FE baseline by more than 5% on average, while using the top 5 MultiModal IR based FE outperforms no FE by close to 10%.

5.2 Backtest on Test set

The following assumptions were made for back testing: (1) We assume no take profit and a stop loss of 0.75%. (2) We use commission rate of 0.04%, equivalent to the maker fee when trading BTC/USDT perpetual in Binance. As the result in figure 2 shows, our model outperforms buy and hold for the duration of the test set. It is notable that the model predicts c_1 (short) effectively when the BTC prices are falling (e.g. around index 50), gaining edge over buy and hold. Also the model predicts c_2 (hold) very well (e.g. around index 100) for periods when there is less volatility and also predicts c_0 (long) around index 200 when the prices actually began to soar.

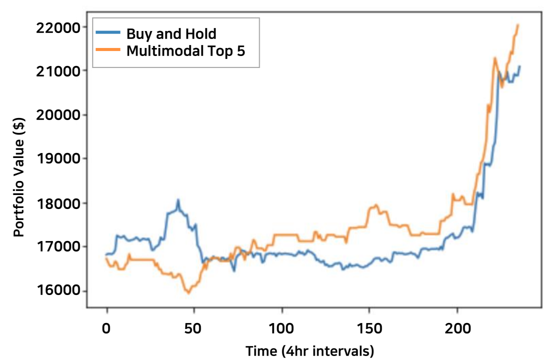

Figure 2: Backtest results.

6 CONCLUSION AND FUTURE WORK

In this research, we investigated how chart pattern matching can be incorporated into a BTC directional prediction model training framework. We presented four different methods of ranking similar chart traces from the past and used the voting information of the labels for the ranked chart data as additional input features to the model. Among these the multimodal embedding based ranking method is the most effective in improving the model's performance and we observed an improvement in accuracy of the directional forecasts. Furthermore the model trained with multimodal embedding based ranking strategy outperformed buy and hold in the test set. Chart pattern matching based feature engineering seems promising and it can be further explored or coupled with other modeling techniques to more accurately forecast the volatile price movement of BTC.

REFERENCES

[1] Naresh Avva. 2015. Pattern-Aided Regression Modelling and Prediction Model Analysis. (2015).

[2] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. (2016).

[3] Ting "Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. "A Simple Framework for Contrastive Learning of Visual Representations". (2020).

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. (2018).

[5] TianYua Gao, XingCheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. (2021).

[6] Xueyuan Gong and Yain-Whar Si. 2013. Comparison of subsequence pattern matching methods for financial time series. In *2013 Ninth International Conference on Computational Intelligence and Security*. IEEE, 154–158.

[7] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. (2020).

[8] Gyeongmin Kim, Minsuk Kim, Byungchul Kim, and Heuiseok Lim. 2023. CBITS: Crypto BERT Incorporated Trading System. *IEEE Access* (2023).

[9] Leland McInnes, John Healy, and James Melville. 2018. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. (2018).

[10] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review* (2008), 21260.

[11] Marc Velay and Fabrice Daniel. 2018. Stock chart pattern recognition with deep learning. *arXiv preprint arXiv:1808.00418* (2018).

[12] Zhihan Yue, Yujing Wang, Duan Juanyong, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. [n. d.]. "Towards Universal Representation of Time Series". *AAAI22* ([n. d.]).

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009