Journal of Combinatorial Optimization (2023) 45:4
https://doi.org/10.1007/s10878-022-00949-9

Bitcoin daily price prediction through understanding blockchain transaction pattern with machine learning methods

Xiao Li[1] · **Linda Du**[2]

Accepted: 18 October 2022 / Published online: 16 November 2022

Abstract
Bitcoin has became one of the most popular investment asset recent years. The volatility of bitcoin price in financial market attracting both investors and researchers to study the price changing manners of bitcoin. Existing works try to understand the bitcoin price change by manually discovering features or factors that are assumed to be reasons of price change. However, the trivial feature engineering consumes human resources without the guarantee that the assumptions or intuitions are correct. In this paper, we propose to reveal the bitcoin price change through understanding the patterns of bitcoin blockchain transactions without feature engineering. We first propose k-order transaction subgraphs to capture the patterns. Then with the help of machine learning models, Multi-Window Prediction Framework is proposed to learn the relation between the patterns and the bitcoin prices. Extensive experimental results verify the effectiveness of transaction patterns to understand the bitcoin price change and the superiority of Multi-Window Prediction Framework to integrate multiple submodels trained separately on multiple history periods.

Keywords Bitcoin · Blockchain · Machine learning · Transaction graph

---

A preliminary version of this work has been published in the proceeding AAIM 2021 (Li and Du 2021).

✉ Xiao Li
Xiao.Li@utdallas.edu

Linda Du
Yingfan.Du@mccombs.utexas.edu

1 The University of Texas at Dallas, Richardson, TX 75080, USA

2 The University of Texas at Austin, Austin, TX 78712, USA

1 Introduction

Since the bitcoin blockchain was announced by Nakamoto in 2008 (Nakamoto 2009), the amazing distributed data processing ability has attracted increasing attention of both public and researchers. The *bitcoin* is originally defined as a special token, namely cryptocurrency, traded in the Bitcoin blockchain, which is a reward to the miner for completing the ledger recording job.

Bitcoin becomes a popular product in public financial market because it can be traded with other products e.g. stocks, gold and crude oil or regular currencies such as US Dollar or British Pound (Vassiliadis et al. 2017). Different from other products, bitcoin is of strikingly volatility in price (Aalborg et al. 2019; Balcilar et al. 2017). Investors have devoted huge amount of investment into bitcoin market seeking for the profiting opportunity from the volatility of bitcoin price (Yermack 2013).

Bitcoin price forecasting models are eagerly desired to provide the suggestions on whether the bitcoin price will rise or fall (Chen et al. 2020b; Yao et al. 2019; Koo and Kim 2021) to help investors make decisions that whether they should buy in or to sell out bitcoins. However, bitcoin price forecasting models usually require well-designed features to understand the reason of bitcoin price change, which is a challenging task. The basic bitcoin price forecasting models only aim to predict the price trend. The results of trend prediction can only present limited information to investors, while investors always desire more accurate and more informative suggestions on the price, so that investors can make further analysis to evaluate how much the price change will impact on their assets. For example, if a trend prediction system suggests that the price will drop, investors may be panic and sold out all their bitcoins. However, if they can know that the price will only drop slightly, investors may choose to wait for a future revival, which can avoid the loss of their asset. However, there are only handful works studying the accurate price prediction problem to predict the exact prices of bitcoin (Abay et al. 2019; Akcora et al. 2018; Mallqui and Fernandes 2019; Cerda and Reutter 2019). Therefore in this paper, we focus on the accurate bitcoin price prediction problem that is much more critical and useful in practice.

To make the predictions of price, machine learning models are widely adopted. Well-designed features are fed into the sophisticated machine learning models. Existing work has proposed various features to encode the latent properties behind the Bitcoin price change from multiple aspects.

Some intuitive features of bitcoin comes from bitcoin blockchain that are the indexes reflecting the transaction information, such as mean degree of addresses, number of new addresses and total coin amount transferred in transactions (Abay et al. 2019). Maesa et al. (2016) build users transferring graph, and analyze the latent features of bitcoin blockchain from a graph perspective. Gyamerah (2021) create 30 technical indicators to represent bitcoin price feature, and Support Vector Regression (SVR) model is applied to make price prediction after the indicators are selected by designed algorithm. Mallqui and Fernandes (2019) consider the bitcoin price is also related to current status of global financial market. They take into account economic indicators to reflect the features of the global financial market, such as crude oil future prices, gold future prices, S&P500 future, NASDAQ future, and DAX index, which are features from financial perspective. Cerda and Reutter (2019) and Shahzad et al.

(2021) introduce public opinion features into bitcoin price prediction through mining the sentiment from social media like Twitter. Yao et al. (2019) attempt to learn the opinion features from news articles. The assumption lays behind these public opinion features is that people's action of buying or selling bitcoin and bitcoin's market value is impacted by the positive and negative opinions delivered by public through news articles and social medias.

Though features from many aspects has been investigated in literature, including blockchain network, financial market information, and even public opinions, it is still unclear what features or factors are useful, and how the bitcoin price is influenced by those features. In addition, manually discovering or creating the features not only relies on heuristics but also consumes notable amount of labour resource. In this paper, we develop features directly from bitcoin blockchain transactions without engineering features outside bitcoin blockchain, e.g. financial market information, and public opinions. The reason that we can abandon those features outside blockchain is that if the external factors, such as public sentiment or news, contribute to the bitcoin price change, they will eventually be reflected by the changes in the transactions and transaction structures of the Bitcoin blockchain. In other words, no matter how the the actions of people are impacted by those features, the different actions taken by people will be reflected by the changes in the Bitcoin blockchain. In this paper, we emphasize that the structure of Bitcoin blockchain encodes abundant transaction patterns information that can interpret the impacts of various factors or features on the bitcoin price change.

We propose blockchain transaction graph to understand transaction patterns. The blockchain transaction graph reveals the patterns of transactions in bitcoin blockchain which is capable to depict market trend and status. As mentioned in Akcora et al. (2018), if the input addresses of a transaction are more than the output addresses, then the transaction is gathering bitcoins, indicating more investment is done. On the other hand, if the input addresses of a transaction are less than the output addresses, then the transaction is distributing the bitcoins, indicating some users leaving the bitcoin market. Therefore by discovering these transaction patterns with proposed bitcoin transaction graph and the prediction framework, we can extract and use valuable information in bitcoin blockchain transaction history that can hardly be discovered by manual feature engineering.

To effectively mining the transaction patterns, we employ the transaction graph to represent the bitcoin blockchain transaction flow. We further propose the k-order transaction subgraph to encode the transaction pattern. With different k, different level of transaction patterns can be captured. Finally the pattern occurrence matrix is proposed to store the frequency of the patterns occurred in blockchain, which can represent the features of blockchain for a given period.

The main contributions of the paper can be summarized as follows:

- We extend the concept of the transaction graph in literature to ***k*-order Transaction Subgraph**, to represent the transaction feature of blockchain at different scopes.
- We propose a transaction subgraph based feature to encode the implicit patterns behind the transactions, then analyze the feature from both semantic view and mathematical view.

– We propose a novel machine learning-based ***Multi-Window Prediction Framework*** that can effectively learn the features from different history periods.
– We evaluate the proposed method on real bitcoin price historical data, and the results demonstrate the superiority comparing to recent state-of-the art methods.

The rest of this paper is organized as follows: First, we review related recent work in Sect. 2. Then in Sect. 3, we propose the k-order transaction subgraph and transaction subgraph feature from both semantic and mathematical views. Next, Sect. 4 presents the Multi-Window Prediction Framework that takes the subgraph features as feed. In Sect. 5 we evaluate the effectiveness of proposed subgraph feature and the prediction framework. Finally, Sect. 6 concludes this paper.

2 Related work

Though blockchain technology is now popular and has been applied in many real-world scenarios (Ding et al. 2022, 2021; Guo et al. 2021b, c, 2022; Luo et al. 2020). The study on the first blockchain system, namely Bitcoin blockchain system is never stopped. One of the most active research topic is bitcoin price prediction. The key issue of bitcoin price prediction or forecasting task is to discover and analysis determinants of bitcoin price. Since Kristoufek (2013) studied the connection between Bitcoin and search queries on Google Trends and Wikipedia, the determinants study has developed rapidly. Influence of social media or public opinions are also studied (Cerda and Reutter 2019; Yao et al. 2019; Mallqui and Fernandes 2021). Balfagih and Keselj (2019) extensively explored the relationship between bitcoin tweets and the prices, which utilizes different language modeling approaches, such as tweet embedding and N-Gram modeling. Mittal et al. (2019) find that there is a relevant degree of correlation of Google Trends and Tweet volume data with the price of Bitcoin, while no significant relation with the sentiments of tweets is discovered. Burnie and Yilmaz (2019) analyzed particular relevance of topics on social media for the high volatility of bitcoin price. Guo et al. (2021a) take the features from both Google Trends and bitcoin blockchain transaction records. However only the transaction amount of each single transaction is considered in the model ignoring many other information encoded in the transactions.

Ciaian et al. (2016) are the first to studies Bitcoin price formation by considering both traditional features in market and digital currencies specific factors from the economical aspects. Aggarwal et al. (2019) attempt to compare the effects of determinants including bitcoin factors, social media and the Gold price. Pieters and Vivanco (2017) study the difference in Bitcoin prices across 11 different markets, and present that standard financial regulations can have a non-negligible impact on the market for Bitcoin.

Georgoula et al. (2015) and Kristoufek (2015) studies the difference of long-term and short-term impact of the determinants on bitcoin price. Kristoufek (2015) points out that time and frequency are both crucial factors for Bitcoin price dynamics since the bitcoin price evolves overtime, and examines how the interconnections from various sources behave in both time and different frequencies. Chen et al. (2020a) analyze the dependence structure between price and its influence factors, and based on copula theory, the bitcoin price has different correlation structures with influence factors.

Bitcoin Blockchain structural information is also mined for discovering the features and determinants of the bitcoin prices. Akcora et al. (2018) propose a bitcoin graph model, where Chainlets is used to represent graph structures in the Bitcoin. A k-chainlet is defined as a subgraph of a bitcoin blockchain that contains exactly k transaction nodes. Akcora et al. (2018) employ both the features derived from chainlets and heuristic features to fed in machine learning model for price prediction. In Akcora et. al's further work (Abay et al. 2019), they propose occurrence matrix and amount matrix to encode the topological features of chainlets.

In this paper, though we adopt the same concept of occurrence matrix to encode the topological features, we design a totally different graph representation model, namely k-order transaction graph to reveal the topological features of Bitcoin Blockchain.

There are also several theoretical (Kyle 1985; Llorente et al. 2002; Schneider 2009) and empirical studies (Balcilar et al. 2017; Koutmos 2018; Naeem et al. 2020) that have looked at Bitcoin transactions focusing on the volume-return causality in the Bitcoin market. However, these studies mostly focus on trading volumes or number of unique bitcoin transactions and utilize traditional regression techniques. In this paper, we extract patterns from bitcoin transactions using the graph models and take our analysis further with machine learning techniques.

The determinants can be considered as features behind the bitcoin price change, then various machine learning methods can be adopted to learn the patterns from the features and make bitcoin price forecasting (Chen et al. 2020b; Yogeshwaran et al. 2019; Sin and Wang 2017). Felizardo et al. (2019) and Chen et al. (2019) compare several popular machine learning methods in bitcoin price prediction task that ARIMA model performs better than Neural network when the price is relatively stable and support vector machine regression is generally the best model among all machine learning methods. Hashish et al. (2019) try to tackle the volatility of cryptocurrencies with Hidden Markov Models and Long Short Term Memory (LSTM) network is adopted to predict the future price movements. Shin et al. (2021) use Ensemble-based LSTM taking history bitcoin price as input, that can predict minute-level bitcoin price. Guo et al. (2021d) combine Multi-scale Residual Convolutional (MRC) block with LSTM fed with bitcoin history price as well as external features such as S&P 500 Index, GVZ, VIX. Nguyen and Le (2019) propose hybrid methods between ARIMA and machine learning to predict the bitcoin next day price. Rajakumar et al. (2022) propose to use Deep Brief Network to solve minute-by-minute bitcoin price. Cavalli and Amoretti (2021) use Convolutional Neural Network (CNN) to predict bitcoin price trend.

3 *k*-order transaction subgraph and subgraph occurrence matrix

Existing works mostly manually define the features based on domain knowledge for the prediction. Although the transparently defined feature are easy to interpret, manually defining the feature is not possible to mining the latent information in the bitcoin blockchain. In order to capture blockchain transaction features, we first define ***transaction graph*** to represent the blockchain transaction information.

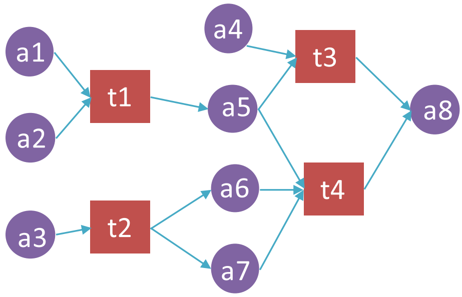


Fig. 1 A simple transaction graph

There are preliminary concepts similar to transaction graph in previous literature (Abay et al. 2019; Maesa et al. 2016), here we formally define the transaction graph as following.

Definition 1 *(Transaction Graph) :* A transaction graph is a directed bipartite graph $G = (A, T, E)$, where A is the set of addresses in Blockchain, T is the set of transactions of blockchain, and E is the set of direct link from $a_i \in A$ to $t_k \in T$, indicating a_i is one of the inputs of t_k, or from $t_k \in T$ to $a_j \in A$, indicating a_j is one of the outputs of t_k.

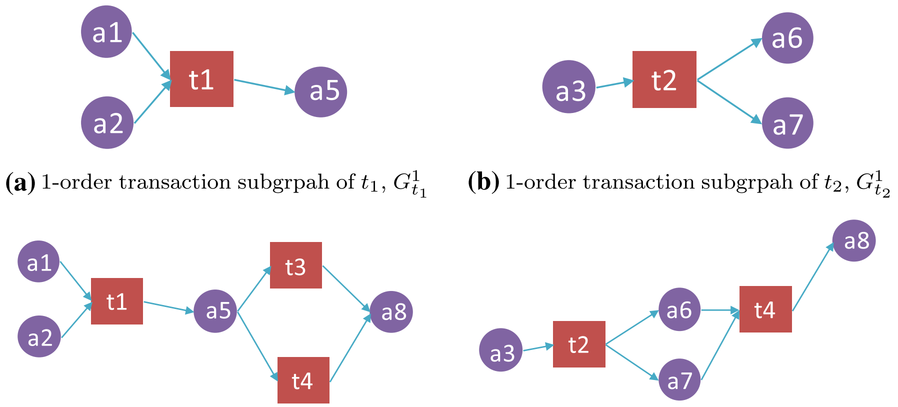
Figure 1 presents an example of a transaction graph formed by 8 addresses and 4 transactions where the directed links represent the flow of bitcoins.

3.1 k-order transaction subgraph

To specify characteristic of each transaction in the transaction graph under different length of bitcoin flow, we define the k-order transaction subgraph of each transaction. For a transaction t_i, we define the k-order transaction subgraph of t_i as a special transaction graph $G_{t_i}^k$ that contains only t_i and the transactions that spend the outputs of t_i in next $k-1$ steps, along with the corresponding addresses connecting to these transactions. The formal mathematical definition is given in Definition 2.

Definition 2 *(K-order transaction subgraph) :* The K-order transaction subgraph of a transaction t_i is a graph $G_{t_i}^k = (A^k, T^k, E^k)$, where $T^k = \{t_j \,|\, \exists\, a_n \in A^{k-1},\ (a_n, t_j) \in E\ and\ \exists (t_l, a_n) \in E^{k-1}\ for\ t_l \in T^{k-1}\}$, $A^k = \{a_n | a_n \in A^{k-1}\ or\ (t_j, a_n) \in E\ where\ t_j \in T^k\}$. Specially, if $k = 1$, $G_{t_i}^1 = (A^1, T^1, E^1)$, where $A^1 = \{a_n | (a_n, t_i) \in E\ or\ (t_i, a_n) \in E\}$, $T^1 = \{t_i\}$ and $E^1 = \{(a_n, t_i)\ or\ (t_i, a_n) | a_n \in A^1\}$.

When $k = 1$, the K-order transaction subgraph of t_i contains only t_i along with its input addresses and output addresses. With k increases, the k order transaction subgraph includes more succeeding transactions of t_i, which will trace further along with the bitcoin flow output by transaction t_i. Figure 2 shows some examples k-order transaction graphs that extracted from Fig. 1. Figure 2a and c shows the 1-order and 2-order transaction subgraph of the transaction t_1, respectively. Figure 2b and d shows the 1-order and 2-order transaction subgraph of the transaction t_2, respectively.



(a) 1-order transaction subgrpah of t_1, $G^1_{t_1}$ **(b)** 1-order transaction subgrpah of t_2, $G^1_{t_2}$

(c) 2-order transaction subgrpah of t_1, $G^2_{t_1}$ **(d)** 2-order transaction subgrpah of t_2, $G^2_{t_2}$

Fig. 2 The 1 order and 2-order transaction subgraph of t_1 and t_2 in Fig. 1

The k-order transaction subgraphs have different patterns by considering the different structures of them. To differentiate the patterns, we use the number of inputs and outputs addresses of the k-order transaction subgraphs. If two k-order transaction graphs have the same number of input addresses and the output address, we consider they are of the same pattern, since they are showing the similar bitcoin flow and similar real-word trading actions.

Briefly, the input addresses of a k-order transaction subgraph $G^k_{t_i}$ are the addresses that give inputs to the transaction t_i, the output addresses of $G^k_{t_i}$ are the addresses that accepts the outputs of the transactions at the last hop in $G^k_{t_i}$. Definition 3 presents the formal definition of input and output address of a K-order transaction subgraph.

Definition 3 *(Input and Output addresses of K-order transaction subgraph) :* The input address set and output address set of K-order transaction subgraph $G^k_{t_i}$ is $\mathcal{I}_{G^k_{t_i}}$ and $\mathcal{O}_{G^k_{t_i}}$, respectively. $\mathcal{I}_{G^k_{t_i}} = \{a_n | \exists (a_n, t_j) \in E^k, t_j \in T^k \textit{ and } \forall t_k \in T^k, (t_k, a_n) \notin E^k\}$. $\mathcal{O}_{G^k_{t_i}} = \{a_n | \exists (t_k, a_n) \in E^k, t_k \in T^k \textit{ and } \forall t_j \in T^k, (a_n, t_j) \notin E^k\}$.

For example, for the 1-order transaction subgraph of t_1 in Fig. 2a, the input addresses are $\{a_1, a_2\} = \mathcal{I}_{G^1_{t_1}}$, and the output address is $\{a_5\} = \mathcal{O}_{G^1_{t_1}}$. For higher orders of transaction subgraph such as 2-order transaction subgraph $G^2_{t_1}$ of t_1 in Fig. 2c, the input and output addresses may not as intuition. Specifically, $\{a_1, a_2\} = \mathcal{I}_{G^2_{t_1}}$ are the input addresses of $G^2_{t_1}$, $\{a_8\} = \mathcal{O}_{G^2_{t_1}}$ is the corresponding output address. Please note that a_5 is not an input nor an output address, the function of a_5 in $G^2_{t_1}$ is only for transition of Bitcoins. Similarly, in Fig. 2d, the input addresses set of $G^2_{t_2}$ is $\{a_3\} = \mathcal{I}_{G^2_{t_2}}$, and the output addresses set of $G^2_{t_2}$ is $\{a_8\} = \mathcal{O}_{G^2_{t_2}}$.

Following the concepts of input addresses set $\mathcal{I}_{G^k_{t_i}}$ and out addresses set $\mathcal{O}_{G^k_{t_i}}$ of k-order transaction graph $G^k_{t_i}$, we now can define the ***pattern*** of $G^k_{t_i}$. The k-order transaction graph of transaction t_i, e.g. $G^k_{t_i}$ belongs to the ***pattern***: $G^k_{(m,n)} = \{G^k_{t_i} || \mathcal{I}_{G^k_{t_i}}| = m, |\mathcal{O}_{G^k_{t_i}}| = n\}$, where $| \cdot |$ denotes the size of a set.

For a given specific period T, we can create a transaction graph $G = (A, T, E)$ to represent the transactions among addresses. Then for each transaction $t_i \in T$, we can derive k-order transaction subgraph $G^k_{t_i}$ for different k. Those obtained k-order transaction subgraphs may belong to different patterns represent different real-world trading scenarios, policies or actions. For the example in Fig. 2, the 2-order transaction subgraph of t_1, $G^2_{t_1}$ belongs to the pattern $G^2_{(2,1)}$, while the 2-order transaction subgraph of t_2, $G^2_{t_2}$ belongs to the pattern $G^2_{(1,1)}$.

We believe these different patterns are able to reveal the characteristics of each transaction in corresponding blockchain during a given period. Moreover, the patterns obtained under different order k can reveal different levels of such information since larger k allows the pattern to trace farther transactions. We denote the pattern based on the number of input addresses and output addresses to make the patters easily encoded into matrices, and be easily adopted as the features of the corresponding transaction graph.

To understand the characteristics bitcoin blockchain during a period, we propose to discover two key factors from patterns of all k-order transaction graph $G^k_{t_i}$ of every transaction t_i in transaction graph G 1) the different types of transaction graph patterns occurred during that period, and 2) the occurrence time of these different patterns. To obtain these two factors simultaneously, We extend the general concept of "occurrence matrix" (Abay et al. 2019) to k-order pattern occurrence matrix. The k order pattern occurrence matrix of a blockchain in a period is denoted as OC^k, where the entry of OC^k is $OC^k_{(m,n)} = |G^k_{(m,n)}|$ which is the number of k-order transaction graphs belonging to the pattern $G^k_{(m,n)}$.

For different k, multiple k-order pattern occurrence matrices can be obtained. We unfold the k-order pattern occurrence matrices to concatenate OC^k for $k = 1, 2, 3, \ldots, s$ as the feature v of blockchain during a given period. In this paper, we aim to study the *Bitcoin Price Prediction* problem by understanding the transaction graph feature v: use the feature vector v that is derived the transaction graph of the bitcoin blockchain historical data in a given period $[t - s, t]$, to predict the bitcoin price at future timestamp $t' = t - s + h$, $P_{t'}$. h is the prediction horizon. Formally, the Bitcoin Price Prediction problem in this paper is defined as Definition 4.

Definition 4 *(Bitcoin Price Prediction) :* Given historical bitcoin price and bitcoin blockchain data in time period $[t - s, t]$, where $s \in N^+$. Let P_t denotes the price of bitcoin at the timestamp t. The bitcoin price prediction problem is to predict the future bitcoin price at timestamp $t' = t + \Delta t$, where $\Delta t \geq 0$, e.g. $P_{t'}$.

3.2 Computation of occurrence matrix

The above sections give a comprehensive interpretation of the occurrence matrix. In this section, we propose an iterative manner for applicable implementation by multiplying matrices to efficiently compute the occurrence matrix.

Fig. 3 The H matrix of transaction graph in Fig. 1

	t1	t2	t3	t4
{a1,a2}	1			
{a3}		1		
{a4,a5}			1	
{a6,a7}				1

Fig. 4 The P matrix of transaction graph in Fig. 1

	t1	t2	t3	t4
a1	1			
a2	1			
a3		1		
a4			1	
a5			1	1
a6				1
a7				1
a8				

Fig. 5 The Q matrix of transaction graph in Fig. 1

	a1	a2	a3	a4	a5	a6	a7	a8
t1					1			
t2						1	1	
t3								1
t4								1

Let $H \in \mathbb{R}^{|T|\times|T|}$ be the matrix denoting the input addresses of each transaction, The entry of H is $H_{A_i,t_i} = 1$, if A_i is the set of input addresses of transaction t_i, otherwise $H_{A_i,t_i} = 0$. Figure 3 shows the H matrix of transaction graph in Fig. 1.

Let $P \in \mathbb{R}^{|A|\times|T|}$ be the matrix denoting the input relationship between each address a_i and each transaction t_j. $P_{i,j} = 1$ if a_i is one of the input addresses of transaction t_j, otherwise, $P_{i,j} = 0$. Figure 4 shows The P matrix of transaction graph in Fig. 1.

Then let $Q \in \mathbb{R}^{|T|\times|A|}$ be the matrix denoting the output relationship between each address a_j and each transaction t_i. $Q_{i,j} = 1$ if a_j is one of the output addresses of transaction t_i, and $Q_{i,j} = 0$ otherwise. Figure 5 shows The Q matrix of transaction graph in Fig. 1.

For calculating the k order occurrence matrix OC^k, we first need to derive the transition matrix $M \in \mathbb{R}^{|T|\times|A|}$ for the k order transaction graph, which is derived through Eq. (1).

$$M^k = H(QP)^{k-1}Q \tag{1}$$

The entry of matrix M^k, $M^k_{A_i,a_j} > 0$ if there is a flow from transactions t_i, whose input addresses set is A_i, to address a_j, otherwise, $M^k_{A_i,a_j} = 0$. In fact, we can easily

$$M^1 = H(QP)^0Q = \begin{array}{c|cccccccc|} & a1 & a2 & a3 & a4 & a5 & a6 & a7 & a8 \\ \hline \{a1,a2\} & & & & & 1 & & & \\ \{a3\} & & & & & & 1 & 1 & \\ \{a4,a5\} & & & & & & & & 1 \\ \{a6,a7\} & & & & & & & & 1 \\ \hline \end{array}$$

Fig. 6 1 order transition matrix M^1 of transaction graph in Fig. 1

$$M^2 = H(QP)^1Q = \begin{array}{c|cccc|} & t1 & t2 & t3 & t4 \\ \hline \{a1,a2\} & 1 & & & \\ \{a3\} & & 1 & & \\ \{a4,a5\} & & & 1 & \\ \{a6,a7\} & & & & 1 \\ \hline \end{array} \times \begin{array}{c|cccc|} & t1 & t2 & t3 & t4 \\ \hline t1 & & & 1 & 1 \\ t2 & & & & 2 \\ t3 & & & & \\ t4 & & & & \\ \hline \end{array} \times \begin{array}{c|cccccccc|} & a1 & a2 & a3 & a4 & a5 & a6 & a7 & a8 \\ \hline t1 & & & & & 1 & & & \\ t2 & & & & & & 1 & 1 & \\ t3 & & & & & & & & 1 \\ t4 & & & & & & & & 1 \\ \hline \end{array}$$

$$= \begin{array}{c|cccccccc|} & a1 & a2 & a3 & a4 & a5 & a6 & a7 & a8 \\ \hline \{a1,a2\} & & & & & & & & 2 \\ \{a3\} & & & & & & & & 2 \\ \{a4,a5\} & & & & & & & & \\ \{a6,a7\} & & & & & & & & \\ \hline \end{array}$$

Fig. 7 2 order transition matrix M^2 of transaction graph in Fig. 1

understand that the $M^k_{A_i,a_j}$ denotes how many possible path from transaction t_i to address a_j in the k order transaction graph of transaction t_i.

Therefore $|A_i|$ is the number of input addresses of the k order transaction graph of t_i, and $\sum_{a_j \in A} \mathbb{I}\{M^k_{A_i,a_j} > 0\}$ is the number of output addresses of the k order transaction graph of t_i. $\mathbb{I}\{*\} = 1$ if the condition $*$ is satisfied, and $\mathbb{I}\{*\} = 0$ otherwise.

Now each entry of OC^k, $OC^k_{(m,n)}$, can be calculated based on the k order transition matrix M^k through Eq. (2).

$$OC^k_{(m,n)} = \sum_{A_i} \mathbb{I}\{|A_i| = m \ \& \sum_{a_j \in A} \mathbb{I}\{M^k_{A_i,a_j} > 0\} = n\}, \tag{2}$$

where A_i is the set of input addresses of transaction t_i, namely $A_i = \{a_k|(a_k, t_i) \in E\}$.

For the simple example in Fig. 1, if $k = 1$, the transition matrix M^1 is illustrated in Fig. 6. Then the occurrence matrix OC^1 can be easily derived.

$OC^1_{(2,1)} = 3$, because $|\{a1, a2\}| = |\{a4, a5\}| = |\{a6, a7\}| = 2$, and $\sum_{a_j \in A} \mathbb{I}\{M^1_{\{a1,a2\},a_j} > 0\} = \sum_{a_j \in A} \mathbb{I}\{M^1_{\{a4,a5\},a_j} > 0\} = \sum_{a_j \in A} \mathbb{I}\{M^1_{\{a6,a7\},a_j} > 0\} = 1$.

$OC^1_{(1,2)} = 1$, because $|\{a3\}| = 1$, and $\sum_{a_j \in A} \mathbb{I}\{M^1_{\{a3\},a_j} > 0\} = 2$. In addition, all the other entries of OC^1 is 0, since there is no other pattern for the 1-order transactions graphs.

If $k = 2$, the calculation of the transition matrix M^2 is illustrated in Fig. 7. Then the occurrence matrix OC^2 can be calculated as follows . $OC^2_{(2,1)} = 1$, since $|\{a1, a2\}| = 2$ and $\sum_{a_j \in A} \mathbb{I}\{M^1_{\{a1,a2\},a_j} > 0\} = 1$. $OC^2_{(1,1)} = 1$, since $|\{a3\}| = 1$ and $\sum_{a_j \in A} \mathbb{I}\{M^1_{\{a1\},a_j} > 0\} = 1$. All the other entries of OC^2 is 0.

The dimension of occurrence matrices may be different for different k-order, and different transactions. However, occurrence matrices of unified size are required for formatting the feature vector of the same size, so that the features can be fed in machine learning based prediction models. According to literature (Akcora et al. 2018), nearly 97.57% transactions have the inputs and outputs sized no greater than 20. Therefore, for the less than 3% left transactions, whose number of inputs or outputs is greater than 20, we manually set number as 20. The k-order occurrence matrix OC^k now can be defined as Eq. (3).

$$OC^k_{(m,n)} = \begin{cases} \sum_{A_i} \mathbb{I}\{|A_i| = m \,\&\, \sum_{a_j \in A} \mathbb{I}\{M^k_{A_i,a_j} > 0\} = n\}, & m < 20, n < 20, \\ \sum_{A_i} \mathbb{I}\{|A_i| \geq 20 \,\&\, \sum_{a_j \in A} \mathbb{I}\{M^k_{A_i,a_j} > 0\} = n\}, & m = 20, n < 20, \\ \sum_{A_i} \mathbb{I}\{|A_i| = m \,\&\, \sum_{a_j \in A} \mathbb{I}\{M^k_{A_i,a_j} > 0\} \geq 20\}, & m < 20, n = 20, \\ \sum_{A_i} \mathbb{I}\{|A_i| \geq 20 \,\&\, \sum_{a_j \in A} \mathbb{I}\{M^k_{A_i,a_j} > 0\} \geq 20\}, & m = 20, n = 20. \end{cases} \tag{3}$$

4 The proposed bitcoin prediction framework

When predicting the future price, we need to make fully use of historical information. However, how much history should be taken into account is a challenging problem. Specifically, incorporating more features from further history does not necessary bring more accuracy prediction results, sometimes even damage the prediction. We demonstrate that this situation exists in Bitcoin Prediction Task in next section. There is no such a fitted length for historical window where the features are extracted can produce consistently best performance. It is challenging to pick a suitable length of history window where features are extracted, since neither of these models can maintain highest accuracy in prediction.

In this paper, instead of using fixed length of history, We propose *Multi-Window Prediction Framework*, which creates submodels to learn different patterns from different historical periods (window) and integrate all the predictions from submodels as final prediction results.

Figure 8 shows the overview of proposed the *Multi-Window Prediction Framework*. M_1 to M_s are s submodels that are trained separately at different length of history window. In this paper, we consider the granularity of timestamp as a day. Specifically, M_1 is a machine learning model trained by the subgraph features obtained from past 1 day, and M_2 is another machine learning model trained by the features obtained from past 2 days. After all submodels are trained, to make price prediction for a specific future time $t' = t + \Delta t$ ($\Delta t \geq 0$), each submodel is able to make separate prediction, then an integrator is applied combine all the separate predictions into one final prediction.

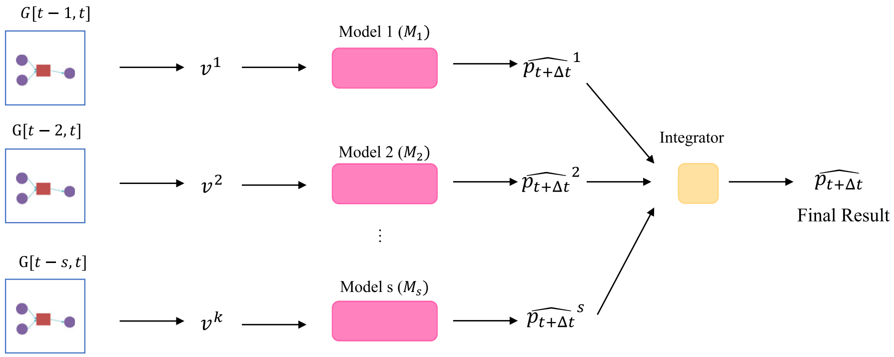
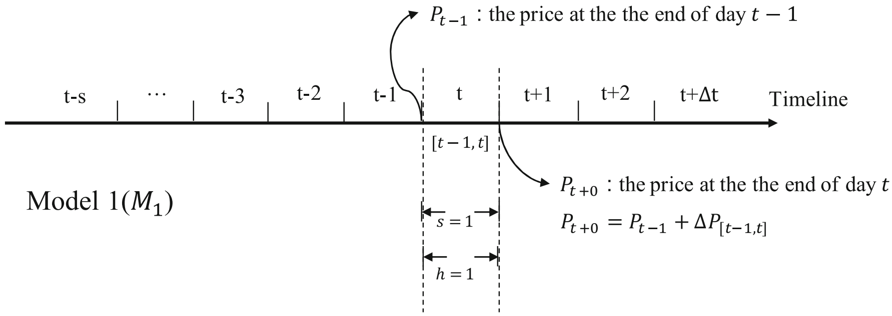


Fig. 8 The overview of multi-window prediction framework

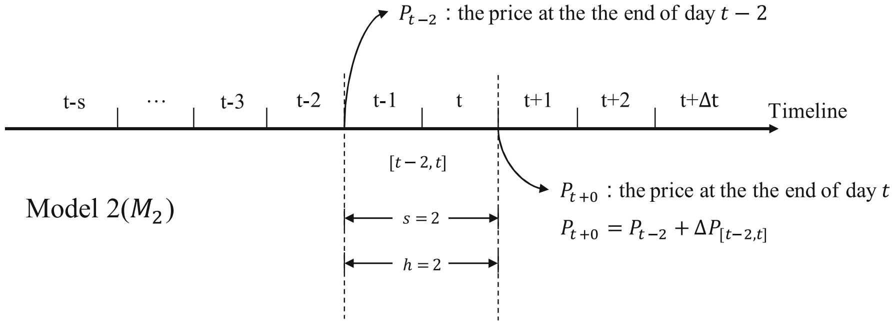
The accuracy of final prediction result apparently depends on the performance of each submodel. Next, we describe how each submodel is trained and make future price prediction. In this paper, we predict the daily end trading price of bitcoin at financial market. The end price of day t' is denoted as $P_{t'}$. Though it is more of intuition to directly predict $P_{t'}$, we argue that it is more reasonable to predict the price difference between $P_{t'}$ and P_{t-s}, denoted as $\Delta P_{[t-s,t']}$, and then calculate the predicted P_t as $\hat{P_t} = P_{t-s} + \Delta \hat{P_{[t-s,t']}}$. The reasons are as following: 1) Since the history price P_{t-s} is known, it should be considered to improve the prediction while traditional way ignores this; 2) The features or patterns occurred during $[t-s, t]$ represent the characteristics only during $[t-s, t]$ in bitcoin market, and those characteristics reflects the reason of price change not the exact price. Therefore, it is more viable to use the obtained features to understand the price change rather than the exact price. In this paper we construct the data sample pairs for model training and testing as $(\mathbf{x}, y)$, where $\mathbf{x}$ is the feature vector extracted from history period $[t-s, t]$, and $y = \Delta P_{[t-s,t']} = P_{t'} - P_{t-s}$. Please note that each submodel will be retrained if it aims to predict different future time since the manner of patterns influencing future price may be different for different future time. We denote the length of future time to be predicted as $h = t' - (t-s) \geq s$, which is also called prediction **horizon**. Figure 9 illustrates several examples of the parameters setting for submodels to make prediction for different future time. In Fig. 9a, submodel M_1 takes features from $[t-1, t]$, to predict $\Delta P_{[t-1,t]}$, and further derive $\hat{P_t}$. Therefore $s = t-(t-1) = 1$ and $h = t-(t-1) = 1$. In Fig. 9b, submodel M_2 takes feature from $[t-2, t]$, and also predicts $\Delta P_{[t-2,t]}$. Therefore $s = t-(t-2) = 2$ and $h = t-(t-2) = 2$. In Fig. 9c, M_2 tries to predict $\Delta P_{[t-2,t+1]}$, hence $s = t-(t-2) = 2$ and $h = (t+1)-(t-2) = 3$.

As mentioned above, an integrator is applied to combine the results from each submodel into the final results. The integrator can be any popular aggregation function such as Min, Max or Average. We define the integrator as a simple linear function that assigns different weights to each submodels, which given by following equation:

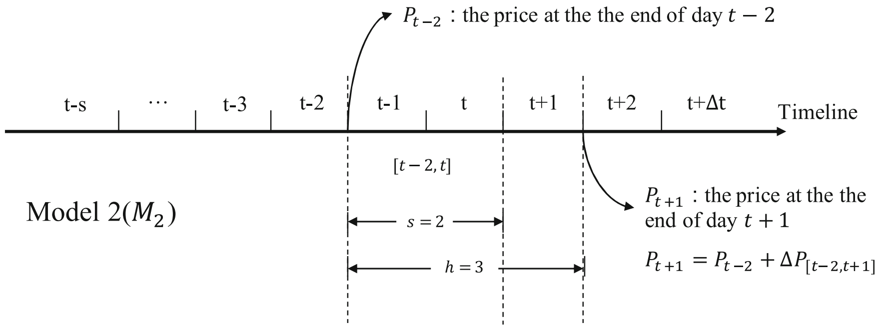
$$\hat{P_{t'}} = r_1 * \hat{P_{t'}}^1 + r_2 * \hat{P_{t'}}^2 + \cdots + r_s * \hat{P_{t'}}^s \tag{4}$$



(a) Settings for subodel 1 (M_1) to predict P_t



(b) Settings for submodel 2 (M_2) to predict P_t



(c) Settings for submodel 2 (M_2) to predict P_{t+1}

Fig. 9 Illustration of settings for different submodels on different prediction task

where $r_1 + r_2 + \cdots + r_s = 1$.

Let W_i be the list of weights in the order $W_i = [r_1, r_2, r_3, \ldots, r_i]$ where $0 \geq i \leq s$. Initially, when the historical window size is 1, that only submodel M_1 is enabled to make the prediction, we define $W_1 = [r_1] = [1.0]$. As the historical window size increases and more submodels are enabled, W_i is defined as Eq. (5):

$$\begin{aligned} W_i[k] &= W_{i-1}[k] \ (0 < k < i-1) \\ W_i[i-1] &= W_{i-1}[i-1] * \alpha \\ W_i[i] &= W_{i-1}[i-1] * (1-\alpha) \end{aligned} \tag{5}$$

where α is the decay factor of weights. Obviously Eq. (5) keeps the property that $\sum_{r_j \in W_i} r_j = 1$ for $i > 0$. α makes the submodel that trained on further history windows has less weights contributing to the final results, which is consistent to the intuition that further history has less impact on the future price, especially for the bitcoin market (Georgoula et al. 2015).

5 Experimental evaluations

In this section, we evaluate the effectiveness of proposed k-order transaction subgraph patterns and the Multi-Window Prediction Framework.

5.1 Data preparation

To conduct the bitcoin price prediction task, the Bitcoin blockchain history data is downloaded from Google Bigquery public dataset $crypto\_bitcoin$.[1] The bitcoin daily closing price history data is provided by Coindesk.[2]

We select 3 intervals for the experiments.

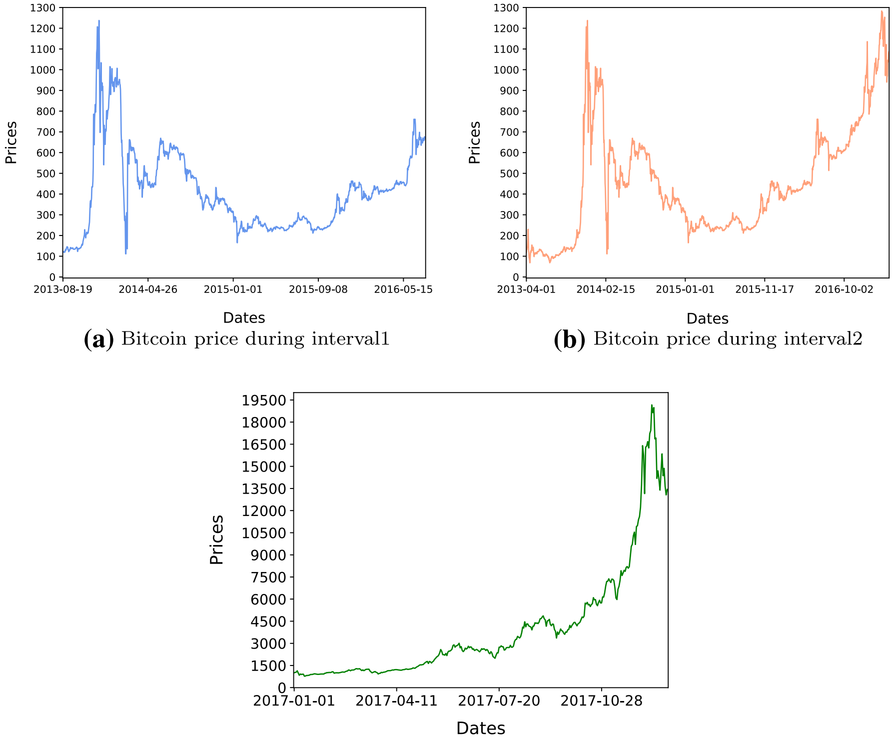
- **Interval 1**: From August 19th, 2013 to July 19th, 2016. This interval has 1065 days, training samples are generated from the first 852 (80%) days, and testing samples are generated from the left 213 (20%) days.
- **Interval 2**: From April 1st, 2013 to April 1st, 2017. This interval has 1461 days, training samples are generated from the first 1022 (70%) days, and testing samples are generated from the left 439 (30%) days.
- **Year 2017**: From January 1st, 2017 to December 31st, 2017. This interval has 365 days, training samples are generated from the first 292 (80%) days, and testing samples are generated from the left 73 (20%) days. This Interval is for demonstrating short term prediction performance with much less training samples.

We adopt two intervals, namely *interval 1* and *interval 2*, and the training/testing ratio settings as the same ones in literature (Mallqui and Fernandes 2019), to create a fair comparison in next sections. The bitcoin price of *interval 1*, *interval 2* and *Year 2017* is presented in Fig. 10a–c, respectively. It is possible to observe, that the bitcoin prices show a high volatility, which indicates that the nature of the bitcoin can hardly be intuitively discovered , therefore the features designed manually may be ineffective.

To evaluate the price prediction accuracy, Mean Absolute Percentage Error (MAPE) is adopted to show the difference between predicted prices and real prices. The MAPE is defined as $MAPE = \frac{1}{N}\sum_{i=1}^{N} \frac{|\hat{p}_i - p_i|}{p_i} \in [0, +\infty)$, where $\hat{p}_i$ is the predicted bit-

[1] Dataset ID is bigquery-public-data: crypto_bitcoin at https://cloud.google.com/bigquery.

[2] https://www.coindesk.com/.



(a) Bitcoin price during interval1

(b) Bitcoin price during interval2

(c) Bitcoin price during 2017

Fig. 10 Bitcoin price at different test intervals

coin price, while p_i is the real price. Obviously, lower MAPE values indicate better prediction accuracy.

Table 1 illustrates how the data samples are generated for training submodels M_1 and M_2 during a given period. One can easily derive the settings to other submodel M_s. After generating all the data samples, the training and testing samples are divided without shuffle according to the ration specified for each dataset described above. All experiments in this paper are done in Python 3.7 at a laptop with Intel Core i7 10875h, 32GB memory and 512G storage. The Neural Network model in experiment is executed with CUDA 11.7 on a RTX 2080 Ti Max-Q graphic card.

5.2 Performance of difference submodel

Tables 3 and 4 show the results of each submodel, M_1 to M_4, adopts the same training strategies while using two different prediction model, namely Support Vector Machine (SVM) and full connected Neural Network (NN) respectively. The detailed model settings are presented in Table 2. The only difference of submodels (M_1 to M_4) in Tables 3 and 4 is the length of historical window when calculating the k-order subgraph features. We set $k = 2$ in all experiments in this article. From the tables, we can

Table 1 Examples for data samples generation

Submodels	Parameter settings	Sample 1	Sample 2	...	Sample n
M_1	Feature extraction period (s = 1)	Day_1	Day_2	...	Day_n
	Target predicting closing price				
	$h = 1$	P_{Day_1}	P_{Day_2}	...	P_{Day_n}
	$h = 2$	P_{Day_2}	P_{Day_3}	...	$P_{Day_{n+1}}$
	...	...	...	...	...
M_2	Feature extraction period (s = 2)	$[Day_1, Day_2]$	$[Day_2, Day_3]$	...	$[Day_n, Day_{n+1}]$
	Target predicting closing price				
	$h = 2$	P_{Day_2}	P_{Day_3}	...	$P_{Day_{n+1}}$
	$h = 3$	P_{Day_3}	P_{Day_4}	...	$P_{Day_{n+2}}$
	...	...	...	...	...

Table 2 The model settings in experiments

Model name	SVM	Neural network
Package	sklearn.svm.SVR	Keras
Parameters	Kernal = rbf Gamma = 0.01 C = 10	Layers: [Inputs,128,64,32,16,1] Activation Function: tanh epoch = 3 k_fold = 2 optimizer = 'rmsprop' loss function = MSE

conclude that including more historical information with bigger s in submodels does not necessarily mean better submodel performance in terms of MAPE. For example in Table 3, M_2 at $h = 2$ results in higher MAPE than M_1 under Interval 1 and Interval 2, even though M_2 takes one more day history information into account. One can identify more similar such cases in Table 4. Therefore it is expected to achieve higher MAPE by taking advantages of different model. By the way, NN model-based prediction can outperform SVM slightly in some cases, such as $h = 4$ in Interval 1, and $h = 2, 4$ in Year 2017, but the margin is not significant.

5.3 Evaluation of α and multi-window prediction framework

Figure 11 shows how the α affect the final prediction of the proposed Multi-Window Prediction Framework. M_1 means only the submodel M_1 is adopted, $M_1 \sim M_2$ means integrator combines the results from both submodels M_1 and M_2, $M_1 \sim M3$ means combining submodels M_1, M_2 and M_3, and so on. For both SVM-based model and NN-based model, when $\alpha > 0.7$ in Interval 1, and $\alpha > 0.75$ in Interval 2, there is at least one combined model outperforms the submodel M_1. This demonstrates that the proposed Multi-Window Prediction Framework can successfully take advantages from submodels and boost the accuracy. Specifically, When $\alpha = 0.85$ the multi-window prediction framework can achieve the most accurate prediction with lowest MAPE value for both SVM-based and NN-based model in both Interval 1 and Interval 2.

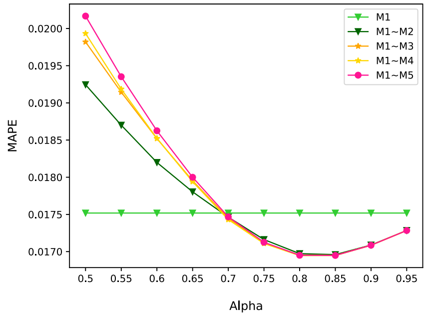
Table 5 shows results of multi-window prediction framework for predict next day bitcoin closing price where $\alpha = 0.85$ and $h = 1$. We first let all submodels trained separately and make prediction on the same testing days. Then the integrator will combine the results from submodels into the final results. Please note $h = 1$ here means to predict the next-day price, for each submodel the settings for s and h may vary according to Fig. 9 to make them predict the same days. In the table, $M_1 \sim M_4$ achieves lowest MAPE values for both Interval 1 and Interval 2. Therefore we can conclude that for predicting next day price, integrating 4 submodels with the longest history window size $s = 4$ is enough for our proposed multi-window prediction framework. The results also verified the intuition that further transaction history impacts less to the future price, which also consistent to the high volatility of bitcoin price. In addition, $M1$ is not related to α, but $M1$ adopting neural network model shows fluctuating MAPE in both Interval 1 and Interval 2, this is because of the instability of neural networks.

Table 3 MAPE of submodels (SVM prediction) for predicting future price

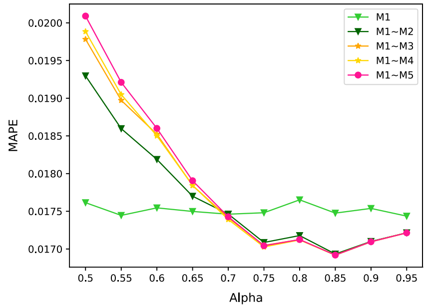
Submodels	Interval 1 (%)					Interval 2 (%)					Year 2017 (%)				
	h=1	h=2	h=3	h=4	h=5	h=1	h=2	h=3	h=4	h=5	h=1	h=2	h=3	h=4	h=5
M1 (s=1)	1.75	2.59	3.15	3.77	4.31	1.74	2.57	3.21	3.78	4.29	4.73	7.09	8.36	10.30	12.20
M2 (s=2)	–	2.61	3.16	3.76	4.29	–	2.58	3.20	3.78	4.29	–	7.01	8.36	10.05	11.90
M3 (s=3)	–	–	3.17	3.76	4.29	–	–	3.20	3.76	4.27	–	–	8.24	9.91	11.70
M4 (s=4)	–	–	–	3.75	4.29	–	–	–	3.78	4.28	–	–	–	9.85	11.60

Table 4 MAPE of submodels (neural network prediction) for predicting future price

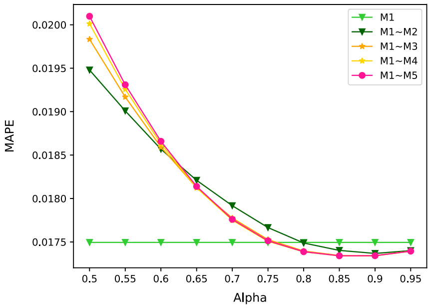
	Interval 1 (%)					Interval 2 (%)					Year 2017 (%)				
	h = 1	h = 2	h = 3	h = 4	h = 5	h = 1	h = 2	h = 3	h = 4	h = 5	h = 1	h = 2	h = 3	h = 4	h = 5
M1 (s = 1)	1.75	2.58	3.15	3.76	4.30	1.74	2.56	3.17	3.75	4.27	4.71	7.02	8.26	10.15	12.06
M2 (s = 2)	–	2.60	3.14	3.74	4.28	–	2.58	3.19	3.76	4.30	–	6.96	8.25	9.88	11.74
M3 (s = 3)	–	–	3.16	3.74	4.28	–	–	3.18	3.74	4.27	–	–	8.23	9.79	11.52
M4 (s = 4)	–	–	–	3.74	4.27	–	–	–	3.76	4.27	–	–	-	9.81	11.50



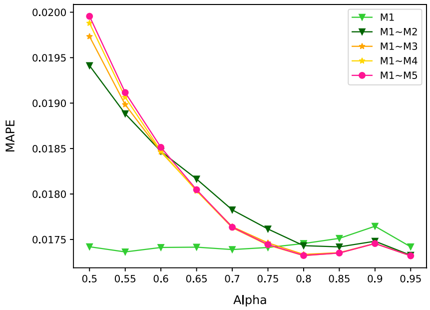
(a) Interval 1: MAPE when sumodels using SVM model



(b) Interval 1: MAPE when sumodels using Nerual Network (NN) model



(c) Interval 2: MAPE when sumodels using SVM model



(d) Interval 2: MAPE when sumodels using Nerual Network (NN) model

Fig. 11 MAPE of multi-window prediction framework when combining different submodels and alpha in interval 1 and interval 2 ($h = 1$)

Table 5 MAPE of multi-window prediction framework when combining different submodels (SVM, $\alpha = 0.85$, $h = 1$)

Submodels integrated	Interval 1 (%)	Interval 2 (%)
M1	1.75	1.74
M1∼M2	1.70	1.73
M1∼M3	1.70	**1.72**
M1∼M4	**1.69**	1.72
M1∼M5	1.70	1.72
Mallqui and Fernandes (2019)-SVM	1.91	1.81

The bold indicates lowest MAPE achieved by least number of models

5.4 Comparison with baseline

Here, we compare our proposed framework with a state-of-art work, namely (Mallqui and Fernandes 2019) where similar next-day bitcoin price prediction task is studied. Mallqui and Fernandes (2019) considers both the factors from Bitcoin blockchain,

Table 6 MAPE of multi-window prediction framework ($M_1 \sim M_4$, SVM, $\alpha = 0.85$)

h	Interval 1 (%)	Interval 2 (%)
1	1.69	1.73
2	3.87	3.95
3	5.20	5.14
4	6.22	6.24
5	6.59	6.56

including history price, volume of trades and financial indicators showing current financial market status such as the Gold price and Nasdaq price. Though multiple machine learning methods are adopted such as regular neural network, regressional neural network and support vector machine (SVM), the SVM regression model presents the best performance in their work. We denoted this best model as Mallqui and Fernandes (2019)-SVM and adopt it for comparison. The results of Mallqui and Fernandes (2019)-SVM are directly taken from their paper since we both conduct the same prediction task on the same intervals. The values are shown at the last line in Table 5. We can observe that both M_1 only and combined models outperform Mallqui and Fernandes (2019)-SVM. Especially, $M1 \sim M4$ achieves 11.5% and 5.0% lower MAPE values than Mallqui and Fernandes (2019)-SVM in Interval 1 and Interval 2 respectively.

5.5 Prediction for different future time

Table 6 shows how the Multi-Window Prediction Framework preforms for predicting different future time. Comparing Tables 6 with 3, though $M_1 \sim M_4$ obtained best prediction accuracy for predicting future time $h = 1$ among any individual submodel, it can not outperform M_1 for any other future time $h > 1$. This indicates that when predicting further future (later than 1 day), introducing more history features into the prediction does not benefit Multi-Window Prediction Framework. This also demonstrates it is extremely hard to predict more than 2-days future price of bitcoin.

6 Conclusion

In this paper, we proposed to understand bitcoin price change though mining the patterns in transactions in Bitcoin blockchain. We first proposed to use transaction graph to represent the relations among transactions. Then k-order transaction graphs is proposed to encode the transaction pattern. Machine learning-based Multi-Window Prediction Framework is further developed to predict bitcoin price fed with k-order transaction graph features, which is able to capture the patterns from multiple history periods by adjusting the history window. Extensive experiments are conducted to demonstrate the advantage of taking multiple history periods into account. The proposed Multi-Window Prediction Framework also outperforms recent state-of-art method on the same intervals, demonstrating the effectiveness of mining the features from transaction graph patterns.

Funding The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Data availability The raw data used in this paper are available by public as described in Sect. 5. The intermediate data generated during and/or analysed during the algorithm process are available at request by contacting with the first author.

Declarations

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Conflict of interest The authors declare that they have no relevant financial or non-financial interests.

References

Aalborg HA, Molnár P, de Vries JE (2019) What can explain the price, volatility and trading volume of bitcoin? Financ Res Lett 29:255–265

Abay NC, Akcora CG, Gel YR, Kantarcioglu M, Islambekov UD, Tian Y, Thuraisingham BM (2019) ChainNet: learning on blockchain graphs with topological features. In: Wang J, Shim K, Wu X (eds) 2019 IEEE International Conference on Data Mining, ICDM 2019, Beijing, China, 8–11 Nov 2019. IEEE, pp 946–951. https://doi.org/10.1109/ICDM.2019.00105

Aggarwal A, Gupta I, Garg N, Goel A (2019) Deep learning approach to determine the impact of socio economic factors on bitcoin price prediction. In: 2019 twelfth international conference on contemporary computing, IC3 2019, Noida, India, 8–10 Aug 2019. IEEE, pp 1–5. https://doi.org/10.1109/IC3.2019.8844928

Akcora CG, Dey AK, Gel YR, Kantarcioglu M (2018) Forecasting bitcoin price with graph chainlets. In: Phung DQ, Tseng VS, Webb GI, Ho B, Ganji M, Rashidi L (eds) Advances in knowledge discovery and data mining—22nd Pacific–Asia conference, PAKDD 2018, Melbourne, VIC, Australia, 3–6 June 2018, Proceedings, Part III, Lecture Notes in Computer Science, vol 10939. Springer, pp 765–776. https://doi.org/10.1007/978-3-319-93040-4_60

Balcilar M, Bouri E, Gupta R, Roubaud D (2017) Can volume predict bitcoin returns and volatility? A quantiles-based approach. Econ Model 64:74–81

Balfagih AM, Keselj V (2019) Evaluating sentiment c1assifiers for bitcoin tweets in price prediction task. In: 2019 IEEE international conference on big data (big data), Los Angeles, CA, USA, 9–12 Dec 2019. IEEE, pp 5499–5506. https://doi.org/10.1109/BigData47090.2019.9006140

Burnie A, Yilmaz E (2019) An analysis of the change in discussions on social media with bitcoin price. In: Piwowarski B, Chevalier M, Gaussier É, Maarek Y, Nie J, Scholer F (eds) Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval, SIGIR 2019, Paris, France, 21–25 July 2019. ACM, pp 889–892. https://doi.org/10.1145/3331184.3331304

Cavalli S, Amoretti M (2021) CNN-based multivariate data analysis for bitcoin trend prediction. Appl Soft Comput 101:107065. https://doi.org/10.1016/j.asoc.2020.107065

Cerda GC, Reutter JL (2019) Bitcoin price prediction through opinion mining. In: Amer-Yahia S, Mahdian M, Goel A, Houben G, Lerman K, McAuley JJ, Baeza-Yates R, Zia L (eds) Companion of the 2019 world wide web conference, WWW 2019, San Francisco, CA, USA, 13–17 May 2019,.ACM, pp 755–762. https://doi.org/10.1145/3308560.3316454

Chen C, Chang J, Lin F, Hung J, Lin C, Wang Y (2019) Comparison of forcasting ability between backpropagation network and ARIMA in the prediction of bitcoin price. In: 2019 international symposium on intelligent signal processing and communication systems, ISPACS 2019, Taipei, Taiwan, 3–6 Dec 2019. IEEE, pp 1–2. https://doi.org/10.1109/ISPACS48206.2019.8986297

Chen W, Zheng Z, Ma M, Wu J, Zhou Y, Yao J (2020a) Dependence structure between bitcoin price and its influence factors. IJCSE. pp 334–345. https://doi.org/10.1504/IJCSE.2020.106058
Chen Z, Li C, Sun W (2020b) Bitcoin price prediction using machine learning: an approach to sample dimension engineering. J Comput Appl Math. https://doi.org/10.1016/j.cam.2019.112395
Ciaian P, Rajcaniova M, Kancs A (2016) The economics of bitcoin price formation. Appl Econ 48(19):1799–1815. https://doi.org/10.1080/00036846.2015.1109038
Ding X, Guo J, Li D, Wu W (2021) An incentive mechanism for building a secure blockchain-based internet of things. IEEE Trans Netw Sci Eng 8(1):477–487. https://doi.org/10.1109/TNSE.2020.3040446
Ding X, Guo J, Li D, Wu W (2022) Pricing and budget allocation for IoT blockchain with edge computing. IEEE Trans Cloud Comput. https://doi.org/10.1109/TCC.2022.3150766
Felizardo L, Oliveira R, Del-Moral-Hernandez E, Cozman F (2019) Comparative study of bitcoin price prediction using wavenets, recurrent neural networks and other machine learning methods. In: 6th international conference on behavioral, economic and socio-cultural computing, BESC 2019, Beijing, China, 28–30 Oct 2019. IEEE, pp 1–6. https://doi.org/10.1109/BESC48373.2019.8963009
Georgoula I, Pournarakis D, Bilanakos C, Sotiropoulos DN, Giaglis GM (2015) Using time-series and sentiment analysis to detect the determinants of bitcoin prices. In: 9th Mediterranean conference on information systems, MCIS 2015, Samos, Greece, 2–5 Oct 2015. Proceedings, AISeL, p 20. http://aisel.aisnet.org/mcis2015/20
Guo H, Zhang D, Liu S, Wang L, Ding Y (2021a) Bitcoin price forecasting: a perspective of underlying blockchain transactions. Decis Support Syst 151:113650. https://doi.org/10.1016/j.dss.2021.113650
Guo J, Ding X, Wu W (2021b) A blockchain-enabled ecosystem for distributed electricity trading in smart city. IEEE Internet Things J 8(3):2040–2050. https://doi.org/10.1109/JIOT.2020.3015980
Guo J, Ding X, Wu W (2021c) Reliable traffic monitoring mechanisms based on blockchain in vehicular networks. IEEE Trans Reliab. https://doi.org/10.1109/TR.2020.3046556
Guo Q, Lei S, Ye Q, Fang Z (2021d) MRC-LSTM: a hybrid approach of multi-scale residual CNN and LSTM to predict bitcoin price. In: International joint conference on neural networks, IJCNN 2021, Shenzhen, China, 18–22 July 2021,.IEEE, pp 1–8. https://doi.org/10.1109/IJCNN52387.2021.9534453
Guo J, Ding X, Wu W (2022) An architecture for distributed energies trading in byzantine-based blockchains. IEEE Trans Green Commun Netw 6(2):1216–1230. https://doi.org/10.1109/TGCN.2022.3142438
Gyamerah SA (2021) Two-stage hybrid machine learning model for high-frequency intraday bitcoin price prediction based on technical indicators, variational mode decomposition, and support vector regression. Complexity 2021:1767708:1-1767708:15. https://doi.org/10.1155/2021/1767708
Hashish IA, Forni F, Andreotti G, Facchinetti T, Darjani S (2019) A hybrid model for bitcoin prices prediction using hidden Markov models and optimized LSTM networks. In: 24th IEEE international conference on emerging technologies and factory automation, ETFA 2019, Zaragoza, Spain, 10–13 Sept 2019. IEEE, pp 721–728. https://doi.org/10.1109/ETFA.2019.8869094
Koo E, Kim G (2021) Prediction of bitcoin price based on manipulating distribution strategy. Appl Soft Comput 110:107738. https://doi.org/10.1016/j.asoc.2021.107738
Koutmos D (2018) Bitcoin returns and transaction activity. Econ Lett 167:81–85
Kristoufek L (2013) Bitcoin meets google trends and Wikipedia: quantifying the relationship between phenomena of the internet era. Sci Rep 3(1):1–7
Kristoufek L (2015) What are the main drivers of the bitcoin price? Evidence from wavelet coherence analysis. PLoS ONE 10(4):e0123923
Kyle AS (1985) Continuous auctions and insider trading. Econom: J Econom Soc 53:1315–1335
Li X, Du L (2021) A multi-window bitcoin price prediction framework on blockchain transaction graph. In: Wu W, Du H (eds) Algorithmic aspects in information and management—15th international conference, AAIM 2021, Virtual Event, 20–22 Dec 2021, Proceedings, Lecture Notes in Computer Science, vol 13153. Springer, pp 317–328. https://doi.org/10.1007/978-3-030-93176-6_27
Llorente G, Michaely R, Saar G, Wang J (2002) Dynamic volume-return relation of individual stocks. Rev Financ Stud 15(4):1005–1047
Luo C, Xu L, Li D, Wu W (2020) Edge computing integrated with blockchain technologies. In: Du D, Wang J (eds) Complexity and approximation—in memory of Ker-I Ko, vol 12000. Lecture Notes in Computer Science. Springer, Cham, pp 268–288. https://doi.org/10.1007/978-3-030-41672-0_17
Maesa DDF, Marino A, Ricci L (2016) Uncovering the bitcoin blockchain: an analysis of the full users graph. In: 2016 IEEE international conference on data science and advanced analytics, DSAA 2016, Montreal, QC, Canada, 17–19 Oct 2016. IEEE, pp 537–546. https://doi.org/10.1109/DSAA.2016.52

Mallqui DCA, Fernandes RAS (2019) Predicting the direction, maximum, minimum and closing prices of daily bitcoin exchange rate using machine learning techniques. Appl Soft Comput 75:596–606. https://doi.org/10.1016/j.asoc.2018.11.038

Mallqui DCA, Fernandes RAS (2021) Analysis of technical, economic and social information features to predict the bitcoin price direction for day-trade operations. In: International joint conference on neural networks, IJCNN 2021, Shenzhen, China, 18–22 July 2021. IEEE, pp 1–7. https://doi.org/10.1109/IJCNN52387.2021.9534056

Mittal A, Dhiman V, Singh A, Prakash C (2019) Short-term bitcoin price fluctuation prediction using social media and web search data. In: 2019 twelfth international conference on contemporary computing, IC3 2019, Noida, India, 8–10 Aug 2019. IEEE, pp 1–6. https://doi.org/10.1109/IC3.2019.8844899

Naeem M, Bouri E, Boako G, Roubaud D (2020) Tail dependence in the return-volume of leading cryptocurrencies. Financ Res Lett 36:101326

Nakamoto S (2009) Bitcoin: a peer-to-peer electronic cash system

Nguyen D, Le H (2019) Predicting the price of bitcoin using hybrid ARIMA and machine learning. In: Dang TK, Küng J, Takizawa M, Bui SH (eds) Future data and security engineering—6th international conference, FDSE 2019, Nha Trang City, Vietnam, 27–29 Nov 2019, Proceedings, Lecture Notes in Computer Science, vol 11814. Springer, pp 696–704. https://doi.org/10.1007/978-3-030-35653-8_49

Pieters G, Vivanco S (2017) Financial regulations and price inconsistencies across bitcoin markets. Inf Econ Policy 39:1–14. https://doi.org/10.1016/j.infoecopol.2017.02.002

Rajakumar BR, Binu D, Shaek MR (2022) Optimal prediction of bitcoin prices based on deep belief network and lion algorithm with adaptive price size: optimal prediction of bitcoin prices. Int J Distrib Syst Technol 13(1):1–28. https://doi.org/10.4018/IJDST.296251

Schneider J (2009) A rational expectations equilibrium with informative trading volume. J Financ 64(6):2783–2805

Shahzad MK, Bukhari L, Khan TM, Islam SMR, Hossain MM, Kwak K (2021) BPTE: bitcoin price prediction and trend examination using twitter sentiment analysis. In: International conference on information and communication technology convergence, ICTC 2021, Jeju Island, Republic of Korea, 20–22 Oct 2021. IEEE, pp 119–122. https://doi.org/10.1109/ICTC52510.2021.9620216

Shin M, Mohaisen D, Kim J (2021) Bitcoin price forecasting via ensemble-based LSTM deep learning networks. In: International conference on information networking, ICOIN 2021, Jeju Island, South Korea, 13–16 Jan 2021. IEEE, pp 603–608. https://doi.org/10.1109/ICOIN50884.2021.9333853

Sin E, Wang L (2017) Bitcoin price prediction using ensembles of neural networks. In: Liu Y, Zhao L, Cai G, Xiao G, Li K, Wang L (eds) 13th international conference on natural computation, fuzzy systems and knowledge discovery, ICNC-FSKD 2017, Guilin, China, 29–31 July 2017. IEEE, pp 666–671. https://doi.org/10.1109/FSKD.2017.8393351

Vassiliadis S, Papadopoulos P, Rangoussi M, Konieczny T, Gralewski J (2017) Bitcoin value analysis based on cross-correlations. J Internet Bank Commer 22(S7):1

Yao W, Xu K, Li Q (2019) Exploring the influence of news articles on bitcoin price with machine learning. In: 2019 IEEE symposium on computers and communications, ISCC 2019, Barcelona, Spain, 29 June–3 July 2019. IEEE, pp 1–6. https://doi.org/10.1109/ISCC47284.2019.8969596

Yermack DL (2013) Is bitcoin a real currency? An economic appraisal. Econ Innov eJournal

Yogeshwaran S, Kaur MJ, Maheshwari P (2019) Project based learning: predicting bitcoin prices using deep learning. In: Ashmawy AK, Schreiter S (eds) IEEE global engineering education conference, EDUCON 2019, Dubai, United Arab Emirates, 8–11 April 2019. IEEE, pp 1449–1454. https://doi.org/10.1109/EDUCON.2019.8725091