ChainNet: Learning on Blockchain Graphs with Topological Features

Nazmiye Ceren Abay
Computer Science
University of Texas at Dallas
Richardson, TX, USA
ncabay@utdallas.edu

Cuneyt Gurcan Akcora
Computer Science
University of Texas at Dallas
Richardson, TX, USA
cuneyt.akcora@utdallas.edu

Yulia R. Gel
Statistics
University of Texas at Dallas
Richardson, TX, USA
ygl@utdallas.edu

Umar D. Islambekov
Statistics
University of Texas at Dallas
Richardson, TX, USA
umard@utdallas.edu

Murat Kantarcioglu
Computer Science
University of Texas at Dallas
Richardson, TX, USA
muratk@utdallas.edu

Yahui Tian
Statistics
University of Texas at Dallas
Richardson, TX, USA
yahui.tian@outlook.com

Bhavani Thuraisingham
Computer Science
University of Texas at Dallas
Richardson, TX, USA
bhavani.thuraisingham@utdallas.edu

***Abstract*—With emergence of blockchain technologies and the associated cryptocurrencies, such as Bitcoin, understanding network dynamics behind Blockchain graphs has become a rapidly evolving research direction. Unlike other financial networks, such as stock and currency trading, blockchain based cryptocurrencies have the entire transaction graph accessible to the public (i.e., all transactions can be downloaded and analyzed). A natural question is then to ask whether dynamics of the transaction graph impacts price of the underlying cryptocurrency. We show that standard graph features such as degree distribution of the transaction graph may not be sufficient to capture network dynamics and its potential impact on fluctuations of Bitcoin price. In contrast, topological features computed from the blockchain graph using the tools of persistent homology, are found to exhibit higher utility for predicting Bitcoin price dynamics.**

***Index Terms*—blockchain, bitcoin, persistent homology, graph**

I. Introduction

Recent jumps of Bitcoin price have led to ever growing debates with respect to the future of Bitcoin and cryptocurrencies and its potential impact on global financial markets [13]. One interesting aspect of popular cryptocurrencies, such as Bitcoin, is that each transaction is recorded on a distributed public ledger, called blockchain. The recorded transactions can be then accessed and analyzed by anyone. Furthermore, all of the transactions could be represented by a graph referred to as the "blockchain graph". Existence of the blockchain graph raises important questions such as "How does the blockchain graph structure impact the underlying cryptocurrency price?"

In this paper, we focus on addressing this question by proposing approaches to represent blockchain graph patterns; and we use these patterns to build machine learning models for Bitcoin price prediction.

First approach that comes to mind to leverage the blockchain graph structure is to extract traditional graph features such as degree distribution, motif counts and clustering coefficients, and to use these graph features in machine learning models such as Random Forest for assessment of their utility in price forecasting.

As already observed by previous studies (e.g., [8, 12, 16]), and also confirmed by our experimental results, these standard graph based features are insufficient to capture important properties such as transaction volumes, transaction amounts, and their relationships with the underlying graph structure. Since these basic approaches do not provide conclusive insights into the blockchain graph dynamics and its impact on cryptocurrency price, we propose novel techniques inspired by topological data analysis (TDA) and, particularly, persistent homology that account for these higher order interactions.

Persistent homology allows us to extract topological information from a blockchain graph and unveil some critical characteristics behind its functionality. Most notably, persistent homology captures interactions of the graph components at a multi-scale level which are otherwise largely inaccessible with conventional analytic methods. Such an approach provides the following important benefits. First, we systematically account for changes in the blockchain graph topology and geometry at different scales, both in terms of transaction patterns and associated transaction volumes. Second, by computing topological features for a range of scale values we bypass the problem of optimal scale selection. That is, instead we systematically derive topological information from the blockchain graph and use its change dynamics for cryptocurrency price prediction. Third, the multi-scale approach permits us to effectively distinguish true topological features from noisy ones in a robust way based on the extent of feature lifespan across scale values. Furthermore, a few studies on the application of TDA to other types of networks show that persistent homology-based features outperform conventional graph features such as betweenness centrality, clustering coefficient and degree centrality in network classification and segmentation [7].

Our contributions can be summarized as follows:

- To our knowledge, we are the first ones to introduce persistent homology to cryptocurrency predictive analytics. Furthermore, we couple homology-based topological features of Blockchain with machine learning techniques to predict Bitcoin prices.

- We introduce a novel concept of a *Betti derivative*. Betti derivatives capture the rate of changes that occur in the topological structure of the blockchain graph. We show predictive utility of the Betti derivatives in forecasting Bitcoin prices.
- Using extensive empirical analysis, we show that machine learning models incorporating our proposed persistent homology-based methodology can significantly outperform (i.e., up to 38% improvement in root mean squared error) models which use only past price and standard features such as total transaction count.

An extended version of this work with detailed explanations of our algorithmic model can be found online. [1]

II. Learning Graph Based and Topological Features

Problem Statement: Let $x_t \in \mathbb{R}^d$ be a set of features computed on the Bitcoin blockchain. Let $(x_1, y_1), \ldots, (x_t, y_t)$ be the observed data where $Y = \{y_1, \ldots, y_t\}$ are the corresponding Bitcoin prices in dollars. At a time point t, estimate the Bitcoin price $y_{t'}$ where $t' > t$.

We provide two solutions to our research problem: *graph filtration (FL)* and the *Betti sequences*. The first approach is based on graph filtration. That is, we filter the transaction network with increasing thresholds of Bitcoin amounts, and create multiple realizations of the network. Afterwards, we merge these realizations to train a model. The second approach uses topological summaries to capture persistent features in terms of Betti sequences and Betti derivatives.

The Betti approach is based on rigorous mathematical foundations of algebraic topology and provides a multi-lens view of the system, whereas the graph filtration is a heuristic that allows manually selecting amount thresholds and associated filtering of the network. Next, we describe these two approaches in details.

A. Learning Graph Representations

We first introduce existing blockchain network models and explain their shortcomings. Next we describe our substructure model of the blockchain graph and extract *graph filtration* features.

In a typical blockchain graph such as the one used by Bitcoin, an owner of multiple addresses can combine them in a transaction and send coins to multiple output addresses. Therefore, the Bitcoin blockchain consists of two types of nodes: transactions, and addresses that are input/output of transactions (e.g., see Figure 1). In our approach, we follow [1] and construct a heterogeneous Blockchain graph with both address and transaction nodes.

With its input and output addresses, each transaction represents an immutable decision that is encoded as a substructure on the blockchain graph. Recently, [1, 2] proposed to study such blockchain substructures in the form of **chainlets**.

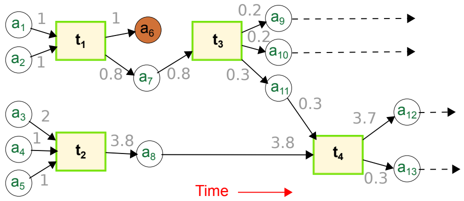


Fig. 1: A Bitcoin graph with 4 transactions and 13 addresses. Amounts on edges show currency transfers. The difference between input and outputs amounts, if exists, shows the transaction fee collected by miners.

The chainlet approach of [1] aims to transfer the ideas of network motifs [14] to blockchain graphs. That is, by counting frequency of certain shapes, a blockchain graph can be summarized with chainlet densities. However, while the chainlet approach of [1] is found to be promising in describing dynamics of the blockchain graph, it has two major shortcomings. First, [1] focuses only on the basic case of $k = 1$, or 1-chainlets. Indeed, as the k value increases, k-chainlets encode higher order structures on the graph and the number of distinct shaped chainlets also increases. Second, even in the basic case of 1-chainlets, [1] disregards such critical information as amounts of coins transferred from its inputs to outputs. In this paper, we address the second shortcoming and incorporate the key information on the transferred amounts into analysis of blockchain substructures.

a) Occurrence and Amount Matrices: On the Bitcoin network, the output and input addresses of a transaction t_n are defined as a list of addresses $|\Gamma^o_n| \geq 1$ and $\left|\Gamma^i_n\right| \geq 1$, respectively. An address $i_a \in \Gamma^i_n$ has an associated coin amount $A(i_a)$ that t_n receives. The output amount of a transaction t_n is defined as the sum of outputs from all input addresses $\mathcal{A}^o(n) = \sum_{i_a \in \Gamma^i_n} A(i_a)$. Considering all transactions T, we define the maximum number of inputs, $i_{max} = \underset{t_n \in T}{argmax}(\left|\Gamma^{\rm i}_{\rm n}\right|)$ and outputs $o_{max} = \underset{t_n \in T}{argmax}(|\Gamma^{\rm o}_{\rm n}|)$.

We then encode chainlet substructures with two dimensions: for $|i|$ *input* addresses and $|o|$ *output* addresses, the chainlet is denoted as $\mathbb{C}_{i \to o}$. The blockchain graph can be then represented in a form of two matrices, that is, the occurrence $\mathcal{O}_{[i_{max} \times o_{max}]}$ and amount $\mathcal{A}_{[i_{max} \times o_{max}]}$ matrices, where the cell of i-th row and o-th column represents information on the substructure $\mathbb{C}_{i \to o}$.

b) Graph Filtration (FL): Given the amount and occurrence information, a natural combination of them entails filtering the occurrence matrix with user defined thresholds on amounts, or filtering the amount matrix with user defined thresholds on occurrences. In both cases, the user defined threshold implies a heuristic aspect.

FL creates multiple occurrence matrices of a Bitcoin network at a given time period, and uses them as the feature set to train a prediction model. At a given time period t, chainlets

[1] https://arxiv.org/abs/1908.06971

of the time period are iterated over with a set of thresholds. A chainlet $\mathbb{C}_{i \to j}$'s occurrence is recorded in the associated occurrence matrix $\mathcal{O}^\epsilon$ if the amount transferred by the chainlet $amount(\mathbb{C}_{i \to j}) \geq \epsilon$. The process is repeated for all inputted data. Resulting occurrence matrices are row-wise concatenated and output as the FL feature set for time period t (i.e., x_t).

The FL captures persistent graph substructures by retaining edges among nodes according to a set of scale values. For a scale value $\epsilon \in \epsilon_{1,\ldots,S}$, we only record the occurrence of chainlet substructures, if the amount transferred by the substructure is $\geq \epsilon$.

B. Learning Topological Representations

TDA is an emerging field at the intersection of algebraic topology and computational geometry providing methods to systematically study the topological and geometric structure underlying data [4]. In this context, these structures are commonly analyzed via the multi-scale-based framework of persistent homology. Below we outline its main steps. The primary idea is to assess which topological features remain persistent over a larger set of scales and hence, e.g., in the case of the Blockchain network, are likely to play a significant role in its functionality.

Let $\mathbb{X} = \{X_1, \ldots, X_n\}$ be a set of data points in a metric space (e.g., the Euclidean space). Select a scale ϵ_k and form a graph G_k with the associated adjacency matrix $A = \mathbb{1}_{d_{ij} \leq \epsilon_k}$, where d_{ij} is the distance between points X_i and X_j. Changing the scale values $\epsilon_1 < \epsilon_2 < \ldots < \epsilon_N$ results in a hierarchical nested sequence of graphs $G_1 \subseteq G_2 \subseteq \ldots \subseteq G_N$ that is called a *graph filtration*.

Next, to glean the intrinsic geometry underlying the data from the graph filtration, we associate an *(abstract) simplicial complex* with each G_k, $k = 1, \ldots, N$. These constructs can be thought of as higher order analogues of graphs having both the topological and combinatorial structure [4]. The latter serves well for the computational purposes to extract various topological summaries from data. A major advantage of the multi-lens perspective is that it avoids the issue of searching for an optimal scale value and associated feature engineering.

The choice of a simplicial complex depends on the complexity of the data and which topological features one is interested in highlighting. The *Vietoris-Rips* (VR) simplicial complex is one of the most popular choices in TDA due to its easy construction and computational advantages (e.g., [4]).

Armed with the associated VR filtration, $VR_1 \subseteq VR_2 \subseteq \ldots \subseteq VR_N$, we can track qualitative topological features such as connected components, loops and voids that appear and disappear as we move along the filtration.

In our analysis, we use the Betti sequences as summaries of persistent homology calculations which encode the counts of these features at increasing scale values. Their individual elements are called the *Betti numbers* that are computed for each value of the scale:

$$\boldsymbol{\beta}_p = (\beta_p(\epsilon_1), \beta_p(\epsilon_2), \ldots, \beta_p(\epsilon_N)), \quad p = 0, 1, \ldots, K,$$

where $\beta_p(\epsilon_k)$ is the p-th Betti number of the simplicial complex at scale ϵ_k. The Betti numbers for small p have a simple interpretation. For instance, β_0 is the number of connected components; β_1 is the number of loops; β_2 is the number of voids etc.

1) Betti Sequences for a Blockchain Network: Although the Betti sequences provide a non-parametric solution to combine information on edge distance with node connectedness, the computational complexity of Betti calculations prohibits their usage in large networks. For example, for simplicial complexes of dimension 2, "currently no upper bound better than a constant times n^3 is known" [6]. For Betti numbers $\beta_{p>3}$, the complexity becomes too restrictive. The problem is compounded in the Bitcoin network since address reuse is discouraged. As such, every day brings $\geq$ 500K new nodes to the network. Betti number computations on such large networks is unfeasible.

To solve the complexity issues, we propose a novel approach that computes the Betti sequences on a network of $N \times N$ nodes where N is the size of the amount matrix $\mathcal{A}$ (See Section II-A). Each of the N^2 unique chainlets (e.g., $\mathbb{C}_{2 \to 3}$) creates a node in the new network, where edge distance between two nodes is computed with a suitable 'distance' d. We describe the main steps as follows:

Given a heterogeneous Blockchain network with transferred bitcoins on edges,

1) All transferred amounts are converted from Satoshis to bitcoins (dividing by 10^8), then added one (so that the values after taking logarithm are non-negative) and log-transformed: $a' = \log(1 + a/10^8)$, where a is an amount in Satoshis.
2) For each chainlet of a given time period, we compute the sample q-quantiles for the associated log-transformed amounts [10]: a k-th q-quantile, $k = 0, 1, \ldots, q$, is the amount $Q(k)$ such that
$$\sum_{i=1}^{\tau} \mathbb{1}_{y_i < Q(k)} \approx \frac{\tau k}{q} \text{ and } \sum_{i=1}^{\tau} \mathbb{1}_{y_i > Q(k)} \approx \frac{\tau(q-k)}{q},$$
where τ is the total number of transactions. The (dis)similarity metric d_{ij} between chainlet nodes i and j is defined as the quantile-based distance $d_{ij} = \sqrt{\sum_{k=0}^{q}[Q_i(k) - Q_j(k)]^2}$.
3) We construct a sequence of scales $\epsilon_1 < \epsilon_2 < \ldots < \epsilon_S$ covering a range of distances during the entire 365-day period. For each ϵ_k, we build the corresponding VR complex whose 0-simplices are single chainlets and 1-simplices are pairs of chainlets with distance $\leq \epsilon_k$. As a result, we obtain the filtration of VR complexes $VR_1 \subseteq VR_2 \subseteq \ldots \subseteq VR_S$.
4) Armed with the VR filtration, we then compute $x_t = \{\beta_0(\epsilon_1), \ldots, \beta_0(\epsilon_S); \beta_1(\epsilon_1), \ldots, \beta_1(\epsilon_S)\}$.

In constructing the new network, we use and hence retain the amount information from the Blockchain network. Furthermore, each node type (chainlet substructure) encodes the number of inputs and outputs in a transaction. This way, we combine distance (computed from transferred coins) with edge connectedness while restricting the network size. Our new TDA approach can work with networks of any size, and our experimental results (See Section III) show predictive power of its topological features.

2) Betti derivatives: The graph of the p-th Betti sequence is often referred to as the *p-th Betti curve*. Analysis of the Betti curves allows us to assess dynamics of essential topological features as a function of the scale. Furthermore, to assess the rate of changes in topological features of the Blockchain graph, we introduce a novel concept of *Betti derivatives* up to order $\ell > 0$ on VR filtrations:

$$\Delta^{\ell}\beta_p(\epsilon_k) = \Delta^{\ell-1}\beta_p(\epsilon_{k+1}) - \Delta^{\ell-1}\beta_p(\epsilon_k),$$

where $k = 1, 2, \ldots, S-1$, $p = \{0, 1, \ldots\}$ values are determined by how many Betti numbers we choose to use, and S is the number of filtration steps. These finite differences are analogues of derivatives for smooth functions. The inclusion of the rates of change of the Betti curves is intended to systematically capture dynamics of essential topological features and to enhance the predictive power.

III. Experiments

In this section, we show the performance of predictive models in our ChainNet framework.

A. Data

We downloaded and parsed the entire Bitcoin transaction graph from 2009 January to 2018 December. Using a time interval of 24 hours, we extracted daily transactions on the network and created the Bitcoin graph. Our Bitcoin price (USD) data is downloaded from blockchain.com which aggregates prices from worldwide online exchanges.

a) Filtration data.: We analyzed Bitcoin transactions to find an appropriate dimension N for the occurrence matrix. We chose $N = 20$, because $N = 20$ enables to distinguish a sufficiently large number (i.e., 400) of chainlets, and still offers a dense matrix. Our models achieved a satisfactory performance with $\epsilon \in \{0, 10, 20, \ldots, 50\}$ scales in the graph filtration.

b) Betti and Betti Derivative Data.: We use the Betti numbers estimation routine of the Perseus [15] software which provides an efficient algorithm to compute the Betti numbers and persistent intervals.

We used $S \in \{50, 100, 200 \text{ and } 400\}$ as the filtration length. Overall, we find no improvement in prediction accuracy for $S > 400$. Furthermore, there is no single optimal value of S to be used in all statistical and machine learning models.

To decrease computational costs, in the present study, we focus on VR complexes of dimension one. This implies that the loops are formed by three or more nodes, which in turn leads to a general negative association between the Betti-0 and Betti-1 curves – as ϵ increases, more simplices are added to the complex, thereby reducing the number of connected components and increasing the number of loops.

In addition to FL and Betti related features, we also experimented with basic features: price, mean degree of addresses (MeanDegree), number of new addresses (NumNewAddress), mean and total coin amount transferred in transactions (meanTxAmount and TotalTxAmount, respectively) and address network average clustering coefficient (ClusCoeff). Among these, we only found Price and TotalTx to be useful predictors and included them in our models.

B. Setting for Feature Time Series

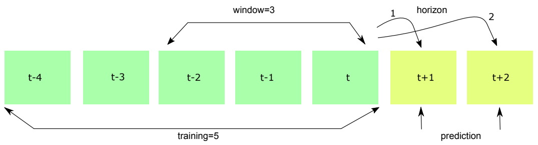


Fig. 2: The sliding window based regressor model. The example model trains with data from the last $m = 5$ days, and uses the data from t, $t-1$ and $t-2$ (window=3) to make a prediction for either day $t+1$ (horizon=1) or day $t+2$ (horizon=2).

Given the features, we employ a time based approach to predict the Bitcoin price, as shown in Figure 2. Our goal is to catch trends in the price data, based on the observation that price movements in the preceding days are a good indicator of future prices.

ChainNet employs three time related concepts: training length, window (lag) and horizon. Training length is the number of past time periods whose data we use to train our model. Window is the number of past time periods whose data we use to predict Bitcoin price. Horizon is the number of days whose price we predict ahead.

In the most basic case of prediction horizon $h = 1$ and prediction window $w = 1$, the model learns to predict the price of day $\hat{y}_{t+1}$ by using the data x_t of day t. Similarly, for any window w, the model uses data from $\{x_{t-w}, \ldots, x_t\}$ to predict the price $\hat{y}_{t+h}$.

Input is time indexed data points and output is the model parameters trained on the given input. For given window w and horizon h values, time series data is processed to utilize the history of the current day, t. Each x_t is replaced by the successive values of time series between $t-w-h$ and $t-h$. Newly generated $\hat{x_t}$ and its corresponding price, y_t, is appended to the train list. After all days are iterated on, dimension reduction is applied to the generated $\hat{x}_{train}$ to obtain compensated data. At the end, the model is optimized with the previously obtained train data and the algorithm returns the obtained model parameters for out-of-sample predictions.

We consider the following two parameters in all predictive models: window $w \in \{3, 5, 7\}$, horizon $h \in \{1, 2, 5, 7, 10, 15, 20, 25, 30\}$, training length $l \in \{25, 50, 100, 200\}$. As the interaction of horizon, window and training length parameters may exhibit nonlinear effects on the prediction, we conduct a grid search by varying all parameters, and report the predicted price values for the best model.

An important point in our sliding prediction approach is that, we train a model per each prediction. As a result, we train a model 365 times to predict Bitcoin prices in 2017. We chose this setting because gain results improved over a batch prediction model. As we model data with low dimensional features, the cost of this approach was negligible.

C. Statistical and Machine Learning Models

We evaluate ChainNet performance by using one statistical and four machine learning models: ARIMAX [3], XGBT [5], Random Forest [9], Gaussian Processes [17] and ENET [18].

a) Parameter Settings for Models: For the hyper-parameter tuning of ARIMAX, the orders for auto-regression and moving average terms are chosen from $\{0, 1, 2\}$. For the tree based approaches such as XGBT, RF, generated number of trees are chosen from {10, 50, 100, 200, 300, 400, 500, 1000}. For the learning rate of XGBT, we tried values from $\{0.01, 0.1, 1.0\}$. ENET regularization parameters for L1 and L2 and penalty constants are selected from {0.0001, 0.001, 0.01, 0.1, 1.0, 10.0} and {0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1.0}. In hyper-parameter tuning of GP, regression types, correlation types, and regularization parameters are chosen from {constant, linear, quadratic}, {absolute exponential, squared exponential, generalized exponential, cubic, linear}, {0.001, 0.01, 0.1, 1.0, 10.0} respectively.

b) High Dimensionality: Since we use a windowed (lagged) history of the data, dimensionality of the training data increases rapidly. We ameliorate the effects of high dimensionality by applying Principal Component Analysis (PCA) [11] to the lagged feature sets of FL, Betti and Betti derivative; we use PCA to map the high dimensional data into low dimensional data with the dimension of $d_2 \in \{5, 10, 15, 20\}$.

D. Baseline Performance

The simplest baseline for ChainNet can be constructed by training models on Price and TotalTx in a sliding window prediction scheme. We did not use other baseline features such as mean degree (see Section III-A0b) since adding those features reduces performance of the baseline models. We train baseline models without reducing the dimensionality, because input features are very few; for $w = 3$, the models use 6 features in training. We assess model performance with root mean squared error (RMSE) as follows: $RMSE = \sqrt{1/|T| \sum_{t \in T} (y_t - \hat{y}_t)^2}$, where $|T|$ is the number of days, $\hat{y}_t$ is the predicted price and y_t is the true observed price on the t^{th} day.

In our rolling predictive framework, we achieve the best results with a training length of 100 days, that is, each considered model is adaptively re-estimated for each y_t using data from the previous 100 days. We only report the best results from each model with the hyper-parameter optimization.

Figure 3 shows the performance of the five models in prediction. ARIMAX has the worst performance for $h > 7$, whereas Gaussian Process (GP) has the best RMSE values overall. We note that as the window value increases, performance does not improve. This implies that considering past information on price and total number of transactions does not deliver improvement in forecasting accuracy. In fact, from window 3 to 7, the RMSE values of the best model, GP, is approximately similar while $h < 10$. For $h > 10$, the RMSE values decrease 13% from window 3 to 7.

E. ChainNet Model Performance

In this section, we provide performance of the predictive models built with FL, Betti and Betti derivative features. *Our hypothesis is that adding these features will increase model performance*, i.e., RMSE in predictions will decrease over their associated baseline values. Due to space limitations, results of RF, GP and ENET comparisons are excluded. We refer the reader to the extended version of the paper [2]for these results.

a) Performance Gain: In our analysis, we report the percentage predictive gain, or decrease in $RMSE$ for a specific machine learning model m w.r.t. its baseline model m_0 as $\Delta_m(w, h) = 100 \times \big(1 - RMSE_m(w, h)/RMSE_{m_0}(w, h)\big)$, where $RMSE_{m_0}(w, h)$ and $RMSE_m(w, h)$ are delivered by a baseline model m_0 and a competing model m, respectively.

Figure 4 shows that XGBT predictions improve for increasing horizons, but decrease for $h > 15$. Specifically $h = 1$ predictions reach a positive gain only in XGBT $w = 7$. XGBT also offers the best gains for $h = 2$, but its performance deteriorates for $h > 15$.

In constructing the XGBT model, the boosting approach focuses on examples that increase the error rate of objective function at each step. We hypothesize that this specific focus is the reason for XGBT's better performance.

The highest gain values for $h \leq 7$ are achieved in XGBT Betti models for $w = 7$ (38% in Figure 4c). Our heuristic approach, FL, has an interesting trend; its usage in models leads to better gains for higher horizons. In turn, Betti models yield higher gain values for short horizons. Considering these results, ChainNet can use Betti and Betti derivatives for short $(h < 10)$ term prediction, and use FL for $h > 15$.

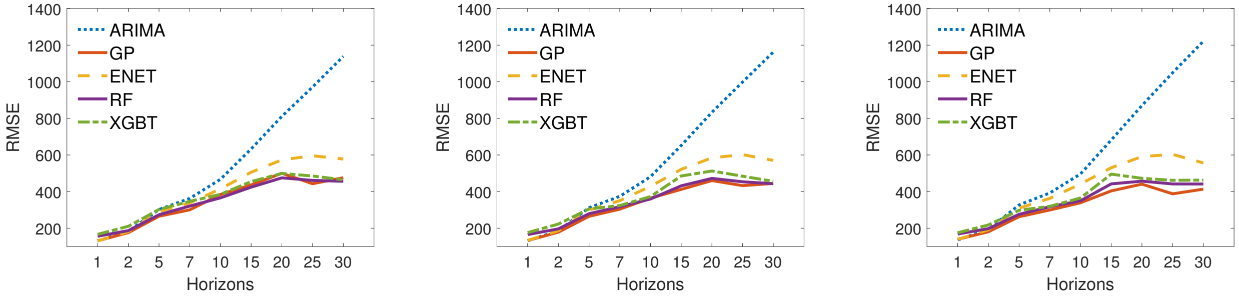
An important result is that next day predictions ($h = 1$) do not improve significantly (i.e., at most 2% in Figure 4c) with ChainNet features. Hence, topological and graph based signals in the blockchain tend to deliver a lower causal affect in the very short term forecasting horizon.

In summary, our findings offer evidence that higher order topological features of the Bitcoin transaction graph, described via Betti characteristics and FL, exhibit a high predictive utility for Bitcoin price dynamics, particularly for medium and long term forecasting horizons.

IV. Conclusion

ChainNet is a price prediction platform that utilizes topological characteristics of a blockchain graph. ChainNet builds topological constructs over a graph and computes quantitative summaries in the form of the Betti sequences and Betti derivatives which are then used in model building for the Bitcoin price prediction. Our results on the full Bitcoin network show that in less than 7 day ahead predictions, Betti models bring a prediction gain of almost 40% over baseline approaches.

[2] https://arxiv.org/abs/1908.06971

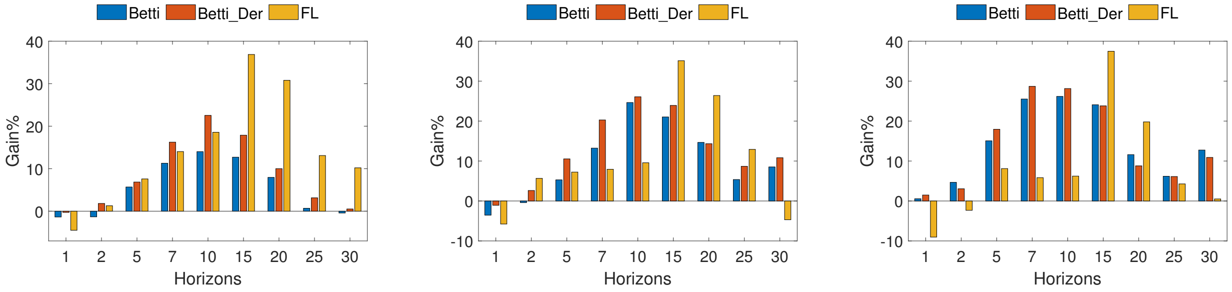


(a) Window=3

(b) Window=5

(c) Window=7

Fig. 3: RMSE of sliding window based predictions of 2017 Bitcoin prices in different window and horizon values.



(a) Window=3

(b) Window=5

(c) Best gain results: Window=7

Fig. 4: Extreme Gradient Boosting (XGBT) performance.

Acknowledgments

This research was supported in part by NIH 1R01HG006844, NSF CNS-1111529, CICI-1547324, IIS-1633331, DMS-1736368 & ARO W911NF-17-1-0356.

References

[1] Akcora CG, Dey AK, Gel YR, Kantarcioglu M (2018) Forecasting bitcoin price with graph chainlets. In: PAKDD, pp 1–12

[2] Akcora CG, Dixon MF, Gel YR, Kantarcioglu M (2018) Bitcoin risk modeling with blockchain graphs. Economics Letters 173:138 – 142

[3] Box GE, Jenkins GM, Reinsel GC, Ljung GM (2015) Time series analysis: forecasting and control. John Wiley & Sons

[4] Chazal F, Michel B (2017) An introduction to Topological Data Analysis: fundamental and practical aspects for data scientists. ArXiv e-prints pp 1–38

[5] Chen T, Guestrin C (2016) Xgboost: A scalable tree boosting system. In: The 22nd SIGKDD, pp 785–794

[6] Edelsbrunner H, Parsa S (2014) On the computational complexity of betti numbers: reductions from matrix rank. In: The 25th ACM-SIAM SODA, pp 152–160

[7] Garg A, Lu D, Popuri K, Beg MF (2016) Cortical geometry network and topology markers for parkinsons disease. arXiv preprint:161104393 pp 1–10

[8] Greaves A, Au B (2015) Using the bitcoin transaction graph to predict the price of bitcoin. No Data

[9] Ho TK (1995) Random decision forests. In: The 3rd ICDAR, vol 1, pp 278–282

[10] Hyndman RJ, Fan Y (1996) Sample quantiles in statistical packages. The American Statistician 50(4):361–365

[11] Jolliffe I (2011) Principal component analysis. In: Int. Encyclopedia of Stat. science, Springer, pp 1094–1096

[12] Kurbucz MT (2019) Predicting the price of bitcoin by the most frequent edges of its transaction network. Economics Letters p 108655

[13] Mattila J, et al. (2016) The blockchain phenomenon–the disruptive potential of distributed consensus architectures. Tech. rep., The Research Inst. of the Finnish Economy

[14] Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chklovskii D, Alon U (2002) Network motifs: Simple building blocks of complex networks. Science 298(5594):824–827

[15] Nanda V (2017) Perseus: the persistent homology software. http://peoplemathsoxacuk/nanda/perseus/indexhtml

[16] Swanson T (2014) Learning from bitcoin's past to improve its future

[17] Williams CK, Rasmussen CE (1996) Gaussian processes for regression. In: NIPS, pp 514–520

[18] Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. JRSS Ser B 67(2):301–320