Technical Report: Ethereum Token Price Anomaly Prediction with Topological Depth Curves

Yitao Li* Umar D. Islambekov* Cuneyt G Akcora* Ekaterina Smirnova†
Yulia R. Gel* Murat Kantarcioglu*

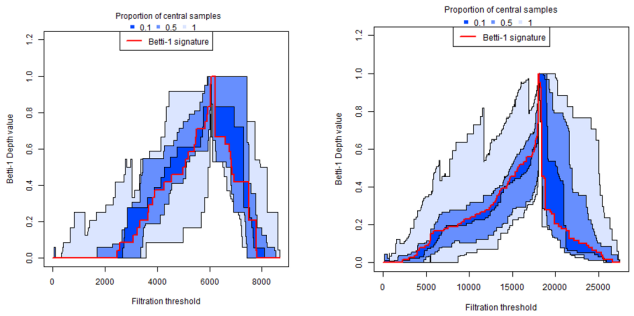
Abstract

Recently, the blockchain based cryptocurrencies and crypto tokens have started to attract significant interest. Crypto tokens that are sold on existing blockchains such as Ethereum have been used to raise significant funding for many start-ups. At the same time, many crypto tokens have failed and resulted in significant financial loss for their investors. This raises an important question: Can we predict the anomalous crypto tokens using the transaction graph data stored on the blockchain?

Unfortunately, due to dynamic and sparse nature of the crypto token transaction graphs, existing graph analysis techniques are not directly applicable. Instead, we propose novel techniques based on topological data analysis and functional data depth that allow us to extract features that are useful for anomaly prediction. Our extensive empirical analysis show that the proposed techniques significantly outperform baseline models.

1 Introduction

Blockchain has started to revolutionize many fields ranging from e-payments to digital ownership management. In addition to blockchain based cryptocurrencies such as Bitcoin and Ethereum, there have been significant interest in initial coin offerings (ICOs). ICOs (many of them offered using Ethereum blockchain) have enabled start-ups and organizations to raise capital by selling digital coins that allow recipients to use the promised service if and when available. For example, one of the successful Ethereum ICOs was Binance. Using a Ethereum smart contract,[1] Binance sold coins that allow recipients to pay exchange fees, withdrawal fees, and all other possible transaction expenses on the Binance cryptocurrency trading platform. One interesting aspect of Ethereum ICOs is that, using the transaction data that is kept on the Ethereum blockchain, it is possible to observe all the coin buying and selling activity and represent all token transactions as edges that connect two investor nodes on a graph.



(a) Power ledger (b) Tronix

Figure 1: Betti signatures of two Blockchain tokens. The signatures can be used to visualize anomalous behavior in the token's network.

Although the Binance coin was very successful, 46% of 2017's ICOs have already crashed [**?**]. This raises some important research questions such as: 1) Can we predict which ICOs are likely to fail by analyzing the underlying transaction graph kept on the blockchain? 2) Can we predict potential anomalies in the coin pricing using transaction graph analysis?

Answers to these questions are crucial for enhancing our understanding of the important ICO trends. Fundraising activity for supporting many start-ups has already moved to blockchain platforms. For example, ICOs accounts for 45% of the funding raised in initial public offerings in the 2nd quarter of 2018 [**?**]. Due to high failure rates of the ICOs, investments that are made without sufficient prior information, other than a trust in the core developer team, have resulted in substantial losses. Hence, it is of vital importance to develop early warning systems to predict which tokens are likely to fail and to spot anomalies as early as possible.

However, identification and prediction of price anomalies using the Ethereum network features pose several graph mining challenges. First, the underlying transaction graph is very sparse and dynamic. Nodes

*University of Texas at Dallas
†Virginia Commonwealth University

[1] Ethereum Smart contracts are Turing complete programs that are executed on the Ethereum blockchain.

Copyright © 20XX by SIAM
Unauthorized reproduction of this article is prohibited

(i.e., account addresses) appear and disappear (i.e., no future transaction) daily, while the number of transactions widely fluctuates across days. Hence, due to sparseness and the dynamic nature of the transaction network, standard global graph features (e.g., clustering coefficient) may not be a feasible indicator of an anomalous token activity for a given token transaction graph. Second, the overall market sentiment and transactions involving other tokens can heavily impact the token price. For example, theDAO hack [**?**] targeting an existing smart contract shook investor morale and resulted in price plummeting for almost all the coins. As a result, conventional graph analysis using techniques such as k-core does not work effectively due to factors such as low node degrees and clustering coefficients – thereby requiring development of *novel graph theoretic tools that are suitable for analysis of time-varying, highly irregular, and very sparse networks.*

We propose to address the above-mentioned challenges by introducing the arsenal of topological data analysis (TDA) tools into blockchain analytics. In particular, our approach is based on the premise that an anomalous situation in a transaction network must be reflected in its underlying topology. To study the network topology and the anomalies associated with it, we blend concepts from algebraic topology and functional data depth. The former is used to encode the mesoscopic, or multi-lense topological structures underlying the Ethereum transaction network, while the latter provides a framework to measure the relative anomalousness of the local topology. More specifically, we introduce a novel concept of *Betti functions* that allow us to efficiently and effectively track a token's transaction network that is weighted and multi edged over time. Furthermore, coupling analysis of persistent homology (PH) with functional data depth, we develop a new notion of *Betti signatures* that offers an insight on the most illustrative, or baseline behavior of a token — as a result, we can more efficiently detect anomalous token activity. Finally, we combine edge counts (i.e., number of transactions between nodes) and edge weights (i.e., transaction amounts) to create filtering techniques and efficiently analyze persistent homology via the proposed Betti functions on Ethereum graph. The resulting Betti functions are then incorporated into a time series model to predict potential future price anomalies.

The importance of our methodology and findings can be summarized as follows:

• We develop a novel functional summary for persistent homology: a **Betti function**. The new Betti functions open the door to systematic integration of TDA and functional data analysis for complex network analytics.

• We propose a new measure of the most illustrative, or "normal" behavior on the Ethereum transaction network: a **Betti signature**. Betti signatures allow us in a data-driven way to quantify and visually assess differences between normal and anomalous transaction activity, as we show in Fig. 1 for two Ethereum tokens.

• We develop a filtering approach that significantly reduces the (prohibitively high) computational costs of TDA. We report the first results where TDA tools can be adopted in large networks while preserving the performance.

• We report the first results for crypto-token price anomaly prediction, and show that token networks contain adequate information to model external price arbitration in the real world. As the crypto-token ICOs have reached $12B in the first half of 2018 [**?**], our prediction results have important real-life implications in start-up funding. As such, we are developing a web service at `EthereumCurves.Github.io` to facilitate investments from the Blockchain community.

2 Related work

We outline four relevant research areas: Ethereum graph analysis, Blockchain price prediction and anomaly detection, as well as TDA summaries.

Ethereum graph analysis. Differing from crypto-currencies (e.g., Bitcoin) where each transaction can have multiple inputs and outputs [**?**], Ethereum transactions transfer ether or tokens from one address to another. As such, Ethereum lends itself to traditional network analysis. For instance, [**?**] studied empirical properties of Ethereum and [**?**] explored token networks, in terms of degree distribution, power laws and clustering. However, there are yet no results that employ network tools for Ethereum price analytics.

Cryptocurrency price prediction. Analyzing transactions and addresses to track the Bitcoin economy has become an important research direction. A time series prediction approach by [**?**] uses a Bayesian optimized RNN and LSTM network with varying degrees of success. Blockchain features, such as average transaction amount, are also shown to exhibit mixed performance for cryptocurrency price forecasting [**?**]. Various blockchain graph characteristics, such as average degree, can be used as prediction features [**?**]. Recently, [**?**] employed blockchain motifs, termed chainlets, as features to predict Bitcoin price. However, all the mentioned approaches are carried out to track a single cryptocurrency. In contrast, our goal is to track multiple cryptoassets at the same time.

Blockchain anomaly detection. Blockchain addresses can be linked to identify people behind suspicious transaction patterns in cryptocurrencies [**?**]. The pattern is usually defined as a repeating shape that in-

Copyright © 20XX by SIAM
Unauthorized reproduction of this article is prohibited

volves moving coins from a (black) address to an online exchange, where the coins can be cashed out without being confiscated by authorities. The black address that starts the transaction chain may be related to money laundering [**?**] and ransomware payment [**?**]. There exists ample evidence of these anomalies in the transaction network [**?**]. A more recent approach found anomalies in Bitcoin price by linking addresses to transactions in time [**?**]. In contrast, we do not assume any prior knowledge about pattern shapes or addresses; our unsupervised data depth approach tracks token networks for price anomalies.

Topological Data Analysis. TDA is an emerging field at the interface of algebraic topology, statistics, and computer science. The rationale is that the observed data are sampled from some metric space and the underlying unknown geometric structure of this space is lost due to sampling. The key idea is to recover the underlying lost topology [**?**]. PH is one of the tools enabling to characterize a topological data structure under varying scales of dissimilarity. The most widely used topological summaries of persistent features are the Betti numbers, barcode plots, persistent diagrams, and persistent landscapes [**?**]. However, barcode plots and persistent diagrams cannot be easily used in machine learning models [**?**]. Differing from these approaches, we propose Betti functions, which can be directly integrated with functional data analysis tools.

3 Background on Ethereum and CryptoTokens

The Ethereum project [**?**] was created in July 2015 to provide Smart Contract functionality on a blockchain. Smart Contracts are self-executing Turing complete software codes which are replicated across a blockchain network. Smart contracts are created and put to a blockchain address by its developers. Smart Contracts ensure unstoppable, deterministic code execution that can be verified publicly. Some smart contracts implement mechanisms that allow trading digital assets, known as crypto-tokens, on the blockchain. We will refer to such a smart contract as a **crypto-token** contract, and use the term token interchangeably. A token is traded publicly among blockchain nodes, and may have an associated dollar value which is arbitrated by token demand and supply in the real world. Online exchange websites, such as `CoinMarketCap.com`, can be used to track current valuations of tokens. Most tokens have a fixed supply that is set at the time when a token contract is created. As the supply is fixed, the value of a token is mostly determined by its demand.

Utility of a token is a hotly debated topic; although some tokens are used to buy services in real life, most tokens have no intrinsic value. For example, the Cryptokitties is used to purchase, collect, breed and sell various types of "virtual cats".

Table 1: Symbols table.

Symbol	Explanation
$\tilde{\omega}$	extension of ω
F	feature matrix
h	prediction horizon (in days)
δ	min price change for anomaly
ϵ	scale parameter
$\mathcal{C}_\epsilon$	simplicial complex at scale ϵ
R_t	Price return for day t
PN	Normalized price
VR_ϵ	Vietoris-Rips complex at scale ϵ
β_p,$\mathcal{B}_p$,$\mathcal{B}_p^s$	Betti-p number, function and signature
RD and MBD	rolling and modified band depth

As most blockchains, the Ethereum blockchain attaches an *input data* field to each transaction to store log data. Smart contract transactions use this input data field to transmit messages; creating a transaction with input data to a Smart Contract is analogous to passing variables to a function. The earliest versions of Smart Contracts were developed without a common standard for transactions, input messages and functions contained in the contract. As such, each time a user wanted to transmit a message to the smart contract, she needed to know which message structure to use. Recently, the community has proposed standards, such as ERC20 (`theethereum.wiki/w/index.php/ERC20_Token_Standard`) and ERC223, which define a common list of rules for tokens to follow within the larger Ethereum ecosystem.

For example, according to the ERC20 standard, each token must implement a number of standard functions, such as *totalSupply()*. This naming standard allows exchanges, users and developers to create transactions for tokens in an automated manner. For example, when address A_1 wants to transfer its tokens (e.g., 20 OmiseGo tokens) to address A_2, it creates a transaction where the "to" address is the OmiseGo address, and the input data contains the message "transfer(A_2, x=20)".

Dataset. We created our Ethereum dataset by installing the official Ethereum Wallet and downloading all blocks. We used the EthR (`github.com/BSDStudios/ethr`) library to query Ethereum blocks through the Go Ethereum Client (i.e., Geth). Our dataset contains all the Ethereum data from 2015 July to 2018 May 6th, with a total of 5.5 million blocks.

By parsing the data, we discovered 1.7K ERC20 tokens which had more than 10K transactions. We included an ERC20 token in our analysis if it had

Copyright © 20XX by SIAM
Unauthorized reproduction of this article is prohibited

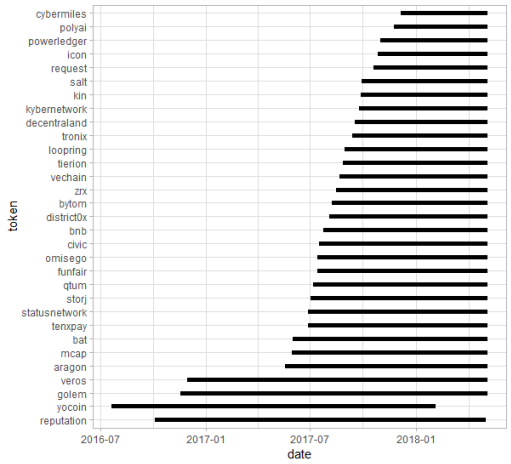


Figure 2: Ethereum token start dates.

more than $100M in market value, as reported by the `EtherScan.io` online explorer. This selection resulted in 31 tokens. Our selection was motivated by a desire to have verifiable prediction results on valuable tokens which will not fail and disappear in a short time.

On average, each token has a history of 297 days, with minimum and maximum of 151 and 576 days, respectively. The first dates of tokens on the Ethereum blockchain are reported in Figure 2. Among the ERC20 tokens, we focus on transfer and approve+transferFrom functions which transfer tokens between two addresses. In the next section, we will detail our graph constructs.

4 Methodology

In this section, we explain the proposed methodology for price anomaly prediction for Ethereum crypto-tokens.

Problem Statement: Given the transaction network of an Ethereum token and time series of the token price in fiat currency, predict whether the token price will change more than δ, $|\delta| > 0$, in the next h days. Identify the maximum horizon value h such that the prediction accuracy is at least ρ.

Our goal is to infer topological information from a graph G at multiple resolutions (i.e., at a mesoscopic level), using the notion of persistent homology, and relate it to organization and functionality of the Ethereum network. Tab. 1 gives the symbols we use in this section. We start by detailing persistent homology and associated summaries.

4.1 Persistent Homology and Its Summaries

Suppose $G = (V, E, \omega)$ is a weighted graph, where V and E are the set of nodes and edges, respectively. $\omega : E \to \mathbb{R}^+$ is a weight function encoding similarity between two nodes connected by an edge. To account for dissimilarity between two disconnected nodes, we introduce the weight $\tilde{\omega} : V \times V \to \mathbb{R}^+$

$$\tilde{\omega}_{uv} = \begin{cases} \omega_{uv} & (u,v) \in E \\ \infty & (u,v) \notin E. \end{cases}$$

In the context of Ethereum network, we define ω_{uv} as $\omega_{uv} = \Big[1 + (A_{uv} - A_{min}) \cdot (a-b)/(A_{max} - A_{min})\Big]^{-1}$, where A_{uv} is the amount of transferred tokens by transactions between nodes u and v; A_{min} and A_{max} are the smallest and the largest transaction amounts respectively. We use $a = 10$ and $b = 1$ to map weights to an interval of $[0.1, 1]$.

The key rationale behind our approach is to vary a weight threshold parameter and to study which topological features appear and disappear at increasing level of dissimilarity between nodes of Ethereum network. Features detected over a wider range of dissimilarities are deemed to be "true" Ethereum patterns and to exhibit a higher role in anomaly detection. This task turns into a computationally feasible combinatorial problem which is solved using the mathematical formalism of persistent homology (PH). The most important aspect of PH is that it allows to analyze data at multiple spatial resolutions in a unified way, and to bypass a subjective selection of the dissimilarity parameter or searching for its optimal value. To find PH, we first need to convert our data into a family of abstract simplicial complexes, indexed by dissimilarity measure.

Definition 1. (Abstract simplicial complex) *Let X be a discrete set. An abstract simplicial complex is a collection $\mathcal{C}$ of finite subsets of X such that if $\sigma \in \mathcal{C}$ then $\tau \in \mathcal{C}$ for all $\tau \subseteq \sigma$. If $|\sigma| = p+1$, then σ is called a* p-simplex.

Intuitively, a simplicial complex is a representation of X as a collection of points, intervals, triangles and their higher order counterparts. *Vietoris-Rips* is a widely used simplicial complex due to its easy construction and fast computational implementation [**?**].

Definition 2. (Vietoris-Rips complex) *Let X be a discrete set in some metric space. A Vietoris-Rips complex on X at dissimilarity scale $\epsilon \geq 0$ is denoted by VR_ϵ, is an abstract simplicial complex whose p-simplices, $p = 0, \ldots, d$, consist of points which are pairwise within distance of ϵ. Here, d is called the dimension of the complex.*

Now, we fix an increasing sequence of scales $\epsilon_1 < \epsilon_2 < \ldots < \epsilon_n$ and construct a chain of nested VR complexes called a *finite VR filtration* $VR_{\epsilon_1} \subseteq VR_{\epsilon_2} \subseteq \ldots \subseteq$

Copyright © 20XX by SIAM
Unauthorized reproduction of this article is prohibited

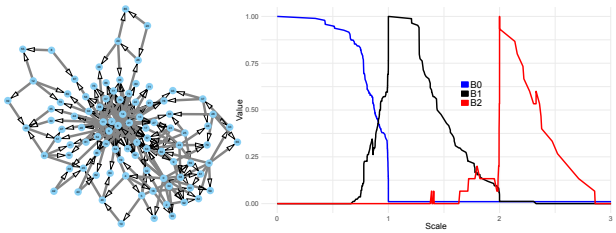


Figure 3: Betti functions for Tronix token network.

VR_{ϵ_n}, where VR_{ϵ_k}, $k = 1, \ldots, n$, is a VR complex on V such that $VR_{\epsilon_k} = \{\sigma \subset V | \tilde{\omega}_{uv} \leq \epsilon_k, \forall u, v \in \sigma\}$.

Armed with the VR filtration, we now offer a formal multi-lense glimpse into the Ethereum network geometry and track topological features that appear and disappear with an increasing scale ϵ_k. Analysis of evolution of such topological features shed light on organization of the Ethereum transaction network. That is, we can expect that features with a longer lifespan, i.e. *persistent features*, have a higher role in explaining functionality of the Ethereum network than features with a shorter lifespan. These short term features are regarded as *topological noise*. We use persistent features to distinguish anomalous dynamics in token transaction activities.

To extract summaries of such topological features at a mesoscopic level, we use *Betti numbers*.

Definition 3. (Betti number) *Betti-p number of a simplicial complex $\mathcal{C}$ of dimension d, denoted by $\beta_p(\mathcal{C})$, is defined as*

$$\beta_p(\mathcal{C}) = \begin{cases} \#\ \textit{of connected components of } \mathcal{C} & p = 0 \\ \#\ \textit{of 1-D holes or tunnels of } \mathcal{C} & p = 1 \\ \#\ \textit{of 2-D holes or cavities of } \mathcal{C} & p = 2 \\ \cdots & \\ \#\ \textit{of d-D holes of } \mathcal{C} & p = d \end{cases}$$

That is, sequence of *Betti numbers* represents the counts of different topological features of the simplicial complex (for an illustrative example see Fig. 4). In this context, we introduce a novel notion of **Betti functions** which relate these counts to the scale parameter viewed as continuum. Sequence of Betti numbers are hence finite dimensional realizations of Betti functions.

Let $\{\mathcal{C}_\epsilon\}_{\epsilon \in \mathbb{R}^+}$ be a *continuous filtration* of simplicial complexes i.e., $\mathcal{C}_\epsilon \subseteq \mathcal{C}_{\epsilon'}$ for any $0 \leq \epsilon < \epsilon'$.

Definition 4. (Betti function) *The Betti-p function $\mathcal{B}_p : \mathbb{R}^+ \to \{0, 1, 2, 3, \ldots\}$, $p = 0, \ldots, d$, associated with $\{\mathcal{C}_\epsilon\}_{\epsilon \in \mathbb{R}^+}$ is defined as*

$$\mathcal{B}_p : \epsilon \mapsto \beta_p(\mathcal{C}_\epsilon).$$

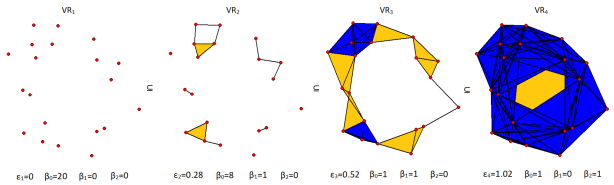


Figure 4: Betti numbers of nested VR complexes of dimension 3 at increasing scales ϵ. The point cloud is sampled from an annulus with outer radius 1 and inner radius 0.5. Here, 0-simplices are depicted by red points, 1-simplices by black edges, 2-simplices by yellow triangles and 3-simplices by blue tetrahedrons. For ϵ of 0.28, the number of connected components β_0 is 8, the number of 1-D holes β_1=1 (formed by four edges at the top) and the number of 2-D holes β_2=0.

The Betti functions can be regarded as a functional summary statistic of the network's topological structure. In contrast to other summaries for topological features such as a barcode [**?**], the newly proposed notion of Betti functions can not only be easily incorporated into machine learning models but also provide a systematic linkage with the tools of functional data analysis. In particular, due to the functional dependency among Betti numbers at different scales, it is important to view $\{\mathcal{B}_p(\epsilon_k)\}_{k=1}^n$ as a realization of Betti function $\mathcal{B}_p$ as opposed to a vector in $\mathbb{R}^n$. This point of view allows us to utilize methods from functional data analysis such as a concept of *functional data depth*.

Example. Consider the Betti functions in Fig. 3 that are computed for a daily network (left) of the Tronix token. The $\mathcal{B}_0$ function reaches 0 at ϵ of 1, whereas $\mathcal{B}_1$ and $\mathcal{B}_2$ functions reach non-zero values around 0.7 and 1.5, respectively. We find visible dependencies among Betti functions. Note that as the dissimilarity scale ϵ increases, $\mathcal{B}_0$ values tend to disappear and, reverse, $\mathcal{B}_1$ appears. Similarly, a transition from $\mathcal{B}_1$ to $\mathcal{B}_2$ occurs at ϵ of 2. The shape of these functions encode important information about the token network.

4.2 Data depth of Betti functions Let $\{(G_t, \tilde{\omega}_t)\}_{t=1}^T$ be a time series of weighted graphs and $\{\mathcal{B}_{p,t}\}_{t=1}^T$ be the associated sequence of Betti functions (see Section 4.1). To assess which Betti functions (or equivalently which transaction networks) are anomalous relative to others, we employ the notion of data depth.

Definition 5. (Data Depth) *If $\mathcal{X}$ is a Banach space (e.g., $\mathbb{R}^n$) and $\mathcal{F}$ is a set of probability distributions on its Borel subsets, then a data depth is a function $D : \mathcal{X} \times \mathcal{F} \to [0, 1]$ such that $D(\cdot|F)$ is a center-outward*

Copyright © 20XX by SIAM
Unauthorized reproduction of this article is prohibited

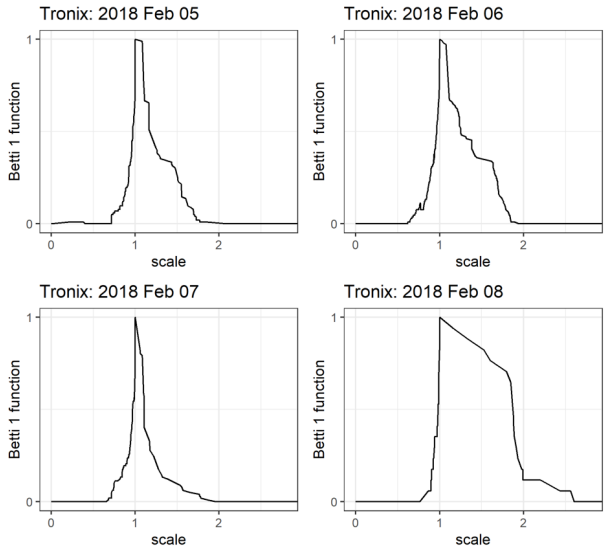


Figure 5: Betti functions of the Tronix token. Figure 1b shows the corresponding Betti signature.

ordering of elements of $\mathcal{X}$ with respect to F. The depth of $y \in \mathcal{X}$ with respect to $\{y_i\}_{i=1}^m \subseteq \mathcal{X}$ is denoted by $D(y|y_1, y_2, \ldots, y_m)$ and is defined as $D(y|\hat{F}_m)$, where $\hat{F}_m$ is the empirical distribution of $\{y_i\}_{i=1}^m$.

Since we focus on Betti functions, we resort to functional depth functions (where $\mathcal{X}$ is a Banach space of functions). Of them, the modified band depth (MBD) [**?**] is particularly suitable for identifying anomalies as it accounts for both the shape and magnitude of the curves. Additionally, MBD is robust and enjoys fast computational implementation. However, our framework is general enough and in principle can take in any other concept of functional data depth.

Intuitively, MBD measures the extend to which data point y lies within a given set $\mathcal{Y}$. More precisely, it is the average of proportions of times for which y lies in the bands determined by all possible pairs in $\mathcal{Y}$. MBD enables us to order a set of functions in $[0, 1]$-scale, where the depth values closest to zero and one correspond to the most anomalous and central functions, respectively. We introduce a concept of *Betti signature* which is defined as the deepest or most central Betti function.

Definition 6. (BETTI SIGNATURE) *For a given collection of Betti functions $\{\mathcal{B}_{p,t_1}, \mathcal{B}_{p,t_2}, \ldots, \mathcal{B}_{p,t_m}\}$, their Betti signature is defined as*

$$\mathcal{B}_p^s := \underset{\mathcal{B}_{p,t} \in \{\mathcal{B}_{p,t_1}, \ldots, \mathcal{B}_{p,t_m}\}}{argmax} MBD(\mathcal{B}_{p,t}|\mathcal{B}_{p,t_1}, \ldots, \mathcal{B}_{p,t_m})$$

Consider Betti functions $\{\mathcal{B}_{p,t}\}_{t=1}^T$ associated with an evolving token transaction network over days $t = 1, 2, \ldots, T$. For instance, Figure 5 shows the Betti functions of the Tronix token in four consecutive days. Although each day visually looks different, the network February 8th presents a clear anomaly in terms of its shape. To measure how the Betti functions change over time and compare with the ones prior to them, we calculate the MBD depth of each day's Betti function with respect to those of the past w days. To this end, we introduce a notion of rolling depth (RD):

$$RD_w(\mathcal{B}_{p,t}) := MBD(\mathcal{B}_{p,t}|\mathcal{B}_{p,t}, \mathcal{B}_{p,t-1}, \ldots \mathcal{B}_{p,t-w+1}).$$

The concept of RD echoes the rolling window approaches used to detect signals of short and long term trends in algorithmic trading and to construct stock price indicators such as Percentage Price Oscillator and Moving Average Convergence Divergence [**?**].

4.3 Anomaly Detection with Topological Features To predict whether the Ethereum token price will change more than $\delta, |\delta| > 0$ within the next h days horizon, we combine the graph topological features defined in Sections 4.2 and 4.3 with traditional network summaries. We start by defining the absolute price return of a token on day t as $\mathrm{R}_t = (Price_t - Price_{t-1})/(Price_{t-1})$. Then, we label a day t as anomalous if there is a significant change in token's price. More specifically, if $|\mathrm{R}_t| \geq \delta$ where $\delta > 0$ is a user-defined threshold (i.e., magnitude of a price shock), then t is considered as an anomalous day. We build one predictive model for each token and examine performance for different prediction horizons $h > 0$.

Our token-based price anomaly detection methodology for Ethereum crypto-tokens problem is summarized as follows. For each day, t, with available token data, we calculate the binary flag variable with values equal to *true* if abnormal price change ($|\mathrm{R}_t| \geq \delta$), was detected in at least one of the next h days (i.e., days $t+1, t+2, \ldots, t+h$) and *false* otherwise. Here, $t = 1, \ldots, T_k$ is the set of dates for which we have the k^{th} token data. For day t, we compute (1) the token's normalized open price, $\mathrm{PN} = Price_t / \max\{Price_1, \ldots, Price_{T_k}\}$. Next, we construct the user transactions network G for k^{th} token on day t. From G we calculate (2) the number of user transactions E (the number of edges in the network). Next, we induce a sub-network G' by selecting K users who have the most edges in the network G. From G', we calculate (3) 7-day rolling depth values for Betti numbers (β_0, β_1, and β_2), denoted as $RD_7(\mathcal{B}_0)$, $RD_7(\mathcal{B}_1)$ and $RD_7(\mathcal{B}_2)$, respectively. Alg. 1 details the curve generation process.

Rationale behind our modeling approach is that network topological features, summarized in terms of RD of Betti functions, will add an important layer of in-

Copyright © 20XX by SIAM
Unauthorized reproduction of this article is prohibited

Table 2: Model descriptions.

Model	Explanation	F: Input
$M1$	Baseline Model	PN, E
$M2$	Betti 0	PN, E, $RD_7(\mathcal{B}_0)$
$M3$	Betti 0, 1	PN, E, $RD_7(\mathcal{B}_0), RD_7(\mathcal{B}_1)$
$M4$	Full model	PN, E, $RD_7(\mathcal{B}_0), RD_7(\mathcal{B}_1)$, $RD_7(\mathcal{B}_2)$
$M5$	Betti 1	PN, E, $RD_7(\mathcal{B}_1)$
$M6$	Betti 2	PN, E, $RD_7(\mathcal{B}_2)$
$M7$	Betti 0, 2	PN, E, $RD_7(\mathcal{B}_0), RD_7(\mathcal{B}_2)$
$M8$	Betti 1, 2	PN, E, $RD_7(\mathcal{B}_1), RD_7(\mathcal{B}_2)$

formation that can be missed by the traditional network summaries. Hence, to test the improvement in anomaly prediction due to adding the network topological features, we evaluate predictive performance of the four models listed in Table 2 (see [**?**] for all considered models), using the traditional (token price and the number of transactions) and topological variables (rolling depth values of Betti functions). We fit each of the four models using the `randomForest` package in `R`. Parameter values in these prediction models is described in Section 5. We use the first 2/3 of a token's timeline period as training and the remaining 1/3 of the period as the test data set.

Algorithm 1 Betti Curves Generation

procedure CURVE(G: token graph, K: filter, d: Betti dimension max, w: window)
  induce graph G' for $top-K$ nodes
  compute $\tilde{\omega}_{uv}$ for each $e=(u,v)\in G'$
  for Betti dimension p $=\{0,\ldots,$d$\}$ **do**
    for each day $G'_t \in G'$ **do**
      compute $\mathcal{B}_{p,t}$
      $F_p^t \leftarrow RD_w(\mathcal{B}_{p,t})$
  return feature matrix F

5 Experimental settings

This section explains settings for the four components of our work.

Betti Signatures. We compute the Betti functions for up to $p=2$ (i.e., $\mathcal{B}_{0,.}$, $\mathcal{B}_{1,.}$ and $\mathcal{B}_{2,.}$) by using GUDHI which is a generic open source C++ library for Topological Data Analysis.

In Section 4.3, we outline a filtered network approach to compute Betti functions, where only the top-K nodes and their transactions are used. We experimented with $k=50,60,\ldots,180$ and found that the best results are attained for k of 150. Due to space limitations, results with other k values are provided in the technical report [**?**]. Our data indicates that even for the most traded tokens such as Tronix and Bat, top 150 nodes in daily networks create 75% and 80% of all edges,

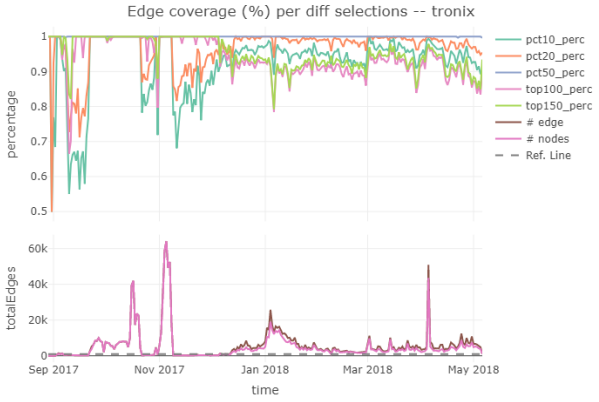


Figure 6: Selection: Tronix

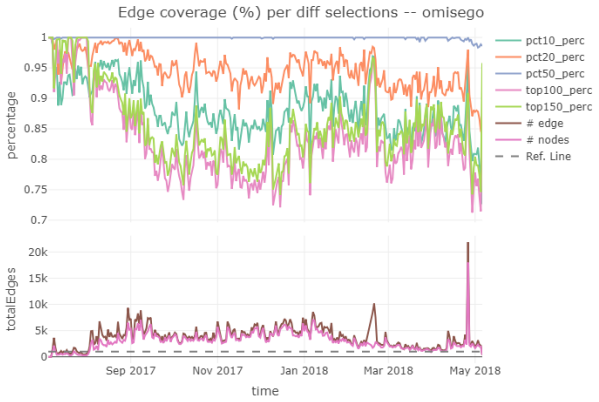


Figure 7: Selection: Omesigo

respectively. The filtered node approach effectively removes $20-25\%$ of the edges in Betti calculations, which results in more competitive performance.

The selection method is compared with top 100, top 150, top 10%, top 20% and top 50% here. Our conclusion is that the top 150 selection works very well if daily nodes is less than 1000. At the worst case, it still includes 80% edges. Even for most intensive tokens such as tronix and omesigo, top 150 is covering at least 75% edges. If the top player's size is greater than 150, it will occasionally cause betti computation error due to hardware limitation. Therefore, we choose 150 as the efficient and safe selection method.

Prediction Models. In our study, each Random Forest model used ntree= 500 trees, and sampling all rows of the dataset is done with replacement. Number of variables used at each split (mtry argument in `randomForest` function) are for each of the four models is the floor of number of features (as given by Table 2)(i.e., $\lfloor\sqrt{|F|}\rfloor$).

Anomaly Definition. In Section 4.3 we define anomalous days based on a price change threshold δ.

Copyright © 20XX by SIAM
Unauthorized reproduction of this article is prohibited

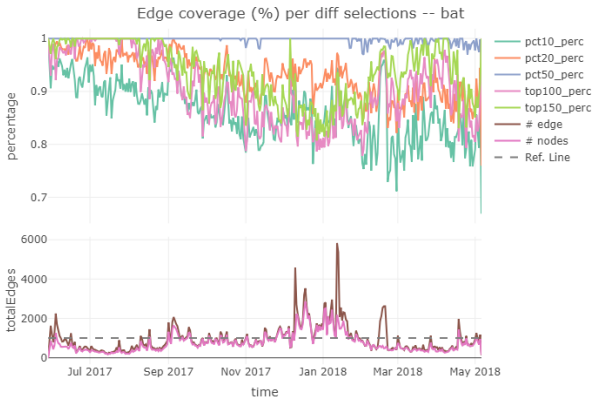


Figure 8: Selection: Bat

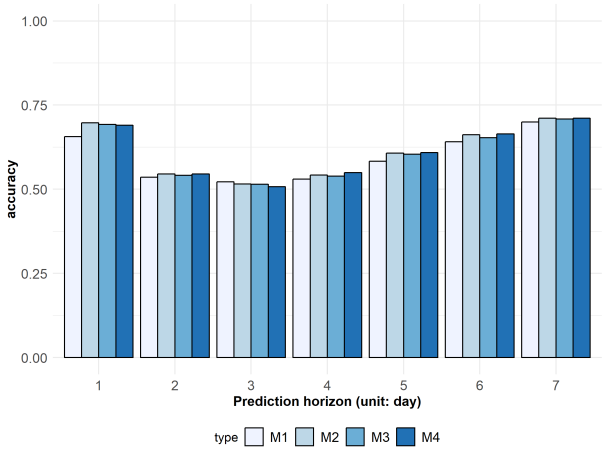


Figure 9: Accuracy: $\delta = 0.1$

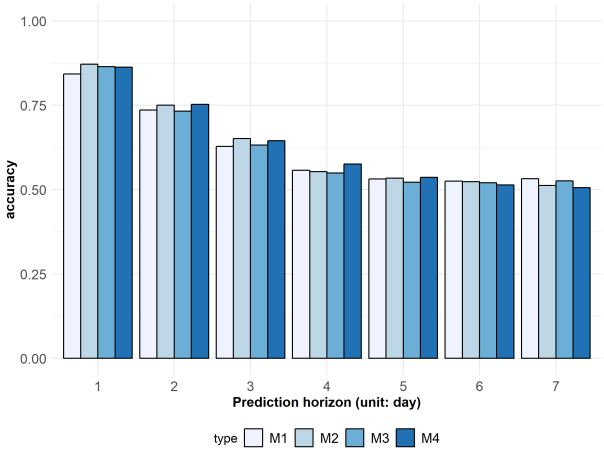


Figure 10: Accuracy: $\delta = 0.15$

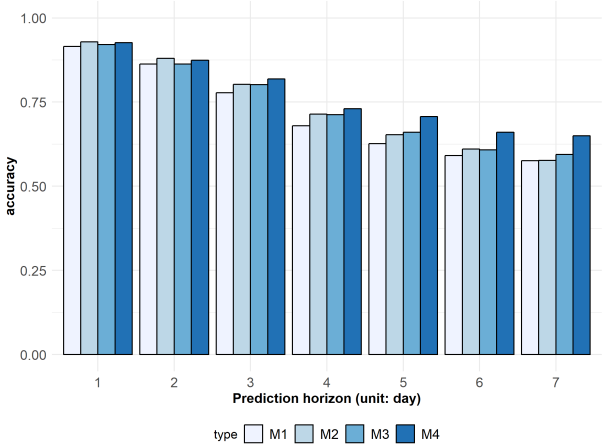


Figure 11: Accuracy: $\delta = 0.2$

In Figure 12 we show the distribution of price change (i.e. PC) in 31 token networks for the interval $-0.5 \leq$ PC ≤ 0.5, which contains 98.4% of all values in 9042 days.[2] Token networks have a mean of 0.08 in price return, and the Veros token has the maximum return of 6.49. In order to choose the δ value, we experimented with $\delta = 0.05, 0.1, \ldots, 0.5$. Our results show that the model accuracy and δ values are positively correlated. However, there is a trade-off between accuracy and number of predicted anomalies. We set $\delta = 0.25$ for better accuracy, but in [**?**] we offer a detailed discussion for all the considered δ values.

When δ is lower. All models seem to have close accuracy performance. But at certain horizons, the topological model can have better accuracy. However, when δ increase to 0.2, this pattern becomes more obvious that topological models have higher accuracy.

Performance metrics. In our results, the accu-

[2] 1.6% of price changes are outside the $[-0.5, +0.5]$ range.

Copyright © 20XX by SIAM
Unauthorized reproduction of this article is prohibited

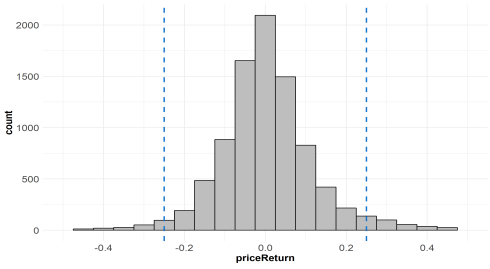


Figure 12: A histogram of daily Ethereum token price changes ($\mu = 0.08$).

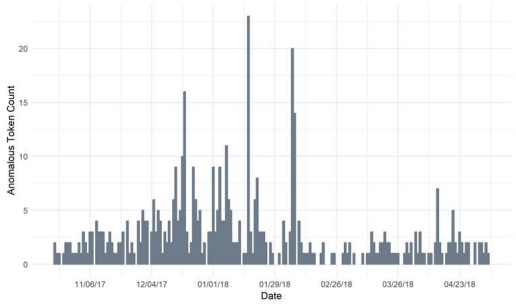


Figure 13: Number of (price) anomalous tokens in time.

racy is calculated as $(TP+TN)/(TP+FP+TN+TP)$, where TP, TN, FP and FN refer to *true-positive*, *true-negative*, *false-positive* and *false-negative*, respectively. Precision and sensitivity are defined as $TP/(TP+FP)$ and $TP/(TP+FN)$ respectively. The specificity is calculated as $(TN)/(TN+FP)$.

6 Experimental Results

In this section, we demonstrate the effectiveness of our approach under different settings. In the interests of providing scientifically reproducible results, all code and datasets used in this work are available at EthereumCurves.Github.io. Due to space limitations, throughout this text we cite [**?**] to inform the reader about a detailed version of the discussed results.

6.1 Model performance We predict price anomalies in 31 token networks, where a total of 9042 days are predicted as anomalous (anomaly:true) or non-anomalous (anomaly:false). In 145 of these days, a true price anomaly occurs, as defined by a price return of more than 25%. Mean and median numbers of anomalies are 6.59 and 2 per token, respectively. The Veros token had a maximum of 46 anomalies. Nine tokens do not have any price anomalies in their test period (the last 1/3 of their timeline). Number of anomalous tokens are given in Fig. 13. In the days leading up to 2018 January, token prices saw big increases; as shown in the figure, on some days more than 20 tokens had $> 25\%$ price returns. In this period (Oct-Dec 2017) price of the Ethereum currency, ether, increased from \$305 to \$1389. In 2018 Jan we see token prices decreasing sharply, but unlike the increase period, we observe fewer (≤ 7) anomalies in tokens on the same day.

Fig. 15 shows the number of anomaly:true predictions by models. M2, M3 and M4 (Betti models) predict the same 52 days as anomalous. A further 9 days are predicted anomalous by a single Betti model only. Betti models make a lower number of anomaly (i.e.,true) predictions compared to the baseline M1 model, which uses

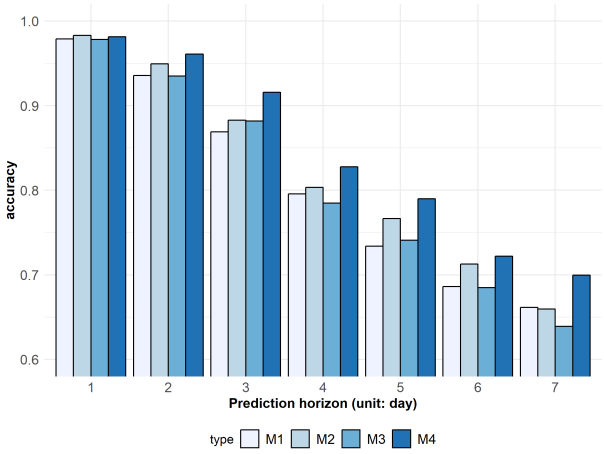


Figure 14: Accuracy

the price and transaction information only. For up to three day horizons in Fig. 14, all models have accuracy > 0.7. The figure shows that compared to other models, the M4 (full) model has the least deteriorating performance as horizon increases from 1 to 7. The accuracy results offer evidence that Betti models are more conservative in making anomalous day predictions, and their accuracy is better than the baseline model M1.

We report the sensitivity results in Fig. 16a, which show that model M3 has the highest sensitivity for all horizons. Other than at $h = 1$, Betti models consistently outperform the baseline model M1. Fig. 16b gives the specificity results. In the figure model M4 has the best values for $h > 1$. However, one of the Betti models, M3, performs worse than M1.

As M4 differs from M3 in its use of $\mathcal{B}_2$, we find the conflicting performance of M3 and M4 interesting. Sensitivity and specificity measure performance with regard to true positives and true negatives, respectively. As such, these results indicate that the use of $\mathcal{B}_2$ in M4 decreases performance in true negative predictions, but helps in predicting true positives. In other words, an increase in two dimensional holes on networks can be

Copyright © 20XX by SIAM
Unauthorized reproduction of this article is prohibited

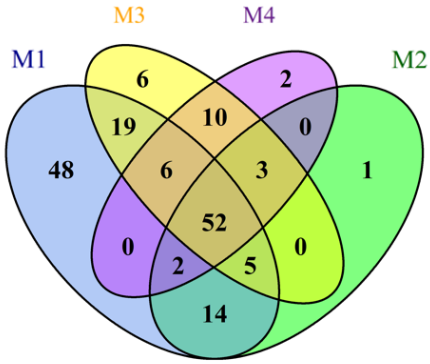
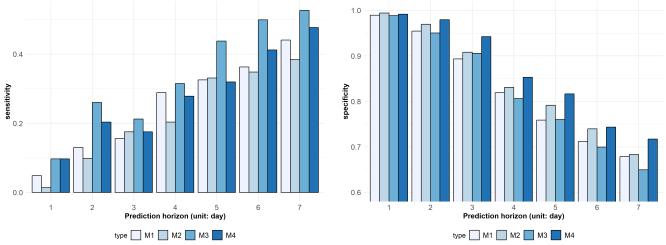


Figure 15: A Venn diagram for the number of predicted anomalies in all token networks (for $h = 1$). Intersecting regions indicate agreement on predictions.

used as a predictor of anomalies.



(a) Sensitivity (b) Specificity

Figure 16: Performance for increasing horizon values.

6.2 Predictability in Token Networks Fig. 20a overlays the accuracy of model M4 ($h = 1$) with the average normalized prices of 31 tokens. The confidence bands around central lines indicate the spread in predictions and prices. Price increases of late 2017 are very prominent in the figure, as well as the Ethereum price crash of Jan-Feb 2018. From March 2018, prediction accuracy becomes more stable, hovering just above 0.9. We show the accuracy for each token in Table 4 [**?**].

Although predicting true negatives (non-anomalous days) is useful, the most important task of anomaly detection is to predict true anomalies well in advance. The unbalanced nature of our dataset complicates this task; only 1.58% of all days are true anomalies, limiting the training cases to a few days per each token.

To account for the unbalanced data problems, we use precision to track model performance for true positives. We reach the highest average precision of 0.393 per token in the M4 model ($h = 2$) [**?**]. We outline two discussion points to explain this precision result.

Temporal effects: Our models use 2/3 of a token's

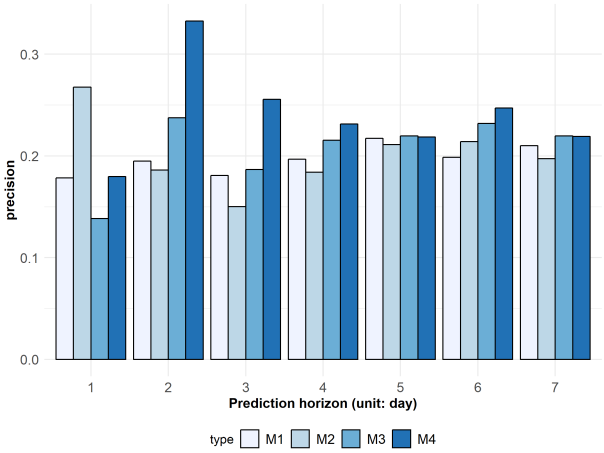


Figure 17: Precision: $\delta = 0.25$

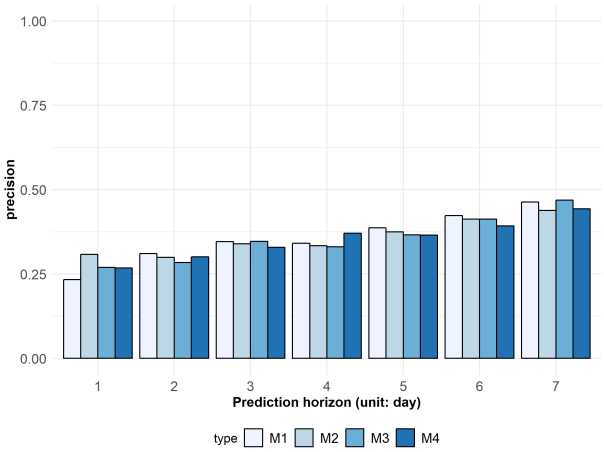


Figure 18: Precision: $\delta = 0.15$

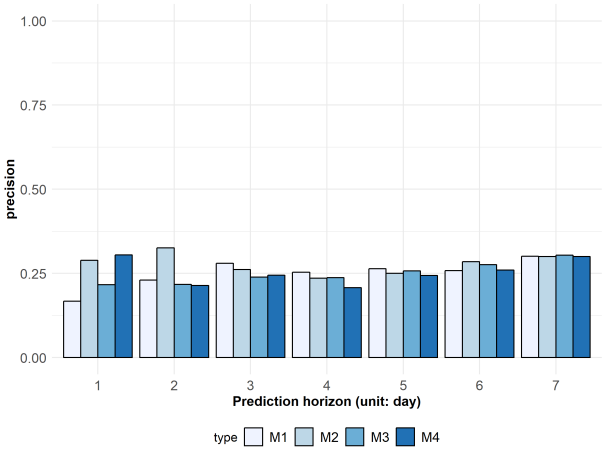


Figure 19: Precision: $\delta = 0.2$

Copyright © 20XX by SIAM
Unauthorized reproduction of this article is prohibited

lifetime for training, and the remaining 1/3 for test. An issue that complicates predictions is the temporal changes that the Ethereum blockchain has undergone during our experiments. As seen in Fig. 13, training (second half of 2017 for most tokens) and test (2018 Jan. and onward) periods have drastically different temporal patterns, in terms of reduced price in fiat currency and node activity on the blockchain. Tokens had fewer anomalies in the test period, and nine of them did not have any. A solution to this changing data behaviour issue is to analyze the blockchain continuously and identify limited time regions to train from as blockchain platforms mature and tokens accumulate more data. Currently, our EthereumCurves website [**?**] is designed to serve the community for this purpose.

Global view: The blockchain ecosystem, specifically Blockchain platforms, are still in infancy. As the ecosystem is still maturing, an event in an obscure part of the ecosystem can affect multiple tokens at once. An example of this is the *theDAO* incident, where a token's misfortunes had the Ethereum platform almost collapse. In addition to negative developments, perceived good news lead to change in activity and price as well. In Fig. 13, we see that there are days when more than 20 tokens had price anomalies (in increases) when Ethereum was gaining recognition in the world (late 2017). As *external events create sudden price changes (where the token network contains no preceding signal), training and testing for anomalies becomes a difficult task.* Once the ecosystem matures and each token moves by events related to its own development, we expect our models to capture price anomalies much better. Fig. 20b offers evidence for this assumption. In the later days of 2018, the figure shows that more anomalies could be predicted, compared to the early days of 2017. The model did not predict more than one anomaly per day in 2017, whereas 2018 has multiple days where we correctly predict two or three anomalies.

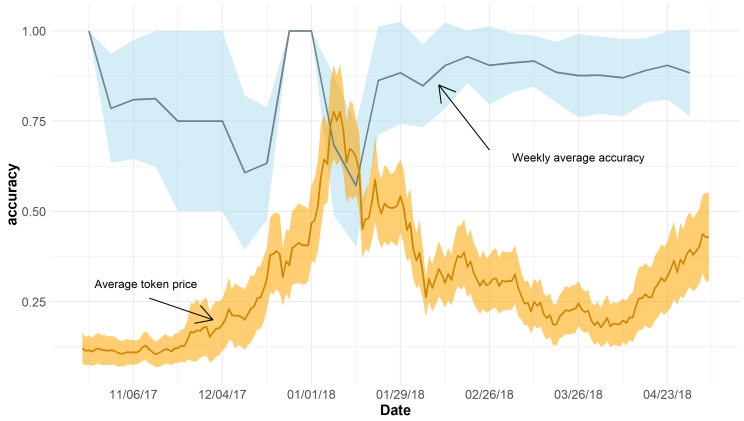
7 Conclusions

Our results offer strong evidence that Ethereum token anomalies can be predicted by analyzing token networks. Although data quality issues exist, we believe that effective analytic tools based on persistent homologies and functional data depth ideas can be developed for this burgeoning ecosystem. We present EthereumCurves as the first step in this direction. Our results indicate that topological features can bring significant improvement in predicting price anomalies in tokens.

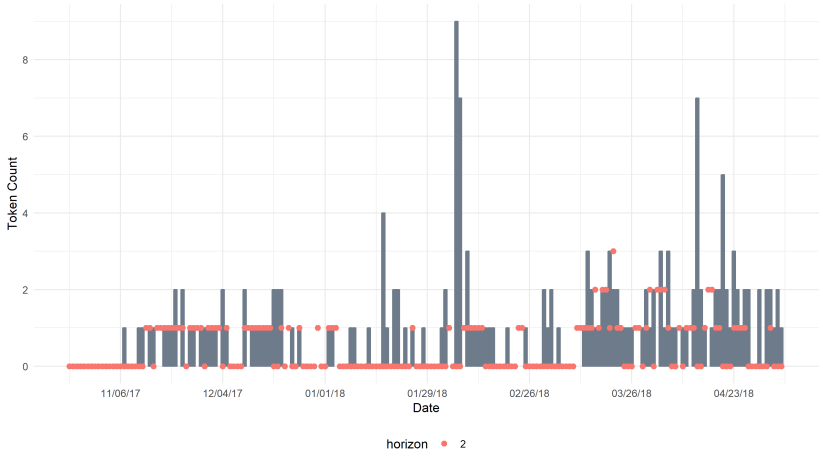
Table 3: Accuracy of models for all tokens ($h = 1$). The tokens are ordered by the average edge count $\bar{E}$ in their daily networks.

token	**M1**	**M2**	**M3**	**M4**	$\bar{E}$
tronix	0.937	0.987	0.987	0.987	5198.2
omisego	1.000	1.000	1.000	0.990	3027.7
mcap	0.939	0.948	0.887	0.904	1502.1
storj	0.990	0.990	0.990	0.990	1224.3
salt	1.000	1.000	1.000	1.000	1214.6
statusnetwork	1.000	1.000	1.000	1.000	1095.2
bnb	0.990	0.990	0.990	0.990	1089.5
golem	0.972	0.972	0.972	0.978	1065.0
powerledger	1.000	1.000	1.000	1.000	939.0
zrx	0.989	0.989	0.989	0.989	905.4
tenxpay	1.000	1.000	1.000	1.000	899.9
cybermiles	0.961	0.980	0.961	0.980	872.7
civic	1.000	1.000	1.000	1.000	864.3
vechain	0.977	0.977	0.977	0.977	851.7
kybernetwork	0.987	0.987	0.987	0.987	792.5
icon	0.908	0.908	0.892	0.923	783.5
bat	0.982	0.982	0.982	0.982	773.5
bytom	0.967	0.967	0.967	0.967	733.5
request	1.000	1.000	1.000	1.000	680.5
qtum	0.980	0.980	0.980	0.980	661.6
funfair	0.990	0.990	0.990	0.990	615.9
loopring	0.988	0.988	0.988	0.988	571.4
tierion	0.965	0.965	0.965	0.965	517.8
district0x	0.957	0.957	0.957	0.957	510.8
kin	1.000	1.000	1.000	1.000	455.5
polyai	0.463	0.667	0.463	0.574	380.1
veros	0.478	0.593	0.558	0.558	369.1
decentraland	0.987	0.987	0.987	0.987	351.6
aragon	0.975	0.983	0.950	0.958	329.2
yocoin	0.716	0.716	0.716	0.716	247.5
reputation	0.976	0.976	0.976	0.976	190.5

Copyright © 20XX by SIAM
Unauthorized reproduction of this article is prohibited



(a) Accuracy of model M4 ($h = 3, \delta = 0.25$), and average price for 31 tokens in time.



(b) True (bars) and predicted (dots) anomalies in the testing period of Model M4 for $h = 2$.

Figure 20: Model performance for horizons.

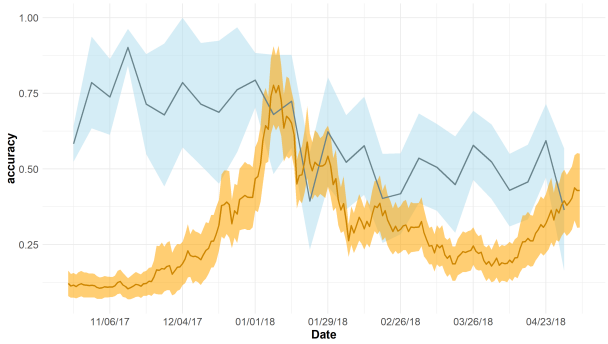


Figure 21: Accuracy of model M4 ($h = 3, \delta = 0.1$), and average price for 31 tokens in time.

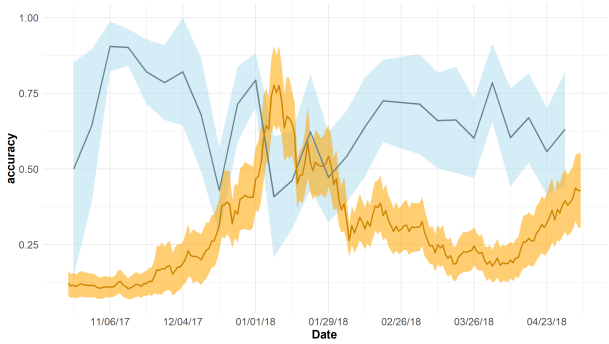


Figure 22: Accuracy of model M4 ($h = 3, \delta = 0.15$), and average price for 31 tokens in time.

Copyright © 20XX by SIAM
Unauthorized reproduction of this article is prohibited

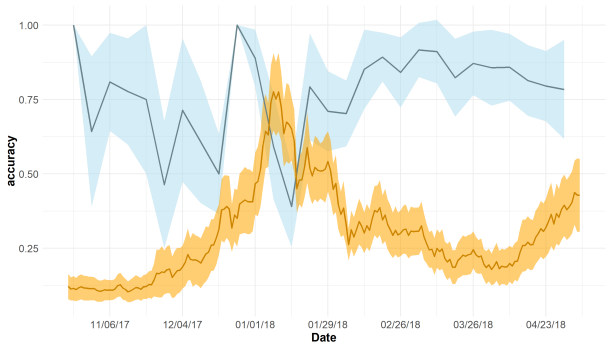


Figure 23: Accuracy of model M4 ($h = 3, \delta = 0.2$), and average price for 31 tokens in time.

Table 4: Accuracy of models for all tokens ($h = 3$). The tokens are ordered by the average edge count $\bar{E}$ in their daily networks.

token	**M1**	**M2**	**M3**	**M4**	$\bar{E}$
tronix	0.709	0.848	0.810	0.949	5198.2
omisego	0.420	0.530	0.620	0.770	3027.7
mcap	0.826	0.870	0.800	0.843	1502.1
storj	0.942	0.933	0.933	0.865	1224.3
salt	1.000	0.986	0.986	1.000	1214.6
statusnetwork	0.981	0.990	0.990	1.000	1095.2
bnb	0.865	0.917	0.875	0.875	1089.5
golem	0.888	0.916	0.927	0.933	1065.0
powerledger	1.000	0.968	1.000	1.000	939.0
zrx	0.798	0.910	0.876	0.933	905.4
tenxpay	1.000	0.990	1.000	0.952	899.9
cybermiles	0.902	0.941	0.941	0.941	872.7
civic	1.000	1.000	1.000	1.000	864.3
vechain	0.701	0.609	0.621	0.805	851.7
kybernetwork	0.961	0.974	0.974	0.961	792.5
icon	0.831	0.677	0.615	0.677	783.5
bat	0.947	0.947	0.947	0.947	773.5
bytom	0.769	0.813	0.780	0.923	733.5
request	0.985	1.000	1.000	1.000	680.5
qtum	0.853	0.833	0.941	0.941	661.6
funfair	0.960	0.950	0.950	0.960	615.9
loopring	0.929	0.952	0.940	0.929	571.4
tierion	0.871	0.882	0.894	0.894	517.8
district0x	0.871	0.871	0.849	0.871	510.8
kin	1.000	1.000	1.000	1.000	455.5
polyai	0.685	0.667	0.704	0.648	380.1
veros	0.637	0.690	0.611	0.619	369.1
decentraland	0.923	0.936	0.949	0.949	351.6
aragon	0.588	0.597	0.555	0.613	329.2
yocoin	0.526	0.483	0.474	0.474	247.5
reputation	0.944	0.944	0.944	0.944	190.5

Copyright © 20XX by SIAM
Unauthorized reproduction of this article is prohibited