

Topological Anomaly Detection in Dynamic Multilayer Blockchain Networks

Ofori-Boateng, D.¹✉, Segovia Dominguez, I.², Akcora, C.³, Kantarcioglu, M.,
and Gel, Y.R.²

¹ Portland State University, USA; dorcas.oforiboateng@pdx.edu

² University of Texas at Dallas, USA

³ University of Manitoba, Canada

Abstract. Motivated by the recent surge of criminal activities with cross-cryptocurrency trades, we introduce a new topological perspective to structural anomaly detection in dynamic multilayer networks. We postulate that anomalies in the underlying blockchain transaction graph that are composed of multiple layers are likely to also be manifested in anomalous patterns of the network shape properties. As such, we invoke the machinery of clique persistent homology on graphs to systematically and efficiently track evolution of the network shape and, as a result, to detect changes in the underlying network topology and geometry. We develop a new persistence summary for multilayer networks, called stacked persistence diagram, and prove its stability under input data perturbations. We validate our new topological anomaly detection framework in application to dynamic multilayer networks from the Ethereum Blockchain and the Ripple Credit Network, and demonstrate that our stacked PD approach substantially outperforms state-of-art techniques.

Keywords: Anomaly Detection · Dynamic Multilayer Network · Blockchain Transaction · Topological Data Analysis · Clique Persistent Homology.

1 Introduction

Due to the recent spike in popularity of crypto assets, detecting anomalies in time evolving blockchain transaction networks has gained a new momentum. Here anomaly detection in dynamic graphs can be broadly defined as the problem of identifying instances within a sequence of graph observations where changes occur in the underlying structure of the graph. Indeed, these anomalies have significant implications, ranging from emergence of new ransomware (e.g., collecting ransom via cryptocurrencies) to financial manipulation. For example, in blockchain transaction networks, e.g., Ethereum, more frequent than expected appearance of particular subgraphs may indicate newly emerging malware or price pump-and-dump trading [70]. Similarly, as recently shown by [68], the flow of coins on the Bitcoin graph provides important insights into money laundering schemes. As criminal, fraudulent, and illicit activities on blockchains continue to rise, with already stolen \$1.4B only in 2020, cryptocurrency criminals increasingly employ

cross-cryptocurrency trades to hide their identity [53]. As such, [72] have recently shown that the analysis of links across multiple blockchain transaction graphs is critical for identifying emerging criminal and illicit activities on blockchain. However, while there exists a plethora of methods for network anomaly detection in single layer networks [57,30,56], there is yet *no single* method designed to detect anomalies in dynamic multilayer networks.

Why TDA? Motivated by the problem of tracking financial crime on blockchains, we develop a state-of-the-art methodology for anomaly detection on multilayer networks using Topological Data Analysis (TDA). Since crime on blockchains such as money laundering tends to involve multiple parties who possibly move funds across multiple cryptocurrency ledgers, one of our primary goals is to identify anomalous patterns in higher order graph connectivity. We postulate that anomalous higher order patterns can be detected using geometric and topological inference on graphs, that is, via a systematic analysis of the graph shape. To explore latent graph shape, we invoke the TDA machinery of the clique persistent homology (PH). PH allows to systematically infer qualitative and quantitative multi-lens geometric and topological structures from data directly and, hence, to enhance our understanding on the hidden role of geometry and topology in the system organization [13,18,67]. As a result, it may be intuitive to hypothesize that there shall be an intrinsic linkage between changes in the underlying graph structure and changes in the network shape which are then reflected in the extracted network topological characteristics. However, to the best of our knowledge, this paper is the first attempt to introduce TDA to anomaly detection in dynamic multilayer networks.

Why Ethereum and Ripple? Using the Blockchain global events timeline [69], we validate our methodology in application to anomaly detection in two multilayer blockchain network types, Ethereum and Ripple. While cryptocurrencies have already been adopted in payments, the recent surge in financial blockchain activity is largely due to platforms, such as Ethereum, which have brought algorithmic trading of digital assets by using Smart Contracts (i.e. short software code on the blockchain) in what is called Decentralized Finance [21]. Assets include cryptocurrencies and crypto tokens as well. Hence, a given address (i.e. a node) may participate in transactions of multiple digital assets. Looking at an individual asset transaction network alone (i.e. a single layer of the transaction graph) may provide a limited view. As a result, we need to consider multiple layers (e.g., a layer for each crypto token) and their interactions to detect anomalies. Resulting multilayer networks and participant activities are temporal, nuanced in the traded assets (e.g., coins, or fiat currencies), rich in network patterns and encode a new wave of financial heart-beat. The Ripple Credit Network transactions also comprise cross-border remittance transfers and even fiat currency trades, allowing trading Ether, Bitcoin and other currencies on its system.

Our contributions, both in application and theory, are as follows:

- 1 To the best of our knowledge, this is the first paper on anomaly detection in dynamic multilayer networks.

- 2 Our new methodology is based on the notion of clique persistent homology. To quantify topology of multilayer graphs, we introduce a multidimensional multi-set object, called the *stacked persistence diagram* (SPD). We prove that SPD is robust against minor input data perturbations w.r.t. bottleneck distance.
- 3 In the absence of the state-of-the-art anomaly detection methods for dynamic multilayer networks, we benchmark our topological anomaly detection (TAD) tool against a multiple testing framework, based on the strongest state-of-the-art (SOTA) methods for anomaly detection in single layer networks. To control for family wise error rate (FWER) in the multiple testing framework, we use Bonferroni correction. We show that TAD substantially outperforms all competitors based on SOTA single layer solutions and the additional technique based on graph embedding.
- 4 We demonstrate utility of TAD on Ethereum and Ripple blockchains, where digital assets worth billions of US Dollars are traded daily. We provide Blockchain benchmark data for anomaly detection on multilayer networks which is the first benchmark multilayer network dataset with ground-truth events, thereby further bridging AI with crypto-finance.

2 Related Work

Graph-Based Anomaly Detection: Over recent years, there has been an increase in application of anomaly detection techniques for single layer graphs in interdisciplinary studies [73,28]. For example, [41] employed a graph-based measure (DELTA CON) to assess connectivity between two graph structures with homogeneous node/edge attribution, and identified anomalous nodes/edges in the sequence of dynamic networks based on similarity deviations. With DELTA CON, an event is flagged as anomalous if its similarity score lies below a threshold. In turn, [66] devised a likelihood maximization tool that extracts a "feature" vector from individual networks, and uses dissimilarity between successive networks snapshots to classify anomalous or normal/regular events. Procedure of [77] segments network snapshots into separate clusters, infers local and global structure from individual nodes and their distribution via community detection and chronological ordering of the results in an effort to single-out potential anomalies. An online algorithm for detecting abrupt edge weight and structural changes in dynamic graphs has been recently introduced by [71], but the method requires a pre-training data set to identify tuning parameters. In turn, [48,61,7] discuss detection of malicious nodes in multiplex/multilayer networks. Finally, [25] proposed a score test for change point detection in multilayer networks that follow a multilayer weighted stochastic block model (SBM). However, the SBM assumption is infeasible for financial networks. To our knowledge (see also the reviews by [30,56]), *there is no existing anomaly detection method designed for dynamic multilayer networks.*

Blockchain: Blockchain graphs have been extracted and analyzed for price prediction [34,2,42], measurement studies [65,43] and e-crime detection [20,3]. Graph anomalies have been tracked to locate coins used in illegal activities,

such as money laundering and blackmailing [55]. These findings are known as taint analysis [24]. Typically, a set of features are extracted from the blockchain graph and used in Machine Learning (ML) tasks. Here we bypass such a feature engineering step in learning on Blockchain networks. Ethereum structure has been analyzed by [31,43], while anomalies in Ethereum token prices have been evaluated using TDA tools [46]. In turn, Ripple has been assessed for its privacy aspects [51] and for health of the credit network [50]. However, multilayer analysis of blockchains have not been studied before.

TDA: Multiple recent papers show utility of TDA for developing early warning signals for crashes in the cryptocurrency market [33], cryptocurrency price analytics [46], and ransomware detection on blockchain transaction graphs [3]. While TDA (as any other tool) cannot be viewed as a universal solution, TDA allows us to assess graph properties which are invariant under continuous deformations; hence it is likely to be one of the most robust tools for blockchain data analytics [75]. TDA has been employed for visual detection of change points in single layer graphs [36]. In the multilayer network context, TDA has been used primarily for centrality ranking [62], including analysis of connectivity in the multiplex banking networks [22], and clustering [74]. Application of TDA to anomaly detection in multilayer networks is yet an unexplored area.

Multilayer Network Benchmark Data: Multilayer networks receive an increasing attention in the last few years, due to their flexibility of modeling interconnected systems [4]. There also exist several data repositories with multilayer graphs, e.g. [29,5], but neither of them have publicly available benchmark data on multilayer graphs with ground truth for anomaly detection.

3 The Mechanism of Persistent Homology

Topology is the study of shapes. TDA and, in particular, *persistent homology* (PH) provides systematic mathematical means to extract the intrinsic shape properties of the observed data \mathcal{X} (in our case \mathcal{X} is a multilayer graph but \mathcal{X} can be a point cloud in Euclidean or any finite metric space) that are invariant under continuous transformations. The key postulate is that \mathcal{X} are sampled from some metric space \mathcal{M} whose properties are lost due to sampling. The goal of PH is then to reconstruct the unknown topological and geometric structure of \mathcal{M} , based on systematic shape analysis of \mathcal{X} . In this paper, we introduce the PH concepts to analysis of dynamic multilayer networks, starting by providing background on PH on graphs.

Definition 1. Let $\mathcal{G} = (V, E, \omega)$ be a (weighted) graph, with vertex set V , edge set $E = \{e_1, e_2, \dots\} \subseteq V \times V$, edge weights $\omega = \omega(e) : E \rightarrow \mathbb{Z}^+$ for all $e \in E$.

At the initial stages of PH, we select a certain threshold $\nu_* > 0$, and then we generate a subgraph $\mathcal{G}_* = (V, E_*, \omega_*)$, such that $E_* = \{e \mid \omega(e) \leq \nu_*\}$, and $\omega_*(e) = \omega(e)$, for all $e \in E_*$. Then the observed graph \mathcal{G}_* is equipped with a basic combinatorial object known as an *abstract simplicial complex*. Formally, a simplicial complex is defined as a collection \mathcal{C} of finite subsets of $V(\mathcal{G})$ such

that if $\sigma \in \mathcal{C}$ then $\tau \in \mathcal{C}$ for all $\tau \subseteq \sigma$. The basic unit of simplicial complexes is called the *simplex*, and if $|\sigma| = m + 1$ then σ is called an m -simplex. Specific to our analysis, we use a simplicial complex type called the *clique complex* to systematically and efficiently extract topological features from the observed \mathcal{G} . A clique complex $\mathcal{C}(\mathcal{G}_*)$ is a simplicial complex with a simplex for every clique (i.e., a set of vertices of \mathcal{G}_* such that any two points in the clique are adjacent) in \mathcal{G}_* . Furthermore, a k -clique community is formed whenever two k -cliques share $k - 1$ vertices ($k \in \mathbb{Z}^+$). With a range of thresholds $\nu_1 < \dots < \nu_n$, we can obtain a hierarchically nested sequence of graphs $\mathcal{G}_1 \subseteq \dots \subseteq \mathcal{G}_n$ for any graph \mathcal{G} , where each individual subgraph will generate its own clique complex. Subsequently, the procedure which generates complexes from the nested sequence $\mathcal{G}_1 \subseteq \dots \subseteq \mathcal{G}_n$ is known as the *network filtration*, and the resultant complex generated by \mathcal{G} is called a *filtered complex* [78]. Particular to cliques, we construct clique complexes and then obtain the *clique filtration* $\mathcal{C}(\mathcal{G}_1) \subseteq \dots \subseteq \mathcal{C}(\mathcal{G}_n)$.

The mechanism of *clique persistent homology* involves tracking clique complexes over the filtration and quantifying lifespan of topological features/shapes such as loops, holes, and voids that appear and disappear at various thresholds ν_* [79,60]. We say that a topological feature is born at the i -th filtration step if it appears in $\mathcal{C}(\mathcal{G}_i)$, and the topological feature dies at the j -th filtration step, if it disappears at $\mathcal{C}(\mathcal{G}_j)$. Hence, the lifespan of a topological feature is $\nu_j - \nu_i$. The primary objective of TDA will then be to assess which topological features/shapes persist (i.e. have longer lifespan) over the clique filtration and, hence, are likelier to contain important structural information on the graph, and which topological features have shorter lifespan. The latter features are typically referred to as topological noise.

One of the most widely used topological summaries is the *persistence diagram* (PD) [13,78]. The PD is a collection of points $(v_i, v_j) \in \mathbb{R}^2$ with each point corresponding to a topological feature, and the x - and y -coordinates representing birth and death times for the topological feature. Similarity between any two PDs, D_a and D_b , can be computed using the Wasserstein (W_r) or the Bottleneck distances (W_∞):

$$W_r(D_a, D_b) = \left(\inf_{\eta} \sum_{x \in D_a} \|x - \eta(x)\|_\infty^r \right)^{1/r}, \quad W_\infty(D_a, D_b) = \inf_{\eta} \sup_{x \in D_a} \|x - \eta(x)\|_\infty.$$

Here $r \geq 1$, η ranges over all bijections from $D_a \cup \Delta$ to $D_b \cup \Delta$, counting multiplicities, with $\Delta = \{(x, x) | x \in \mathbb{R}\}$ and $\|z\|_\infty = \max_i |z_i|$ [40,67]. We evaluate both distances in the methodological development of the TAD.

4 Persistence Methodology for Network Anomaly Detection

We now introduce the new topological method (TAD) for anomaly detection on multilayer graphs and support its design with relevant theoretical guarantees. Table 1 details all notations we introduced, and we use the terms graph and network interchangeably.

Table 1: Description of Symbols Used

Symbol	Description
\mathcal{X}	point cloud
\mathcal{M}	metric space
$\mathcal{G}; V, E$	graph; node and edge lists
$\omega(e)$	weight of edge e
\mathcal{C}	simplicial complex
σ	simplex
$\mathcal{C}()$	clique complex
D_a	persistence diagram a
W_r, W_∞	Wasserstein and Bottleneck distances
L, T	number of layers and time intervals
\mathcal{G}^l	graph of a single layer l
ω^{l+}	geodesic edge weight
$\mathbb{D}()$	distance between two persistence diagrams
\mathcal{G}_t^{l+}	geodesically densified graph of layer l at time t
d_{GH}	Gromov-Hausdorff distance
\mathbb{F}	persistence module
$d_i()$	interleaving distance
k_{max}	maximum clique size

Definition 2 (Multilayer network). A multilayer network, $\mathbb{G} = (\mathcal{G}^1, \dots, \mathcal{G}^L)$, is a graph structure that consists of L non-overlapping graph layers, where each layer is modeled with a (weighted) graph $\mathcal{G}_i = (V_i, E_i, \omega_i)$, with $i = 1, \dots, L$.

Problem Statement: Let $\{\mathbb{G}_t\}_{t=1}^T = \{(\mathcal{G}_t^1, \dots, \mathcal{G}_t^L)\}_{t=1}^T$ be a T sequence of multilayer networks observed over time t , with $1 \leq t \leq T < \infty$. The objective is to locate a time point $t^* < T$, such that an event within the time range $[t^* - m, t^* + m]$, for $0 \leq m < t^*$ causes the structure and shape of \mathbb{G}_{t^*} to differ from the structural properties of the earlier observed networks $\mathbb{G}_1, \dots, \mathbb{G}_{t^*-1}$. With this search, we include anomalies which cause: ❶ the network system to experience a brief shock at t^* , and ❷ a permanent change in the network system until the next $t^* + m$.

Main Idea: Conceptually, TAD method is designed to associate anomalies in the sequence of multilayer networks to anomalies identified from the time series of their topological summaries. In addition, we introduce our new idea of a specialized persistence diagram for multilayer networks known as the *stacked persistence diagram* (SPD).

Definition 3 (Stacked Persistence Diagram (SPD)). For a multilayer network $\mathbb{G} = (\mathcal{G}^1, \dots, \mathcal{G}^L)$, we define the associated PD of \mathbb{G} as $D_{\mathbb{G}} = (D_{\mathcal{G}^1} \oplus \dots \oplus$

$D_{\mathbb{G}^L}$), i.e. $D_{\mathbb{G}}$ is created as a direct sum of all PDs $D_{\mathbb{G}^l}$ associated with each single intra-/inter-layer network $\mathcal{G}^l \subseteq \mathbb{G}$, for $1 \leq l \leq L$.

Why Do We Stack PDs and Why Not to Average PDs? As our primary focus here is on anomaly detection in multilayer graphs, our goal is to *simultaneously capture joint dynamics* of topological properties exhibited by each graph layer within the interconnected system. As such, currently existing methods based on averaging PDs and their vectorizations [52,8] which are developed for analysis of a single, possibly time-varying object, are not feasible in our context. That is, averaging PDs of the two distinct layers may be viewed as averaging PDs, extracted from apples and oranges. In turn, our idea is to jointly track dynamic topological properties which are demonstrated by apple and orange trees over the same time period, and the SPD structure is motivated by the notion of direct sums of multiple vector spaces which serve as mathematical formalization of very different objects.

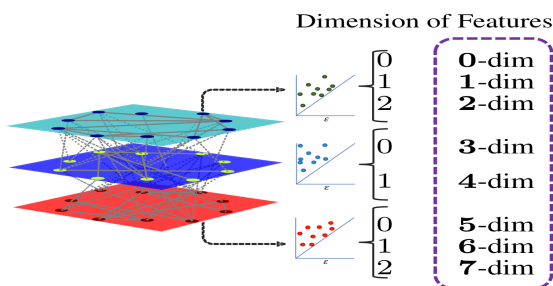


Fig. 1: An example of the formation of the SPD for a multilayer network. The multilayer network has 3 layers with PDs that have unequal topological features (3 in the first, 2 in the second, and 3 in the third). Although the first and third layer PDs contain information about 3-dimensional topological features, they have unequally-positioned points. Essentially, the SPD for the multilayer network will contain information about 7 classes of topological features.

Geodesic Densification of Blockchain Graphs: Dynamic networks such as Blockchain transaction graphs tend to be sparse, because a node (i.e. an address) can be inexpensively created without proving identity, which allows users to hide their transactions behind new addresses for privacy and security purposes. Furthermore, blockchain communities (e.g. Bitcoin) encourage one-time-use addresses (i.e. creating a new address every time a transaction is created). As a result, a sparse and constantly evolving network structure emerges, making it difficult to rely on conventional network connectivity (i.e. adjacency matrix). To address this limitation, we replace the (weighted) adjacency matrix of the single layer graph \mathcal{G}^l of \mathbb{G} with the (weighted) geodesic distance (GD) matrix [9] which redefines the edge weights ω^l as $\omega^{l+} = \sum_{e \in E(P_{uv})} \omega(e)$, where P_{uv} is the shortest path length between vertex pair u, v . This densification reconnects node

pairs that have a common path. Paths encode useful information because nodes (i.e. addresses) may merge their coins into a single address to sell them to leave the Blockchain (and thus pay less transaction fees).

The proposed TAD framework operates according to the following order:

$$\{\mathbb{G}_t\}_{t=1}^T \xrightarrow{T\text{-step}} \{\mathbb{D}(D_{\mathbb{G}_{t-1}}, D_{\mathbb{G}_t})\}_{t=2}^T \xrightarrow{AD\text{-step}} \{t_1^*, \dots\},$$

where $\mathbb{D}(D_{\mathbb{G}_{t-1}}, D_{\mathbb{G}_t})$ is any suitable distance metric between two persistence diagrams $D_{\mathbb{G}_{t-1}}$ and $D_{\mathbb{G}_t}$. Note that this distance can either be the Bottleneck or the r -th Wasserstein distance.

T – step: At this step, we implement the clique PH to convert the sequence of multilayer networks $\{\mathbb{G}_t\}_{t=1}^T$ into a sequence of SPDs. This involves the transformation of all the (weighted) adjacency matrices of \mathcal{G}_t^l into \mathcal{G}_t^{l+} , followed by the filtration of persistent topological features by using clique PH.

AD – step: While TAD method can be integrated with any user-preferred outlier or change point detection algorithm for univariate time series, we adopt the recently proposed seasonal extreme studentized deviate test S-ESD [63,37]. For an observed time series, S-ESD filters out the seasonal component, piecewise approximates the long-term trend component (in order to decrease the instances of false positives) and then incorporates robust statistical learning to identify the location of anomalies. S-ESD is our choice due to its sensitivity to both global anomalies irrespective of seasonal trends and intra-seasonal local anomalies. We provide pseudocode for TAD below, and discuss its computational complexity as well.

Algorithm 1: Topological anomaly detection in multilayer networks (TAD)

Input : Sequence of L -multilayer graphs $\{\mathbb{G}_t\}_{t=1}^T = \{\mathcal{G}_t^1, \dots, \mathcal{G}_t^L\}_{t=1}^T$.
Output : Anomalous events $\{t_1^*, \dots\}$.

- 1 **for** $t \leftarrow 1 : T$ **do**
- 2 **for** $l \leftarrow 1 : L$ **do**
- 3 Compute GD matrix \mathcal{G}_t^{l+} for \mathcal{G}_t^l
- 4 Generate the PD $\mathcal{D}_{\mathcal{G}_t^{l+}}$ for \mathcal{G}_t^{l+}
- 5 **end**
- 6 Obtain SPD $\mathcal{D}_{\mathbb{G}_t}$ by chronologically stacking PDs from $\mathcal{D}_{\mathcal{G}_t^{1+}}$ to $\mathcal{D}_{\mathcal{G}_t^{L+}}$
- 7 **end**
- 8 **for** $t \leftarrow 2 : T$ **do**
- 9 With suitable distance metric (\mathbb{D}), obtain similarity between $\mathcal{D}_{\mathbb{G}_{t-1}}$ and $\mathcal{D}_{\mathbb{G}_t}$
- 10 **end**
- 11 With S-ESD, detect anomalies (t_1^*, \dots) from the series $\{\mathbb{D}(\mathcal{D}_{\mathbb{G}_1}, \mathcal{D}_{\mathbb{G}_2}), \dots,$
- 12 $\mathbb{D}(\mathcal{D}_{\mathbb{G}_{T-1}}, \mathcal{D}_{\mathbb{G}_T})\}$

Computational Complexity for TAD procedure

Algorithm 1 details the computation of PDs based on finding and merging clique communities. The available clique implementation of [60] (which we utilize here) is highly efficient and has a complexity of $\mathcal{O}(\kappa\varphi^{-1}(\kappa))$ for persistent homology computation, where κ is the number of edges and $\varphi^{-1}(\cdot)$ is the extremely slow-growing inverse of the Ackermann function [35]. It works jointly with one of the many clique percolation algorithms [32]. To compute the Wasserstein distance metric, we use the efficient polynomial time algorithm [54].

4.1 Theoretical Properties of the Stacked Persistence Diagram

As shown by [16], the conventional PD of an object (i.e. a single layer graph or point cloud) is stable under minor data perturbations. Noting that SPD is derived from the direct sum of the persistence modules corresponding to each layer in \mathbb{G} and using the Isometry theorem for individual persistence modules [17], we derive similar theoretical guarantees for SPD.

Theorem 4 (Stability of SPD). *Let $\mathbb{G}_X = \{\mathcal{G}_X^1, \dots, \mathcal{G}_X^L\}$ and $\mathbb{G}_Y = \{\mathcal{G}_Y^1, \dots, \mathcal{G}_Y^L\}$ be two multilayer networks generated from the same space of L -multilayer networks. Then*

$$W_\infty(D_{\mathbb{G}_X}, D_{\mathbb{G}_Y}) \leq \max_{1 \leq l \leq L} (d_{GH}(\{\mathcal{G}_X^l, \omega_{\mathcal{G}_X^l}\}, \{\mathcal{G}_Y^l, \omega_{\mathcal{G}_Y^l}\}))$$

where W_∞ is the Bottleneck distance and d_{GH} is the Gromov-Hausdorff distance.

Proof for Theorem 4 is in the Appendix. Theorem 4 implies that the proposed new SPD $D_{\mathbb{G}}$ (see Definition 3) for any multilayer network \mathbb{G} is robust with respect to W_∞ under minor input data perturbations. As a result, Theorem 4 provides theoretical foundations to our TAD idea. Hence, under the null hypothesis of no anomaly, we expect to observe similar SPDs over dynamic multilayer networks $D_{\mathbb{G}_t}$, while a noticeable difference between two adjacent SPDs is likely to be a sign of anomaly. Note that stability of SPD in terms of W_1 requires vectorization of SPD and Lipschitz continuity of the associated vectorization. While such vectorization approaches are highly successful for image and graph learning (see, e.g., [138, 76]), our preliminary studies show lack of sensitivity of such vectorization techniques in conjunction with network anomaly detection.

5 Experiments on Blockchain Networks

5.1 Experimental Setup

Baseline Algorithms: We compare performance of TAD method against the following strong state-of-the-art (SOTA) algorithms for anomaly detection on single layer networks: ① DeltaCon by [41] (which we label DC) for weighted/unweighted networks, ② Scan Statistics algorithm by [19] (which we label gSeg) for unweighted networks, ③ Edge monitoring method with Euclidean distance by

[66] (which we label EMEu) for weighted networks, and ④ Edge monitoring method with Kullback-Leibler divergence by [66] (which we label EMKL) for weighted networks. Finally, we also considered an embedding-based algorithm for anomaly detection. That is, we tracked Frobenius norms among embeddings of multilayer blockchain graphs at each time snapshot, delivered by the one of the most widely used algorithms for multilayer graph embedding, MANE of [45]. This ⑤-th approach is denoted by Graph-Em. We provide a brief description of the mechanism for each method in the Appendix. For all competing methods, we use the default parameters reported in the literature. Wherever applicable, we set a standard level of significance α of 0.05.

Since all competing methods are designed for single layer networks, we implement them (individually) w.r.t. each l layer in all the multilayer graphs $\{\mathcal{G}_t\}_{t=1}^T$ and then combine the detected results, while correcting for the multiple hypothesis testing framework. In the Appendix, we provide two types of multiple hypotheses that specifies how we retain anomalies for the sequence of multilayer graphs and these include: ① keep all anomalies identified from at least one $\{\mathcal{G}_t^l\}_{t=1}^T$, ② keep all anomalies that are commonly identified from all $\{\mathcal{G}_t^l\}_{t=1}^T$. We provide results for choice (1), and defer the results for (2) to the Appendix. Additionally, we construct a single layer version of TAD (which we call S-TAD) and apply this to the same single layers. To be precise, our improvised S-TAD will extract PDs from each l -layer, and without creating SPDs, apply the chosen distance metric to consecutive PDs to obtain a time series of topological summaries for the sequence $\{\mathcal{G}_t^l\}_{t=1}^T$. Therefore, our evaluation will investigate the performance of TAD method against the performance of the chosen techniques (DC, gSeg EMEu, EMKL, Graph-Em), and S-TAD when the (un)weighted multilayer networks are viewed as a multiple hypothesis.

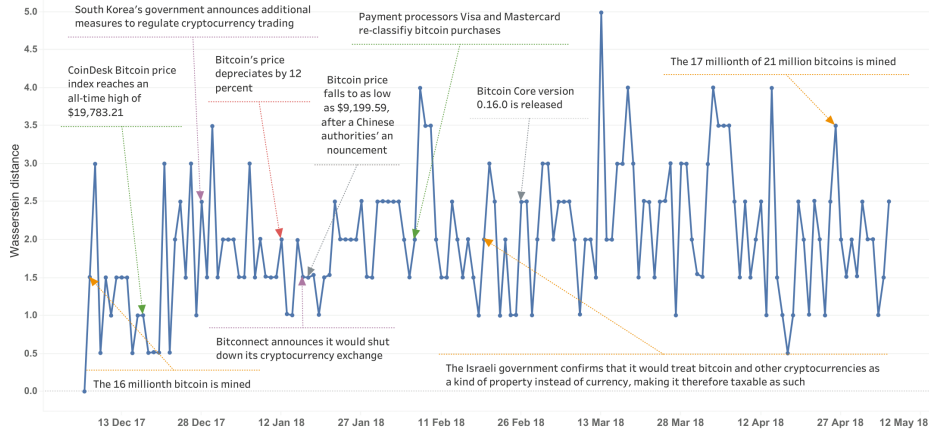


Fig. 2: Anomalous events detected by TAD for the multilayer Ethereum network.

Topological Distances in TAD We have experimented with various topological metrics, particularly, W_∞ bottleneck and W_1 Wasserstein distances. While our preliminary results do not indicate that W_1 yields substantial gains over W_∞ (i.e. 70% of the true anomalous events are detected regardless of the distance choice), W_1 tends to be slightly more sensitive than W_∞ . As such, we proceed with W_1 as the primary choice and consider $W_1(D_{G_{t-1}}, D_{G_t})$, between consecutive SPDs $D_{G_{t-1}}$ and D_{G_t} for $2 \leq t \leq T$. We apply the TAD technique to two input data types: weighted and unweighted multilayer networks. Edge weight is defined as a number of transactions between nodes.

Reproducibility and replicability The anonymized codes and data sets for this project are available at <https://github.com/tdagraphs>.

5.2 Ethereum Token Networks

Data set: The Ethereum blockchain was created in 2015 to implement Smart Contracts, which are Turing complete software codes that execute user defined tasks. Among many possible tasks, contracts are used to create and sell digital assets on the blockchain. The assets can be categorized into two categories: ① Tokens whose prices can fluctuate; ERC20 or ERC721 [65], ② Stablecoins whose prices are pegged to an asset such as USD [49] (these are also ERC20 tokens). Token networks are particularly valuable because each token naturally represents a network layer with the same nodes (addresses of investors) appearing in the networks (layers) of multiple tokens. For our experiments, we extract token networks from the publicly available Ethereum blockchain, and use the normalized number of transactions between nodes as the edge weights. By principle, a token network is a directed, weighted multigraph where an edge denotes the transferred token value. Although address creation is cheap and easy, most blockchain users use the same address over a long period. Furthermore, the same address may trade multiple tokens. As a result, the address appears in networks of all the tokens it has traded. From our data set timeline, we only include tokens reported by the [Etherscan.io](https://etherscan.io) online explorer to have more than \$100M in market value. Eventually, the data set contains 6 tokens, and on average, each token has a history of 297 days (minimum and maximum of 151 and 576 days, respectively). Note that each token has a different creation date, hence token networks have non-identical lifetime intervals.

Ground Truth: As ground truth, we adopt and curate Blockchain events from Wikipedia [69], which lists and explains major events since 2008. In total, there are 72 events that have shaped blockchain networks — some of them in adverse (see the supplementary material for the complete list). However, token networks cannot detect events before 2015 because Ethereum and its tokens did exist before then. Hence, our experiments focused on at most 32 (out of the 72) the token transaction events.

Results: Table 2 presents summary statistics for the weighted token multilayer network analysis against the three single-layer SOTA solutions (i.e. DC, EMEu, EMKL) and our topological S-TAD method. We find that TAD delivers lower FP values. In addition, we notice that TAD achieves a significantly higher accuracy

Table 2: Anomaly detection performance for the weighted Ethereum blockchain and Ripple currency networks.

	Ethereum					Ripple				
	S-TAD	DC	EMEu	EMKL	TAD	S-TAD	DC	EMEu	EMKL	TAD
TP	15	52	3	5	10	95	105	10	10	16
FP	28	69	3	5	2	260	283	40	32	9
TN	99	30	132	130	135	872	837	1152	1161	1187
FN	10	1	14	12	5	35	37	60	59	50
Acc.	0.750	0.539	0.888	0.888	0.954	0.766	0.746	0.921	0.928	0.953

(> 7% of what EMEu/EMKL report). This is evidenced by the detected points in Figure 2. From Figure 3 we notice again that the TAD yields substantially higher Precision (66% more than what DC gets) and F1 (> 23% of what DC gets) values, implying that TAD tends to be substantially more efficient in locating relevant anomalies within the multilayer graph sequence than its competitors. In addition, we find that the performance results of the Graph-Em method in Table 3 and Figure 4 are substantially worse than the ones delivered by our proposed TAD. This phenomenon can be explained by higher data aggregation typically performed by graph embedding tools which results in lower sensitivity to anomalous changes in the graph structure. Altogether, these results indicate that TAD tends to be the most preferred tool for identifying anomalies in the multilayer network setting. Table 3 presents experimental results for the anomalous event detection in the unweighted multilayer Ethereum blockchain networks. We find that TAD delivers the highest detection accuracy (0.954, which is about 20% greater than what gSeg yields). In addition, we notice that TAD attains the lowest FP value (about 10% of what gSeg obtains) and the highest TN value (about 27% more than what gSeg gets). In turn, Figure 4 suggests that TAD yields the highest precision (93% greater than DC) and the highest F1 score (23% more than DC). These findings suggest that the new TAD method tends to be the most accurate approach for flagging relevant anomalous events.

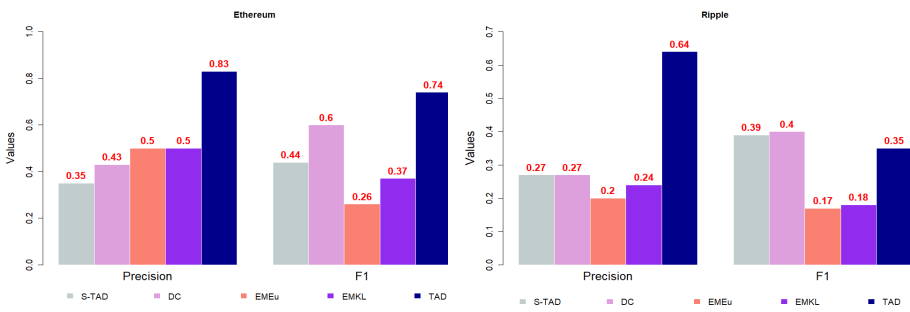


Fig. 3: Precision and F1 scores for the weighted Ethereum blockchain and Ripple currency networks.

Table 3: Anomaly detection performance for the unweighted Ethereum blockchain networks.

	Ethereum					Ripple				
	S-TAD	DC	gSeg	Graph-Em	TAD	S-TAD	DC	gSeg	Graph-Em	TAD
TP	17	52	14	3	10	80	105	17	0	11
FP	28	69	21	11	2	241	283	56	1	23
TN	97	30	106	126	135	900	837	1130	1195	1173
FN	10	1	11	12	5	41	37	59	66	55
Acc.	0.750	0.539	0.789	0.849	0.954	0.777	0.746	0.909	0.947	0.938

5.3 Ripple Currency Networks

Data set: The Ripple Credit Network was created to facilitate remittance across countries, but the network has transitioned to a blockchain-like structure where network approved entities (e.g., banks) issue currencies in I-Owe-You notes, and addresses can trade these currencies in blocks (which are called ledgers). On the Ripple network any real life asset, such as Chinese Renminbi or US Dollar, can be issued by certain participants only but traded by all addresses (nodes). In terms of regulatory issues by governments and price movements, Ripple is a part of the Blockchain ecology and the networks are impacted by the global events such as government regulations and trade volume increases [50]. We use the official Data API (<https://xrpl.org/data-api.html>) and extract the five most issued fiat currencies on the Ripple network: JPY, USD, EUR, CCK, CNY. We construct a multilayer network from the payment transactions of the five currencies that covers a timeline of Oct-2016 to Mar-2020. Similar to the Ethereum token analysis, we use the normalized number of transactions between nodes as the edge weights.

Ground Truth: As ground-truth, we use the same events described in the Ethereum token network experiments. However, since the Ripple data set has a longer temporal span of observations than the Ethereum token networks, there are a total of 66 Blockchain events.

Results: Summaries from Table 2 indicate that TAD attains the highest event detection accuracy (0.953). Furthermore, we find that TAD yields the lowest FP value, which is actually 22.5% of the value by EMEu and about 28% of the value by EMKL. Figure 3 displays detection results for the anomalous events in the multilayer Ripple payment networks. We find that TAD yields the highest precision (more than double what DC/S-TAD get) and is close to the top F1 score. Differing from Ethereum experiment, the best F1 performance is delivered by DC, closely followed by S-TAD and then TAD. Table 3 suggests that TAD delivers the highest detection accuracy (0.938, which is about 3% greater than what gSeg yields) for the unweighted Ripple currency network. In addition, we notice that TAD attains the lowest FP value (about 41% of what gSeg obtains) and the highest TN value (about 3% more than what gSeg gets). In turn, Figure 4 shows that TAD yields the highest precision (about 18% greater than DC) but the lowest F1 score (55% of what DC gets).

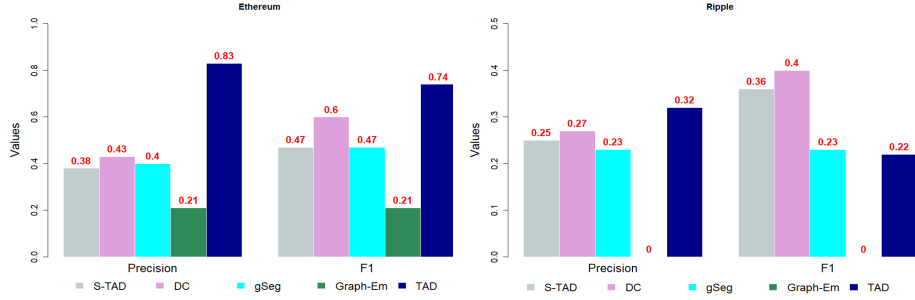


Fig. 4: Precision and F1 scores for the unweighted Ethereum blockchain and Ripple currency networks.

Finally, note that in Ethereum we use 6 tokens, whereas Ripple experiments are performed on 5 currencies. As Tables 3 and 2 suggest, the Ethereum results appear to be better than those of Ripple. That is, detection accuracy substantially improves with a higher number of layers. However, for both cases TAD either outperforms or on par with baseline techniques. The key intuition behind these results is that TAD allows for simultaneous evaluation of subtle changes in multiple homological features both within network layers and across network layers in sparse dynamic environments of blockchain transaction graphs. As such, SPD appears to be more sensitive to subtle changes in the multilayer network structure than competing non-TDA tools.

6 Conclusion

We have proposed the first topological anomaly detection (TAD) framework for dynamic multilayer networks. We have derived stability guarantees of the new topological summary for multilayer graphs, i.e., stacked persistence diagram, which is the key tool behind TAD and validated utility of TAD on two blockchain transaction graphs. Our studies have indicated that TAD yields a highly competitive performance in detecting anomalous events on Ethereum and Ripple blockchains. In the future we plan to advance TAD to anomaly detection in attributed dynamic networks and analysis of evolving communities.

Appendix

Theoretical proof of Theorem 1

To prove Theorem 4.3, we invoke the following arguments. First, we note a relationship between conventional PD of a single layer network and its associated persistence module. Then the results on isometry of PDs and persistent modules allow us to shift our focus from stability of PDs to stability of the associated persistent modules. Second, as no two layers in the multilayer network \mathcal{G} share edges, the persistence module associated with SPD is then a direct sum of persistence modules associated with PDs of single layer networks. Finally, armed with the results of [15,17] on stability of persistence modules and accounting for the direct sum representation of SPD, we arrive to stability of SPD.

Now we state the key mathematical definitions and results which are required for our derivations.

Theorem 5. (Stability of PD, Theorem 3.1 in [15]) For any finite metric spaces $\{X, w_X\}$ and $\{Y, w_Y\}$,

$$W_\infty(D_X, D_Y) \leq d_{GH}(\{X, w_X\}, \{Y, w_Y\}),$$

where d_{GH} is the Gromov-Hausdorff distance [15].

Note that weight functions ω_X and ω_Y correspond to pairwise node distances within networks X and Y respectively.

Definition 6 (Persistence module). A persistence module $\mathbb{F} = \{F_a, f_a^b\}$ is an indexed family of vector spaces ($F_a \mid a \in \mathbb{R}$) together with the double-indexed family of linear maps ($f_a^b : F_a \rightarrow F_b \mid a \leq b$) which satisfy the composition law $f_a^c = f_a^b \circ f_b^c$ whenever $a \leq b \leq c$ and where f_a^a is the identity map of F_a [17].

A persistence module $\mathbb{F} = \{F_a, f_a^b\}$ is **q-tame** if $\text{rank}(f_a^b) < \infty$ for any $a < a + q < b$ [16].

Definition 7 (Interleaving distance). For $\delta \geq 0$, the two modules $\mathbb{F} = \{F_a, f_a^b\}$ and $\mathbb{M} = \{M_a, m_a^b\}$ are δ -**interleaved** if for all $a \in \mathbb{R}$ there exists the collection of linear maps $\Phi : F_a \rightarrow M_{a+\delta}$ and $\Theta : M_a \rightarrow F_{a+\delta}$ such that all diagrams that can be composed out of the maps Φ_a, Θ_a , and the linear maps of \mathbb{F} and \mathbb{M} commute [14,17,10].

Hence, the interleaving distance d_i between \mathbb{F} and \mathbb{M} is defined as

$$d_i(\mathbb{F}, \mathbb{M}) = \inf(\delta \mid \mathbb{F} \text{ and } \mathbb{M} \text{ are } \delta\text{-interleaved}),$$

where \inf represents the infimum.

Theorem 8. (Isometry Theorem, Theorem 4.11 in [17]) Let \mathbb{F} and \mathbb{G} be q -tame persistence modules. Then

$$d_i(\mathbb{F}, \mathbb{G}) = W_\infty(D_{\mathbb{F}}, D_{\mathbb{G}}).$$

Proposition 1 (Proposition 4.5 in [17]) Let $\mathbb{U}_1, \mathbb{U}_2, \mathbb{V}_1$ and \mathbb{V}_2 be persistence modules. Then

$$d_i(\mathbb{U}_1 \oplus \mathbb{U}_2, \mathbb{V}_1 \oplus \mathbb{V}_2) \leq \max\left(d_i(\mathbb{U}_1, \mathbb{U}_2), d_i(\mathbb{V}_1, \mathbb{V}_2)\right).$$

Generally, let $(\mathbb{U}_s \mid s \in S)$ and $(\mathbb{V}_s \mid s \in S)$ be families of persistence modules indexed by the same set S , and let

$$\mathbb{U} = \bigoplus_{s \in S} \mathbb{U}_s \quad \text{and} \quad \mathbb{V} = \bigoplus_{s \in S} \mathbb{V}_s.$$

Then

$$d_i(\mathbb{U}, \mathbb{V}) \leq \sup_{s \in S} \left(d_i(\mathbb{U}_s, \mathbb{V}_s) \mid s \in S \right),$$

where \oplus denotes a direct sum.

Proof of Theorem 1 Let $\mathcal{G}_X = (V_X, \omega_X)$ and $\mathcal{G}_Y = (V_Y, \omega_Y)$ be two single layer networks equipped with clique filtrations \mathcal{F}_X and \mathcal{F}_Y respectively.

Armed with \mathcal{F}_X and \mathcal{F}_Y , we generate the family of persistence modules \mathbb{F}_X and \mathbb{F}_Y , respectively [16,17]. As discussed by [12,23], \mathbb{F}_X and \mathbb{F}_Y induce the corresponding PDs $D_{\mathcal{G}_X}$ and $D_{\mathcal{G}_Y}$. Hence, the topological persistence patterns for \mathcal{G}_X and \mathcal{G}_Y are

$$\mathcal{G}_X \rightarrow \mathcal{F}_X \rightarrow \mathbb{F}_X \rightarrow D_{\mathcal{G}_X} \quad \text{and} \quad \mathcal{G}_Y \rightarrow \mathcal{F}_Y \rightarrow \mathbb{F}_Y \rightarrow D_{\mathcal{G}_Y}.$$

By assumptions, \mathbb{F}_X and \mathbb{F}_Y are q -tame. Hence, in view of the Isometry theorem (Theorem 4.11 of [17]), we can invoke interleaving distance d_i to assess relationship between d_i and W_∞ . Under a Vietoris-Rips filtration (of which clique filtration is a subcase), the Stability of PDs theorem (Theorem 3.1 of [15]) implies that

$$W_\infty(D_{\mathcal{G}_X}, D_{\mathcal{G}_Y}) \leq d_{GH}\left(\{\mathcal{G}_X, \omega_{\mathcal{G}_X}\}, \{\mathcal{G}_Y, \omega_{\mathcal{G}_Y}\}\right), \quad (1)$$

where d_{GH} is the Gromov-Hausdorff distance. By $(\mathcal{G}_X, \omega_{\mathcal{G}_X})$ we mean the induced finite metric space (V_X, ω_X) where the distances between the finite vertex set is induced by the edge weight function ω_X . Similar reference is implied for $(\mathcal{G}_Y, \omega_{\mathcal{G}_Y})$. In turn, the Isometry theorem implies

$$d_i(\mathbb{F}_X, \mathbb{F}_Y) = W_\infty(D_{\mathcal{G}_X}, D_{\mathcal{G}_Y}).$$

Hence, in view of [1], we obtain

$$d_i(\mathbb{F}_X, \mathbb{F}_Y) \leq d_{GH}\left(\{\mathcal{G}_X, \omega_{\mathcal{G}_X}\}, \{\mathcal{G}_Y, \omega_{\mathcal{G}_Y}\}\right). \quad (2)$$

Now consider two multilayer networks $\mathbb{G}_X = \{\mathcal{G}_X^1, \dots, \mathcal{G}_X^L\}$ and $\mathbb{G}_Y = \{\mathcal{G}_Y^1, \dots, \mathcal{G}_Y^L\}$, with $\omega_{\mathbb{G}_X} = \{\omega_{\mathcal{G}_X^1}, \dots, \omega_{\mathcal{G}_X^L}\}$ and $\omega_{\mathbb{G}_Y} = \{\omega_{\mathcal{G}_Y^1}, \dots, \omega_{\mathcal{G}_Y^L}\}$ as the weight functions. Clearly, $\{\mathbb{G}_X, \omega_{\mathbb{G}_X}\}$ and $\{\mathbb{G}_Y, \omega_{\mathbb{G}_Y}\}$ are metric spaces. From the definition of SPD (Definition 3 under ‘‘Persistence Methodology for Network Anomaly Detection’’), \mathbb{G}_X and \mathbb{G}_Y induce SPDs

$$D_{\mathbb{G}_X} = (D_{\mathcal{G}_X^1}, \dots, D_{\mathcal{G}_X^L}) \quad \text{and} \quad D_{\mathbb{G}_Y} = (D_{\mathcal{G}_Y^1}, \dots, D_{\mathcal{G}_Y^L})$$

respectively, and based on [\(1\)](#) and [\(2\)](#) above, for $l \in \{1, \dots, L\}$

$$W_\infty(D_{\mathcal{G}_X^l}, D_{\mathcal{G}_Y^l}) \leq d_{GH}(\{\mathcal{G}_X^l, \omega_{\mathcal{G}_X^l}\}, \{\mathcal{G}_Y^l, \omega_{\mathcal{G}_Y^l}\}),$$

which implies that

$$d_i(\mathbb{F}_{X^l}, \mathbb{F}_{Y^l}) \leq d_{GH}(\{\mathcal{G}_X^l, \omega_{\mathcal{G}_X^l}\}, \{\mathcal{G}_Y^l, \omega_{\mathcal{G}_Y^l}\}). \quad (3)$$

By SPD construction, no edges are shared between any \mathcal{G}_X^i and \mathcal{G}_X^j , $i \neq j$. Hence, the persistence module of \mathbb{G}_X corresponding to $D_{\mathbb{G}_X}$ is the direct sum of all \mathbb{F}_{X^l} :

$$\mathbb{F}_{\mathbb{G}_X} = \bigoplus_{1 \leq l \leq L} \mathbb{F}_{X^l}. \quad (4)$$

According to Proposition 1 above (i.e. Proposition 4.5 in [\[17\]](#)),

$$d_i(\mathbb{F}_{\mathbb{G}_X}, \mathbb{F}_{\mathbb{G}_Y}) \leq \max_{1 \leq l \leq L} (d_i(\mathbb{F}_{X^l}, \mathbb{F}_{Y^l})), \quad (5)$$

and by assumptions, each \mathbb{F}_{X^i} and \mathbb{F}_{Y^j} is q -tame, $i \neq j$. Combining [\(3\)](#) and [\(5\)](#) results in

$$d_i(\mathbb{F}_{\mathbb{G}_X}, \mathbb{F}_{\mathbb{G}_Y}) \leq \max_{1 \leq l \leq L} (d_{GH}(\{\mathcal{G}_X^l, \omega_{\mathcal{G}_X^l}\}, \{\mathcal{G}_Y^l, \omega_{\mathcal{G}_Y^l}\})).$$

Finally, by invoking the Isometry theorem and accounting for the direct sum representation [\(4\)](#), we obtain

$$W_\infty(D_{\mathbb{G}_X}, D_{\mathbb{G}_Y}) \leq \max_{1 \leq l \leq L} (d_{GH}(\{\mathcal{G}_X^l, \omega_{\mathcal{G}_X^l}\}, \{\mathcal{G}_Y^l, \omega_{\mathcal{G}_Y^l}\})),$$

which concludes the proof.

Experimental Set-up: Computational complexity and sampling of cliques

The process of tracking clique communities for high values of k is computationally prohibitive, mainly because the clique decision problem is NP-complete and common algorithms require that all the maximal cliques be found first [\[39, 58\]](#). The aforementioned computational complexity makes it unlikely to apply clique-based methods on the entirety of our large-scale and dense blockchain networks

(the Ethereum and Ripple networks have an average of 442788/1192722 and 71337/922084 nodes/edges, respectively).

To maintain reasonable computation time, we take a subgraph of each network by adopting the maximum weight subgraph approximation method of [64], which restricts the subgraph size to the nodes of the p most active edges. Sampling helps us with efficient computation in certain token networks with large sizes. At the same time, sampling nodes has minimal impact on results since most daily token networks have <100 nodes. For instance, even for the most traded tokens such as Tronix and Bat, top 150 nodes in daily networks form 75% and 80% of all edges, respectively. The filtered node approach effectively removes 20 – 25% of edges in calculations, which reduces computation costs. Hence, with the Ethereum blockchain and Ripple currency networks we only work with 100 and 90 nodes respectively. To mitigate computation efficiency constraints, we set the maximum clique size to $k_{max} = 20$ for the Ethereum network and $k_{max} = 4$ for Ripple data set, and compute clique community PDs from $k = 1$ to $k = k_{max}$ for each network. In addition, an event at time t in the ground truth is marked as detected when a predicted anomaly occurs in $t - 2 \leq t \leq t + 2$; note that trading patterns are variable and may not change the same day as published events in news.

Description of State-Of-The-Art methods

We select the following state-of-the-art methods for network anomaly detection based on their superior performance proven in most recent experimental studies and their theoretical guarantees [71,647]. The DeltaCon (DC) of [41] measures the similarity between edgelist pairs, and classifies the events with similarity scores below a specified threshold as anomalous. The two edge monitoring algorithms (EMEu and EMKL) of [66] account for temporal dependencies by tracking the evolution of the network as a Markov process, thereby focusing on comparison of the probability distribution of edges. The MANE by [45] calculates the relative difference between pre and post multilayer graph embeddings (i.e., relative difference with Frobenius norm). In turn, gSeg of [19] is a nonparametric graph-based method that uses two-sample tests based on scan statistics. Among these benchmark methods, the DC and the gSeg are suited for unweighted graph data, while the DC, EMEu and EMKL are applicable for weighted graphs. Since blockchain edges carry transaction amounts, it is natural to use weighted graph based anomaly detection approaches.

Structure of Multiple Hypotheses Testing

Let $\mathcal{A}^l = \{t_1^l, t_2^l, \dots, t_{\tau_l}^l\}$, $1 \leq \tau_l \leq T$, be a set of anomalous events identified in the T sequence of l -single layer graphs $\mathcal{G}^l \in \mathbb{G}$, $1 \leq l \leq L$. Then we can test for

$$\begin{aligned} H_0 : \mathcal{A}^1 = \dots = \mathcal{A}^L = \emptyset \quad \text{vs} \\ H_a : \exists l \text{ such that } \mathcal{A}^l \neq \emptyset, \quad l \in \{1, \dots, L\}. \end{aligned} \quad (6)$$

The general practice for all statistical hypotheses testing is to control the rate of false positives at a probability known as the level of significance (α). However, the multiple testing approach is significantly prone to varying kinds of false positives rates [44], and the usual kind is the familywise error rate ($FWER$) [26], which is defined as:

$$FWER = P(V \geq 1) = 1 - (1 - \alpha_c)^L,$$

where V is the number of false positives in all L hypotheses, P is probability, L is the number of hypotheses, and α_c is the level of significance that determines the rejection of single hypothesis from the multiple set [26, 59].

We employ a stronger control over $FWER$ by testing H_0 in (6) against an alternative

$$\begin{aligned} H_0 : \mathcal{A}^1 = \dots = \mathcal{A}^L = \emptyset \quad \text{vs} \\ H_a : \exists t \in \mathcal{A}^1 \cap \dots \cap \mathcal{A}^L, \quad 1 \leq t \leq T. \end{aligned} \quad (7)$$

To ensure reliability of conclusions in the multiple testing framework, it is essential to control $FWER$ and to require $FWER \leq \alpha$ [26]. One of the most widely used control procedures is the Bonferroni correction method [27]. Under the Bonferroni correction, we reject the single null hypothesis $H_0 : \mathcal{A}^i = \emptyset$ if and only if $FWER \leq \alpha_c$ where $\alpha_c = (\alpha/L)$.

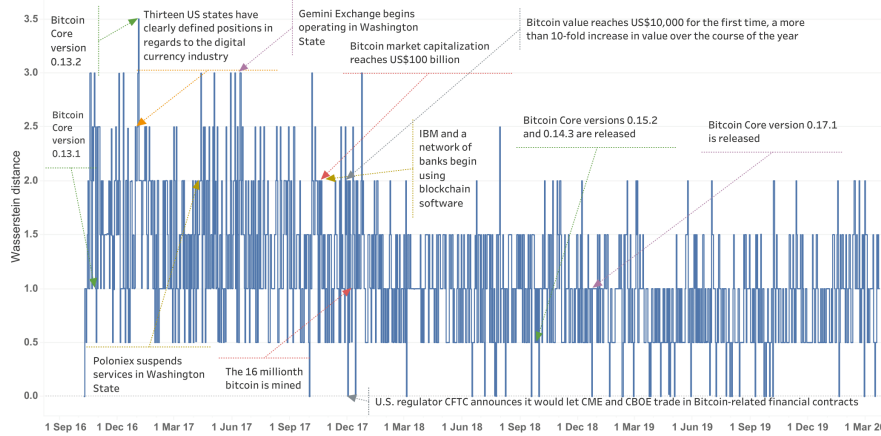


Fig. 5: Anomalous events detected by TAD for the multilayer Ripple currency network.

Results for the stronger controlled $FWER$

At this stage, we test the following multiple hypotheses for the 4 single layer SOTA methods (DC, gSeg, EMEu, EMKL) and our improvised S-TAD method

against the TAD:

$$\begin{aligned} H_0 : \mathcal{A}^1 = \dots = \mathcal{A}^L = \emptyset \quad \text{vs} \\ H_a : \exists t \in \mathcal{A}^1 \cap \dots \cap \mathcal{A}^L, \quad 1 \leq t \leq T. \end{aligned} \quad (8)$$

The performance of all methods (S-TAD, DC, gSeg, EMEu, EMKL, TAD) is summarized as true positive (TP), false positive (FP), true negative (TN), false negative (FN), accuracy (Acc.), precision (Prec.) and F1 score values (F1).

Results (Ethereum blockchain): From Tables 4 and 5, we find that TAD delivers the highest detection accuracy (0.954, which is about 5% greater than what all the other methods deliver) for both the unweighted and weighted networks.

Table 4: Anomaly detection performance for the weighted Ethereum blockchains using strong control of FWER.

	S-TAD	DC	EMEu	EMKL	TAD
TP	0	0	0	0	10
FP	0	0	0	0	2
TN	138	138	138	138	135
FN	14	14	14	14	5
Acc.	0.908	0.908	0.908	0.908	0.954
Prec.	0.000	0.000	0.000	0.000	0.833
F1	0.000	0.000	0.000	0.000	0.741

By comparing the remaining summaries (TP, FP, TN, FN, Prec., F1) to the case we assessed earlier,

we find that with the strong *FWER* setup, there is no common anomalous event located in the sequence of six layer networks (i.e. the setup of the the six hypotheses makes it harder to locate a common event).

Table 5: Anomaly detection performance for the unweighted Ethereum blockchains using strong control of FWER

	S-TAD	DC	gSeg	TAD
TP	0	0	0	10
FP	0	0	0	2
TN	138	138	138	135
FN	14	14	14	5
Acc.	0.908	0.908	0.908	0.954
Prec.	0.000	0.000	0.000	0.833
F1	0.000	0.000	0.000	0.741

Table 6: Anomaly detection performance for the weighted Ripple currency networks using strong control of FWER.

	S-TAD	DC	EMEu	EMKL	TAD
TP	0	0	0	0	16
FP	0	0	0	0	9
TN	1198	1198	1198	1198	1187
FN	64	64	64	64	50
Acc.	0.949	0.949	0.949	0.949	0.953
Prec.	0.000	0.000	0.000	0.000	0.640
F1	0.000	0.000	0.000	0.000	0.352

Results (Ripple currency): Looking at the summaries in Tables 6 and 7, we find that the TAD has the highest accuracy (0.953, which is closely matched by the other methods) for the weighted networks. With the unweighted networks, however, the performance of the other methods (0.949) is slightly higher than the accuracy for TAD (0.938). Similar to the Ethereum blockchain results, we notice that the ability of the competing techniques and our S-TAD are strongly disadvantaged when the multiple hypotheses is setup to strongly control *FWER* (because no unique anomalous event is identified).

Table 7: Anomaly detection performance for the unweighted Ripple currency networks using strong control of FWER

	S-TAD	DC	gSeg	TAD
TP	0	0	0	11
FP	0	0	0	23
TN	1198	1198	1198	1173
FN	64	64	64	55
Acc.	0.949	0.949	0.949	0.938
Prec.	0.000	0.000	0.000	0.324
F1	0.000	0.000	0.000	0.220

One-day Example

TAD computes PDs on geodesic densifications of blockchain graphs, tracking k -clique community persistences based pair-nodes connections. Figure 6 shows a real example on one-day graph of CCK currency, i.e. Ripple currency network. A weighted graph, with twenty connected nodes, is presented in Figure 6a. We compute the geodesic distance between each pair of nodes, i.e. the shortest path length between each node-pair, see Figure 6b. Finally, significant topological summaries are found through the clique community persistence procedure, see Figure 6c, using community sizes from $k = 1$ to $k = 4$. Hence, our methodology

preserves not only direct k -cliques communities present in the network, but also communities from indirect connections.

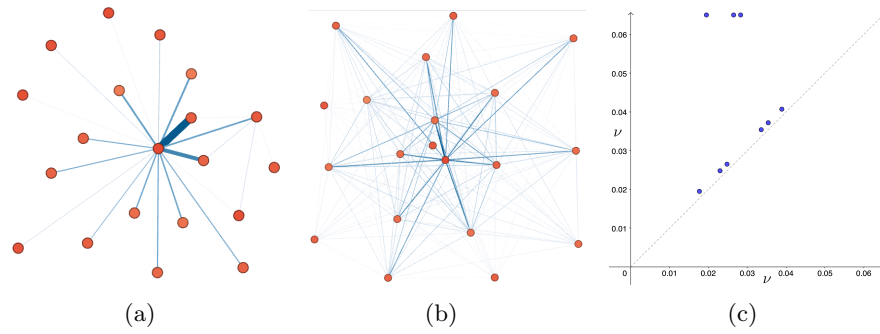


Fig. 6: Processing one-day unilayer graph from Ripple network. (a) One-day graph of CCK currency, twenty connected nodes. (b) Corresponding Geodesic densification. (c) Clique Community Persistence Diagram.

Acknowledgements

This work is supported in part by NSF Grants No. ECCS 2039701, DMS 1925346, CNS 1837627, OAC 1828467, IIS 1939728, CNS 2029661 and Canadian NSERC Discovery Grant RGPIN-2020-05665. The authors would like to thank Baris Coskunuzer for insightful discussions.

References

1. Adams, H., Emerson, T., Kirby, M., Neville, R., Peterson, C., Shipman, P., Chepushtanova, S., Hanson, E., Motta, F., Ziegelmeier, L.: Persistence images: A stable vector representation of persistent homology. *JMLR* **18**(1), 218–252 (2017)
2. Akcora, C.G., Dixon, M.F., Gel, Y.R., Kantarcioglu, M.: Bitcoin risk modeling with blockchain graphs. *Economics Letters* **173**, 138–142 (2018)
3. Akcora, C.G., Li, Y., Gel, Y.R., Kantarcioglu, M.: Bitcoinheist: Topological data analysis for ransomware detection on the bitcoin blockchain. In: *IJCAI*. pp. 1–9 (2020)
4. Aleta, A., Moreno, Y.: Multilayer networks in a nutshell. *Annu. Rev. Condens. Matter Phys.* **10**, 45–62 (2019)
5. Alves, G.A.L., Mangioni, G., Cingolani, I., Rodrigues, A.F., Panzarasa, P., Moreno, Y.: The nested structural organization of the worldwide trade multi-layer network. *Scientific reports* (2019)
6. Arlot, S., Celisse, A., Harchaoui, Z.: A kernel multiple change-point algorithm via model selection. *Journal of Machine Learning Research* **20**(162), 1–56 (2019)
7. Bansal, M., Sharma, D.: Ranking and discovering anomalous neighborhoods in attributed multiplex networks. In: *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, pp. 46–54. ACM (2020)
8. Berry, E., Chen, Y.C., Cisewski-Kehe, J., Fasy, B.T.: Functional summaries of persistence diagrams. *J. Appl. Comput. Topol.* **4**(2), 211–262 (2020)
9. Biasotti, S., Falcidieno, B., Giorgi, D., Spagnuolo, M.: *Mathematical tools for shape analysis and description*. Morgan& Claypool (2014)
10. Bjerkevik, H.B., Botnan, M.B., Kerber, M.: Computing the interleaving distance is np-hard. *Foundations of Computational Mathematics* pp. 1–35 (2019)
11. Bubenik, P.: Statistical topological data analysis using persistence landscapes. *JMLR* **16**(1), 77–102 (2015)
12. Bubenik, P., Vergili, T.: Topological spaces of persistence modules and their properties. *J. of Applied and Comp. Topology* **2**(3-4), 233–269 (2018)
13. Carlsson, G.: Topology and data. *BAMS* **46**(2), 255–308 (2009)
14. Chazal, F., Cohen-Steiner, D., Glisse, M., Guibas, L.J., Oudot, S.Y.: Proximity of persistence modules and their diagrams. In: *SOCG*. pp. 237–246 (2009)
15. Chazal, F., Cohen-Steiner, D., Guibas, L.J., Mémoli, F., Oudot, S.Y.: Gromov-hausdorff stable signatures for shapes using persistence. *Computer Graphics Forum* **28**, 1393–1403 (2009)
16. Chazal, F., Cohen-Steiner, D., Guibas, L.J., Oudot, S.: The stability of persistence diagrams revisited. Tech. rep., CRISAM - Inria Sophia Antipolis (06 2008)
17. Chazal, F., De Silva, V., Glisse, M., Oudot, S.: *The structure and stability of persistence modules*. Springer (2016)
18. Chazal, F., Michel, B.: *An introduction to topological data analysis: fundamental and practical aspects for data scientists*. arXiv:1710.04019 (2017)

19. Chen, H., Zhang, N.: Graph-based change point detection. *The Annals of Stat.* **43**(1), 139–176 (2015)
20. Chen, W., Zheng, Z., Cui, J., Ngai, E., Zheng, P., Zhou, Y.: Detecting ponzi schemes on ethereum: Towards healthier blockchain technology. In: *WWW*. pp. 1409–1418 (2018)
21. Chen, Y., Bellavitis, C.: Blockchain disruption and decentralized finance: The rise of decentralized business models. *J. Bus. Ventur. Insights* **13**, e00151 (2020)
22. de la Concha, A., Martinez-Jaramillo, S., Carmona, C.: Multiplex financial networks: Revealing the level of interconnectedness in the banking system. In: *Complex Networks & Their Applications VI*. pp. 1135–1148 (2017)
23. De Silva, V., Nanda, V.: Geometry in the space of persistence modules. In: *SOCG*. pp. 397–404 (2013)
24. Di Battista, G., Di Donato, V., Patrignani, M., Pizzonia, M., Roselli, V., Tamassia, R.: Bitcoveview: visualization of flows in the bitcoin transaction graph. In: *IEEE VizSec*. pp. 1–8 (2015)
25. Dong, H., Chen, N., Wang, K.: Modeling and change detection for count-weighted multilayer networks. *Technometrics* **62**(2), 184–195 (2020)
26. Dudoit, S., Shaffer, J.P., Boldrick, J.C.: Multiple hypothesis testing in microarray experiments. *Stat. Sci.* pp. 71–103 (2003)
27. Dudoit, S., Van Der Laan, M.J.: Multiple testing procedures with applications to genomics. Springer Science & Business Media (2007)
28. Eswaran, D., Faloutsos, C., Guha, S., Mishra, N.: Spotlight: Detecting anomalies in streaming graphs. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. pp. 1378–1386 (2018)
29. FBK: Multilayer network datasets released for reproducibility (June 2020), <https://comunelab.fbk.eu/data.php>
30. Fernandes, G., Rodrigues, J.J., Carvalho, L.F., Al-Muhtadi, J.F., Proença, M.L.: A comprehensive survey on network anomaly detection. *Telecommun Syst* **70**(3), 447–489 (2019)
31. Ferretti, S., D’Angelo, G.: On the ethereum blockchain structure: A complex networks theory perspective. *Concurrency and Computation: Practice and Experience* (2019)
32. Fortunato, S.: Community detection in graphs. *Physics Reports* **486**(3-5), 75 – 174 (2010)
33. Gidea, M., Goldsmith, D., Katz, Y.A., Roldan, P., Shmalo, Y.: Topological recognition of critical transitions in time series of cryptocurrencies. *Physica A: Stat. Mechanics and its Apps.* p. 123843 (2020)
34. Greaves, A., Au, B.: Using the bitcoin transaction graph to predict the price of bitcoin. *No Data* (2015)
35. Grossman, J.W., Zeitman, R.: An inherently iterative computation of ackermann’s function. *Theoretical Computer Science* **57**(2), 327 – 330 (1988)
36. Hajij, M., Wang, B., Scheidegger, C., Rosen, P.: Visual detection of structural changes in time-varying graphs using persistent homology. In: *2018 IEEE PacificVis*. pp. 125–134 (2018)
37. Hochenbaum, J., Vallis, O.S., Kejariwal, A.: Automatic anomaly detection in the cloud via statistical learning. *arXiv* p. 1704.07706 (2017)
38. Hofer, C.D., Kwitt, R., Niethammer, M.: Learning representations of persistence barcodes. *JMLR* **20**(126), 1–45 (2019)
39. Karp, R.M.: Reducibility among Combinatorial Problems, pp. 85–103. Springer (1972)

40. Kerber, M., Morozov, D., Nigmatov, A.: Geometry helps to compare persistence diagrams. In: ALENEX. pp. 103–112 (2016)
41. Koutra, D., Shah, N., Vogelstein, J.T., Gallagher, B., Faloutsos, C.: Deltacon: Principled massive-graph similarity function with attribution. *ACM TKDD* **10**(3), 1–43 (2016)
42. Kurbucz, M.T.: Predicting the price of bitcoin by the most frequent edges of its transaction network. *Economics Letters* **184**, 108655 (2019)
43. Lee, X.T., Khan, A., Sen Gupta, S., Ong, Y.H., Liu, X.: Measurements, analyses, and insights on the entire ethereum blockchain network. In: *WWW*. pp. 155–166 (2020)
44. Li, J.: Testing each hypothesis marginally at alpha while still controlling fwer: How and when. *Statistics in medicine* **32**(10), 1730–1738 (2013)
45. Li, J., Chen, C., Tong, H., Liu, H.: Multi-layered network embedding. In: *SIAM SDM*. pp. 684–692 (2018)
46. Li, Y., Islambekov, U., Akcora, C., Smirnova, E., Gel, Y.R., Kantarcioglu, M.: Dissecting ethereum blockchain analytics: What we learn from topology and geometry of the ethereum graph? In: *SIAM SDM*. pp. 523–531 (2020)
47. Masuda, N., Holme, P.: Detecting sequences of system states in temporal networks. *Scientific reports* **9**(1), 1–11 (2019)
48. Mittal, R., Bhatia, M.: Anomaly detection in multiplex networks. *Procedia Computer Science* **125**, 609–616 (2018)
49. Moin, A., Sirer, E.G., Sekniqi, K.: A classification framework for stablecoin designs. *arXiv:1910.10098* (2019)
50. Moreno-Sanchez, P., Modi, N., Songhela, R., Kate, A., Fahmy, S.: Mind your credit: Assessing the health of the ripple credit network. In: *WWW*. pp. 329–338 (2018)
51. Moreno-Sanchez, P., Zafar, M.B., Kate, A.: Listening to whispers of ripple: Linking wallets and deanonymizing transactions in the ripple network. *PoPETs* **2016**(4), 436–453 (2016)
52. Munch, E., Turner, K., Bendich, P., Mukherjee, S., Mattingly, J., Harer, J., et al.: Probabilistic fréchet means for time varying persistence diagrams. *Electronic J. of Stat.* **9**(1), 1173–1204 (2015)
53. Nelson, D.: Crypto criminals have already stolen \$1.4b in 2020, says ciphertace (June 2020), <https://www.coindesk.com/author/danielnelsoncoindesk-com>
54. Nocedal, J., Wright, S.J.: *Numerical Optimization*. Springer (2006)
55. Phetsouvanh, S., Oggier, F., Datta, A.: Egret: extortion graph exploration techniques in the bitcoin network. In: *IEEE ICDMW*. pp. 244–251 (2018)
56. Pourhabibi, T., Ong, K.L., Kam, B.H., Boo, Y.L.: Fraud detection: A systematic literature review of graph-based anomaly detection approaches. *Decision Support Systems* p. 113303 (2020)
57. Ranshous, S., Shen, S., Koutra, D., Harenberg, S., Faloutsos, C., Samatova, N.F.: Anomaly detection in dynamic networks: a survey. *Wiley Interdisciplinary Reviews: Computational Statistics* **7**(3), 223–247 (2015)
58. Reid, F., McDaid, A., Hurley, N.: Percolation computation in complex networks. In: *ASONAM*. p. 274–281 (2012)
59. Rempala, G.A., Yang, Y.: On permutation procedures for strong control in multiple testing with gene expression data. *Statistics and its interface* **6**(1) (2013)
60. Rieck, B., Fugacci, U., Lukasczyk, J., Leitte, H.: Clique community persistence: A topological visual analysis approach for complex networks. *IEEE Trans. on Visualization and Computer Graphics* **24**(1), 822–831 (2017)

61. Suárez, G., Gallos, L., Fefferman, N.: A case study in tailoring a bio-inspired cyber-security algorithm: designing anomaly detection for multilayer networks. In: SPW. pp. 281–286 (2018)
62. Taylor, D., Porter, M.A., Mucha, P.J.: Tunable eigenvector-based centralities for multiplex and temporal networks. arXiv:1904.02059 (2019)
63. Vallis, O., Hochenbaum, J., Kejariwal, A.: A novel technique for long-term anomaly detection in the cloud. In: USENIX HotCloud (2014)
64. Vassilevska, V., Williams, R., Yuster, R.: Finding the smallest h-subgraph in real weighted graphs and related problems. In: Automata, Languages and Programming. pp. 262–273. Springer Berlin Heidelberg (2006)
65. Victor, F., Lüders, B.K.: Measuring ethereum-based erc20 token networks. In: IFCA. pp. 113–129. Springer (2019)
66. Wang, Y., Chakrabarti, A., Sivakoff, D., Parthasarathy, S.: Fast change point detection on dynamic social networks. arXiv:1705.07325 (2017)
67. Wasserman, L.: Topological data analysis. *Annu Rev Stat Appl* **5**, 501–532 (2018)
68. Weber, M., Domeniconi, G., Chen, J., Weidele, D.K.I., Bellei, C., Robinson, T., Leiserson, C.E.: Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. preprint arXiv:1908.02591 (2019)
69. Wikipedia contribs: History of bitcoin (November 2013), https://en.wikipedia.org/wiki/History_of_bitcoin. [Accessed 01-March-2020]
70. Xu, J., Livshits, B.: The anatomy of a cryptocurrency pump-and-dump scheme. In: USENIX Security. pp. 1609–1625 (2019)
71. Yoon, M., Hooi, B., Shin, K., Faloutsos, C.: Fast and accurate anomaly detection in dynamic graphs with a two-pronged approach. In: ACM SIGKDD. pp. 647–657 (2019)
72. Yousaf, H., Kappos, G., Meiklejohn, S.: Tracing transactions across cryptocurrency ledgers. In: USENIX Security. pp. 837–850 (2019)
73. Yu, W., Cheng, W., Aggarwal, C.C., Zhang, K., Chen, H., Wang, W.: Netwalk: A flexible deep embedding approach for anomaly detection in dynamic networks. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 2672–2681 (2018)
74. Yuvaraj, M., Dey, A.K., Lyubchich, V., Gel, Y.R., Poor, H.V.: Topological clustering of multilayer networks. *Proceedings of the National Academy of Sciences* **118**(21) (2021)
75. Zhao, D.: An algebraic-topological approach to processing cross-blockchain transactions. arXiv preprint arXiv:2008.08208 (2020)
76. Zhao, Q., Wang, Y.: Learning metrics for persistence-based summaries and applications for graph classification. In: NeurIPS. pp. 9855–9866 (2019)
77. Zhu, T., Li, P., Chen, K., Chen, Y., Yu, L.: Hyper-network based change point detection in dynamic networks. *Semantic Scholar* (2018)
78. Zomorodian, A.: Fast construction of the vietoris-rips complex. *Computers & Graphics* **34**(3), 263–271 (2010)
79. Zomorodian, A.: The tidy set: a minimal simplicial set for computing homology of clique complexes. In: SOCG. pp. 257–266 (2010)