Topological recognition of critical transitions in time series of cryptocurrencies

Marian Gidea

Yeshiva University, Department of Mathematical Sciences, New York, NY 10016, USA

Yuri Katz

S&P Global Market Intelligence, 55 Water Str., New York, NY 10040, USA

Pablo Roldan

Yeshiva University, Department of Mathematical Sciences, New York, NY 10033, USA

Daniel Goldsmith

Yeshiva University, Department of Mathematical Sciences, New York, NY 10033, USA

Yonah Shmalo

Yeshiva University, Department of Mathematical Sciences, New York, NY 10033, USA

Abstract

We analyze the time series of four major cryptocurrencies (Bitcoin, Ethereum, Litecoin, and Ripple) before the digital market crash at the end of 2017 - beginning 2018. We introduce a methodology that combines topological data analysis with a machine learning technique – k-means clustering – in order to automatically recognize the emerging chaotic regime in a complex system approaching a critical transition. We first test our methodology on the complex system dynamics of a Lorenz-type attractor, and then we apply it to the four major cryptocurrencies. We conclude that our methodology can be useful for providing early warning signals for critical transitions in cryptocurrency markets, even when the relevant time series exhibit a non-stationary, highly erratic behavior.

Email addresses: `Marian.Gidea@yu.edu` (Marian Gidea), `yuri.katz@spglobal.com` (Yuri Katz), `pablo.roldan@yu.edu` (Pablo Roldan), `daniel.goldsmith@mail.yu.edu` (Daniel Goldsmith), `yshmalo1@mail.yu.edu` (Yonah Shmalo)

Preprint submitted to Elsevier *June 26, 2018*

Keywords: Cryptocurrency, Critical Transitions, Complex Systems Dynamics, Topological Data Analysis, Financial Time Series, k-Means Clustering
PACS: 05.40.-a, 05.45. Tp

1. Introduction

Critical transitions – abrupt shifts in the state of a system that are triggered by small perturbations – have been observed in many complex systems, including climate, ecosystems, and financial markets; see, e.g., [40, 41], and the references therein. In the case of deterministic systems, critical transitions often appear in the form of catastrophic bifurcations (see, e.g., [27, 16, 31])), while in noisy systems, they are associated with drastic changes in the distribution of a system's states (see, e.g., [24]). Generally, it is quite challenging to identify an approaching critical transition from real-life data, because the state of the system may show little change, and only for a short period, before reaching a critical threshold.

Detection of early signals of an approaching critical transition is particularly difficult in financial time series, where the data is intrinsically very noisy and exhibits strong non-stationarity (see, e.g. [32, 35]). Moreover, since in the financial markets critical transitions can occur with only little warning, it is important to extract information from short-time windows with less than hundred data-points. We would like to stress here that for non-linear dynamical systems and non-stationary stochastic processes with short-time windows, the standard statistical methods may yield unreliable results. The fundamental question is how to extract useful information from such empirical data, leading to construction of an 'early warning system' for financial markets.

The geometric method of topological data analysis (TDA), which is at the core of this paper, is free from any statistical assumptions, and is able to detect critical transitions in complex systems (see, e.g., [2, 3, 22]). The input of the method is a point cloud – a snapshot image of the data – to which a geometric 'shape' is associated, and topological information on that shape is extracted. The output of the method is a *persistence diagram*, or a *persistence landscape*, which represents a summary of all topological features displayed by the data, ranked by the 'resolution level'. The method is robust, that is, small noisy perturbations added to the data results in only small changes in the persistence landscape. For deterministic systems undergoing bifurcations, the TDA is able to recognize a change in the topology of underlying attractors. This change is reflected by the evolving shape of the point-cloud data. The features of that shape are computable via persistent homology, and the changes in the shape can be measured via the norms of the persistence landscapes.

Using TDA to recognize different regimes of behavior of an underlining complex system has received a lot of attention in the last few years. Hence, there is a well-grounded theoretical foundation, as well as numerous experimental results; see, e.g. [2, 3, 37, 38, 28, 29, 42, 33, 30]. However, the application of TDA to financial markets is in the early stages of development; see [21, 22, 45].

In this paper, we apply the TDA-based method to investigate the behavior of the most capitalized digital currencies – Bitcoin, Ethereum, Litecoin, and Ripple – prior to the crash that occurred at the end of 2017 - early 2018. Our interest in cryptocurrencies is motivated by the wild statistical properties of the time series of these assets, which makes them an ideal test-case for application of TDA. The cryptocurrencies market capitalization exploded from around \$19 billion in February 2017 to around \$800 billion in December 2017, with more than 1000 cryptocurrencies currently on the market. Despite the recent downturn, which wiped off over \$550 billion of value, it is expected that market valuation will hit \$1 trillion during 2018. The trading activity of cryptocurrencies appears to be heavily influenced by popularity driven by news and social media, speculative bubbles, fraud, regulations, and policy interventions. Consequently, there is a clear evidence of strong non-stationarity in time series of log-returns of the relevant prices (exchange rates) nominated in U.S. dollars. Also, long-term memory, leverage effect, and volatility clustering have been observed in these time series; see the related works mentioned below.

The pipeline of our approach is comprised of the following steps: (i) we use the time-delay coordinate embedding to obtain a multi-dimensional representation of the time-series, (ii) we apply a sliding window to scan the obtained multi-dimensional time series, (iii) for each sliding window we associate a point cloud and apply a simplicial complex filtration, (iv) we extract topological information from this filtration in the form of a real-valued function – a persistence topological landscape, and (v) for each window we compute the norm of the persistence landscape. Our TDA-based approach shows that, prior to a critical transition, the norms of the persistence landscapes undergo significant changes, and such changes can be detected even with short sliding windows. When the value of an asset undergoes a critical transition, that is, an abrupt change from one regime to a significantly different one, the underlying price dynamics changes from a less turbulent to a more turbulent regimes. This is reflected by a significant change in the shape of the data, which can be recognized by the TDA-based method.

Finally, we compare the derived time series of the norms of persistence landscapes with the time series of the observed log-returns to identify patterns indicative of critical transitions. To capture the relation between the time series of price/return of each asset and the L^1-norms prior to the crash, we use an unsupervised machine learning technique utilizing non-parametric, geometry based k-means clustering method. The main point is that this approach requires no statistical assumptions to be satisfied beforehand, in contrast to most statistical tests. The k-means clustering applied to the data consisting of the normalized log-prices of each asset, the log-returns, and the L^1-norms of the persistence landscapes, yields an automatic identification of topologically distinct regimes emerging before the crash of each asset.

We now mention some related works. A methodology to detect bubbles in the price dynamics of cryptocurrencies (specifically, Bitcoin), using multi-scale analysis, as well as k-means clustering, is presented in [19]. In [9] the predictability of cryptocurrencies time series is investigated through several alternative univariate and multivariate models in point and density forecasting. A general equilibrium monetary model of a cryptocurrency market, which endogenizes the value of a cryptocurrency, and the underlying trading and mining activities, is developed in [13]. A dynamic model of price formation is proposed in [10]. The statistical properties of the Bitcoin time series – are studied in [6]. Statistical properties of the cryptocurrencies are explored in [11], where a wide range of parametric distributions are fitted to data. In [14], twelve GARCH models are fitted to each of the seven most popular cryptocurrencies.

The structure of the paper is as follows. In Section 2, we provide background on the methodology. In Section 3, we test this methodology on a chaotic time series generated by dynamical system undergoing bifurcations (with small additive noise). Our experiments show that that TDA is able to detect critical transitions in the time series, as the L^1-norms follow an increasing trend when the system transitions from a less turbulent to a more turbulent regime. Additional examples of this behavior can be found in [22, 2, 3]. Since these time series are simulated, it is easy to match the changes in the norms with the changes in the parameters driving the system. Topological recognition of critical transitions becomes much more difficult in real-life financial time series with no precise demarcation lines between different regimes. For this reason, we apply a k-means clustering technique to recognize the critical transition. In Section 4, we apply our methodology to the time series of Bitcoin, Ethereum, Litecoin, and Ripple between January 2016 and January 2018 and focus on recognition of different topological patterns associated to the critical transition that lead to the crash. The k-means clustering method to a data set consisting of 3 variables: the normalized value of the price of each asset, the log-return of asset, and L^1-norm of the persistence landscape. The outcome of this automatic classifier is the identification of clusters of topologically distinct regimes, which correspond to relevant time periods prior to the crash. In Section 5, we conclude that the proposed method is able to recognize critical transitions in time-series of the cryptocurrencies under consideration, and can be used to identify specific time-periods associated to such transitions.

The salient features of our approach are the following: (I) the method applies to time series generated by complex systems that exhibit strong non-linearity and non-stationarity, (II) the method is robust under small additive noise, and (III) the method is able to recognize critical transition even when the applied sliding windows are short.

2. Methodology

2.1. Time-delay coordinate embedding

Time-delay coordinate embedding is used to obtain a description of the phase space of a nonlinear dynamical system from the time series of certain observables of the system.

We briefly review the method. Given a discrete dynamical system defined by a diffeomorphism $f : M \to M$ on a D-dimensional manifold M, the evolution of a state $x \in M$ is given by its orbit $\{f^t(x) : t \in \mathbb{Z}\}$. Given an observable $\phi : M \to \mathbb{R}$ of the system, one defines a reconstructed set $\mathscr{A}^d$ consisting of d-dimensional vectors of the form

$$(\phi(f^{-(d-1)}(x)), \phi(f^{-(d-2)}(x)), \ldots, \phi(f^{-1}(x)), \phi(x)),$$

which are obtained by successive evaluations of ϕ along orbit segments

$$\{f^{-(d-1)}(x), f^{-(d-2)}(x), \ldots, f^{-1}(x), x\}.$$

Assume that f, ϕ are C^r-differentiable with $r \geqslant 2$. The classical theorem of Takens [44] says that, if $d \geqslant 2D + 1$, then for generic (C^1-open and dense) sets of f and ϕ, the map

$$x \in M \mapsto \Phi(x) := (\phi(f^{-(d-1)}(x)), \phi(f^{-(d-2)}(x)), \ldots, \phi(f^{-1}(x)), \phi(x))$$

is an embedding. Moreover, the map f on M is topologically conjugate with the left-shift map σ on the reconstructed set $\mathscr{A}^d$, which is defined by

$$(\phi(f^{-(d-1)}(x)), \ldots, \phi(f^{-1}(x)), \phi(x)) \mapsto (\phi(f^{-(d-2)}(x)), , \ldots, \phi(x), \phi(f(x))).$$

Topological conjugacy means that Φ matches orbits of f with orbits of σ, i.e., $\sigma \circ \Phi = \Phi \circ f$.

Takens' Theorem has been extended in [39] to the case of an invariant set (e.g., an attractor) $A \subseteq M$, in which case it is sufficient to consider delay-coordinate vectors of dimension $d \geqslant 2D + 1$, where D is the box-counting dimension of the invariant set. Also, the genericity with respect to f and ϕ is replaced with a concept of 'prevalence'. The reconstructed set $\mathscr{A}^d$ is topologically equivalent to A. An important consequence is that the topology of A can be described in terms of the homology groups of $\mathscr{A}^d$.

Thus, bifurcations of attractors can be detected via changes in the homology groups associated to the time-delay coordinate reconstructed sets.

One problem in practice is that the embedding dimension d is not known beforehand, and the dimension D of the invariant set cannot be estimated without first embedding the time series. There are practical methods to estimate the embedding dimension; see, e.g., [26]. However, in this paper we will only use a fixed, low dimensional embedding.

We end up this quick review by noting that there also exist stochastic versions of Takens Theorem, for stochastically forced systems; see, e.g., [43].

2.2. State space reconstruction from time series

Consider a dynamical system, consisting of a deterministic part plus a stochastic part, and a corresponding time series

$$X = \{x_1, x_2, \ldots, x_N\},$$

of measurements of a certain observable of the system at the time instants $i = 1, 2, \ldots, N$. The underlying dynamical system is a 'black box', i.e., it is unknown to the observer.

Transform the time series X into a sequence of $(N - d + 1)$ d-dimensional delay coordinate vectors

$$\begin{aligned} z_1 &= (x_1, x_2, \ldots, x_d), \\ &\ldots \\ z_i &= (x_i, x_{i+1}, \ldots, x_{i+d-1}), \\ &\ldots \\ z_{N-d+1} &= (x_{N-d+1}, x_{N-d+2}, \ldots, x_N), \end{aligned}$$

where the embedding dimension d is suitably chosen. In our applications below, we will only be interested in detecting $D = 1$-dimensional objects (loops), in which case we can safely choose $d \geqslant 3$, due to Takens' theorem.

The vectors $\{z_1, z_2, \ldots, z_{N-d+1}\}$ form a time-varying point-cloud embedded in $\mathbb{R}^d$. We apply a sliding window $Z_j = \{z_j, z_{j+1}, \ldots, z_{j+w-1}\}$ of size w, for $j \in \{1, \ldots, N-d-w+2\}$, with w sufficiently large with $d \ll w \ll N$. The evolution in time of the system is now described by a cloud of w points in $\mathbb{R}^d$, which changes slowly in time. The size w of the window is chosen empirically depending on the system under investigation. In our case, we choose w to correspond to the length of a typical time-interval on which the system does not change too much.

A model to be considered is that of a dynamical system with a slowly evolving parameter, dressed with noise. In the case when the parameter is frozen and there is no noise, all states of the system approach some fixed attractor, – e.g., an attractive fixed point, or a periodic orbit, or a chaotic attractor –, which is viewed as a quasi-stationary state. In the case when the parameter is slowly varying, but still there is no noise, the attractor is slowly evolving in time (e.g., undergoing bifurcations), and typically changes its topology. The state of the system closely follows the evolving attractor, and undergoes a change in topology as well. If small noise is added, the state of the system will oscillate around the time-varying attractor, but when approaching a bifurcation it can also jump between one attractor and another, thus experiencing a critical transition. An explicit example is given in Section 3. In Section 2.3 we describe how to use topological data analysis – more precisely, persistence landscapes –, to track the changes in the topology of the evolving attractor.

2.3. Topological data analysis of time-varying point-clouds

In this section we briefly review the ideas of persistence homology and persistence landscapes applied to point-clouds. Technical details on this method can be found in, e.g,, [15, 7, 17, 4, 12, 5].

For each instant of time $j \in \{1, \ldots, N - d - w + 2\}$, we consider the corresponding point-cloud $\{Z_j\} \subseteq \mathbb{R}^d$ from Section 2.2. To simplify the notation, let us fix such a point-cloud and denote it by $Z = \{z_1, \ldots, z_w\}$. We associate to it a topological space as follows. Introduce a 'resolution' parameter $\varepsilon > 0$. Define the so called Vietoris-Rips simplicial complex $R(Z, \varepsilon)$, or, simply Rips complex, obtained as follows:

- for each $j = 1, 2, \ldots$, a k-simplex of vertices $\{z_{i_1}, \ldots, z_{i_k}\}$ is part of $R(Z, \varepsilon)$ if and only if the mutual distance between any pair of its vertices is less than ε, that is

$$d(z_{i_j}, z_{i_l}) < \varepsilon, \text{ for all } z_{i_j}, z_{i_l} \in \{z_{i_1}, \ldots, z_{i_k}\}.$$

In other words, a k-simplex is included in $R(X, \varepsilon)$ whenever the vertices of that simplex are 'indistinguishable from one another' at resolution level of ε.

The Rips simplicial complexes $R(Z, \varepsilon)$ form a filtration, that is, $R(Z, \varepsilon) \subseteq R(Z, \varepsilon')$ whenever $\varepsilon < \varepsilon'$. For each such complex, we can compute its k-dimensional homology $H_k(R(Z, \varepsilon))$ with coefficients in some field. Informally, the generators of the 0-dimensional homology group $H_0(R(Z, \varepsilon))$ correspond to the connected components of $R(Z, \varepsilon)$, the generators of the 1-dimensional homology group $H_1(R(Z, \varepsilon))$ correspond to the 'independent loops' in $R(Z, \varepsilon)$, the generators of the 2-dimensional homology group $H_0(R(Z, \varepsilon))$ correspond to 'independent cavities' in $R(Z, \varepsilon)$, etc. In the sequel, we will use only 1-dimensional homology.

The filtration property of the Rips complexes induces a filtration on the corresponding homologies, that is $H_k(R(Z, \varepsilon)) \subseteq H_k(R(Z, \varepsilon'))$ whenever $\varepsilon < \varepsilon'$, for each k.

These inclusions determine canonical homomorphisms $H_k(R(Z,\varepsilon)) \hookrightarrow H_k(R(Z,\varepsilon'))$, for $\varepsilon < \varepsilon'$. Due to this family of induced mappings, for each non-zero k-dimensional homology class α there exists a pair of values $\varepsilon_1 < \varepsilon_2$, such that:

- $\alpha \in H_k(R(Z,\varepsilon_1))$ but is not in the image of any $H_k(R(Z,\varepsilon_1 - \delta))$ under the corresponding homomorphism, for $\delta > 0$,
- the image of α in $H_k(R(Z,\varepsilon'))$ is non-zero for all $\varepsilon_1 < \varepsilon' < \varepsilon_2$, but the image of α in $H_k(R(Z,\varepsilon_2))$ is zero.

In this case, one says that the class α is 'born' at the parameter value $b_\alpha := \varepsilon_1$, and 'dies' at the parameter value $d_\alpha = \varepsilon_2$; the pair (b_α, d_α) represents the 'birth' and 'death' indices of α. The multiplicity $\mu_\alpha(b_\alpha, d_\alpha)$ of the point (b_α, d_α) equals the number of classes α that are born at b_α and die at d_α. This multiplicity is finite since the simplicial complex is finite.

The information on the k-dimensional homology generators at all scales can be encoded in a *persistence diagram* P_k. Such a diagram consists of:

- for each k-dimensional homology class α one assigns a point $p_\alpha = p_\alpha(b_\alpha, d_\alpha) \in \mathbb{R}^2$ together with its multiplicity $\mu_\alpha = \mu_\alpha(b_\alpha, d_\alpha)$;
- in addition, P_k contains all points in the positive diagonal of $\mathbb{R}^2$; these points represent all trivial homology generators that are born and instantly die at every level; each point on the diagonal has infinite multiplicity.

The axes of a persistence diagram are birth indices on the horizontal axis and death indices on the vertical axis.

The space of persistence diagrams can be embedded into a Banach space, whose norm can be used to derive a metric. One such an embedding is based on *persistence landscapes*, consisting of sequences of functions in the Banach space $L^p(\mathbb{N} \times \mathbb{R})$. For each birth-death point $(b_\alpha, d_\alpha) \in P_k$, we first define a piecewise linear function

$$f_{(b_\alpha,d_\alpha)} = \begin{cases} x - b_\alpha, & \text{if } x \in \left(b_\alpha, \frac{b_\alpha+d_\alpha}{2}\right]; \\ -x + d_\alpha, & \text{if } x \in \left(\frac{b_\alpha+d_\alpha}{2}, d_\alpha\right); \\ 0, & \text{if } x \notin (b_\alpha, d_\alpha). \end{cases} \tag{2.1}$$

To a persistence diagram P_k consisting of a finite number of off-diagonal points, we associate a sequence of functions $\lambda = (\lambda_k)_{k\in\mathbb{N}}$, where $\lambda_k : \mathbb{R} \to [0;1]$ is given by

$$\lambda_k(x) = k\text{-max}\{f_{(b_\alpha,d_\alpha)}(x) \,|\, (b_\alpha, d_\alpha) \in P_k\} \tag{2.2}$$

where k-max denotes the k-th largest value of a function. We set $\lambda_k(x) = 0$ if the k-th largest value does not exist. An example of a point cloud, the corresponding persistence diagram and persistence landscape are shown in Fig. 3

Via the above embedding, the persistence landscapes form a subset of the Banach space $L^p(\mathbb{N} \times \mathbb{R})$, where the norm of λ is given by:

$$\|\lambda\|_p = \left(\sum_{k=1}^{\infty} \|\lambda_k\|_p^p \right)^{1/p}. \tag{2.3}$$

Above, $\|\cdot\|_p$ denotes the L^p-norm, $p \geqslant 1$, i.e., $\|f\|_p = \left(\int_{\mathbb{R}} |f|^p\right)^{1/p}$, where the integration is with respect to the Lebesgue measure on $\mathbb{R}$. In the sequel, we will use only the L^1 norm.

The above computation of persistence diagrams and persistence landscapes is to be carried out for every sliding window Z_j. The output is the time series of the L^1-norm of the corresponding persistence landscapes $\{\|\lambda_k^j\|_1\}_{k\geqslant 0}$. This can be summarized by the following

Pipeline:	$Z_j \longrightarrow$	$\{R(Z_j, \varepsilon)\}_\varepsilon \longrightarrow$	$\{H_*(R(Z_j, \varepsilon))\}_\varepsilon \longrightarrow$	$\{\lambda_k^j\}_{k\geqslant 0} \longrightarrow$	$\{\|\lambda_k^j\|_1\}_{k\geqslant 0}$.
	Point-cloud	Rips complex	Homology	Landscape	L^1-norm

A remarkable property that makes persistence homology suitable to analyze noisy data is its *robustness* under small perturbations. Informally, this property says that if the underlying cloud point data set changes only 'little', then the corresponding persistence diagrams changes only by a 'small' distance.

2.4. K-means clustering

To capture in a precise way the relationship between the underlying time series and the L^1-norms, we use an unsupervised machine learning technique utilizing non-parametric, geometry based k-means clustering. The main point is that this method requires no statistical assumptions to be satisfied beforehand, in contrast with most statistical tests.

The basic algorithm starts with k initial centroids (randomly chosen), and assigns each data point to one of the centroids; each collection of points assigned to the same centroid forms a cluster. For each cluster, the squared error is calculated as the sum of the squares of the Euclidean distances between the data points and the cluster centroid. Then the centroid of each cluster is re-calculated and updated, and the process of assigning the data points to clusters is repeated. The procedure is iterated until the outcome stabilizes, i.e., the updated centroids remain the same, and no point changes cluster. This is achieved when the cluster squared error cannot be reduced any further. Different choices of the initial (random) centroids may yield different clusters. To obtain independence of the clusters on the initial choices, one runs a large number of random initial centroids, and then select the one yielding the lowest cluster squared error. The number k of clusters needs to be pre-specified at the beginning of the algorithm. For a survey and references see [25].

As input for the k-means clustering, we use the values of the time series of interest, the first difference in these values (or in the logarithms of these values), and the L^1-norms of the persistence landscapes associated to the time series. The reason for including the first difference in the values of the time series in the classifier resides with the subsequent application of this approach to financial time series, where the first difference corresponds to the return. The output consists of subsets of data for which the trends in the time series are consistent with those in the L^1-norms of the persistence landscapes. We illustrate the application of this method in Section 3, and use it in Section 4.

3. Chaotic time series

We consider time series derived from a *chaotic* dynamical system dressed with noise, on which we test the methodology described above to detect critical transitions. The chaotic system is given by a three-dimensional diffeomorphism that, for certain parameter values, possesses Lorenz-type chaotic attractors [23]. We stress that this system is different from the well-known system of differential equations that generates the classical Lorenz attractor, and hence it has different properties; however, for some parameter values, it mimics very well the motion along continuous trajectories of the classical Lorenz attractor. The diffeomorphism that generates the dynamics is:

$$f(x, y, z) = (y, z, M_1 + Bx + M_2 y - z^2), \tag{3.1}$$

where M_1, B, M_2 are real parameters. For certain values of M_1, B, M_2, all suitable initial conditions approach asymptotically a strange attractor. When M_1 and B are kept fixed and M_2 is varied, the system evolves from a single attractive fixed point to a pair of attractive fixed points, then to a pair of attractive periodic orbits, and eventually ends up with a chaotic attractor. The bifurcation diagram for $M_1 = 0$, $B = 0.7$, and varying M_2, is shown in Fig. 1. Several instances of the attractor are shown in Fig. 2.

We use this model to illustrate the methodology presented in Section 2.3; we generate the attractor for $M_1 = 0$, $B = 0.7$, $M_2 = 0.81$, then we reconstruct the topology

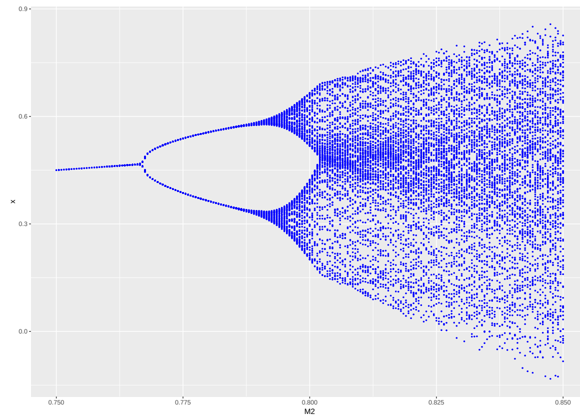

Figure 1: Bifurcation diagram of the Lorenz map, in (M_2, x)-coordinates.

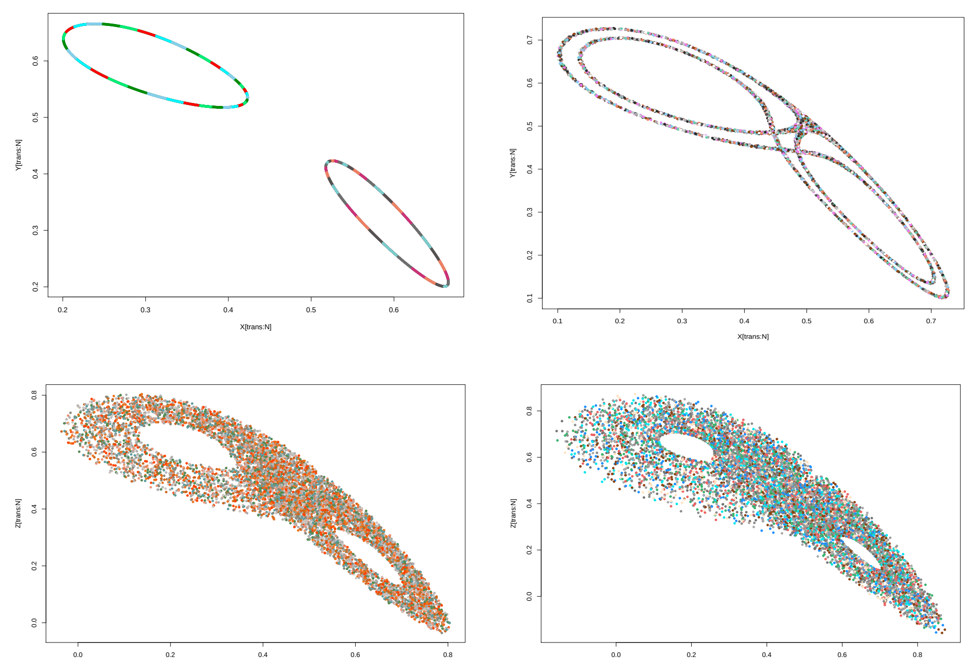

Figure 2: Top: Lorenz attractor for: (i) $M_2 = 0.8$ (top-left), (ii) $M_2 = 0.81$ (top-right), Bottom: (iii) $M_2 = 0.83$ (bottom-left), (iv) $M2 = 0.85$ (bottom-right)

of the phase space from the x-time series, utilizing a 4D embedding. Then we compute the persistence diagram and the corresponding persistence landscape. See Fig. 3.

Now we consider the system with a slowly evolving parameter, that is, at each step of the iteration of (3.1), we increase the parameter M_2 by a small increment. That is:

$$x_{n+1} = y_n,\ y_{n+1} = z_n,\ z_{n+1} = M_1 + Bx_n + M_{2,n}y_n - z_n^2,\ M_{2,n+1} = M_{2,n} + \Delta M_2. \quad (3.2)$$

We note that, starting with some initial condition (x_0, y_0, z_0) and some initial parameter value of M_2, the x-time series generated by (3.2), shown in Fig. 4, follows closely, but not exactly, the bifurcation diagram in Fig. 1, generated by (3.1). The reason for the difference is that at each step of the iteration, the attractor for the corresponding $M_{2,n}$ changes a little, and the one-step iteration of the state (x_n, y_n, z_n) keeps lagging behind the attractor, whose parameter has now changed to $M_{2,n+1}$.

As we are interested in critical transition, we cut the time series at $M_2 = 0.81$, which corresponds roughly to the parameter value corresponding to the bifurcation to a chaotic attractor with frozen parameters. We also dress the system with small Gaussian noise. As before, we extract the x-time series, and we we use 4-dimensional delay coordinate vectors to reconstruct the phase space. We use a sliding window of width $w = 100$ on the sequence of 4-dimensional vectors to generate a sequence of 4-dimensional point clouds, each point cloud consisting of 100 points. On this sequence of cloud points we apply TDA, obtaining the time series of the L^1-norms of the 1-dimensional persistence landscapes. See Fig. 4.

To capture the relation between the x-time series and the L^1-norms, we use the k-means clustering classifier described in Section 2.4. As input we use the values of the x-time series, the first difference in these values, and the L^1-norms of the persistence landscapes. In order to have a finer segmentation of the data, we use a relatively large number of clusters $k = 8$. The clusters are shown in Fig. 5. Cluster 2 appears significant, as it is clearly separated from the others; see Table 6.1. The parameter M_2 corresponding to this cluster ranges between 0.8091 and 0.81. There is an increasing trend of the L^1 norm for this range. See Fig. 6.

The conclusion of this experiments is that TDA is able to detect the bifurcation, even though the size of the sliding window (hence the size of the underlying point cloud) is quite small. The time series of L^1-norms grows with the increasing 'turbulence' in the system even prior to the bifurcation value. This is a well known type of behavior in the theory of critical transitions, where the instability of the system can be amplified by the addition of small noise, thus providing early signs of critical transition. This phenomenon is related to stochastic resonance; see, e.g., [36].

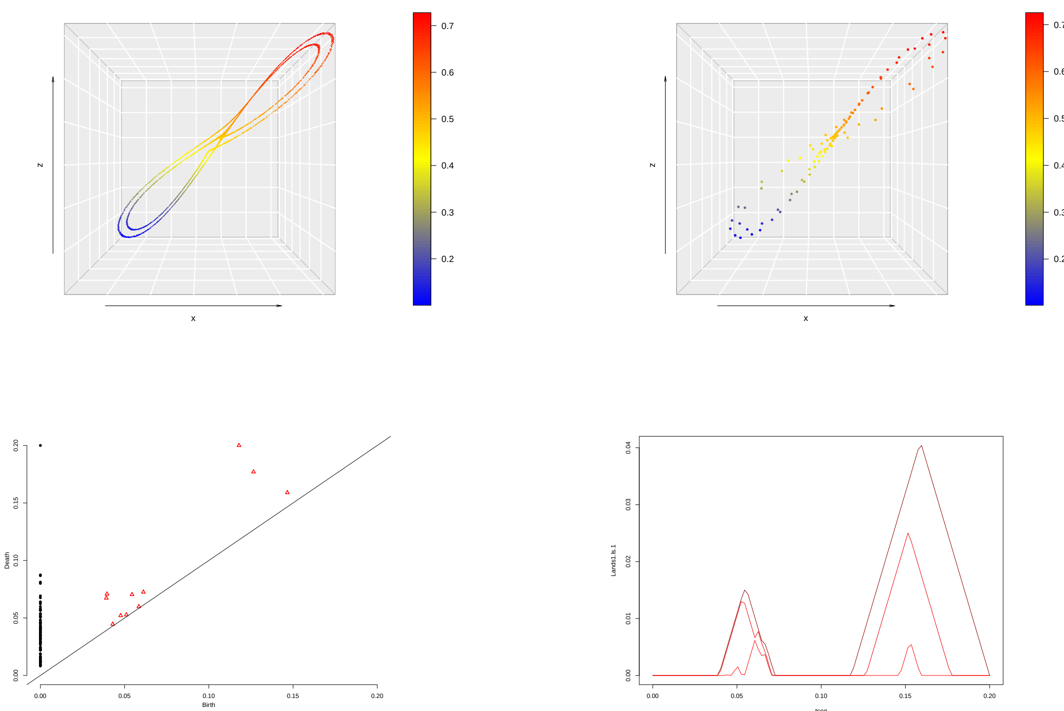

Figure 3: Example: (a) Lorenz-type attractor for $M_1 = 0$, $B = 0.7$, $M_2 = 0.81$, (b) reconstructed Lorenz-type attractor, (c) persistence diagram, where the 0-dimensional generators are displayed as dot symbols (black), and the 1-dimensional generators as triangle symbols (red); the two main triangle symbols correspond to the two periodic orbits (loops); the other triangle symbols correspond to noisy loops in the point-cloud, (d) 1-dimensional persistence landscape $\lambda = (\lambda_k)_k$, for $k = 1, 2, 3$; the two main peaks correspond to the two periodic orbits (loops), and the smaller peaks correspond to noisy loops.

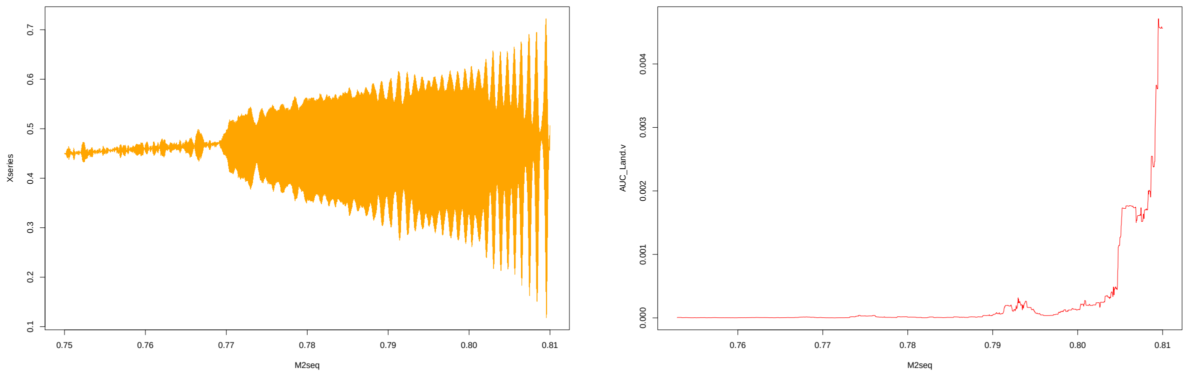

Figure 4: Lorenz map: (a) x-time series, (b) L^1-norms.

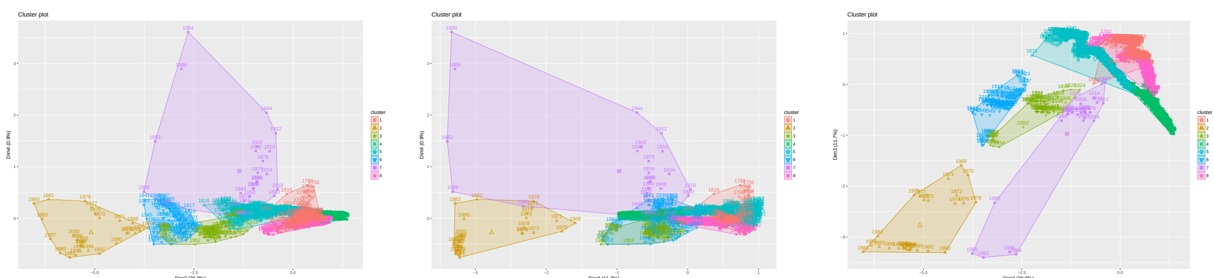

Figure 5: Lorenz map time series k-means clustering: projections onto principal components: $(1, 3)$, $(2, 3)$, $(1, 2)$.

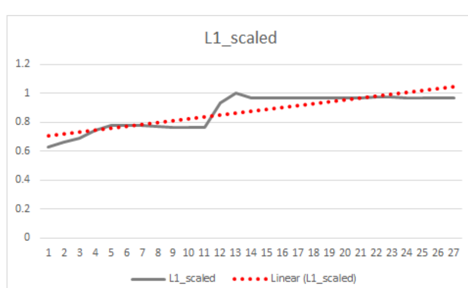

Figure 6: The L^1-norms for the Lorenz map time series, cluster no 2.

4. Analysis of cryptocurrencies

Cryptocurrencies constitute a new type of financial assets. The underlining blockchain technology is robust. However, due to novelty of these financial instruments, the time series of the relevant exchange rates feature wild swings in value. Statistical properties of such assets show strong deviation of distributions of log-returns from normality, and substantial variability of first and second moments in time. Thereby, offering an interesting real-world object for studying 'turbulent' behavior, and an excellent test case for applying TDA for possible detection of critical transitions in financial markets.

We analyze 4 cryptocurrencies – Bitcoin, Ethereum, Litecoin, Ripple – over a period of time from the beginning of 2016 to early 2018. Each of these cryptocurrencies suffered one or two 'crashes', during December 2017 – January 2018. We apply TDA to the time series of the log of prices (in USD) for each of the 4 digital assets, and investigate whether the TDA method is able to recognize changes in the relevant time series prior to these crashes. Therefore, for each of the 4 cryptocurrencies under consideration we identify the one or two most significant peaks in the value prior to the crash, and we cut the time series right at the top of the last peak. The cut-off dates for the 4 time series are slightly different. Also, while Bitcoin and Ethereum show two significant peaks before the crash, Litecoin and Ripple show only one peak. To be precise:

Asset	Starting date	1st peak	2nd peak
Bitcoin	2016-01-01	2017-12-17	2018-01-07
Ethereum	2016-01-01	2018-01-13	2018-01-28
Litecoin	2016-01-01	2017-12-18	
Ripple	2016-01-01	2018-01-03	

For the delay-vector embedding of the time series, we use a dimension $d = 4$, and a size of the sliding window $w = 50$ trading days.

We apply TDA and compute the L^1-norm of the persistence landscapes. We display together the log of the price of the asset, the log-return of the asset, the L^1-norm of the persistence landscapes, and the first difference of the L^1-norms, for each of the 4 assets, in Fig. 7, Fig. 8, Fig. 9, Fig. 10. Inspecting these plots, we notice that L^1-norms tend to peak in the vicinity of the crashes; there is also an increase in the first difference of the L^1-norms. There are other regions in the time series where the price of asset exhibits large swings, which appear to match peaks in the L^1-norms; some of these peaks in the L^1-norms are even larger than those prior to the major crash. This makes it clear that we cannot rely on the L^1-norms alone to recognize approaching major crashes.

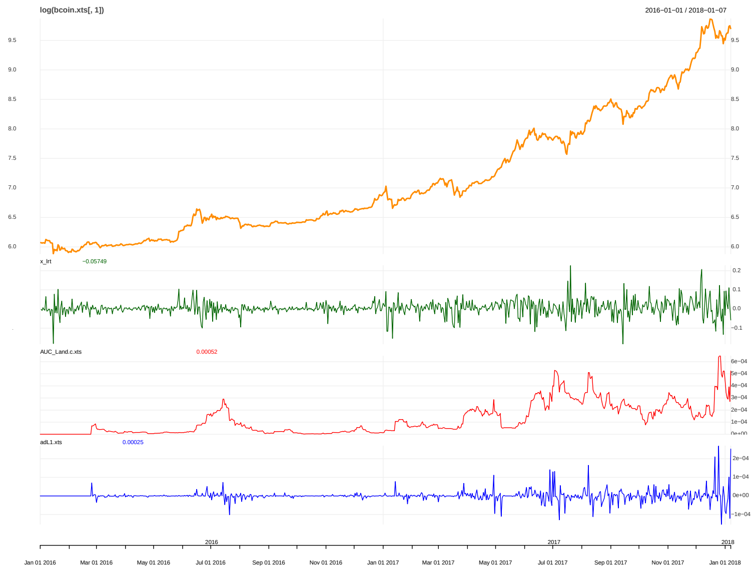

Figure 7: Bitcoin 2016-01-01 – 2018-01-07: log of the values of the asset, the log-return of the asset, the L^1-norm, and the first difference of the L^1-norms.

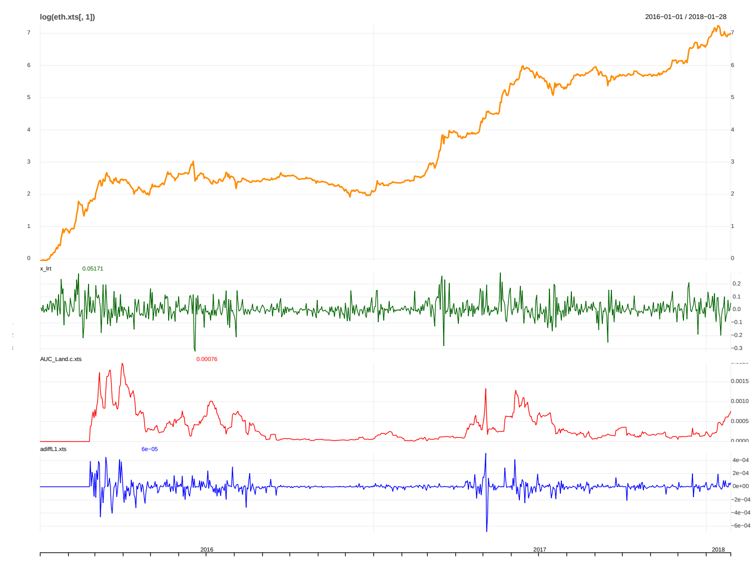

Figure 8: Ethereum 2016-01-01 – 2018-01-28: log of the values of the asset, the log-return of the asset, the L^1-norm, and the first difference of the L^1-norms.

Figure 9: Litecoin 2016-01-01 – 2018-01-06: log of the values of the asset, the log-return of the asset, the L^1-norm, and the first difference of the L^1-norms.

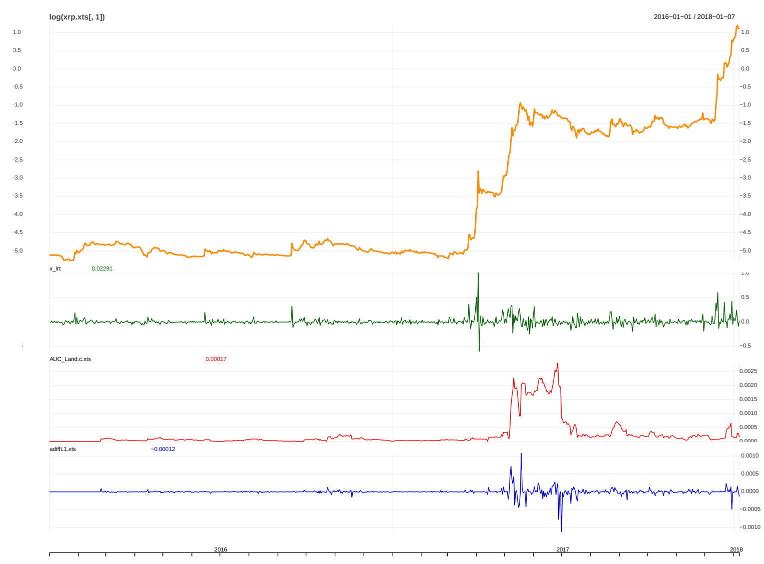

Figure 10: Ripple 2016-01-01 – 2018-01-07: log of the values of the asset, the log-return of the asset, the L^1-norm, and the first difference of the L^1-norms.

To quantify the relationship between the price/return of each asset and the L^1-norms prior to the crash, we use an unsupervised machine learning technique utilizing non-parametric, geometry based k-means clustering, briefly described in Section 2.4. The main point is that this method requires no statistical assumptions to be satisfied beforehand, in contrast with most statistical tests.

As input data we use x = log of the price of the asset (in USD), y = log-return of the asset, and $z = L^1$-norm of the persistence landscape. Thus, our data set consists of points $(x, y, z) \in \mathbb{R}^3$. Each of these times series is normalized to the $[0, 1]$ range. We make an empirical choice of a relatively large number of clusters ($k = 18$), which is bigger than the optimal value obtained from the basic methods: 'elbow', 'silhouette' and 'gap statistic'. The reason of choosing a large k is to obtain a finer segmentation of the data. Smaller values of k yield periods of time that are too large (e.g., larger than the window size $w = 50$), while larger k's yield periods of time that are too small (e.g., a couple of days). The clusters are visually depicted by first performing principal component analysis (PCA) and then projecting onto the principal axes. Thus, for each data set we have 3 possible projections. See Fig. 11, Fig. 12, Fig. 13, Fig. 14. This graphical rendering helps us to identify a small number of significant clusters that are most separated from the others. The corresponding information on these clusters is given in Table 6.2, Table 6.4, Table 6.4, Table 6.5. (Note that the number associated to each cluster is just an identifier assigned automatically.) We note here that these clusters are relatively robust, since running the k-means clustering algorithm for slightly lower or bigger values of k leaves these clusters unchanged.

Now we analyze the clusters for the 4 assets.

For Bitcoin, in Table 6.2, we distinguish two significant clusters. Cluster no. 2 consists of a discontiguous period between December 8, 2017 – January 6, 2018. Within this period, the L^1-norms display an increasing trend between December 8, 2017 – December 15, 2017 – which is just before the 1-st peak of the Bitcoin on December 17, 2017 –, and a second increasing trend between December 18, 2017 – January 3, 2018 – which is just before the 2-nd peak of the Bitcoin on January 7, 2018. Cluster no. 1 consists of a discontiguous period between December 21, 2017 – January 7, 2018. Within this period, the L^1-norms display an overall increasing trend, up to the 2-nd peak of the Bitcoin. Overall, the combined information from these two clusters correctly identifies the critical period before the Bitcoin crash, and within that period the L^1-norm shows increasing trends that reach their maximum both prior to the 1-st peak and prior to the 2-nd peak. See Fig. 15.

For Ethereum, in Table 6.4, we also distinguish two significant clusters. Cluster no. 3 consists of a discontiguous period between January 3, 2018 – January 21, 2018. Within this period, the L^1-norms displays an overall increasing trend, with a sharp increase around the 1-st peak of the Ethereum on January 13, 2018. Cluster no. 13 consists of a discontiguous period between January 12, 2018 – January 28, 2018, before the 2-nd peak of the Ethereum. The L^1-norms follow an increasing trend throughout the whole period. See Fig. 16.

For Litecoin, in Table 6.4, we distinguish one significant cluster. Cluster no. 12 consists of a discontiguous period between November 23, 2017 – December 11, 2018. During this period the L^1-norms show an increasing trend. We note here that this increasing trend continues during the period December 12, 2017 – December 13, 2017 – which is part of cluster no 11 (not shown in the table) –, and plateaus during December 14, 2017 – December 18, 2017, – which is part of cluster 8 (not shown in the table). Thus, the increasing trend occurs over a significant period prior the peak of Litecoin on December 18, 2017, even though this happens across multiple clusters.

For Ripple, in Table 6.5, we also distinguish one significant cluster. Cluster no 8 consists of one discontiguous period between December 22, 2017 – December 29, 2017, where the L^1-norms display a steep increasing. This period precedes the first peak of the Ripple on January 3, 2018. See Fig. 18.

Thus, the k-means clustering applied to the data consisting of the log-price of each asset, the log-return, and the L^1-norms of the persistence landscapes, provides identification of topologically distinct regimes before the crash of each asset. Within each corresponding period of time, we make an empirical observation that the L^1-norms follow an increasing trend, a behavior consistent with the findings in [22].

5. Conclusions

We have presented a TDA-based method to detect critical transitions in complex systems by exploring changes in the topological properties of the relevant time series. The proposed method involves time-delay coordinate embedding, sliding windows, and persistence landscapes. It can be applied to systems that are strongly nonlinear, and performs well even with short-time windows, whereas typical statistical methods are prone to ambiguities. We applied this method to the time series of the most capitalized cryptocurrencies – Bitcoin, Ethereum, Litecoin, and Ripple, – for some interval of time before the crash at the end of 2017 – beginning 2018. Furthermore, we have used a k-means clustering technique to identify topologically distinct regimes in the time series of observed and derived signals for each digital asset, and matched them to relevant time periods prior to the crash. In summary, we conclude that the combination of the TDA-based method and k-means clustering technique has the potential to automatically recognize approaching critical transitions in the cryptocurrency markets, even when the relevant time series exhibit a highly non-stationary, erratic behavior.

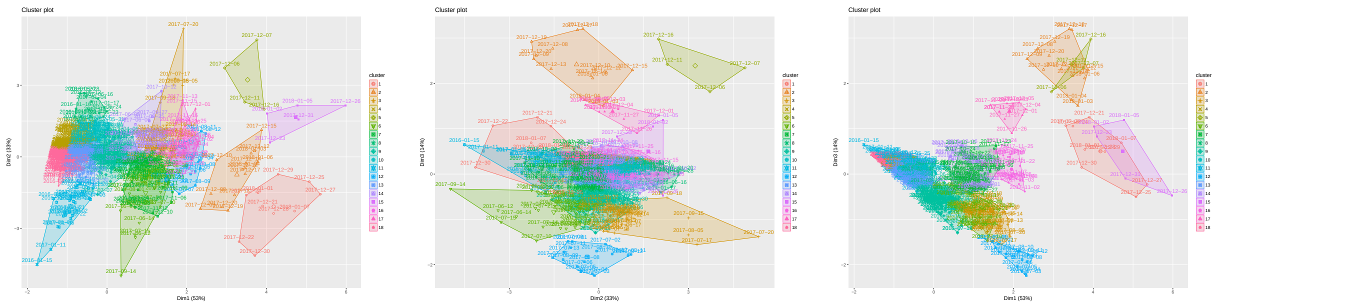

Figure 11: Bitcoin k-means clustering: projections onto principal components: $(1,2)$, $(2,3)$, $(1,3)$.

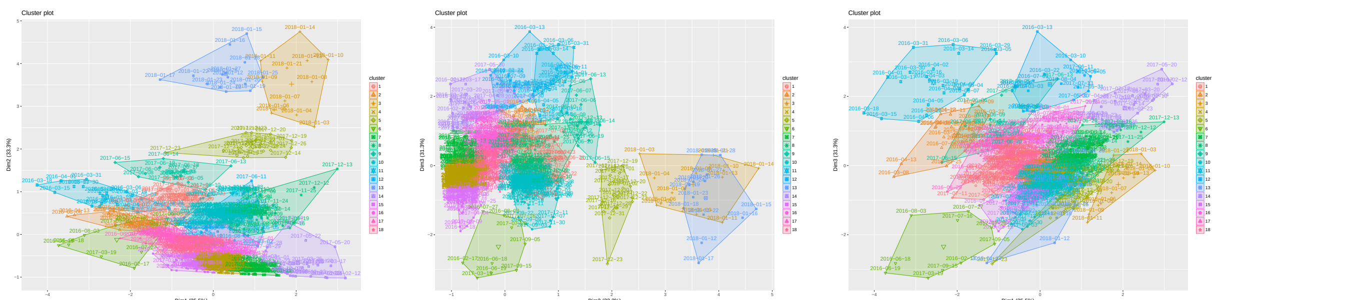

Figure 12: Ethereum k-means clustering: projections onto principal components: $(1,2)$, $(2,3)$, $(1,3)$.

Figure 13: Litecoin k-means clustering: projections onto principal components: $(1,2)$, $(2,3)$, $(1,3)$.

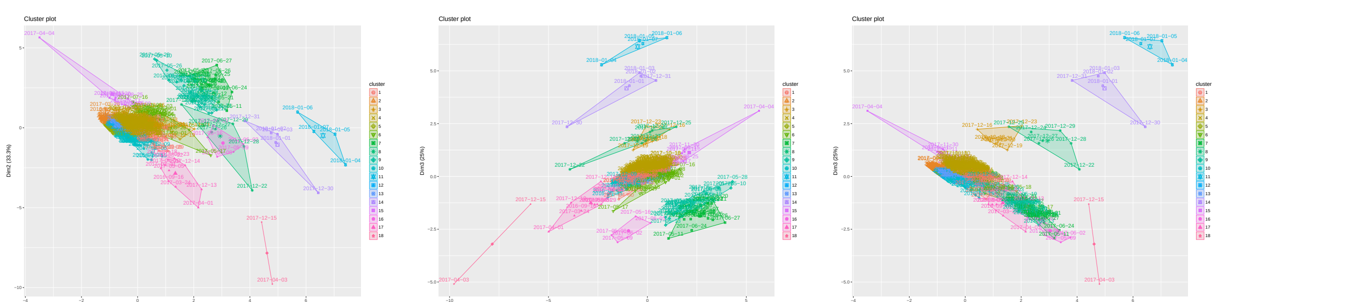

Figure 14: Ripple k-means clustering: projections onto principal components: $(1,2)$, $(2,3)$, $(1,3)$.

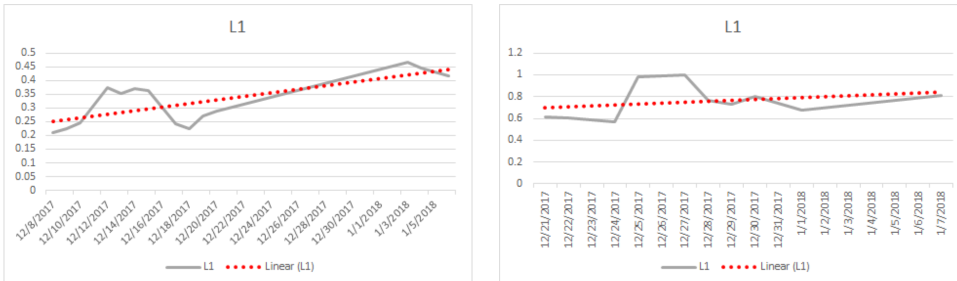

Figure 15: The L^1-norms for Bitcoin: (a) cluster no 2, (b) cluster no 1.

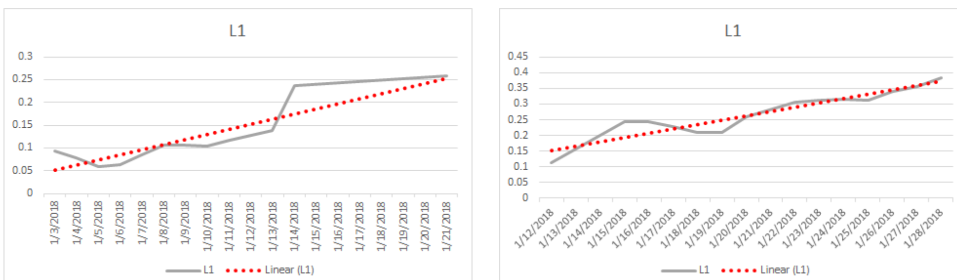

Figure 16: The L^1-norms for Ethereum: (a) cluster no 3, (b) cluster no 13.

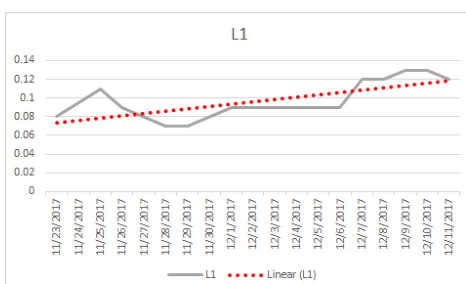

Figure 17: The L^1-norms for Litecoin, cluster no 12.

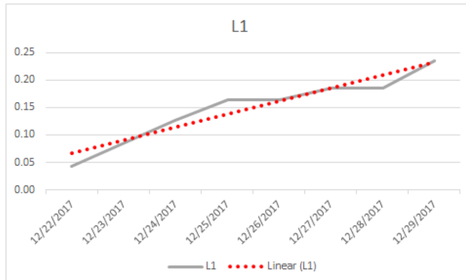

Figure 18: The L^1-norms for Ripple, cluster no 8.

6. Acknowledgement

Research of M.G. was partially supported by NSF grant DMS-0635607 and by the Alfred P. Sloan Foundation grant G-2016-7320. Research of Y.K. was supported by S&P Global Market Intelligence. The views expressed in this paper are those of the authors, and do not necessary represent the views of S&P Global Market Intelligence.

Appendix

Table 6.1. Lorenz map – significant cluster(s)

Cluster 2	x	x differences	L1
0.8091	0.6891	0.1184	0.6304
0.8091	0.5636	0.1669	0.6639
0.8092	0.7176	0.1108	0.6867
0.8092	0.5357	0.1809	0.7448
0.8092	0.7551	0.1004	0.7778
0.8093	0.4968	0.2018	0.7778
0.8093	0.7987	0.0881	0.7778
0.8093	0.4283	0.2388	0.7731
0.8093	0.8590	0.0705	0.7657
0.8094	0.3424	0.3004	0.7663
0.8094	0.9221	0.0520	0.7663
0.8095	0.9800	0.0305	0.9355
0.8095	1.0000	0.0110	1.0000
0.8096	0.9451	0.0000	0.9704
0.8096	0.8425	0.0047	0.9704
0.8097	0.7310	0.0288	0.9704
0.8097	0.2688	0.3064	0.9687
0.8097	0.6464	0.0638	0.9677
0.8098	0.3974	0.2096	0.9680
0.8098	0.5976	0.0967	0.9680
0.8098	0.4953	0.1640	0.9691
0.8099	0.5714	0.1224	0.9719
0.8099	0.5623	0.1411	0.9730
0.8099	0.5619	0.1392	0.9681
0.8099	0.6096	0.1292	0.9681
0.8100	0.5599	0.1502	0.9681
0.8100	0.6430	0.1225	0.9684

Table 6.2. Bitcoin – significant cluster(s)

Cluster 2	price	return	L1
12/8/2017	0.827	0.328	0.209
12/9/2017	0.767	0.266	0.225
12/10/2017	0.773	0.467	0.247
12/12/2017	0.886	0.516	0.373
12/13/2017	0.845	0.335	0.354
12/14/2017	0.852	0.465	0.371
12/15/2017	0.908	0.6	0.362
12/17/2017	0.986	0.414	0.244
12/18/2017	0.98	0.431	0.226
12/19/2017	0.909	0.266	0.27
12/20/2017	0.848	0.281	0.288
1/3/2018	0.778	0.511	0.467
1/4/2018	0.779	0.45	0.447
1/6/2018	0.884	0.475	0.418

Cluster 1	price	return	L1
12/21/2017	0.801	0.311	0.613
12/22/2017	0.711	0.164	0.606
12/24/2017	0.717	0.349	0.568
12/25/2017	0.714	0.437	0.983
12/27/2017	0.791	0.389	1
12/28/2017	0.741	0.291	0.763
12/29/2017	0.741	0.447	0.727
12/30/2017	0.646	0.122	0.806
1/1/2018	0.688	0.367	0.677
1/7/2018	0.833	0.307	0.807

Table 6.3. Ethereum: significant cluster(s)

Cluster 3	price	return	L1
1/3/2018	0.634	0.754	0.093
1/4/2018	0.688	0.663	0.079
1/5/2018	0.698	0.552	0.06
1/6/2018	0.712	0.56	0.064
1/7/2018	0.746	0.605	0.085
1/8/2018	0.829	0.701	0.106
1/9/2018	0.82	0.51	0.107
1/10/2018	0.93	0.736	0.105
1/11/2018	0.907	0.486	0.116
1/13/2018	0.909	0.68	0.139
1/14/2018	1	0.685	0.236
1/21/2018	0.827	0.694	0.258

Cluster 13	price	return	L1
1/12/2018	0.829	0.378	0.112
1/15/2018	0.977	0.489	0.243
1/16/2018	0.925	0.438	0.243
1/17/2018	0.759	0.202	0.229
1/18/2018	0.727	0.456	0.21
1/19/2018	0.736	0.548	0.21
1/20/2018	0.748	0.553	0.258
1/22/2018	0.755	0.378	0.306
1/23/2018	0.718	0.446	0.312
1/24/2018	0.706	0.5	0.314
1/25/2018	0.761	0.65	0.311
1/26/2018	0.753	0.511	0.34
1/27/2018	0.755	0.532	0.355
1/28/2018	0.795	0.613	0.384

Table 6.4. Litecoin: significant clusters

Cluster 12	price	return	L1
11/23/2017	0.19	0.46	0.08
11/25/2017	0.21	0.5	0.11
11/26/2017	0.24	0.58	0.09
11/28/2017	0.25	0.5	0.07
11/29/2017	0.26	0.48	0.07
12/1/2017	0.24	0.45	0.09
12/2/2017	0.27	0.56	0.09
12/3/2017	0.27	0.44	0.09
12/4/2017	0.28	0.44	0.09
12/5/2017	0.28	0.46	0.09
12/6/2017	0.28	0.41	0.09
12/7/2017	0.27	0.4	0.12
12/8/2017	0.27	0.41	0.12
12/9/2017	0.35	0.72	0.13
12/10/2017	0.43	0.65	0.13
12/11/2017	0.41	0.38	0.12

Table 6.5. Ripple: significant cluster(s)

Cluster 8	price	return	L1
12/22/2017	0.35	0.62	0.04
12/24/2017	0.36	0.38	0.13
12/25/2017	0.32	0.30	0.16
12/26/2017	0.34	0.41	0.16
12/27/2017	0.36	0.41	0.19
12/28/2017	0.42	0.47	0.19
12/29/2017	0.44	0.40	0.23

References

[1] Adler, R.J.; Bobrowski, O.; Borman, M.S.; Subag E.; Weinberger, S. (2010). Persistent homology for random fields and complexes. In Borrowing Strength: Theory Powering Applications, A Festschrift for Lawrence D. Brown, IMS Collections 6, 124–143.

[2] Berwald, J.; Gidea, M. (2014). Critical Transitions in a Model of a Genetic Regulatory System. Mathematical Biology and Engineering, 11:723, 2014.

[3] Berwald J.; Gidea, M; Vejdemo-Johansson, M. (2014). Automatic Recognition and Tagging of Topologically Different Regimes in Dynamical Systems. Discontinuity, Nonlinearity and Complexity, Vol. 3.

[4] Bubenik, P. (2015). Statistical topological data analysis using persistence landscapes, J. Machine Learning Research Vol. 16, 77.

[5] Bubenik P.; Dlotko P. (2016). A persistence landscapes toolbox for topological statistics, Journal of Symbolic Computation, Vol. 78, 91.

[6] Drożdż, S., Gębarowski, R., Minati, L., Oświęcimka, P., & Wątorek, M. (2018). Bitcoin market route to maturity? Evidence from return fluctuations, temporal correlations and multiscaling effects. eprint arXiv:1804.05916

[7] Carlsson, G. (2009). Topology and Data, Bull. Amer. Math. Soc., Vol. 46 255.

[8] Casdagli, M.; Eubank, S.; Farmer, J.D.; Gibson, J (2002). State space reconstruction in the presence of noise. Physica D: Nonlinear Phenomena, 51(1–3), 52–98.

[9] Catania, L., Grassi, S. and Ravazzolo, F. (2018). Forecasting Cryptocurrencies Financial Time Series (No. 5/2018).

[10] Catania, L.; Grassi, S. (2017). Modelling Crypto-Currencies Financial Time-Series. Available at SSRN: https://ssrn.com/abstract=3028486 or http://dx.doi.org/10.2139/ssrn.3028486

[11] Chan, S.; Chu, J.; Nadarajah, S.; Osterrieder, J. (2017). A statistical analysis of cryptocurrencies. Journal of Risk and Financial Management, 10(2), p.12.

[12] Chazal, F.; Fasy, B.T.; Lecci, F.; Rinaldo, A.; Wasserman, L. (2015). Stochastic convergence of persistence landscapes and silhouettes, Journal of Computational Geometry Vol. 6, 140–161.

[13] Chiu, J.; Koeppl, T. (2017). The economics of cryptocurrencies - bitcoin and beyond (No. 6688). Victoria University of Wellington, School of Economics and Finance.

[14] Chu, J.; Chan, S.; Nadarajah, S.; Osterrieder, J. (2017). GARCH Modelling of Cryptocurrencies. Journal of Risk and Financial Management, 10(4), p.17.

[15] Cohen-Steiner, D.; Edelsbrunner, H.; Harer, J. (2007). Stability of persistence diagrams, Discrete & Computational Geometry, Vol.37, 103.

[16] Dakos, V.; Carpenter, S.R.; Brock, W.A.; Ellison, A.M.; Guttal, V. et al. (2012) Methods for Detecting Early Warnings of Critical Transitions in Time Series Illustrated Using Simulated Ecological Data. PLOS ONE 7(7): e41010

[17] Edelsbrunner, H.; Harer, J. (2010). Computational topology: an introduction (Amer. Math. Soc.

[18] Holzmann, H.; Vollmer, S. (2008). A likelihood ratio test for bimodality in two-component mixtures with application to regional income distribution in the EU. AStA Advances in Statistical Analysis. Vol. 2, 57-69.

[19] Gerlach, J.-C.; Demos, G.; Sornette, D. (2018). Dissection of Bitcoin's Multiscale Bubble History from January 2012 to February 2018. eprint arXiv:1804.06261

[20] Garland, J.; Bradley, E.; Meiss, J.D. (2016). Exploring the topology of dynamical reconstructions. Physica D: Nonlinear Phenomena, 334, 49–59.

[21] Gidea, M. (2017) Topological Data Analysis of Critical Transitions in Financial Networks. In International Conference and School on Network Science (pp. 47-59). Springer, Cham.

[22] Gidea, M.; Katz, Y. (2018). Topological data analysis of financial time series: Landscapes of crashes. Physica A: Statistical Mechanics and its Applications, Vol. 491, 820–834.

[23] Gonchenko, S.V.; Ovsyannikov, I.I.; Simó, C.; Turaev, D. (2005) Three-Dimensional Hénon-Like Maps and Wild Lorenz-Like Attractors. International Journal of Bifurcation and Chaos, Vol. 15, No. 11, 3493–3508.

[24] Guttal, V.; Jayaprakash, C. (2008). Changing skewness: an early warning signal of regime shifts in ecosystems. Ecology letters, 11(5), pp.450-460.

[25] Jain, A.K. (2010). Data clustering: 50 years beyond K-means. Pattern recognition letters, 31(8), 651–666.

[26] Kennel, M.B.; Brown, R.; Abarbanel, H.D. (1992). Determining embedding dimension for phase-space reconstruction using a geometrical construction. Physical review A, 45(6), p.3403.

[27] Livina V.N.; Lenton T.M. (2007). A modified method for detecting incipient bifurcations in a dynamical system. Geophys Res Lett 34.

[28] Lum, P.Y.; Singh, G.; Lehman, A.; Ishkanov, T.; Vejdemo-Johansson, M.; Alagappan, M.; Carlsson, J.; Carlsson, G. (2013) Extracting insights from the shape of complex data using topology. Scientific Reports, Vol. 3, 1236.

[29] Khasawneh, F.A.; Munch, E. (2016). Chatter detection in turning using persistent homology, Mechanical Systems and Signal Processing, Vol. 70-71, 527.

[30] Khasawneh, F.A.; Munch, E. (2018). Topological Data Analysis for True Step Detection in Piecewise Constant Signals. eprint arXiv:1805.06403

[31] Kozlowska, M.; Denys, M.; Wilinski, M.; Link, G.; Gubiec, T.; Werner, T.R.; Kutner, R.; Struzik, Z.R. (2016). Dynamic bifurcations on financial markets. Chaos, Solitons & Fractals, 88, 126–142.

[32] Livan, G; Inoue J. and Scalas, E (2012). On the non-stationarity of financial time series: impact on optimal portfolio selection. Journal of Statistical Mechanics: Theory and Experiment, Volume 2012, P07025.

[33] Kramar, M.; Levanger, R.; Tithof, J.; Suri, B.; Xu, M.; Paul, M.; Schatz, M.F.; Mischaikow, K. (2016). Analysis of Kolmogorov flow and Rayleigh-Benard convection using persistent homology. Physica D: Nonlinear Phenomena, Vol. 334, 82.

[34] Muldoon, M.R.; MacKay, R.S.; Huke, J.P.; Broomhead, D.S.(1993). Topology from time series. Physica D: Nonlinear Phenomena, Vol. 65, 1–16.

[35] Münnix, M. C., Shimada, T., Schfer, R., Leyvraz, F., Seligman, T. H., Guhr, T., & Stanley, H. E. (2012). Identifying States of a Financial Market. Scientific Reports, 2, 644. http://doi.org/10.1038/srep00644

[36] Nicolis, G.; Nicolis, C.; McKernan, D. (1993). Stochastic resonance in chaotic dynamics. Journal of Statistical Physics, Vol. 70, 125.

[37] Perea, J.A. and Harer, J. (2015). Sliding windows and persistence: An application of topological methods to signal analysis. Foundations of Computational Mathematics, 15(3), 799–838.

[38] Perea, J.A.; Deckard, A.; Haase, S.B.; Harer, J.(2015). SW1PerS: Sliding windows and 1- persistence scoring; discovering periodicity in gene expression time series data. BMC Bioinformatics, Vol. 16.

[39] Sauer, T.; Yorke, J. A.; Casdagli, M. (1991). Embedology. J. Statistical Physics 65:579.

[40] Scheffer, M., et al. (2009). Early-warning signals for critical transitions. Nature, 461.

[41] Scheffer, M.; Carpenter, S.R.; Lenton, T.M.; Bascompte, J.; Brock, W.; Dakos, V.; Van de Koppel, J.; Van de Leemput, I.A.; Levin, S.A.; Van Nes, E.H.; Pascual, M. (2012). Anticipating critical transitions. Science, 338(6105), pp.344-348.

[42] Seversky, L.M.; Davis, S.; Berger, M. (2016). On time-series topological data analysis: New data and opportunities. In Computer Vision and Pattern Recognition Workshops (CVPRW), 2016 IEEE Conference on 2016 Jun 26 (pp. 1014-1022). IEEE.

[43] Stark, J.; Broomhead, D.S.; Davies, M.E.; Huke, J. (2003). Delay embeddings for forced systems. II. Stochastic forcing. Journal of Nonlinear Science, 13(6), 519–577.

[44] Takens, F. (1981). Detecting strange attractors in turbulence. In *Dynamical systems and turbulence, Warwick 1980, Volume 898*, Lecture Notes in Mathematics, Volume 898, Springer-Verlag, 1981, 366–381.

[45] Truong, P. (2017). An exploration of topological properties of high-frequency one-dimensional financial time series data using TDA. KTH Royal Institute of Technology.