Applied Mathematics and Computation 430 (2022) 127237

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Cryptocurrency price analysis with ordinal partition networks

Zahra Shahriari[a], Fahimeh Nazarimehr[a], Karthikeyan Rajagopal[b], Sajad Jafari[a,c], Matjaž Perc[d,e,f,g,*], Milan Svetec[d]

[a] Department of Biomedical Engineering, Amirkabir University of Technology (Tehran polytechnic), Iran
[b] Centre for Nonlinear Systems, Chennai Institute of Technology, Chennai, India
[c] Health Technology Research Institute, Amirkabir University of Technology (Tehran polytechnic), Iran
[d] Faculty of Natural Sciences and Mathematics, University of Maribor, Koroška cesta 160, 2000 Maribor, Slovenia
[e] Department of Medical Research, China Medical University Hospital, China Medical University, Taichung 404332, Taiwan
[f] Alma Mater Europaea, Slovenska ulica 17, 2000 Maribor, Slovenia
[g] Complexity Science Hub Vienna, Josefstädterstraße 39, 1080 Vienna, Austria

ARTICLE INFO

Article history:
Received 20 March 2022
Revised 28 April 2022
Accepted 6 May 2022

Keywords:
Complex system
Nonlinear dynamics
Time series
Ordinal partition network
Cryptocurrency

ABSTRACT

The time series of cryptocurrency prices provide a unique window into their value and fluctuations. In this study, an ordinal partition network is constructed using the price signals, and its features are extracted to investigate the variations. Our research shows that the proposed method indeed works well for analyzing price fluctuations. We apply the method to ten digital coins, including Bitcoin, Binance coin, and XRP. In particular, the permutation entropy and clustering coefficient are investigated using the minimum, maximum, mean, and the geometric mean functions for the inbound, outbound, and loop directions. We find that the global clustering coefficient using the minimum function for triplets in a loop in one direction is the best measure in terms of predictive power and insight.

1. Introduction

Cryptocurrency trading has been a hot topic in recent years [1]. Among all cryptocurrencies, Bitcoin has attracted lots of attention and has the highest percentage of transactions [2–4]. Cryptocurrency is a decentralized medium of exchange that uses cryptographic functions to conduct financial transactions [5]. Many studies have been done on predicting the price of digital cryptocurrencies to analyze the relationship of the coin’s current, past, and future prices [6–8]. There are business reasons that cache the speculators’ attention to each of the currencies [9,10]. Studying the time series of the currencies can also help to understand changes in price trends and their efficiencies [11–13]. Due to the very complex and chaotic behavior of the cryptocurrency’s price [14], they can be analyzed using complex network methods. In this case, the complete information about the structure of the dynamical network which generates these signals is not available. So, the only way to obtain information about the system is to analyze the output time series.

Over time, various methods have been proposed for time series analysis [15,16]. Some methods such as the calculation of the correlation matrix [17], regression analysis of time series [18], singular spectrum analysis [19,20], stochastic neural networks [7], calculating the distribution of values in the time series [21], studying autocorrelation properties of the time series, and the variation of statistical properties [22] were introduced decades ago and are still used today. In recent years,

[*] Corresponding author.
E-mail address: matjaz.perc@gmail.com (M. Perc).

https://doi.org/10.1016/j.amc.2022.127237
0096-3003/© 2022 Elsevier Inc. All rights reserved.

much more advanced methods like deep learning techniques [23], multi-scale event synchronization [24], and Wavelet-based multi-scale analysis [25,26] have been proposed, providing more accurate and advanced time series analysis. The first step in using these methods is noise reduction to clean the signal. Various methods have been proposed for this purpose. For example, using iterative decomposition methods for creating an appropriate phase space from data [27], analysis with wavelet soft threshold technique [28], and least squares support vector machines (LS-SVM) in multidimensional recurrent vector [29]. On the other hand, studying complex systems using networks has been of great interest in recent years [30,31].

In this study, an ordinal partition network (OPN) [32] is generated using the time series to analyze the cryptocurrencies prices. It is a signal-based method and does not need the dynamical equations of the system or the graph structure. Recently weighted networks have attracted lots of interest [33,34]. So, many studies have been done on generalizing network measures to complex weighted networks [35,36]. The clustering coefficient is one of the crucial characteristics in complex network theory [37]. The clustering coefficient was initially developed for binary, undirected graphs. It has recently been extended to weighted, undirected networks [38]. Another network measure is permutation entropy [39,40].

In this paper, an OPN is generated from the cryptocurrency signals, and its features are investigated by clustering coefficient and permutation entropy. Various clustering coefficients and permutation entropy of this network are calculated to extract information about digital currencies' price and their changes. The remaining part of the paper is organized as follows. At first, the applied method for generating the OPN network is explained. Then, the measures to analyze this network are described. Section 3 gives the results of applying this method for Bitcoin and some other cryptocurrencies prices, and the paper is concluded in Section 4.

2. Material and methods

This section discusses a method for generating an OPN of a signal. Then, some network measures for comparing the networks are presented.

2.1. Ordinal partition network

In some cases, a hypothetical network is constructed to analyze a signal. Then the analysis is performed based on the network parameters. A network is a set of nodes and edges that demonstrate the relationship between any two nodes [41]. In real-world applications, the number of nodes in a dynamical system's graph is equal to the number of its time series, but the number of nodes can be chosen manually when an OPN is constructed from the signals [42]. Although this network is not similar to the original network that generated the signals (in terms of structure and the concept of network nodes and edges), it can provide helpful information to analyze them.

To generate an OPN, a window of length L is considered for sliding over the time series. In each window, m points of time series with a τ-points gap between every two consecutive selected points are considered to generate the network. Therefore, for each time series, a vector is obtained as 1, which is a sample of the windowed data $(i = 1, 2,)$.

$$z^{(i)} = (x_i, x_{i+\tau}, x_{i+(m-1)\tau}) \tag{1}$$

The window size equals $L = (m-1)\tau$. Using this vector, the related ordinal network can be mapped to a symbol:

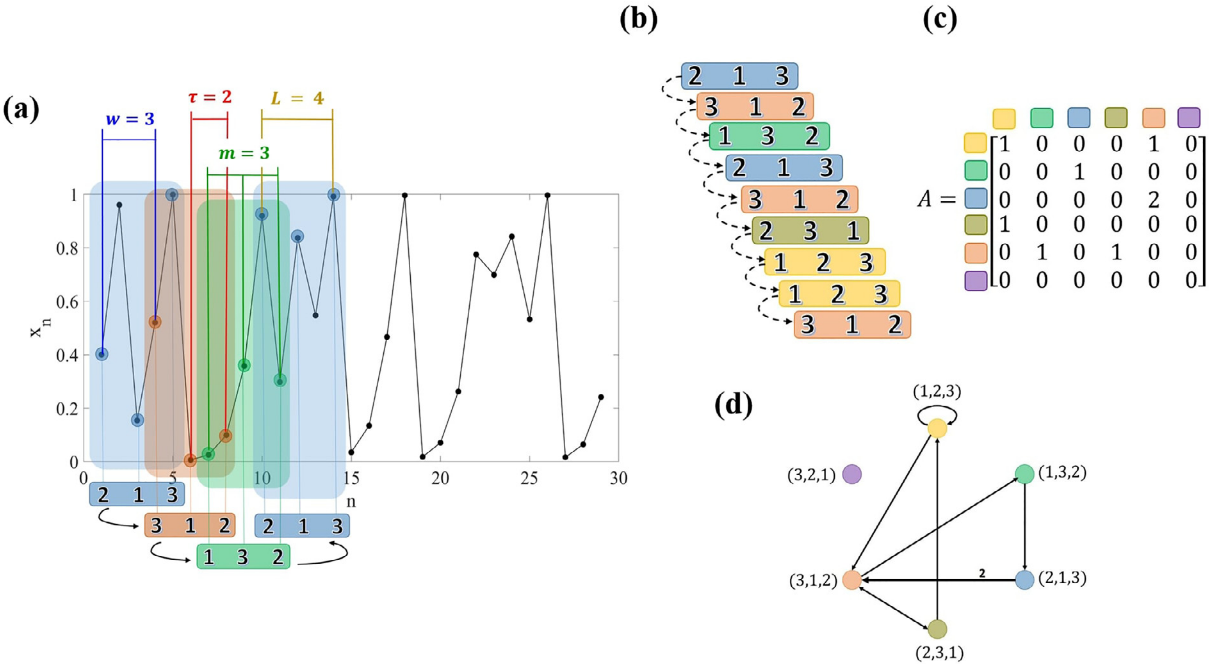
$$o^{(i)} = (\pi_1, \pi_2, \ldots, \pi_\omega) \tag{2}$$

where $o^{(i)}$ is the corresponding ordinal partition for each vector $z^{(i)}$, and each element of this vector displayed with the symbol π_k, such that $\pi_k \in \{1, 2, \ldots, m\}$, $\pi_k \neq \pi_l \iff k \neq l$ and $\pi_k < \pi_l \iff x_k > x_l \quad \forall x_k, x_l \in z_n$. By showing the non-overlapping points of each two windows with $w \in [1, L]$, it can be concluded that each time series with N points will be divided into $\left\lfloor \frac{N-(m-1)\tau-1}{w} \right\rfloor + 1$ widows. So the time series will have the highest number of windows for $w = 1$ and the lowest number of windows for $w = L$.

To construct a network from the time series, each node of the network is associated with the order of $z^{(i)}$ in the selected window. A directed connection between nodes for each transition from one state to another is considered. This network shows the changes between order patterns in the sampled time series. In this work, the structure presented in Fig. 1 is used to generate a network. In each window, only the relative magnitude of m points is considered. This method was first introduced by Small in 2013 [43]. By applying this method, the m points in each window are ordered in one of the $m!$ different possible states. Then sequential ordinal symbols are organized for each scalar time series, and a dynamical network is generated. The network structure is impressed by the number of sampled points in each window (m), the gap points (τ), and the overlap of windows ($L - w$).

Various methods have been proposed to analyze a dynamic network, such as calculating the minimum spanning tree with Kruskal [44], Prim [45], or Boruvka's [46] algorithm, which are mostly used in fuzzy lattices. Other methods are finding the shortest paths with Dijkstra [47] or similar algorithms [48]. Understanding the individual dynamics of the network's nodes and their collective behaviors is essential in all methods [22]. In this work, the clustering coefficient as one of the characteristics of the network is used to analyze the time series. Then permutation entropy is calculated to measure the complexity of the signal, and finally, the relationship between these two criteria is discussed.

Here, the opening price of ten different digital coins: Cardano (ADA), Bitcoin Cash (BCH), Binance Coin (BNB), Bitcoin (BTC), Dogecoin (DOGE), Polkadot (DOT), Ethereum (ETH), Litecoin (LTC), Uniswap (UNI), and XRP (XRP) from 1 October



$$A = \begin{array}{c|cccccc} & 🟨 & 🟩 & 🟦 & 🟫 & 🟧 & 🟪 \\ \hline 🟨 & 1 & 0 & 0 & 0 & 1 & 0 \\ 🟩 & 0 & 0 & 1 & 0 & 0 & 0 \\ 🟦 & 0 & 0 & 0 & 0 & 2 & 0 \\ 🟫 & 1 & 0 & 0 & 0 & 0 & 0 \\ 🟧 & 0 & 1 & 0 & 1 & 0 & 0 \\ 🟪 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Fig. 1. The process of constructing an OPN network using a sample time series; a) finding the ordinal pattern by the embedding vectors with dimension m, lag τ, the number of non-overlapping points of neighbor windows w, and the length of each window L; the ordinal patterns are found based on the amplitude rank of the selected samples; b) organizing ordinal symbols of the sequential windows; c) generating network with adjacency matrix A that each of its nodes is a unique ordinal pattern, and the edges are assigned between nodes based on the transitions between ordinal patterns in the time series; d) schematic of the constructed network;.

2020 to 31 March 2021 is investigated. The time resolution of the cryptocurrencies data is 1 second. All price data have been taken from https://www.binance.com. A weighted directed network is generated with six dynamical nodes from each time series. Then the networks are compared by calculating the clustering coefficient and permutation entropy of the designed network. These methods are discussed in the following subsection.

2.2. Measures

2.2.1. Clustering coefficient

A graph is represented by a set of nodes and edges. For an undirected unweighted graph with no self-loop, the local clustering coefficient for each node v_i is defined with the fraction of connected neighbors as follows Newman [49]:

$$c_i = \frac{t_i}{\frac{d_i(d_i-1)}{2}} = \frac{2t_i}{d_i(d_i-1)} \tag{3}$$

where t_i is the number of links between d_i neighbors of node v_i, d_i denotes the degree of node v_i. c_i, as the local clustering coefficient of node v_i, shows the ratio of the number of triangles (to which v_i belongs) to the number of connected triplets:

$$c_i = \frac{\Sigma\tau_\triangle}{\Sigma\tau} \tag{4}$$

$\Sigma\tau$ shows the total number of triplets and $\Sigma\tau_\triangle$ is a subset of $\Sigma\tau$ that close with an additional third edge. It can be said that a triple of nodes (v_j, v_i, v_k) is connected to v_i if v_j is connected to v_i, v_i is connected to v_k, and $j < k$. For a specific node v_i, $\frac{d_i(d_i-1)}{2}$ is the maximum number of possible interconnections among the neighbors of the node. A triangle is defined as a connected triple (v_j, v_i, v_k) in which v_j and v_k are connected. c_i is always in the interval [0,1]. It measures the level of cohesiveness around any given node or local interconnectedness of the network. If the neighbors of node i are not interconnected at all, then c_i will be equal to 0, while $c_i = 1$ expresses the case that all the neighbors are interconnected.

The network's global or average clustering coefficient is equal to the average of local clustering coefficients of all the nodes in the network, which is defined as Bagler [50]:

$$\bar{c} = \frac{1}{n}\sum_{v_i \in V} c_i \tag{5}$$

where V denotes the collection of all the nodes of the network.

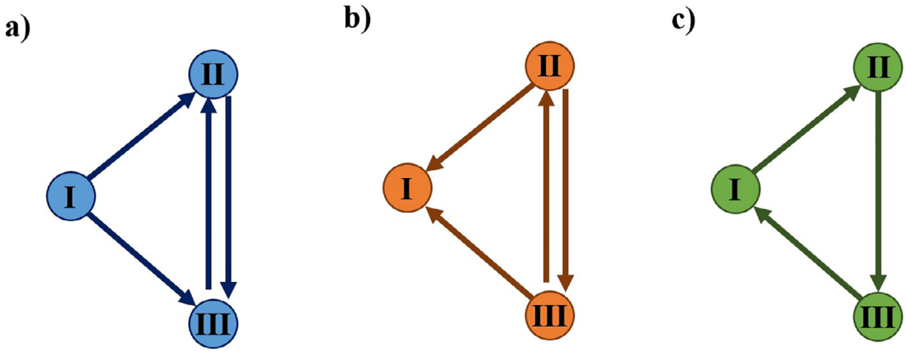


Fig. 2. Three methods to select triangles from triplets to calculate clustering coefficients: a) outbound; b) inbound; and c) loop direction; For the variation of cryptocurrency prices, the outbound clustering coefficient means that by changing prices from order (I) to orders (II) and (III), what is the possibility of moving directly between the two orders (II) and (III). In contrast, the input clustering coefficient shows that by varying the coin price from orders (II) and (III) to order (I), what is the possibility of moving directly between the two orders (II) and (III).

The presence of weights alters the definition of the standard clustering coefficient. The weighted clustering coefficient can be defined as follows:

$$C_\omega = \frac{total\, number\, of\, closed\, triplets}{total\, number\, of\, triplets} = \frac{\sum_{\tau_\triangle} \omega}{\sum_\tau \omega} \tag{6}$$

The method of considering the weight of the edges in calculating the clustering coefficient depends on the research problem. Depending on the issue, each of the functions of minimization, maximization, averaging, or geometric mean can be used to consider the weight in calculating the clustering coefficient of the network. The cost and benefits of these methods should be compared to choose a solution for the problem. For example, in the averaging method, the difference between the weights of the different edges in a triangle is not considered. Also, very heavy or very light edges can affect the results. The geometric averaging method is less sensitive than averaging to outliers and mitigates the effect of very large weights. Maximizing or minimizing are methods that consider extremes. Therefore, if the weight of one edge is too high or too low, the answer will be the same regardless of the weight of the other edges. These functions are selected according to the problem [51]. It should be noted that only if the minimization method is used, the weighted clustering coefficient remains in the interval $[0, 1]$. If other functions apply to the weight of the edges, depending on the weight of the third edge that results in the closed triangle, the clustering coefficient can be smaller or larger than 1.

On the other hand, the definition of the clustering coefficient can also be affected in the directed graph. In this case, the same as for the weight of a graph, the definition of the clustering coefficient depends on the research problem. One of the most common methods is calculating the clustering coefficient separately for the inbound or outbound directions. The main problem is choosing the method to consider the third side, which turns a triplet into a closed triangle (Fig. 2 a and b).

The most common way to calculate the clustering coefficient in directed networks is to consider both directions for closing the triangle. In this case, the clustering coefficient is defined as follows Clemente and Grassi [52]:

$$C_i^{in} = \frac{2}{k_i^{in}(k_i^{in} - 1)} \sum_{j,k} a_{ji} a_{ki} \frac{a_{ji} + a_{ki}}{2} \tag{7a}$$

$$C_i^{out} = \frac{2}{k_i^{out}(k_i^{out} - 1)} \sum_{j,k} a_{ji} a_{ki} \frac{a_{ji} + a_{ki}}{2} \tag{7b}$$

where C_i^{in} and C_i^{out} are inbound and outbound clustering coefficients, respectively. a_{ij} shows the number of links from the node v_i to node v_j. k_i^{in} and k_i^{out} represent in-degree and out-degree for edges pointing in and out respectively and are defined as follows:

$$k_i^{in} = \sum_{j=1}^{n} a_{ji} \tag{8a}$$

$$k_i^{out} = \sum_{j=1}^{n} a_{ij} \tag{8b}$$

Another way to calculate the clustering coefficient in directed graphs is to consider triplets with opposite directions and closed triangles in one order (Fig. 2 c). This method is represented for the first time in this study.

Put it in a nutshell, the following methods can be used to define the clustering coefficient in a weighted directional graph:

1. Minimization, maximization, averaging or geometric mean of weights for the inbound or outbound directions.
2. Counting the number of available paths to cross a triplet or triangle (using the minimum, maximum, average, or geometric mean function)

In this paper, some networks are generated that show the trend of price changes for digital coins. The nodes represent an ordering for the price changes, and the edges demonstrate a transition from one order to another. Considering Fig. 2a for the variation of cryptocurrency prices, the outbound clustering coefficient means that by changing prices from order (I) to orders (II) and (III), what is the possibility of moving directly between the two orders (II) and (III). Similarly, the input clustering coefficient (Fig. 2b) shows that by varying the coin price from orders (II) and (III) to order (I), what is the possibility of moving directly between the two orders (II) and (III). Each method of minimization, maximization, averaging, or the geometric mean of weights can be used in a specific problem. For instance, averaging function shows how many times it is switched between the two modes (II) and (III) on average.

On the other hand, minimization indicates the least number of times the price order switches between two different orders. Since the clustering coefficient indicates the degree of nodes association to a specific node, the self-loops have no role in its definition by using any functions and in any direction. Suppose the clustering coefficient is considered in a closed triangle in a loop direction. In that case, the probability of minimum maximum average geometric mean for returning paths to a specific order after passing through the other two orders is considered. This method properly predicts how likely a particular order will rise again after two stages. For example, calculating the minimum probability that the price will rise again after ascending for a while.

2.2.2. Permutation entropy

Permutation Entropy (PE) is a robust complexity measure for a dynamical system. It captures the permutation patterns and ordinal relations among the individual values of a given time series. Then, a probability distribution of the ordinal patterns is extracted. Ordinal patterns are used to reveal the hidden primary patterns of the system. This criterion is non-parametric. So, it can be calculated directly from the time series and does not require a parametric model. It is also robust against noise, making it significantly efficient in extracting the dynamic content of nonlinear time series [53]. The PE analysis started in one-dimensional time series. The PE can be computed by partitioning the time series into a matrix of overlapping column vectors and creating an OPN. In this way, the m-dimensional vectors are mapped into unique permutations of the ordinal rankings. Bandt and Pompe have defined the permutation entropy of a time series, that is, the Shannon entropy of the corresponding set of ordinal symbols s [54,55] :

$$h^{PE} = -\sum_{i=1}^{m!} p_i \log_2 p_i \tag{9}$$

where $p_i = P(S = s_i)$ shows the probability mass function for $s_i \in s$. s is the set of ordinal symbols in the symbolic dynamics S, and s_i is equal to the relative outbreak of each symbol. The P matrix is considered a stationary distribution of the Markov chain of S. So each element of the probability mass function could be defined as:

$$p_{i,j} = \frac{a_{i,j}}{\sum_j a_{i,j}} \tag{10}$$

where $a_{i,j}$ shows the number of transitions from state i to state j between the $m!$ states for each node. As a result, the probability mass function can be approximated as follows:

$$p_i \approx \frac{\sum_j a_{i,j}}{\sum_k \sum_j a_{k,j}}. \tag{11}$$

The PE measure can also be normalized such that:

$$h^{PE}_{norm} = -\frac{1}{\log_2 m!} \sum_{i=1}^{m!} p_i \log_2 p_i \tag{12}$$

which is restricted between 0 and 1. The maximum value for the permutation entropy ($h^{PE}_{norm} = 1$) occurs in a state where the distribution of the probability mass function is uniform, and the probability of occurrence of all events in the sample space is equal. On the other hand, the minimum value ($h^{PE}_{norm} = 0$) is when only one of the sample space events always occurs ($p_{i=j} = 1$), and the probability of other events occurring is zero ($p_{i\neq j} = 0$).

In this paper, the permutation entropy and the global clustering coefficient are calculated in different directions (inbound and outbound). Various functions (minimum, maximum, mean, and geometric mean) are applied for all the coins considering different window size values.

3. Results and discussions

In this research, the price of cryptocurrencies is used from the https://www.binance.com website. The cost of each coin is based on the price of Tether, which is always around $1. So, Tether is considered a reference for other digital currencies' prices in this work.

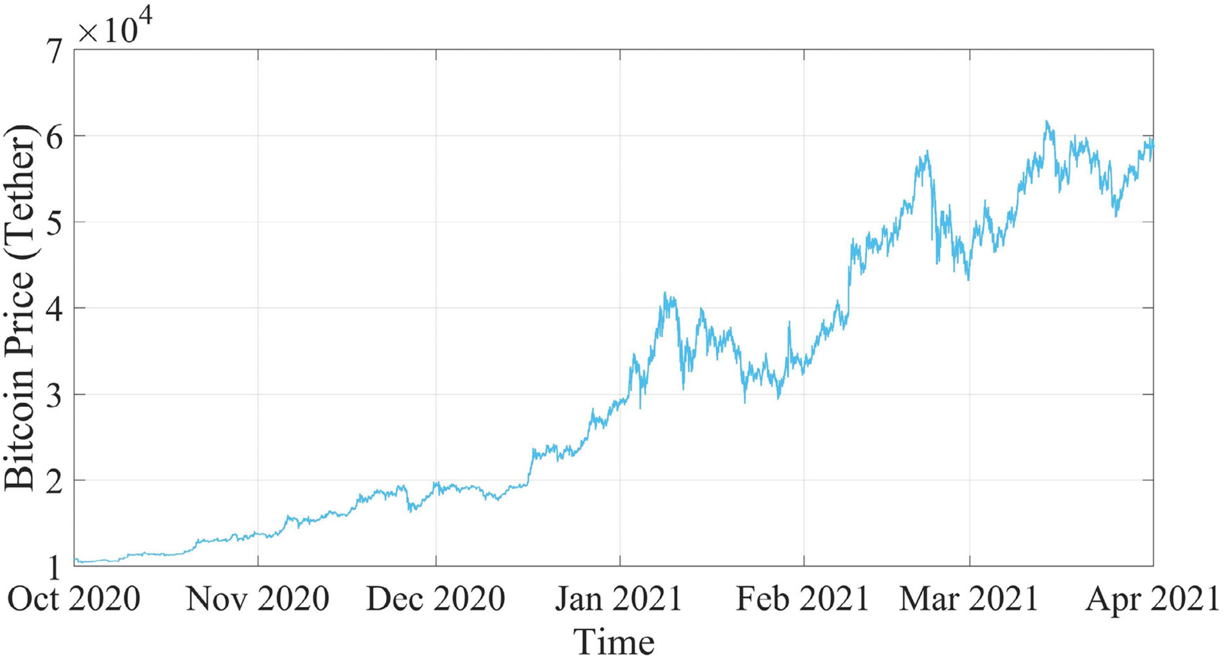


Fig. 3. Bitcoin price from October 2020 to the end of March 2021 in terms of Tether price; Fluctuations in this coin's price increase over time, and it generally has an upward trend during this period.

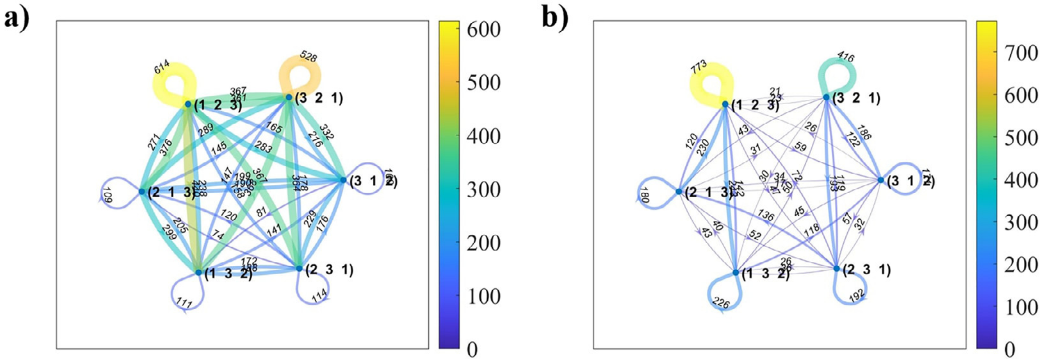


Fig. 4. The constructed weighted directed network of the Bitcoin signal from October 2020 to March 2021 a) with the small window $w = 30$, $\tau = 20$; b) with the large window $w = 60$, $\tau = 180$; The results show that the weight difference between the network's links increases by enlarging the window size, and the network becomes more heterogeneous.

3.1. Bitcoin signals

Fig. 3 shows the time series of Bitcoin prices over six months from the beginning of October 2020 to the end of March 2021 in terms of Tether. As can be seen in this picture, fluctuations in this coin's price increase over time, and it generally has an upward trend during this period.

Here, a network is generated that indicates the bitcoin price trend during this period using its time series. Parameters of OPN are important in the generated network. For example, if the window size is large, the minor fluctuations are not visible in the generated network. Conversely, choosing a small window for creating the network explains more details of price fluctuations. Here, among the three parameters for network formation (m, τ, and w), the number of points inside each window is constant $m = 3$. The variation of the other two parameters, the distance between the points inside each window (τ), and the number of non-overlapping points of the side windows (w) are studied.

The network of bitcoin price signals in these six months with two different sets of window size parameters is shown in Fig. 4. Part (a) of the figure shows the network where the parameters are set as $w = 30$ and $\tau = 20$. So the window size is relatively small. In this case, the window length ($L = (m-1)\tau$) is equal to 40 seconds, and the overlap of adjacent windows ($L - w$) is 10 seconds. In part (b) of the figure, the parameters are set as $w = 60$ and $\tau = 180$, making the window size larger and equal to 6 minutes. The overlap of adjacent windows is also assumed larger and equal to 5 minutes. The

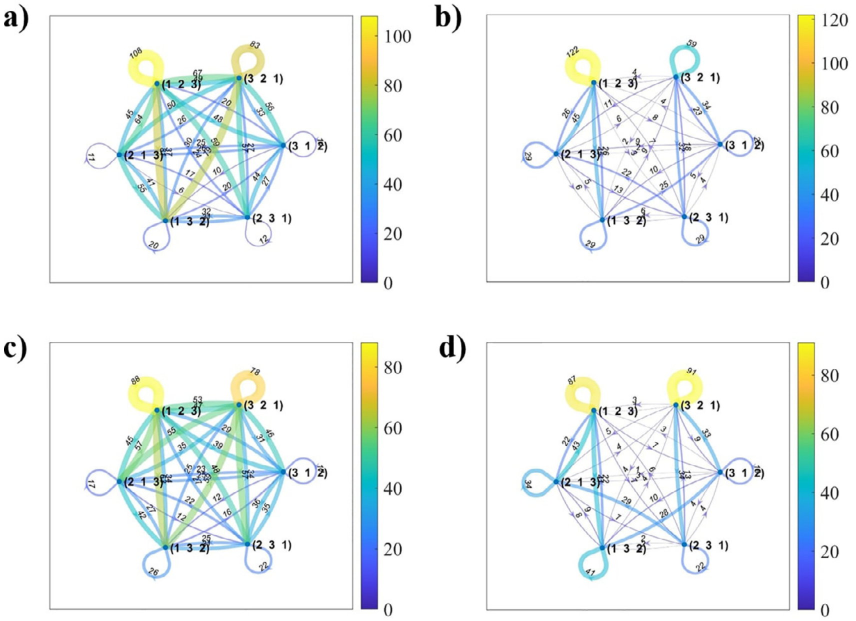


Fig. 5. The constructed weighted directed network from the Bitcoin time series for a) the price in November 2020 with a small window size; b) the price in November 2020 with a large window size; c) the price in February 2021 with a small window size; d) the price in February 2021 with a large window size; The results show that in both months, the network becomes more heterogeneous by enlarging the window size.

comparison between these two networks shows that the fluctuation details are more observable in part (a) of the figure. Part (b) only shows considerable fluctuations in the time series ranking. It means that as the size of the window increases, the information about the details of the time series' fluctuation is lost. Also, the number of network links is reduced by increasing the parameter w. On the other hand, in Fig. 4 b, the percentage of self-loops of node (1, 2, 3) to the total number of links is more significant than Fig. 4 a. This difference is because of the large window size selected in Fig. 4 b. The network links represent the general ranking of the signal, not the details. It causes the weight of the network links to become more unbalanced by enlarging the window size. Therefore, the limited number of links has more weight. As described earlier, this causes a decrease in the permutation entropy of the network. The permutation entropy of the network in Fig. 4 a is 0.9791, and in Fig. 4 b, it is equal to 0.9678, which means that a slight difference in the entropy shows a considerable difference in network balance. In the rest of the paper, the same two sets of parameters are used to illustrate the large and small window sizes. So, the small window is $w = 30$ and $\tau = 20$, and the large window is $w = 60$ and $\tau = 180$.

By generating the OPN for the time series of one month, similar variations can be observed in the results of small and large window sizes networks. Fig. 5 shows the generated networks of the Bitcoin price signals in November 2020 (a, b) and February 2021 (c, d) with different window sizes. The network presents fewer details in both months by enlarging the window size (b, d). Increasing the window size increases the ratio of self-loops with ascending (1,2,3) or descending (3,2,1) trend to the total number of links. There are also differences between the networks of different months with the same parameters. For instance, the ratio of loops on the uptrend node to the total links in November is higher than in February. It shows that compared to February, the price of bitcoin has been rising more often in November, which was also shown in Fig 3. Price changes had an upward trend for almost the entire month of November, but in the second half of February, the downward trend in prices can be seen. In this part, a decrease in permutation entropy can also be seen with the inhomogeneity of the network of each month. The value of the permutation entropy In Fig. 5 a is 0.9750, in Fig. 5 b is 0.9678, in Fig. 5 c is 0.9818, and Fig. 5 d is 0.9729. In both the November and February months, a decrease in permutation entropy is observable by enlarging the window size. Also, it can be understood that, in general, the network is more homogenous in February 2021 than in November 2020.

To study the variations of the Bitcoin price signal, the time series are divided for each week. The network is generated for each week's signal, and some measures are used to compare the various week's networks. Fig. 6 shows the permutation entropy and global clustering coefficients calculated with different methods. The minimum, maximum, mean, and geometric mean functions for the inbound, outbound, and loop directions have been used to calculate the network's global clustering coefficients. The results show that the global clustering coefficients of each function for the inbound and outbound directions are very similar. According to the definition of these variables, this fact seemed predictable. In the description of inbound and outbound clustering coefficients, the weight of the third link, which closes a triplet and converts it to a triangle, was considered in both directions. However, in the definition of loop clustering coefficients, the direction of the third link is considered only if it forms a directional ring. The window size is crucial in network generation. Due to the definition of global

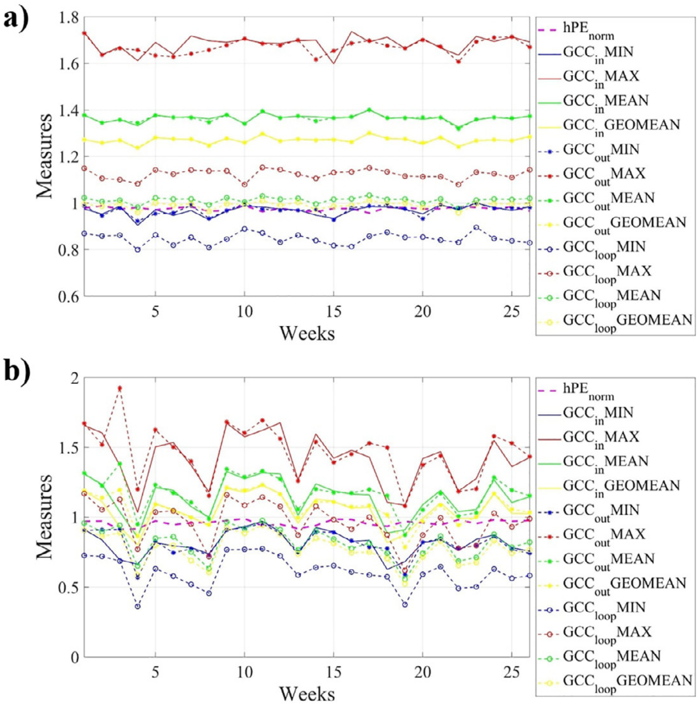


Fig. 6. Normalized permutation entropy and global clustering coefficients with the minimum, maximum, mean, and geometric mean functions for the inbound, outbound, and loop directions for the constructed network of bitcoin time series by changing week for a) small window size; b) large window size; The results show that, in general, increasing the window size causes an increment in the variance of the used criteria for studying the network.

clustering coefficient with the minimum function, its results are always less than one. Since the normalized permutation entropy (which is always between 0 and 1) is used for the network analysis in this study, this can be an advantage for using the minimum function in calculating the clustering coefficient that both are in the same range. It can also be seen that by enlarging the window size, the average of the permutation entropy of the network decreases. In Fig. 6a, the amount of permutation entropy varies in the interval [0.9568, 0.9894], and its average is equal to 0.9753. However, in Fig. 6b, this quantity changes between 0.9132 and 0.9894, and on average, it is equivalent to 0.9582. It means that by varying the window size, the network homogeneity is changed and causes changes in the permutation entropy.

Comparing Figs. 3 and 6 shows that when the price of Bitcoin has increased with a sharp slope, the quantity of the minimum clustering coefficient increases. For example, in Fig. 6, in both parts (a) and (b) with the small and large window sizes, it can be seen that in the 5th, 9th, and 23rd week, the minimum clustering coefficient obtained in all directions increased significantly. It is a unique feature of the clustering coefficient obtained with the minimum function. The zoomed view of the time series of these weeks is shown in Fig. 7. Comparison of parts (a) and (b) of Fig. 6 indicates that as the size of the window increases, the variance of the clustering coefficients and the variance of the permutation entropy increase sharply. Expanding the window size can also reduce the magnitude of clustering coefficients. In addition, it can be seen that increasing the length of the window reduces the difference between different clustering coefficients by various algorithms. In addition, the trend of changes and their fluctuations become more similar.

In Fig. 8, a network is generated for each month's signal, and the above measures are calculated for each of them. In general, the charts showing the measures for months are more uniform with minor fluctuations than the weeks' charts. It shows that due to the more occurrence of trend changing with increasing the length of the time series, the difference in the magnitude of clustering coefficients and permutation entropy of various networks decreases in different months. Again, it can be seen that in minimum and geometric mean functions, increasing the window size increases the variance of the criteria changes. The changes in the clustering coefficient obtained using the minimum function can be a good indicator for the severe changes in the bitcoin price.

3.2. Different coins

In this section, the OPNs are generated for the top 10 of the most-selling cryptocurrencies in the six months from the beginning of October 2020 to the beginning of April 2021 by small ($w = 30, \tau = 20$) and large ($w = 60, \tau = 180$) window sizes. Fig. 9 shows the normalized permutation entropy and clustering coefficients with minimum function in various directions in parts (a) and (b) and the maximum clustering coefficient in parts (c) and (d) for these networks. The values of the minimum clustering coefficients and the normalized permutation entropy are always less than or equal to 1. The values of

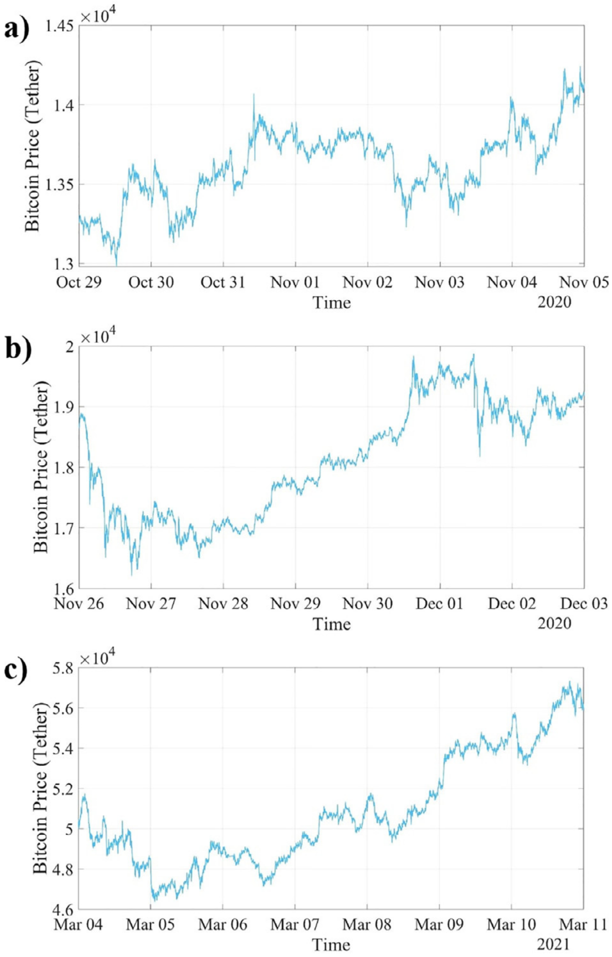


Fig. 7. Zoomed view of time series of bitcoin price in a) 5th-week. b) 9th-week; c) 23rd-week. An increase in the price of bitcoin can be seen in all of these weeks.

the maximum clustering coefficients are always greater than or equal to 1. Furthermore, for all cryptocurrencies, the value of the minimum and maximum clustering coefficients in the loop is always smaller than the inbound and outbound directions. Also, the difference between the maximum clustering coefficient for the loop and the maximum inbound and outbound clustering coefficients is more significant than the difference between the minimum clustering coefficient for the loop and the minimum inbound and outbound clustering coefficient. The reason is that the weight of the third link that closes the triangle is considered in only one direction in the loop method. In the minimum and maximum cases, the sum of the two sides is considered as the weight, which can be seen in calculating the clustering coefficient with the maximum function.

Part (a) of Fig. 9 shows that in the small window size, the value of the outbound minimum clustering coefficient is greater than the inbound minimum clustering coefficient and the network permutation entropy for all cryptocurrencies. While in part (b), for large window size, this phenomenon is quite vice versa. The outbound minimum clustering coefficient value is always less than the inbound minimum clustering coefficient and the network permutation entropy for all cryptocurrencies. Although the increase in window size causes a reduction in the quantity of all minimum clustering coefficients for all currencies, the entropy of the network does not change significantly, and the trend of changes is different for various cryptocurrencies. In both parts (a) and (b) of Fig. 9, DOGE has the highest permutation entropy compared to other cryptocurrencies. It means that the time series of this coin has more homogeneous fluctuations and does not have an ascending or descending trend in a long time. A comparison of parts (c) and (d) shows that increasing the window size for the network construction reduces the difference in the inbound and outbound maximum clustering coefficients. Unlike the minimum clustering coefficient, the maximum clustering coefficient values increase for all currencies by expanding the window size.

Fig. 10 shows the global clustering coefficients with mean and geometric mean functions in all directions for the same networks used in Fig. 9. The calculated clustering coefficients with mean and geometric mean functions depend much on

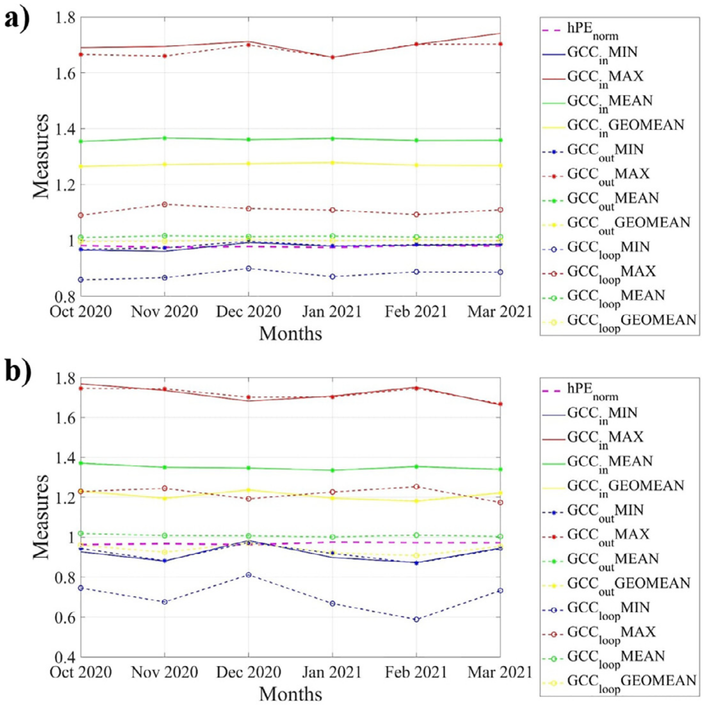


Fig. 8. Normalized permutation entropy and global clustering coefficients with the minimum, maximum, mean, and geometric mean functions for the inbound, outbound, and loop directions for the constructed network from bitcoin time series in each month for a) small window size; b) large window size; The results show that the criteria' variances for months are much minor than the weeks.

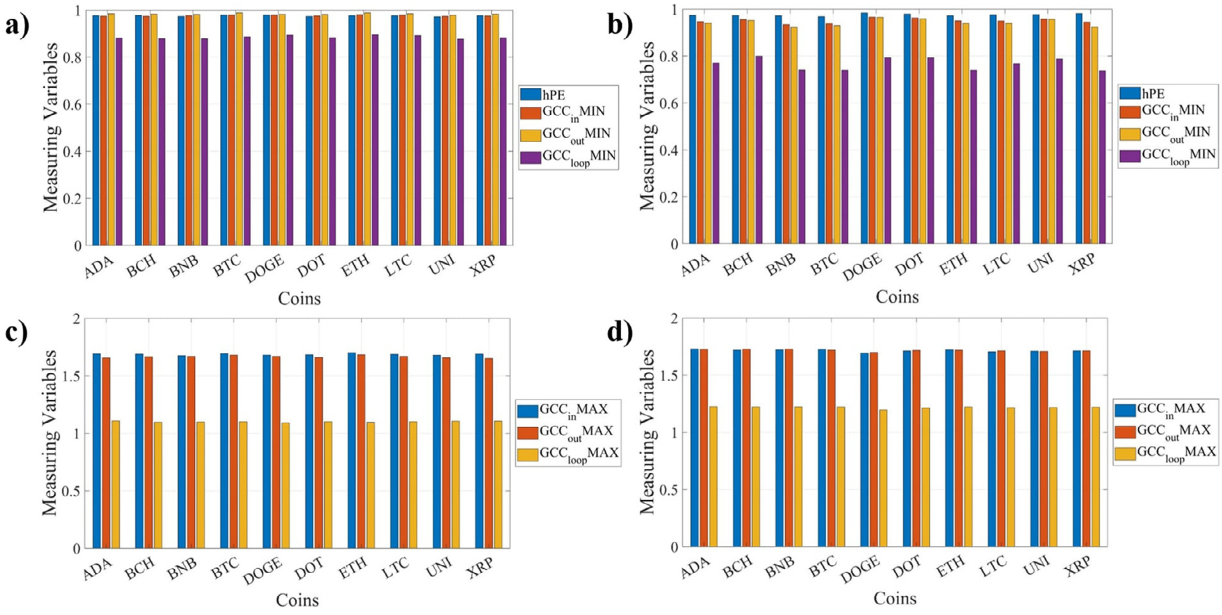


Fig. 9. Different measures of the 10 top of the most-selling cryptocurrencies' price from October 2020 to the end of March 2021; a) permutation entropy and global clustering coefficients with minimum function in all directions for the network with small window size; b) permutation entropy and global clustering coefficients with minimum function in all directions for the network with large window size; c) global clustering coefficients with maximum function in all directions for the network with small window size; d) global clustering coefficients with maximum function in all directions for the network with large window size; The results show that the changes in various criteria with minimum and maximum functions in inbound and outbound directions are different with increasing window size.

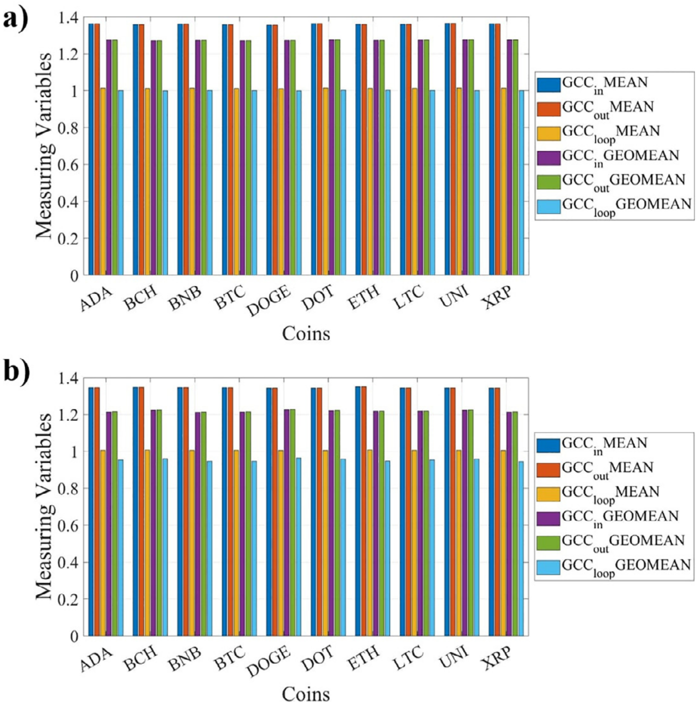


Fig. 10. The global clustering coefficients with mean and geometric mean functions in all directions for the network of the 10 top of the most-selling cryptocurrencies' price from October 2020 to the end of March 2021 with a) small window size; b) large window size; The results show the dependency of the calculated clustering coefficients with mean and geometric mean functions on window length.

the window length used to generate the network. The calculated clustering coefficients with geometric mean functions rely more on the window length used to generate the network than cases with the mean function.

Summarizing the results of Figs. 6, 8, and 9, the best benchmark for analyzing these networks and specifying the differences between them is the clustering coefficient calculated with the minimum function and in the form of a loop. It can also be concluded that by considering a large window for network generation and ignoring the small details of signal fluctuations, these values can be more appropriate criteria for analyzing networks.

Figs. 11 and 12 show the same procedure used in Fig. 9 if only a specific part of the time series is used to generate the network. Fig. 11 presents the permutation entropy and calculated clustering coefficients with minimum function for November 2020 and February 2021, and Fig. 12 shows calculated clustering coefficients with maximum function for the same networks. These figures show that although the variance of the calculated measures for different coins increases with the decrement of the time series length, the main features of each measure are almost the same. It shows that the measures which are mentioned above can be used in different situations for different signal lengths.

Fig. 13 shows an example of comparing two parts of the time series for two different coins. BNB and XRP prices in November 2020 are considered for this comparison. In parts (a) and (b), the time series of BNB and XRP prices are presented. The price of BNB is constantly fluctuating during this period, while XRP's price has an upward trend for most of this time and its fluctuations are less than BNB. Parts (c) and (d) show the OPN networks constructed from the time series. It shows that the percentage of self-loops on the upward node (1, 2, 3) for XRP (127) is more than BNB (100). Parts (e) and (f) represent the calculated measures for the OPN networks. They illustrate that in the second week, when the prices of both cryptocurrencies are almost stable, it seems that all of the ranking states occur almost equal. So, a decrease in the quantity of all clustering coefficients can be seen compared to the first week. PEs for the first and second weeks are 0.9564,0.9371 for BNB, and 0.9849,0.9660 for XRP coins, respectively. So, a decrease in the PE can be observed in the second week compared to the first one.

In this study, the proposed method was investigated on 10 different cryptocurrencies, and in all cases, the results were very effective for analyzing the time series of cryptocurrencies' prices. In most parts of the paper, the main focus has been on bitcoin to present the results. Then, to show how this method works on other cryptocurrencies, an example of XRP and BNB price analysis has been presented in Fig. 13. A summary of the work on the other cryptocurrencies was shown in Figs. 9 to 12. In this study, the focus was only on the price analysis of cryptocurrencies. In future works, we plan to apply this method on other financial instruments such as stocks or bonds to check its efficiency.

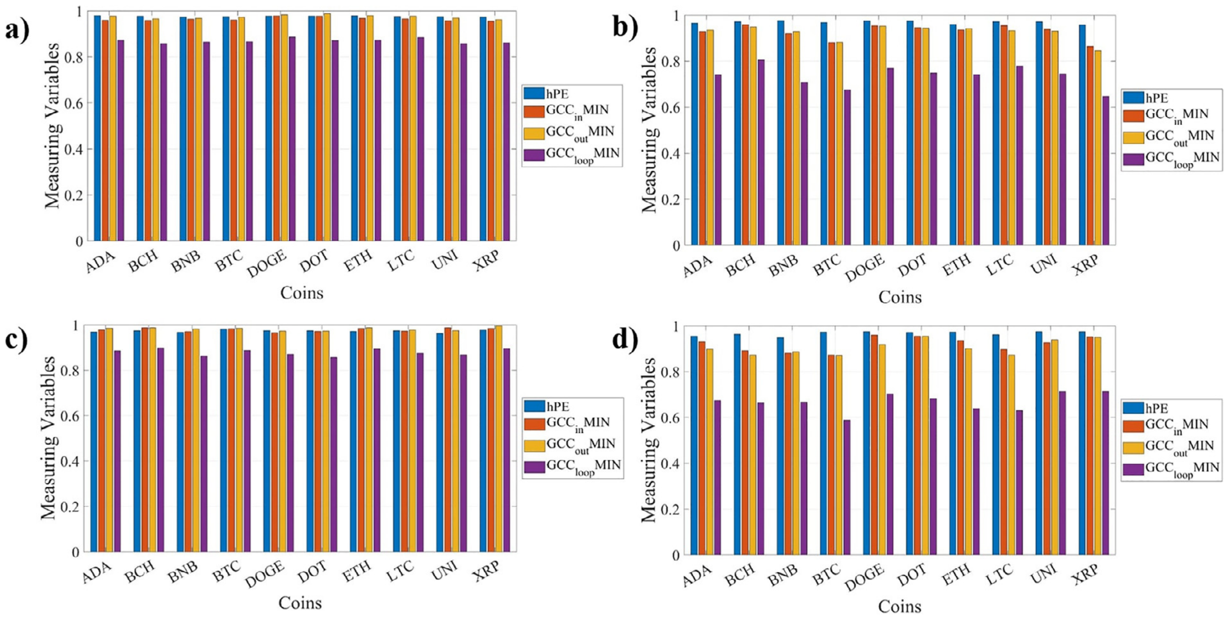


Fig. 11. Permutation entropy and global clustering coefficients with minimum function in all directions for the network of the 10 top of the most-selling cryptocurrencies' price; a) in November 2020 with small window size; b) in November 2020 with large window size; c) in February 2021 with small window size; d) in February 2021 with large window size; The results show that the variance of the calculated measures increases with the decrement of the time series length.

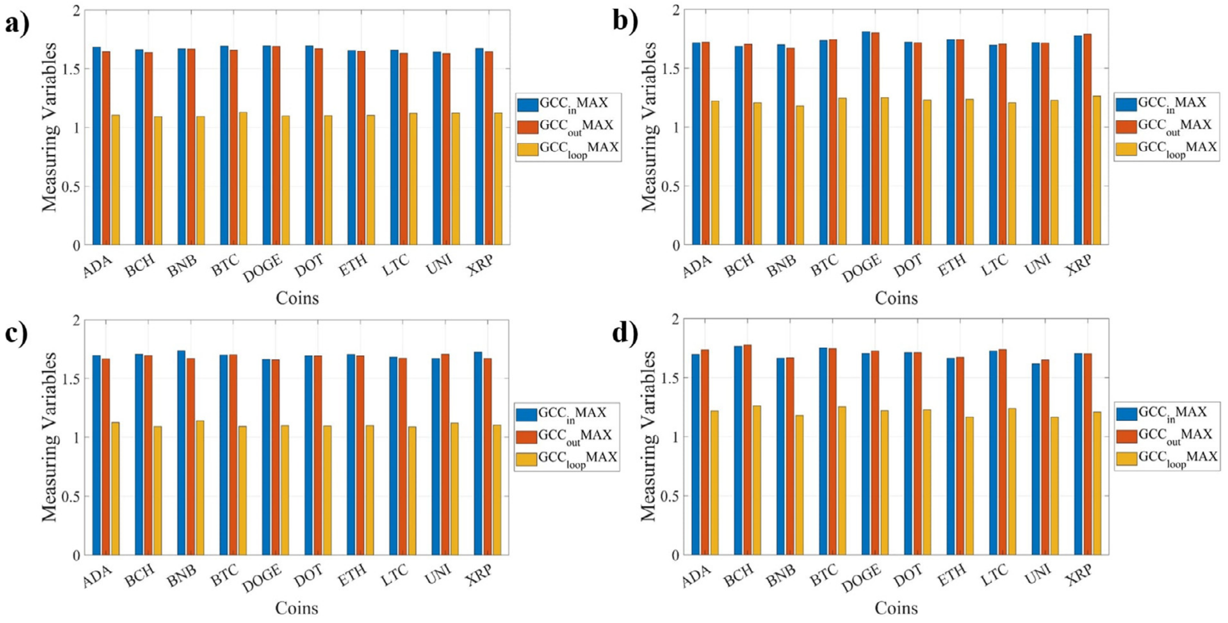


Fig. 12. Permutation entropy and global clustering coefficients with minimum function in all directions for the network of the 10 top of the most-selling cryptocurrencies' price; a) in November 2020 with small window size; b) in November 2020 with large window size; c) in February 2021 with small window size; d) in February 2021 with large window size; The results show that the variance of the calculated measures increases with the decrement of the time series length.

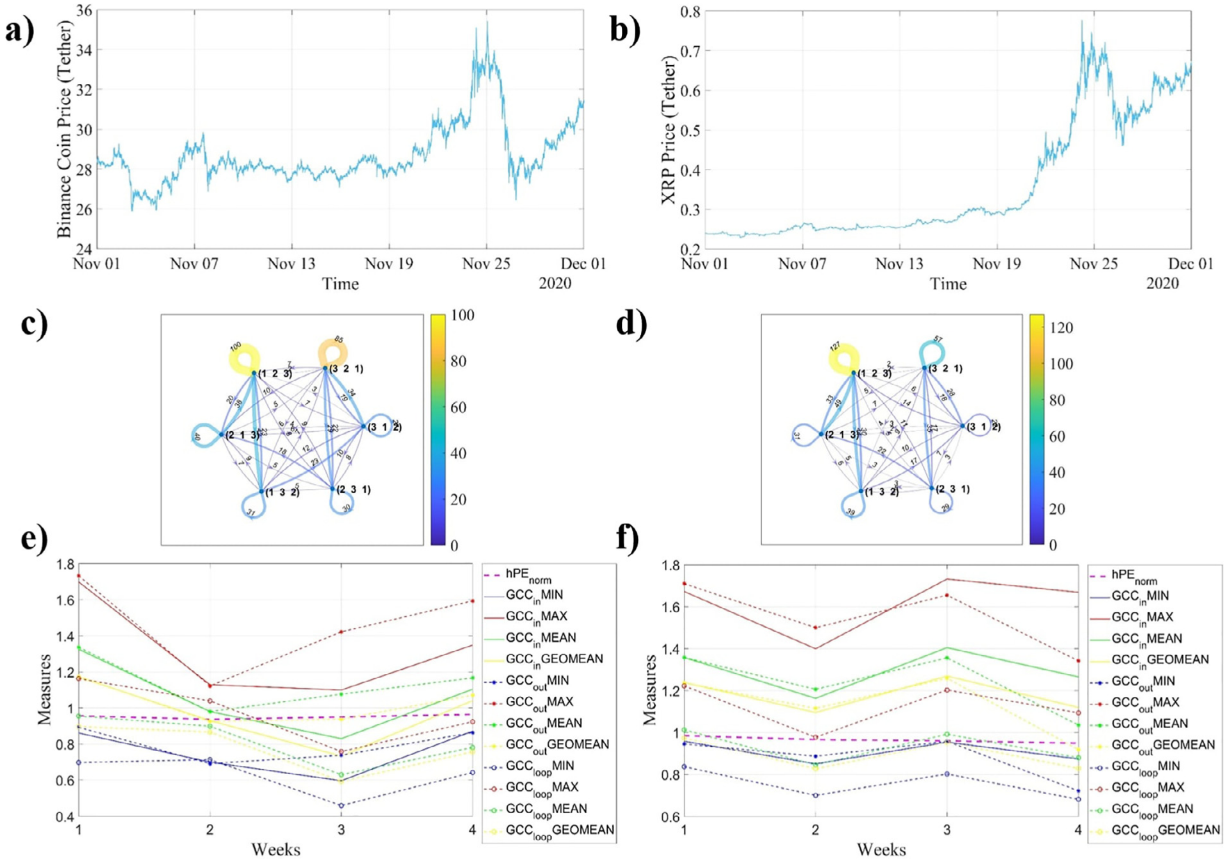


Fig. 13. A comparison between time series, networks, and calculated measures of BNB and XRP cryptocurrencies prices in November 2020; a) time series of BNB cryptocurrency in November 2020; b) Time series of XRP cryptocurrency in November 2020; c) the constructed network from the time series of part (a) with a large window; d) the constructed network from the time series of part (b) by using a large window; e) calculated measures for the network of BNB cryptocurrency prices in each week of November 2020; f) calculated measures for the network of XRP cryptocurrency prices in each week of November 2020;.

4. Conclusion

In this paper, some measures for analyzing cryptocurrency signals were discussed by generating ordinal partition networks. These measures included permutation entropy and clustering coefficient of the network by using the minimum, maximum, mean, and geometric mean functions for the inbound, outbound, or loop directions. Using these methods, the fluctuations of the cryptocurrency signals were analyzed in various periods. The permutation entropy was an excellent measure to represent the homogeneity of the generated network. Also, global clustering coefficients were exciting criteria to analyze the cryptocurrency signals. The results indicated that for cryptocurrency price analysis, the best measure is the global clustering coefficient calculated by using the minimum function for triplets that form a loop in one direction.

Moreover, the consideration of appropriate window size was an essential subject for obtaining an optimal outcome. This study can be applied to analyze many different signals like biological data such as ECG and EEG or atmospheric data such as temperature, pressure, and wind, in cases with insufficient knowledge about the complex network that generates these signals. In future works, we will try to predict the variations of the cryptocurrencies using their analyzed trends and networks.

Data Availability Statement

The codes and the data used in this paper can be found in the following link:
https://drive.google.com/drive/folders/1XmzdfdZvRktuh3HbQpygdzBShhRL6H4r?usp=sharing.

References

[1] I. Makarov, A. Schoar, Trading and arbitrage in cryptocurrency markets, J. Financ. Econ. 135 (2) (2020) 293–319.
[2] A. Urquhart, The inefficiency of bitcoin, Econ. Lett. 148 (2016) 80–82.
[3] S. Nadarajah, J. Chu, On the inefficiency of bitcoin, Econ. Lett. 150 (2017) 6–9.

[4] R. Böhme, N. Christin, B. Edelman, T. Moore, J.E. Perspect, Bitcoin: Economics, technology, and governance, 2015, 29, 2, 213–38
[5] G. Hileman, M. Rauchs, Global cryptocurrency benchmarking study, Cambridge Centre Alternat. Finance 33 (2017) 33–113.
[6] J. Abraham, D. Higdon, J. Nelson, J. Ibarra, Cryptocurrency price prediction using tweet volumes and sentiment analysis, SMU Data Sci. Rev. 1 (3) (2018) 1.
[7] P. Jay, V. Kalariya, P. Parmar, S. Tanwar, N. Kumar, M. Alazab, Stochastic neural networks for cryptocurrency price prediction, IEEE Access 8 (2020) 82804–82818.
[8] M.M. Patel, S. Tanwar, R. Gupta, N. Kumar, A deep learning-based cryptocurrency price prediction scheme for financial institutions, J. Inf. Secur. Appl. 55 (2020) 102583.
[9] C. Lamon, E. Nielsen, E. Redondo, Cryptocurrency price prediction using news and social media sentiment, SMU Data Sci. Rev. 1 (3) (2017) 1–22.
[10] I.M. Sifat, A. Mohamad, M.S.B.M. Shariff, Lead-lag relationship between bitcoin and ethereum: evidence from hourly and daily data, Res. Int. Bus. Finance 50 (2019) 306–321.
[11] A. Dutta, S. Kumar, M. Basu, A gated recurrent unit approach to bitcoin price prediction, J. Risk Financ. Manage. 13 (2) (2020) 23.
[12] H.Y. Sigaki, M. Perc, H.V. Ribeiro, Clustering patterns in efficiency and the coming-of-age of the cryptocurrency market, Sci. Rep. 9 (1) (2019) 1440.
[13] L.G. Alves, H.Y. Sigaki, M. Perc, H.V. Ribeiro, Collective dynamics of stock market efficiency, Sci. Rep. 10 (1) (2020) 21992.
[14] S. Lahmiri, S. Bekiros, Cryptocurrency forecasting with deep learning chaotic neural networks, Chaos, Solitons Fractals 118 (2019) 35–40.
[15] L. Chen, J. Sun, K. Li, Q. Li, Research on the effectiveness of monitoring mechanism for æyield to pedestriang based on system dynamics, Physica A 591 (2022) 126804.
[16] K. Li, H. Xu, X. Liu, Analysis and visualization of accidents severity based on lightgbm-tpe, chaos, Solitons Fractals 157 (2022) 111987.
[17] Y. Yang, H. Yang, Complex network-based time series analysis, Physica A 387 (5–6) (2008) 1381–1386.
[18] E. Parzen, An approach to time series analysis, Ann. Math. Stat. 32 (4) (1961) 951–989.
[19] N. Golyandina, Particularities and commonalities of singular spectrum analysis as a method of time series analysis and signal processing, Wiley Interdiscip. Rev. Comput. Stat. 12 (4) (2020). E1487
[20] R. Vautard, M. Ghil, Singular spectrum analysis in nonlinear dynamics, with applications to paleoclimatic time series, Physica D 35 (3) (1989) 395–424.
[21] S. Aminikhanghahi, D.J. Cook, A survey of methods for time series change point detection, Knowl. Inf. Syst. 51 (2) (2017) 339–367.
[22] M.B. Shrestha, G.R. Bhatta, Selecting appropriate methodological framework for time series data analysis, J. Finance Data Sci. 4 (2) (2018) 71–89.
[23] P. Lara-Benítez, M. Carranza-García, J.C. Riquelme, An experimental review on deep learning architectures for time series forecasting, Int. J. Neural Syst. 31 (03) (2021) 2130001.
[24] J. Kurths, A. Agarwal, R. Shukla, N. Marwan, M. Rathinasamy, L. Caesar, R. Krishnan, B. Merz, Unravelling the spatial diversity of indian precipitation teleconnections via a non-linear multi-scale approach, Nonlinear Process. Geophys. 26 (3) (2019) 251–266.
[25] C.M. Chou, Wavelet-based multi-scale entropy analysis of complex rainfall time series, Entropy 13 (1) (2011) 241–253.
[26] A. Agarwal, R. Maheswaran, N. Marwan, L. Caesar, J. Kurths, Wavelet-based multiscale similarity measure for complex networks, Eur. Phys. J. B 91 (11) (2018) 296.
[27] J. Lardiès, Modal parameter identification by an iterative approach and by the state space model, Mech. Syst. Sig. Process. 95 (2017) 239–251.
[28] J. Hou, P. Guo, A. Xu, D. Ning, S. Chen, Z. Zhang, Y. Gong, Y. Chen, H. Tian, H. Du, An adaptive noise reduction method based on improved dislocation superposition method for abnormal noise fault component of automotive engine, Shock Vib. 2021 (2021) 6998232.
[29] J. Sun, C. Zheng, Y. Zhou, Y. Bai, J. Luo, Nonlinear noise reduction of chaotic time series based on multidimensional recurrent ls-svm, Neurocomputing 71 (16–18) (2008) 3675–3679.
[30] A.A. Pessa, R.S. Zola, M. Perc, H.V. Ribeiro, Determining liquid crystal properties with ordinal networks and machine learning, Chaos, Solitons Fractals 154 (2022) 111607.
[31] A. Agarwal, L. Caesar, N. Marwan, R. Maheswaran, B. Merz, J. Kurths, Network-based identification and characterization of teleconnections on different scales, Sci. Rep. 9 (1) (2019) 8808.
[32] K. Sakellariou, T. Stemler, M. Small, Markov modeling via ordinal partitions: an alternative paradigm for network-based time-series analysis, Phys. Rev. E 100 (6) (2019) 062307.
[33] J.-H. Li, J.-L. Liu, Z.-G. Yu, B.-G. Li, D.W. Huang, Synchronizability analysis of three kinds of dynamical weighted fractal networks, Int. J. Mod. Phys. B 35 (23) (2021) 2150243.
[34] X. Yao, Y. Liu, Z. Zhang, W. Wan, Synchronization rather than finite-time synchronization results of fractional-order multi-weighted complex networks, IEEE Trans. Neural Netw. Learn. Syst. (2021) 1–12.
[35] S. Srinivasan, C. Dickens, E. Augustine, G. Farnadi, L. Getoor, A taxonomy of weight learning methods for statistical relational learning, Mach. Learn. (2021) 1–40.
[36] T. Opsahl, F. Agneessens, J. Skvoretz, Node centrality in weighted networks: generalizing degree and shortest paths, Soc. Netw. 32 (3) (2010) 245–251.
[37] J. Saramäki, M. Kivelä, J.-P. Onnela, K. Kaski, J. Kertesz, Generalizations of the clustering coefficient to weighted complex networks, Phys. Rev. E 75 (2) (2007) 027105.
[38] G. Fagiolo, Clustering in complex directed networks, Phys. Rev. E 76 (2) (2007) 026107.
[39] C. Bandt, B. Pompe, Permutation entropy: a natural complexity measure for time series, Phys. Rev. Lett. 88 (17) (2002) 174102.
[40] Y. Cao, W.-w. Tung, J. Gao, V.A. Protopopescu, L.M. Hively, Detecting dynamical changes in time series using the permutation entropy, Phys. Rev. E 70 (4) (2004) 046217.
[41] M. Li, Y. Fan, J. Chen, L. Gao, Z. Di, J. Wu, Weighted networks of scientific communication: the measurement and topological role of weight, Physica A 350 (2–4) (2005) 643–656.
[42] J. Zhang, J. Zhou, M. Tang, H. Guo, M. Small, Y. Zou, Constructing ordinal partition transition networks from multivariate time series, Sci. Rep. 7 (1) (2017) 1–13.
[43] M. Small, Complex Networks from Time Series: capturing Dynamics, in: 2013 IEEE International Symposium on Circuits and Systems (ISCAS), IEEE, 2013, pp. 2509–2512.
[44] B. Munier, M. Aleem, M.A. Islam, M.A. Iqbal, W. Mehmood, A fast implementation of minimum spanning tree method and applying it to kruskals and prims algorithms, Sukkur IBA j. Comput. Math. Sci. 1 (1) (2017) 58–66.
[45] W. Wamiliana, A. Asmiati, M. Usman, A. Hijriani, W.C. Hastono, Comparative analysis of some modified prims algorithms to solve the multiperiod degree constrained minimum spanning tree problem, Indian J. Sci. Technol. 11 (11) (2018) 1–6.
[46] O. Swathika, S. Hemamalini, Graph theory and optimization algorithms aided adaptive protection in reconfigurable microgrid, J. Electr. Eng. Technol. 15 (1) (2020) 421–431.
[47] M. Akram, A. Habib, J.C.R. Alcantud, An optimization study based on dijkstra algorithm for a network with trapezoidal picture fuzzy numbers, Neural Comput. Appl. 33 (4) (2021) 1329–1342.
[48] F. Sapundzhi, M. Popstoilov, Optimization algorithms for finding the shortest paths, Bulg. Chem. Commun. 50 (2018) 115–120.
[49] M.E. Newman, The structure and function of complex networks, SIAM Rev. 45 (2) (2003) 167–256.
[50] G. Bagler, Analysis of the airport network of india as a complex weighted network, Physica A 387 (12) (2008) 2972–2980.
[51] T. Opsahl, P. Panzarasa, Clustering in weighted networks, Soc. Netw. 31 (2) (2009) 155–163.
[52] G.P. Clemente, R. Grassi, Directed clustering in weighted networks: a new perspective, chaos, Solitons Fract. 107 (2018) 26–38.
[53] M. Henry, G. Judge, Permutation entropy and information recovery in nonlinear dynamic economic time series, Econom. 7 (1) (2019) 10.
[54] O.A. Rosso, H. Larrondo, M.T. Martin, A. Plastino, M.A. Fuentes, Distinguishing noise from chaos, Phys. Rev. Lett. 99 (15) (2007) 154102.
[55] M. McCullough, M. Small, H. Iu, T. Stemler, Multiscale ordinal network analysis of human cardiac dynamics, Philos. Trans. R. Soc. A 375 (2096) (2017) 20160292.