This is a repository copy of *Network Entropy using Edge-based Information Functionals*.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/159487/

Version: Accepted Version

Article:
Aziz, Furqan, Hancock, Edwin R orcid.org/0000-0003-4496-2028 and Wilson, Richard Charles orcid.org/0000-0001-7265-3033 (2020) Network Entropy using Edge-based Information Functionals. Journal of Complex Networks. ISSN 2051-1329

https://doi.org/10.1093/comnet/cnaa015

Reuse
Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

Preprint Submitted to Journal of Complex Networks (2020) Page 1 of 25
doi:10.1093/imaiai/drn000

Network Entropy using Edge-based Information Functionals

FURQAN AZIZ*,
Centre for Computational Biology, University of Birmingham, Birmingham, UK, B15 2TT
*f.aziz@bham.ac.uk
EDWIN R. HANCOCK
Department of Computer Science, University of York, York, UK, YO10 5GH
edwin.hancock@york.ac.uk

AND

RICHARD C. WILSON
Department of Computer Science, University of York, York, UK, YO10 5GH
richard.wilson@york.ac.uk

[Received on 26 March 2020]

In this paper we present a novel approach to analyze the structure of complex networks represented by a quantum graph. A quantum graph is a metric graph with a differential operator (including the edge-based Laplacian) acting on functions defined on the edges of the graph. Every edge of the graph has a length interval assigned to it. The structural information contents are measured using graph entropy which has been proved useful to analyze and compare the structure of complex networks. Our definition of graph entropy is based on local edge functionals. These edge-functionals are obtained by a diffusion process defined using the edge-based Laplacian of the graph using the quantum graph representation. We first present the general framework to define graph entropy using heat diffusion process and discuss some of its properties for different types of network models. Second, we propose a novel signature to gauge the structural complexity of the network and apply the proposed method to different datasets.

Keywords: Network Entropy; Edge-based Laplacian; Diffusion process on graph

2000 Math Subject Classification: 34K30, 35K57, 35Q80, 92D25

1. Introduction

Information theoretical methods have been widely used to analyze complex systems represented by means of graphs. A complex system is composed of many components which may interact with each other in different possible ways [35]. Examples of such systems include biological networks [47], social networks [27], document networks [17], computer networks [48], and citation networks [31]. Graphs provide a natural way to represent these systems, where nodes of the graph correspond to components while edges of the graph represent interactions. These graphs are commonly known as complex networks in the literature [35]. The term is used to differentiate a graph representing a complex system from the one that is randomly generated. Once a system is modelled using a complex network, the network can not only be used to obtain important information about the structure and the behaviour of the system, but it can also be used to compare different systems to assess their similarities [56].

One of the challenging problems in network analysis is the quantification of the complexity of a network [1]. The complexity of a network is usually defined as a summary of the amount of the information

© The author 2020. Published by Oxford University Press on behalf of the Institute of Mathematics and its Applications. All rights reserved.

encoded by a complex network. A number of measures have been proposed that allow us to estimate the complexity of a network. These can be broadly divided into two categories, that is a) randomness complexity and b) statistical complexity. Randomness complexity aims to probabilistically characterise the degree of disorganisation of a network [28]. This is usually estimated by the non-trivial structure of the network, such as the degree distribution, clustering coefficients or betweenness centrality. Here Shannon entropy provides a convenient way to estimate the required complexity [9]. However, since randomness complexity is usually based on a probability distribution over the node statistics, one of the shortcomings with this measure is that it does not capture the global structure of a network. Statistical complexity, on the other hand, is defined using the notion of a processes rather than a measure. In order to overcome the locality problem with randomness complexity, statistical complexity aims to measure the regularities beyond randomness thus allowing more global structures to be probed. For example Passerini et al. [39] have used the eigenvalues of the normalized Laplacian of a graph to define the entropy of a graph. They have shown that this quantity is related to the number of connected components, long paths and nontrivial symmetries in the graph.

1.1 *Related Literature*

One of the earliest attempts at computing network entropy was proposed by Körner [28]. This involves computing the entropy of a vertex packing polytope assuming there is a probability distribution associated with the vertices of a graph. Körner entropy is linked to the chromatic number of the graph, the computation of which is in general an NP-complete problem. In [39] Passerini and Severini have proposed the use of Von Neumann entropy that was inspired by a connection between quantum information and graph theory. Their definition of graph entropy was based on the eigenvalues of the normalised discrete Laplacian matrix of a graph. They have shown that the quantum entropy can be used as a measure of regularity for graphs, i.e., regular graphs in general have higher entropy. Lin et al. [22] have approximated the Von Neumann entropy by replacing the Shannon entropy by a quadratic counterpart. Since the von Neumann entropy is computed from the eigenvalues of the Laplacian, it generally requires cubic computational time in the number of nodes. The approximate entropy [22] on the other hand can be computed in quadratic time. In [1] Ananad et al. have investigated the relationship between the Von Neumann entropy and the Shannon entropy and have provided some interesting results. Another important measure, that is based on spectral graph theory, is the use of Laplacian energy[21], i.e., the sum of absolute differences between the eigenvalues and the average vertex degree. The network heterogeneity index, proposed by Estrada [51] is another important addition to the graph-spectral literature that is computed using the differences in degrees for all pairs of connected vertices. In related work [13] Estrada has defined a topological index as a measure of the degree of folding of a protein to characterise 3D molecular structures. This index has found multiple applications in the study of protein function [16]. In a related work [2], we have used frequencies of closed walks and cycles to estimate the entropy of a graph.

An alternative approach to measure the complexity of a network is the decomposition approach. Here a graph is divided into substructures by grouping vertices. The entropy of the graph is then computed using local vertex density. In [9, 50], Dehmer has generalised this idea and has proposed a framework that is based on local vertex functionals obtained by calculating a j-sphere using Dijkstra's algorithm. They have also investigated the relationship between the proposed measure and other entropic measures and have provided some numerical results to demonstrate the usefulness of the proposed measure. In a related work, Estarada [14] et al. have defined graph entropies using random walks on a graph referred to as walk entropies. The walk entropies are also based on the concept of local vertex probabilities. The local vertex probabilities are defined by selecting a closed random walk starting and ending at the same node. They have shown that walk entropies behave differently for regular and non-regular graphs.

1.2 *Contributions*

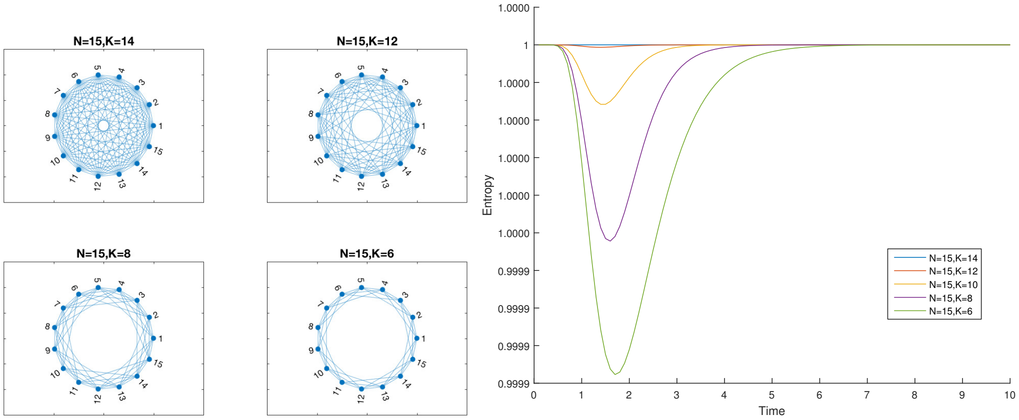
In this paper we propose a novel decomposition framework for defining the entropy of a graph. Our idea is based on the fact that the edges of a graph can capture the connectivity structure conveying more information when compared to the vertices of the graph. For example, in [4], we have shown that a backtrackless walk (i.e., a random walk on an oriented line graph, one in which the original edges are represented as new vertices, and the new edges indicate the coincidence of the original edges) is more powerful in distinguishing cospectral graphs when compared to a random walk on the original graph. In another work Dutta et. al. [11] have shown that a backtrackless walk on a graph can be used to match graphs with higher accuracy when compared to a random walk on the graph. Motivated by this, here we divide the graph into substructures by grouping the edges of a graph and then use local edge probabilities to compute the entropy of graph.

To define a local information functional for an edge, we use a heat diffusion process that is computed using the edge-based Laplacian of a graph. Based on this definition, we then compute the Shannon entropy of a graph. In [3], we have used the edge-based heat diffusion process to characterize the points on the surface of a three-dimensional shape represented using a mesh that approximates its surface. In a related work [5], we have used the solution of a wave equation defined using the edge-based Laplacian to study the properties of a complex network. Here, the initial solution was a Gaussian wave packet residing on a single edge of a graph and we have studied its solution to characterize the shape of a complex network.

To illustrate why this might be advantageous, Figure 1(b) compares the value of the entropy computed using the proposed measure at different time scales between the different *k*-regular graphs of Figure 1(a), each with 15 vertices. A *k-regular graph* is a graph with *n* vertices, where each vertex is connected with *k* neighbours. Note that at small scale and large scale, all the graphs are also similar as the local structures at these scales are similar. At intermediate scales, their local structure differs and therefore we obtain different entropy values. In particular, the positions of the minimum entropy and the corresponding values for different graphs are different. Later, in our experimental evaluation, we use these values to estimate the graph complexity.

In order to compare graphs for similarity, we propose a feature descriptor (ED_{EB}) that embeds a graph into a higher-dimensional feature space. The descriptor is developed by sampling the values of entropy computed using heat diffusion process defined using edge-based operator at different scales. Since it is based on the heat diffusion processes on a graph, it *hears the shape of a graph [49]*. The reason for using different time-scale is to obtain a signature that captures both local and global structure of the graph. Experimental evaluation on real-world data demonstrates that the proposed method is expressive and it outperforms state-of-the-art works.

The remainder of the paper is organized as follows. In Section 2 we define quantum graphs and provide some definitions and relevant basic results that are required for the developments presented in this paper. Section 3 proposes a general framework to define the Shannon entropy of a graph based on its edges. Finally we propose a global signature that can be used to characterize the global structure of a graph. In Section 4 we provide experimental evaluation and empirically demonstrate the properties of the proposed measure on different types of graphs. We also demonstrate the usefulness of the proposed



(a) Example graphs

(b) Entropy

FIG. 1. Comparison of regular graphs with same number of vertices at multiple scales.

method in characterizing graphs and compare its performance with alternative methods. Section 5 provides conclusion and gives some possible future directions.

2. Preliminaries

To make this paper self contained, in this section we give some basic definitions and provide some basic results.

2.1 *Definitions*

A *graph* (or a simple graph) $G = (V, E)$ consists of a finite nonempty set V of *vertices* and a finite set E of unordered pairs of vertices, called *edges*. A *directed graph* or *digraph* $D = (V_D, E_D)$ consists of a finite nonempty set V_D of vertices and a finite set E_D of ordered pairs of vertices (i.e., oriented edges), called *arcs*. The *oriented line graph* $OL(G) = (V_O, E_O)$ is constructed by replacing each arc of $D(G)$ by a vertex. These vertices are connected if the head of one arc meets the tail of another, except that reverse pairs of arcs are not connected, i.e. $((u,v),(v,u))$ is not an edge [41]. A *random walk* on the vertices of a line graph corresponds to a random walk on the edges of the original graph, while a random walk on the vertices of the oriented line graph corresponds to a *backtrackless walk* on the edges of the original graph (See [4] and the references cited therein). A graph is usually represented by an *adjacency matrix* of size $|V| \times |V|$, whose $(i,j)^{th}$ entry is 1 if $(v_i, v_j) \in E$. The *Laplacian matrix* is a convenient matrix representation of a graph which is defined as the $L = D - A$. Here D is the diagonal matrix, whose i^{th} diagonal entry represents the degree of the vertex v_i. The Laplacian matrix is sometimes referred to as the discrete Laplacian, as it is an approximation of continuous Laplacian to a discrete domain, i.e., the vertices of the graph.

A *metric graph* [18] is a geometric realisation of a graph, where each edge is assigned a real length interval. In this paper though, our focus is on graphs with uniform length intervals. A *quantum graph* [30] is a metric graph equipped with the operator $\mathscr{H}$ (Hamiltonian) that acts as the negative second order derivative along the edge intervals and satisfies some suitable boundary conditions at the vertices.

The *edge-based Laplacian* [19] is defined on a metric graph and it satisfies the Neumann-Kirchhoff boundary condition. This condition states that sum of the outward-pointing gradients must be zero. Therefore this operator is more closely related to the Laplacian defined in calculus and can be used to simulate a number of process over a graph.

2.2 *Eigen-system of the Edge-based Laplacian*

We now review the eigen-system of the edge-based Laplacian developed in [19] and in our earlier work [55]. Let $G = (V,E)$ be a graph with geometric realization $\mathscr{G}$ that assigns a unit length interval to every edge of the graph. We associate an edge variable x_e with each edge, $e = (u,v)$, that represents the standard coordinate on the edge interval with $x_e(u) = 0$ and $x_e(v) = 1$. The eigenfunctions of the edge-based Laplacian are of two types, i.e., a) vertex-supported eigenfunctions and b) edge-interior eigenfunctions. The vertex supported eigenfunctions are determined by the eigenfunctions of the normalised adjacency matrix of the graph, while the edge-interior eigenfunctions are determined by the adjacency matrix of the associated oriented line graph. Let A be the adjacency matrix of the graph G, and $\tilde{A}$ be the row normalised adjacency matrix. i.e., the $(i,j)th$ entry of $\tilde{A}$ which is given as

$$\tilde{A}(i,j) = \frac{A(i,j)}{\sum_{(k,j)\in E} A(k,j)}.$$

Let (ϕ_v, λ) be an eigenvector-eigenvalue pair for this matrix. Let ω^2 and ϕ_e denote the edge-based eigenvalue and eigenfunction respectively. Then the vertex-supported eigenpairs of the edge-based Laplacian are given as follows:

1. For each (ϕ_v, λ) with $\lambda \neq \pm 1$, we have a pair of eigenvalues ω^2 with $\omega = \cos^{-1}\lambda$ and $\omega = 2\pi - \cos^{-1}\lambda$. Since there are multiple solutions to $\omega = \cos^{-1}\lambda$, we obtain an infinite sequence of eigenfunctions; if $\omega_0 \in [0,\pi]$ is the principal solution, the eigenvalues are $\omega = \omega_0 + 2\pi n$ and $\omega = 2\pi - \omega_0 + 2\pi n, n \geqslant 0$. The eigenfunctions are $\phi_e = C(e)\cos(B(e) + \omega x_e)$ where

$$C(e)^2 = \frac{\phi_v^2 + \phi_u^2 - 2\phi_v\phi_u\cos(\omega)}{\sin^2(\omega)}$$

$$\tan(B(e)) = \frac{\phi_v\cos(\omega) - \phi_u}{\phi_v\sin(\omega)}$$

   There are two solutions here, $\{C, B_0\}$ or $\{-C, B_0 + \pi\}$ but both give the same eigenfunction. The sign of $C(e)$ must be chosen correctly to match the phase.

2. $\lambda = 1$ is always an eigenvalue of $\tilde{A}$. We obtain a principle frequency $\omega = 0$, and therefore since $\phi_e = C\cos(2n\pi x_e)$ and so $\phi_v = \phi_u = C$, which is constant on the vertices.

3. $\lambda = -1$ is an eigenvalue of $\tilde{A}$, if the graph is bipartite. We obtain a principle frequency $\omega = \pi$, and therefore since $\phi_e = C\cos(\pi x_e + 2n\pi x_e)$ and so $\phi_v = C$ and $\phi_u = -C$, which means the eigenfunction is constant with an alternating sign on both sides of bipartition.

The edge-interior eigenfunctions are those eigenfunctions which are zero on vertices and therefore must have a principle frequency of $\omega \in \{\pi, 2\pi\}$. These eigenfunctions can be determined from the eigenvectors of the adjacency matrix of the oriented line graph.

1. The eigenvector corresponding to eigenvalue $\lambda = 1$ of the oriented line graph provides a solution in the case $\omega = 2\pi$. In this case we obtain $|E| - |V| + 1$ linearly independent solutions.

2. The eigenvector corresponding to eigenvalue $\lambda = -1$ of the oriented line graph provides a solution in the case $\omega = \pi$. If the graph is bipartite, then we obtain $|E| - |V| + 1$ linearly independent solutions. If the graph is non-bipartite, then we obtain $|E| - |V|$ linearly independent solutions.

This comprises all the principal eigenpairs of the edge-based Laplacian.

3. Graph Entropy

In this section we propose a general framework to define graph entropy that is based on edge-based operators of a graph. Our approach is closely related to the framework proposed by Dehmer [9]. While Dehmer's definition of entropy is based on a probability distribution of the vertices of a graph, our work is based on a derived probability distribution over the edges of a graph. More specifically, we use a heat diffusion process over the edges of a graph to extract a local subgraph and use the process to define a probability distribution over the edges of a graph. Next we define the Shannon entropy using the obtained probability distribution. The heat diffusion process over the edges of the graph is defined using the edge-based Laplacian and is related to the random walks on the edges of the graph.

To commence, we define the concept of information functional for an edge of a graph.

Definition 3.1 Let $G = (V, E)$ be a graph. For an edge $e_i \in E$, we define:

$$p(e_i) := \frac{f(e_i)}{\sum_{j=1}^{|E|} f(e_j)} \tag{3.1}$$

where f represents an information functional for an edge e satisfying the condition that $f(e_i) > 0, \forall e_i \in E$. Since $p(e,i) > 0, \forall i$ and $p(e_1) + p(e_2) + ... + p(e_{|E|}) = 1$, therefore we can interpret the quantities $p(e_i)$ as edge probabilities.

Based on this definition of the edge probability distribution, we can define the Shannon entropy of a graph.

Definition 3.2 Let $G = (V, E)$ be a graph and let f be an arbitrary information functional such that $f(e_i) > 0, \forall e_i \in E$. We define the entropy of G by

$$I_f(G) = -\sum_{i=1}^{|E|} \frac{f(e_i)}{\sum_{j=1}^{|E|} f(e_j)} \log\left(\frac{f(e_i)}{\sum_{j=1}^{|E|} f(e_j)}\right) \tag{3.2}$$

We now define an information functional which is based on a heat diffusion process defined over the edges of a quantum graph. The heat equation on the graph is given by

$$\frac{\partial H_t}{\partial t} = -\Delta_E H_t, \tag{3.3}$$

where Δ_E is the edge-based Laplacian and H_t is the fundamental solution of the heat equation also referred to as heat kernel. The heat kernel has the following eigen-decomposition:

$$H_t(x_e, x_d) = \sum_{\omega} \exp(-\omega_i^2 t)\phi_i(x_e)\phi_i(x_d) \tag{3.4}$$

where (ϕ_i, ω_t^2) are the eigenpairs of the edge-based Laplacian. The heat kernel can be thought of as the amount of heat that is transferred from x_e to x_f in time t given a unit heat source at x_e.

In this paper our goal is to use $H_t(x_e, x_d)$ to define the entropy of a graph. We restrict the heat kernel to the temporal domain only and discard its entire spatial domain. In [46], Sun et al have used a similar measure that uses vertex-based Laplacian to define a unique signature for every point on the surface of a 3D shape. They have shown that despite restricting the heat kernel to its temporal domain, it captures most of the salient properties of heat kernel under mild assumptions. Since the edges of a metric graph are real length intervals, to define an analogous measure based on the edges of the graph, we therefore take the integral of the heat kernel restricted to temporal domain over its edge-interior points, i.e.,

$$\tau_t(e) = \int_0^1 H_t(x_e, x_e) dx_e.$$

Substituting the value of H_t from Equation 3.4 into above equation, we obtain

$$\tau_t(e) = \sum_{\omega}\sum_{n} \int_0^1 e^{-(\omega+2n\pi)^2 t} C_{e,\omega}^2 \cos^2\left[B_{e,\omega} + (\omega + 2n\pi)x_e\right] dx_e. \tag{3.5}$$

The value of $\tau_t(e)$ characterises the amount of heat residing on the edges of the graph at a particular moment in time. To compute $\tau_t(e)$, we need to compute its values for the set of all edge interior eigenvalues and eigenfunctions. Consider, for example, the case when the eigenvalues are constant, i.e., $\omega = 0, n \geqslant 0, B_{e,\omega} = 0$. We evaluate the two cases separately, i.e., the case when $n = 0$ and the case when $n > 0$. When $n = 0$, then $C_{e,\omega} = \frac{1}{|E|}$, and therefore

$$\tau_t^{(1)}(e) = \frac{1}{|E|}.$$

On the other hand, when $n > 0$, we have $C_{e,\omega} = \frac{2}{|E|}$, and therefore

$$\tau_t^{(1)}(e) = \frac{1}{|E|}\sum_{n>0} e^{-(2n\pi)^2 t}.$$

Combining the above two equations, we obtain

$$\tau_t^{(1)}(e) = \frac{1}{|E|} + \frac{1}{|E|}\sum_{n>0} e^{-(2n\pi)^2 t} = \frac{1}{|E|}\sum_{n\geqslant 0} e^{-(2n\pi)^2 t}.$$

In a manner similar to above, we need to evaluate all the cases and the combine the results to obtain the value of $\tau_t(e)$. Table 1 lists all the possible solutions. Here $\tau_t^1(e)$ corresponds to the case when eigenvalues are constant, while $\tau_t^2(e)$ corresponds to the case when eigenvalues are computed from the adjacency matrix of the oriented line graph. The third term $\tau_t^3(e)$ comes from the constant eigenvalue for the bipartite graph, i.e., the case when $\lambda = -1$. The last two terms, $\tau_t^4(e)$ and $\tau_t^5(e)$ corresponds to the edge-interior eigenfunctions.

The exact solution of heat kernel depends both upon whether the graph being considered is bipartite or not, and upon the number of edges and vertices in the graph. Here we give solutions for some of the special cases of graph structure.

1. For a bipartite graph with $|E| > |V|$, the solution is

$$\tau_t(e) = \tau_t^{(1)}(e) + \tau_t^{(2)}(e) + \tau_t^{(3)}(e) + \tau_t^{(4)}(e) + \tau_t^{(5)}(e).$$

   Here τ^3 is due to the constant eigenfunction corresponding to the principal frequency $\omega = \pi$.

Term	ω	Solution
$\tau_t^{(1)}(e)$	0	$\frac{1}{\lvert E\rvert}\sum_{n\geqslant 0} e^{-(2n\pi)^2 t}$
$\tau_t^{(2)}(e)$	$\cos^{-1}\lambda$	$\sum_{\omega\in\Omega}\sum_n \frac{e^{-(\omega+2n\pi)^2 t}C^2_{e,\omega}}{4(\omega+2n\pi)}\{\sin(2B_{e,\omega}+2\omega)-\sin(2B_{e,\omega})+2(\omega+2n\pi)\}$
$\tau_t^{(3)}(e)$	π	$\frac{1}{\lvert E\rvert}\sum_{n\geqslant 0} e^{-(\pi+2n\pi)^2 t}$
$\tau_t^{(4)}(e)$	2π	$\sum_\omega\sum_{n\geqslant 0}\frac{1}{2}C^2_{e,\omega}e^{-(2n\pi)^2 t}$
$\tau_t^{(5)}(e)$	π	$\sum_\omega\sum_{n\geqslant 0}\frac{1}{2}C^2_{e,\omega}e^{-(\pi+2n\pi)^2 t}$

Table 1. List of Solutions

2. For a non-bipartite graph with $|E| > |V|$, the solution is

$$\tau_t(e) = \tau_t^{(1)}(e) + \tau_t^{(2)}(e) + \tau_t^{(4)}(e) + \tau_t^{(5)}(e),$$

   and so the term τ^3 is omitted, because a non-bipartite graph has no principle frequency $\omega = \pi$ and $B \neq 0$.

3. Note that the presence of the terms τ^4 and τ^5 depends on the number of edges and vertices in the graph, and also upon whether the graph is bipartite or not. So, for a graph which is a cycle of odd length, the solution is

$$\tau_t(e) = \tau_t^{(1)}(e) + \tau_t^{(2)}(e) + \tau_t^{(5)}(e),$$

   while for a cycle of even length, the solution is

$$\tau_t(e) = \tau_t^{(1)}(e) + \tau_t^{(2)}(e) + \tau_t^{(3)}(e) + \tau_t^{(4)}(e) + \tau_t^{(5)}(e).$$

4. For a tree (which is always bipartite), we have

$$\tau_t(e) = \tau_t^{(1)}(e) + \tau_t^{(2)}(e) + \tau_t^{(3)}(e),$$

   i.e., both τ^4 and τ^5 are omitted. This is because for a tree $|E| = |V| - 1$, and so the multiplicity of the edge-interior eigenfunctions is zero.

Once we compute the value of $\tau_t(e)$, for a particular t, we define the edge-based heat kernel entropy of the graph.

Definition 3.3 Let G = (V,E) be a graph and let $\tau_t(e) > 0$ is the information functional which represents the amount of heat released by the edge e at time t given a unit heat source at e. The entropy of the network, at a specific time t, is then defined as:

$$I_\tau(G,t) = -\sum_{i=1}^{|E|} \frac{\tau_t(e_i)}{\sum_{j=1}^{|E|}\tau_t(e_j)} \log\left(\frac{\tau_t(e_i)}{\sum_{j=1}^{|E|}\tau_t(e_j)}\right). \tag{3.6}$$

The term $\sum_{j=1}^{|E|} \tau_t(e_j)$ is called the heat kernel trace. Note that as t increases, the value of the edge-based heat kernel entropy approaches $\log(|E|)$. This is proved in the following theorem.

THEOREM 3.1 Let G = (V,E) be a graph and $\tau_t(e)$ represents the amount of heat released by the edge e at time t given a unit heat source at e. Then, $I_\tau(G,t)$ approaches $\log(|E|)$ as $t \to \infty$, i.e.,

$$\lim_{t\to\infty} I_\tau(G,t) = \log(|E|).$$

Proof. When $t \to \infty$, $e^{-(2n\pi)^2 t} \to 0$. This means when $t \to \infty$, $\tau_t^{(1)}(e) = \frac{1}{|E|}$ while $\tau_t^{(2)}(e) = \tau_t^{(3)}(e) = \tau_t^{(4)}(e) = \tau_t^{(5)}(e) = 0$. Therefore for large values of t, we have

$$I_\tau(G, t \to \infty) = -\sum_{i=1}^{|E|} \frac{1}{|E|} \log\left(\frac{1}{|E|}\right) = \log(|E|)$$

□

Note that in [15] Estrada has shown that when $t \to \infty$, the walk entropy is maximum, i.e., $\log(|V|)$. On the other hand, in our case the maximum entropy is $\log(|E|)$ which for any graph is attained when $t \to \infty$. In the experimental section of this paper, we will show that the difference between the maximum entropy and the minimum entropy can be used as a useful tool to characterize the structure of a complex network.

In order to compute similarity (or dissimilarity) between two graphs, we define a novel global feature descriptor that can be used to embed the graph in a higher-dimensional feature space. The proposed signature is obtained by sampling the values of $I_\tau(G,t)$ at different time scales.

Definition 3.4 Let G = (V,E) be a metric graph and let $I_\tau(G,t)$ be its edge-based heat kernel entropy for a particular time t. We define an entropy based signature of the graph as

$$ED_{EB}(G) = [I_\tau(G,t_0), I_\tau(G,t_1), ..., I_\tau(G,t_n)], \tag{3.7}$$

where t_0, t_1, t_2,..., t_n are the times at which the samples of $I(G,t)$ are taken. To obtain a size invariant representation, the feature vectors are normalized by dividing each component of the feature vector by $\log(|E|)$.

Note that for small values of t, the value of $\tau_t(e)$ is related to local structure of the graph, while for large values of t it is related to the global properties [46]. In our experimental evaluation we choose $t_0 = 4\log(10)/\omega_{max}^2$ and $t_0 = 4\log(10)/\omega_{min}^2$ and uniformly sampling 100 points in the interval between t_0 and t_{100}. Note that in [46], Sun et al. have used the same time limits to define a local signature for a vertex of a mesh. Here we are using the same limits to define a novel global signature that can be used to compare graph structures.

4. Experiments

In this section we present some empirical results and demonstrate some of the properties of the proposed framework. We test our method on both synthetic data as well as real-world data. The purpose of the experiments on synthetic data is to investigate the properties of the entropy as a function of time for graphs that are generated using different network models.

4.1 *Synthetic Data*

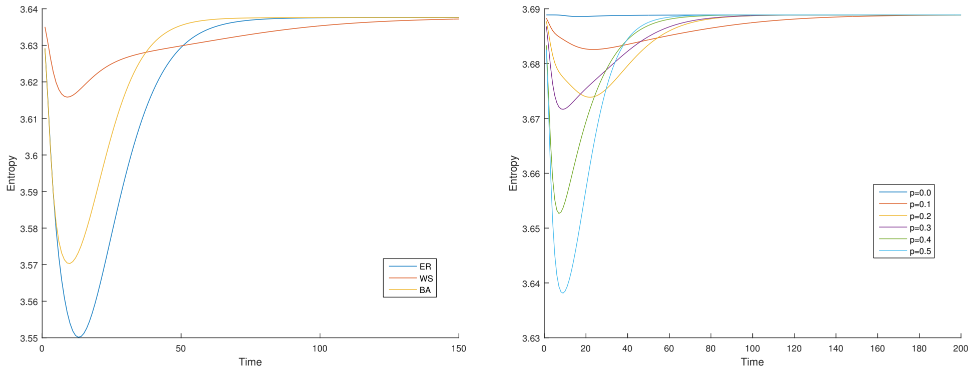
In this section we perform experiments on synthetically generated graphs. We commence by demonstrating that the proposed definition of network entropy is consistent with the other definitions of graph entropy, such as random walk entropies [15]. In other words, we show that as with the vertex-based measures of graph entropy, the proposed method can also capture the regularity structure of a graph. For this purpose, we generate graphs according to following three different network models.

Erdős-Rényi model (ER) [12]: An ER graph $G(n, p)$ is constructed by making pairwise connections between n vertices randomly with probability p. In other words each edge is included in the graph with probability p independent from every other edge.

Watts and Strogatz model (WS) [54]: A WS graph $G(n, k, p)$ is constructed in the following way. First we construct a regular ring lattice on the nodes. Then for a graph with n vertices, each vertex is connected to its k nearest vertices on the ring, with $k/2$ connections on each side of the ring. Then for every vertex we take each edge and rewire it with probability p.

Barabási-Albert model (BA) [6]: A BA graph $G(n, n_0, m)$ is constructed from an initial fully connected graph with n_0 vertices. New vertices are added to the graph one at a time. Each new vertex is connected to m previous vertices with a probability that is proportional to the number of pre-existing links for each node.

To commence, we generate a graph with 20 vertices for each of the above models. We choose the parameters for each model in such a way so that all three graphs have the same number of edges. The objective here is to investigate how the shape of the entropy verse time curve varies for graphs of the same size but that are generated according to different graph or network models. Figure 2(a) shows the resulting entropy curves as a function of time for all the three models with $t = 0$ to $t = 15$. The curve for the ER graph is shown in blue, that for the WS graph in red and that for BA graph in yellow.



(a) Entropy curves for networks generated according to three different network models

(b) Entropy curves for networks generated according to WS models

FIG. 2. Entropy curves for randomly generated networks

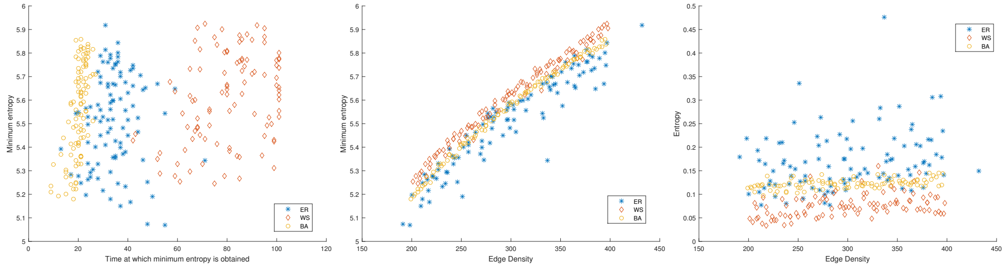
The plot shows that the ER graphs (blue) have the largest variation in the entropy, while the graph generated according to WS model (red) has lowest variation in entropy. This suggests that a graph with a more irregular structure exhibits higher variation in the shape of entropy verses time curve. To further investigate the relationship between the entropy verses time curve and the regularity structure of the corresponding graph, we generate a WS graph with $n = 20$, $k = 4$ and with 0 rewiring probability. This will produce a k-regular graph with 20 vertices, 40 edges and $k = 4$. Next we generate 5 different WS graphs by keeping the values of n and k fixed, while increasing the rewiring probability, subject to the constraint that the number of edges remains the same. Figure 2(b) plots the resulting entropies. It is clear from the figure that as the rewiring probability increases, the variation in the shape of entropy verses time curve also increases. We may, therefore, conclude that a graph with a more regular structure will exhibit a small change in the shape of the entropy verses time curve and will have higher entropy as compared to a graph having the same number of vertices and edges but with a more irregular structure.

Another important observation that can be made from Figure 2 is that the entropy curve of a graph is smooth and it has a local minima (except for a complete graph whose entropy curve is constant). We define ENT_{min} as the minimum value of the entropy curve of a graph, i.e.,

$$ENT_{min} = \min_{t \in (0,\infty)} I_{\tau}(G,t).$$

The reason for defining this graph invariant is that its value is different for networks that are generated according to different network models. This is indeed the case for the entropy curves shown in Figure 1 and 2. It can be seen that the difference between the maximum entropy of the graph (which is $\log|E|$) and ENT_{min} is minimum for a regular graph and as the irregularity in the graph increases the difference also increases. To measure the structural contents of graphs with different sizes, we define a size invariant complexity measure, the edge-based entropy ENT_{EB}, of a network as $ENT_{EB} = \log(|E|) - ENT_{min}$. According to this definition, a graph with more irregular structure will have high entropy as compared to a graph with a more regular structure. Therefore a complete graph will have zero entropy.

To analyze the behaviour of the above defined invariant, we apply it to synthetic graphs. For this purpose we have generated 100 graphs for each model with $101, 102, \ldots, 200$ vertices. We have chosen the remaining parameters in such a way so that all the three types of graphs with the same number of vertices have approximately the same number of edges. For ER models we choose $p = 10/n$, where n is the number of vertices in the graph, for WS models we choose $p = 0.25$ and $k = 8$, and for BA models we choose $n_0 = 5$ and $k = 4$. Figure 3(a) plots minimum entropy against the diffusion time at which this minimum entropy value was obtained.



(a) ENT_{min} vs. T_{min}

(b) ENT_{min} vs. $|E|$

(c) $|E|$ vs. ENT_{EB}

FIG. 3. Relationship between different invariants.

The embedding results of Figure 3(a) show that both the diffusion time of the entropy and the minimum value of the entropy can be used to characterise the structure of a graph. Note that the difference between the entropy value for the graphs generated according to same model is due to the difference in sizes of the graphs. This is demonstrated in Figure 3(b), where we have plotted the edge density against the minimum entropy of the graph, which shows a linear relation between the edge density and minimum entropy is linear. It also suggests that an ER graph has lowest value of minimum entropy when compared to graphs generated according to alternative network models. Finally, Figure 3(c) plots the value of ENT_{EB} against the edge density of the graph. This clearly shows that the value of ENT_{EB} is high for irregular networks when compared to a network with a more regular structure with the same graph size.

We now demonstrate the power of the proposed method in characterising the structure of networks. For this purpose, we generate different graphs according to six different network models. These include the three models already discussed at the beginning of this section (i.e., BA, ER, and WS) and the following three models.

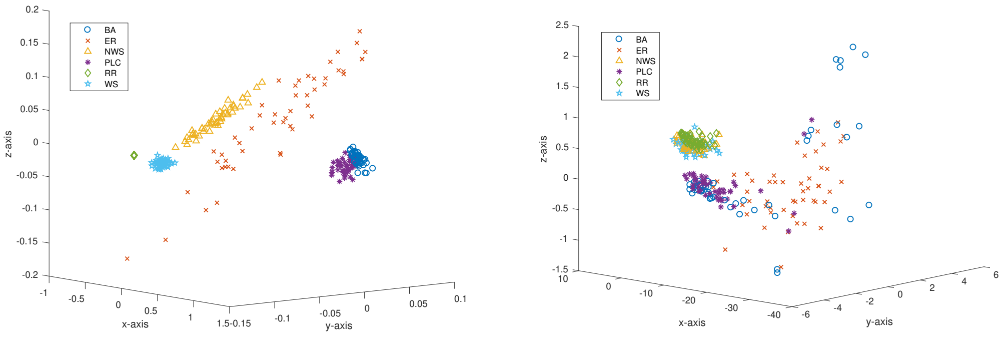
Newman Watts Strogatz model (NWS) [36]: The model is a variant of Watts–Strogatz network, in which edges are added between pairs of nodes in the same way as in a WS network but without removing edges from the underlying lattice.

Power-Law Cluster (PLC) [25]: Power-law cluster (Holme and Kim network model) is essentially BA model with an extra step that each random edge is followed by a chance of making an edge to one of its neighbours too (and thus a triangle). This algorithm improves on the BA model in the sense that it enables a higher average clustering of nodes to be attained if desired.

Random Regular Graphs(RR) [45]: This method generates a random k-regular graph with n nodes that has no loops or parallel edges.

For each of the six different models, we generate 50 different graphs with 100 nodes each. We choose the parameters in such a way that the number of edges in each graph remains roughly the same. For each graph we compute a feature vector using Equation 3.7. To visualise the results, we have applied principal component analysis (PCA) to the resulting feature vectors. *PCA* is one of the most commonly used statistical techniques for reducing the dimensionality of data. It is a statistical procedure that projects data with possibly correlated variables into a finite dimensional linear subspace while keeping the fraction of the original data variance as high as possible. In the new projection space, the greatest variance by any projection of the data comes to lie along the first coordinate axis (called the first principal component), the second greatest variance lies along the second coordinate axis, and so on. The resulting embedding in a 3D feature space is shown in Figure 4(a).

The results shown in Figure 4(a) suggest that the proposed feature vector consisting of the entropies at different time scales can be used as a powerful tool to distinguish between networks that are generated according to different network models. Note that, except for the ER model, all the remaining five models show a small intra-class variation. It is also interesting to note that the graphs generated according to the BA and PLC models are clustered in close proximity to each other. This is due to the fact that these graphs share some common properties. A similar observation can be made about the graphs generated according to WS and NWS models. The RR graphs, on the other hand, show very high intra-class variation. However, they are mapped to different points in the feature space. Finally, due to their completely random structure, the ER graphs show very high intra-class variation. To analyze the performance of the proposed method, we have performed a similar experiment, but with the graph entropy computed using the vertex-based heat kernel. In other words, we have generated a feature vector in a similar way as we did using Equation 3.7. However, instead of using the edge-based Laplacian, here we have used the traditional vertex-based Laplacian. The resulting embeddings, in this case, are shown in Figure 4(b), which show that the traditional vertex-based Laplacian fails to capture the structure of



(a) Edge-based method

(b) Vertex-based method

FIG. 4. Entropies for three different type of graphs

the graphs generated according to different network models.

We conclude this section by applying the proposed method to the *SYNTHIE* dataset [33], a publicly available synthetic dataset. This dataset consists of 400 graphs, subdivided into four classes, with 15 real-valued node attributes. In our experiments we have ignored the labels. The average number of nodes and the edges in these graphs are 95.00 and 172.93 respectively. We have computed a feature vector by sampling the edge-based graph entropy using Equation 3.7. We have also constructed a similar feature vector that is based n the vertex-based (combinatorial) Laplacian of the graph. To compare the performance of the two methods, we have applied 10-fold cross validation using Support Vector Machine (SVM) with a Gaussian kernel. The proposed method achieved an accuracy of 47.75%, while with the entropies computed using vertex-based Laplacian we obtained only 43.9% accuracy.

4.2 *Social Networks*

This section presents experiments on data extracted from social networks. For this purpose, we choose following four publicly available simple undirected networks [32]:

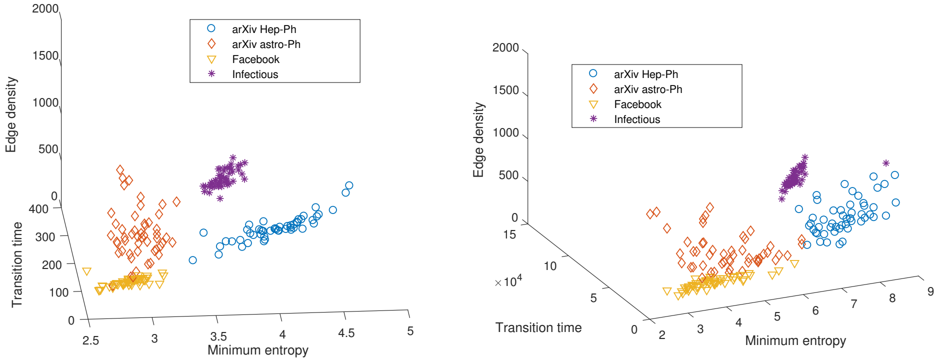
1) arXiv Hep-ph: The collaboration network of authors of scientific papers from the arXiv's high energy Physics section. An edge between two authors represents a common publication.

2) arXiv Hep-ph: Another collaboration network of authors of scientific papers from the arXiv's Astro Physics section.

3) Facebook: This network contains a subset of friendship data of Facebook users. Here a node represents a user and an edge represents friendship.

4) Infectious: This network describes the face-to-face behaviour of people during the exhibition INFECTIOUS: STAY AWAY in 2009 at the Science Gallery in Dublin. Nodes represent exhibition visitors; edges represent face-to-face contacts that were active for at least 20 seconds.

We select 50 subgraphs from each of the above networks. This is done by randomly selecting between 200 to 300 nodes and their associated edges subject to the connectivity constraint. For each of the graphs, we have computed ENT_{EB}, the corresponding transition time, and the edge densities. Using these three measures, we have embedded the graphs in a three dimensional space spanned by minimum entropy, transition time, and edge-density. Figure 5(a) shows the resulting embedding.



(a) Edge-based method

(b) Vertex-based method

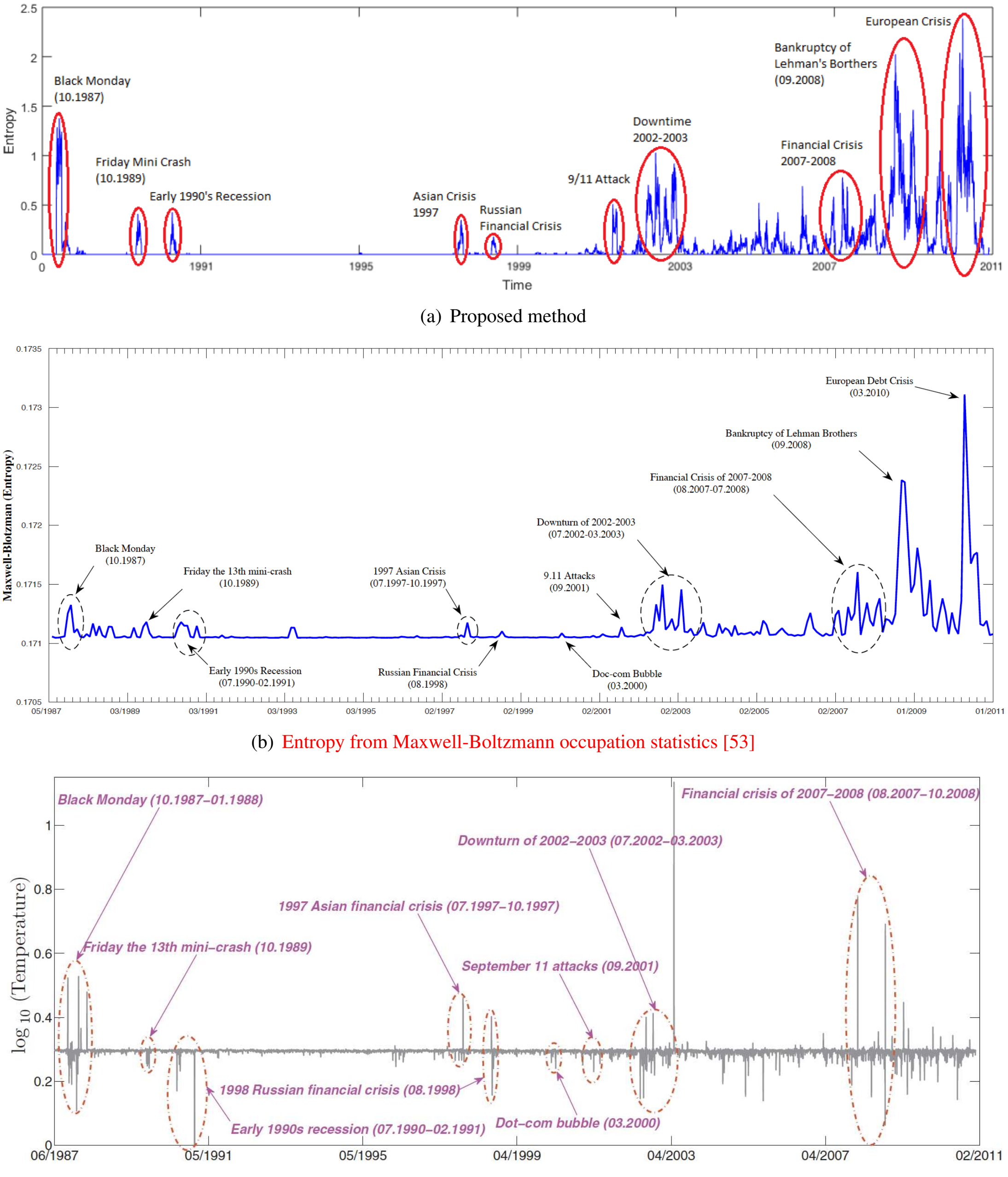
FIG. 5. PCA embeddings of feature vectors

The embedding results shown in Figure 5(a) demonstrate the ability of the proposed entropy measure to distinguish between graphs. We have performed a similar experiment with graph entropies computed using vertex-based Laplacian. The embedding results are shown in Figure 5(b). To compare the performance of the two methods, we have applied 10-fold cross validation using SVM. The proposed method achieved an accuracy of 83.5%, while with vertex-based Laplacian, we obtained only 80.2% accuracy. This shows that the proposed method can quantifying the structure of a network with higher accuracy.

4.3 *Financial data*

In this experiment we explore whether the proposed measure can be used as a tool for understanding the evolution of a complex network with time. We show that our proposed method is consistent with alternative methods reported in the literature, including [53] and [57], in identifying financial crises as periods of anomalous network structure. For this purpose we choose the publicly available New York Stock Exchange (NYSE) dataset. This dataset consists of the daily prices of different stocks traded continuously on the New York Stock Exchange for a 25 year span from January 1986 to February 2011. A total of 347 stock were selected from this set. Here, the nodes of the network represent the stocks. An edge is present, if the correlation between the closing prices of the stock over a 28 days window exceeds a threshold. is above a threshold. We select an empirical value of 0.85 as a threshold value. This was done under the assumption that, at any given time, a particular stock must interact with another stock. A new network is generated by sliding the two windows by one day and repeating the process. In this way a total of 5977 time-varying networks are generated. For each of the generated networks, we have computed the entropy difference. Figure 6(a) plots the ENT_{EB} for all the networks.

The results suggest that the proposed measure can be used as an effective tool for understanding the time evolution of real world complex networks. As Figure 6 shows, the entropy shows high variation during significant financial events including Black Monday, Friday Mini Crash and 9/11 attack. Note that following the 9/11 attack, the US economy was affected by the terrorism which has suffered the



(a) Proposed method

(b) Entropy from Maxwell-Boltzmann occupation statistics [53]

(c) Thermodynamic temprature [57]

FIG. 6. NYSE

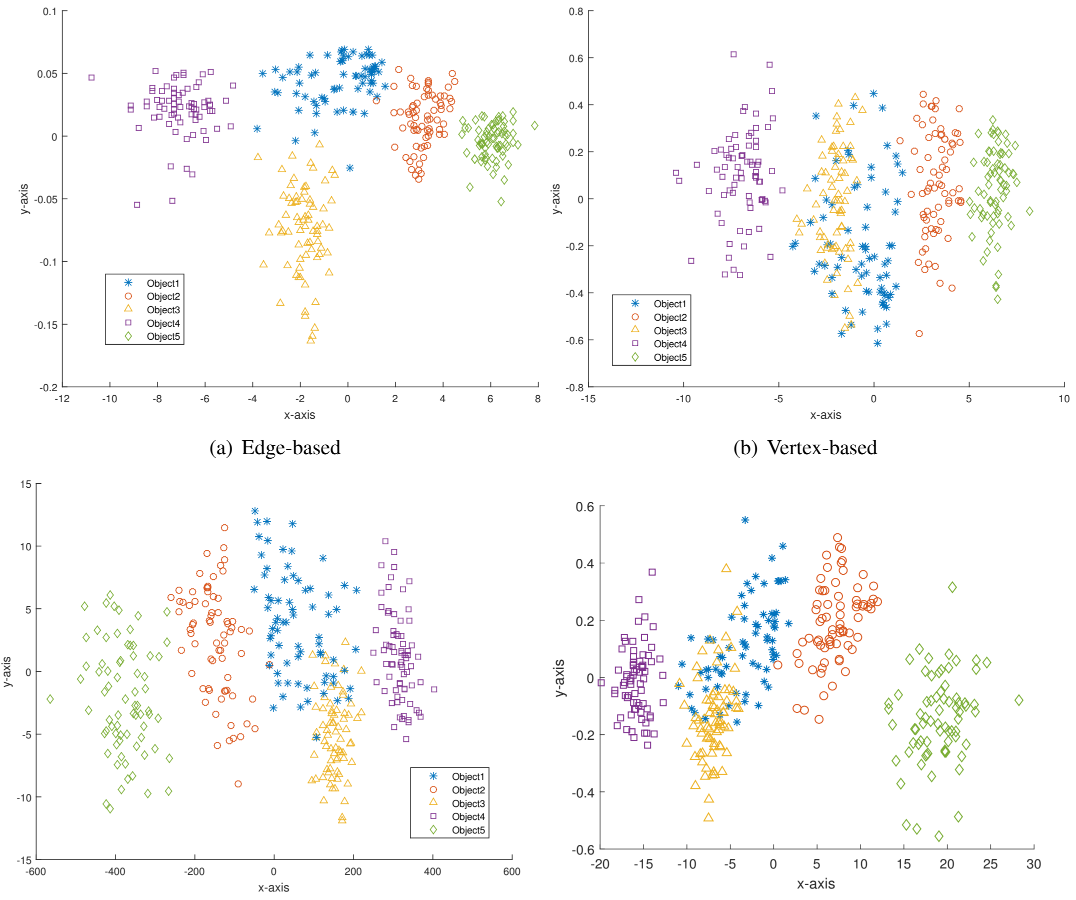
investor’s confidence. This resulted in significant variance in the structure of the entire market. For the purposes of comparison, we also show the results obtained using the entropy computed from Maxwell-Boltzmann occupation statistics for the node-based Laplacian eigenvalues [53] and results produced by the analysis of thermodynamic temperature (rate of change of energy with entropy) [57]. These results are shown in Figure 6(b) and 6(c) respectively. They show that our proposed method is consistent in performance with the alternative methods. In [53], Jianjia et. al. have shown that entropy measures such as von-Neumann entropy and Fermi-Direc entropy cannot detect these events. Similarly, Ye et al [57] have shown that both the node-based heat kernel signature and the wave kernel signature fail to characterise the dynamic behaviour of time series data. Note that all the three methods suffer from few true negatives and false positives. However all the methods have successfully captured the major financial crises in the data.

4.4 *View-based Object Recognition*

In this section we empirically demonstrate that the proposed entropy can be used as a helpful tool to analyze the geometrical and topological properties of a point set adjacency graph. To this end, we choose graphs that are extracted from the arrangement of corner points in 2D views of 3D objects taken from the **COIL** [34] (Columbia Object Image Library) image library. This library consists of 20 different 3D objects each with 72 2D views. These views are obtained from equally spaced directions over 360°in azimuth angle around the object. In our experiments we have selected 5 different objects with all 72 views. To construct graphs over these images, we have applied the **Harris corner detector** [23]. This is used to extract a list of candidate feature points corresponding to the sharp corners in the 3D object. We treat these feature points as the seed points of a Voronoi tessellation of the image plane. The region adjacency graph for Voronoi polygon is the Delaunay triangulation of the original feature points. A **Delaunay triangulations (DT) [10]** for a set P of points in a Euclidean space is a triangulation, DT(P), such that no point in P is inside the circumcircle of any triangle in DT(P). The nodes of the graph represent the feature points and the edges represent adjacencies of the Voronoi regions containing the points.

To evaluate the performance of the proposed method we compare it with two alternate methods i.e., a) feature vectors composed of **Ihara coefficients** [41] and b) the **random walk kernel**. Ihara coefficients are related to prime cycle frequencies in a graph. These cycle frequencies are computed using first few coefficients of the reciprocal of the Ihara zeta function of the graph, commonly referred to as Ihara coefficients. We use three coefficients as proposed by Peng in [41]. Note that Ihara coefficients are considered a powerful tool to capture the cyclic structure of graphs [41]. However they many not perform well on graphs with branches [4]. The random walk kernel compares two graphs based on the number of matching random walks of different lengths in the two graphs. Additionally, we also compare it with a similar invariant computed from the vertex-based Laplacian of the graph.

To commence, we compute the feature vectors for Delaunay triangulations using Equation 3.7. To visualise the feature vectors, we embed them in a low dimension feature space. Figure 7(a) shows the resulting embedding of the feature vectors for all the five classes. To compare the visualisation results, we compute feature vectors for the alternate methods and embed them in a two dimensional feature space using principal component analysis. Figure 7 compares the resulting embedding for each of the methods. To compare the performance of the proposed method, we apply 10-fold cross validation using Support Vector Machine (SVM) with Gaussian kernel. Here we have used the first two principal components that correspond to the highest variance components. Table 2 reports the accuracies of the proposed method and the alternate state-of-the-art methods.



(a) Edge-based

(b) Vertex-based

(c) Ihara Coefficients

(d) Random Walks

FIG. 7. PCA embeddings of feature vectors

Table 2. Accuracies of different methods on COIL

Method	**DT**	**GG**
Edge-based	**96.67 %**	**94.72 %**
Vertex-based	89.17%	90.56%
Ihara Coefficients	92.78%	91.11%
Random Walks	91.39%	93.89%

The visualisation results shown in Figure 7 and the accuracies reported in Table 2 show that the proposed method can give high performance as compared to the state-of-the-art methods. The proposed method also outperforms Ihara Coefficients [4, 41], which is considered a powerful tool in characterising cyclic graphs.

We now compare the performance of the proposed method on Gabriel graphs extracted from the same classes of objects. The Gabriel graph for a set of n points is a subset of Delaunay triangulation, in which an edge connects two data points v_i and v_j for which there is no other point v_k inside the open ball whose diameter is the line connecting points v_i and v_j. The reason for this modification is to sparsify the edges in the group so as to reduce the frequency of cycles of smaller lengths and to introduce some branches in the graph. The purpose of the experiments on Gabriel graphs are two fold. First, since a Gabriel graph is a subset of Delaunay triangulation, it allows us to investigate the performance of the proposed method under controlled structural modification. Second, a Gabriel graph may have branches which allows us to compare the performance of the proposed method on non-cyclic graphs. As with Delaunay triangulations, we compute feature vectors for the Gabriel graphs using both the proposed methods and the alternate methods and then apply 10-fold cross validation using a support vector machine (SVM) with a Gaussian kernel. The resulting accuracies are shown in Table 2, which show that the proposed edge-based entropy functional still gives superior performance as compared to alternate methods.

In our final experiment we explore wether ENT_{EB} can be used to characterise real world graphs that are constructed according to different network models. We have already shown in our previous experiments that ENT_{EB} can be used as an effective tool to distinguish synthetic graphs that are generated according to different network models. We have also demonstrated the effectiveness of ENT_{EB} in characterising the time evolving networks. We now show that ENT_{EB} can also be used to distinguish between Delaunay triangulations and Gabriel graphs. For this purpose, we extract Delaunay triangulations and Gabriel graphs for each of the 72 views of all the 20 objects. Next we compute ENT_{EB} for each of the extracted graphs. Figure 8 plots the resulting values of ENT_{EB}. Here the x-axis represents the ordinal number for different object views. Here, all the 72 views of each of the objects are arranged in blocks for each object in turn.

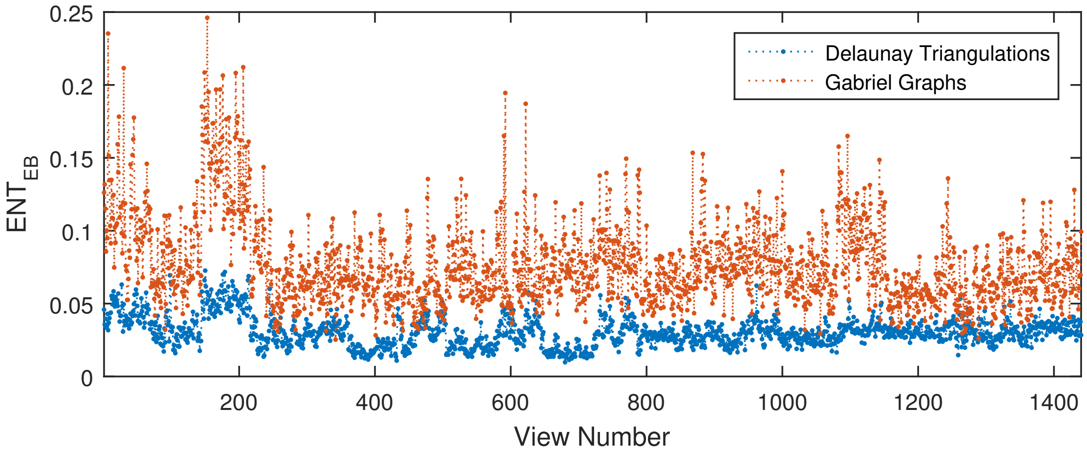


FIG. 8. Entropy traces for Del and GG

There are a number of important observations that can be drawn from the data plotted in Figure

8. First, the values of ENT_{EB} are considerably lower for Delaunay triangulations than for Gabriel graphs. This suggests that a Delaunay triangulation has a more regular structure as compared to a Gabriel graph. Note that, as mentioned earlier, the edges of the Gabriel graph are a subset of those contained in the Delaunay triangulation. It also has a smaller number of triangles as compared to a Delaunay triangulations. Moreover, a Gabriel graphs may also have branches (i.e., nodes with degree one) whilst Delaunay triangulations are graphs where each vertex is of degree at least 2. For these reasons, a Gabriel graph is expected to have a more irregular structure when compared to a Delaunay triangulation. This shows that the value of ENT_{EB} can be used as an effective tool to measure the complexity of a network. Secondly, It is also worth noting that the values of ENT_{EB} for Delaunay triangulations exhibits small intra-class variation as compared to Gabriel graphs. This is again due to the fact that Delaunay triangulations have a more regular structure and are expected to have same complexity for all the views of the same object. Finally, the values of ENT_{EB} for Delaunay triangulations are also very consistent and exhibit small variations across different classes. This suggests that our method is consistent in detecting the structure of real world networks.

4.5 *Bioinformatics*

In our this set of experiments we apply the proposed method on graphs extracted from bioinformatics datasets. For this purpose we choose 7 different publicly available benchmark datasets[1]. A brief description of each of the dataset is given below.

MUTAG [44]: The Mutagenesis dataset consists of a set of chemical compounds. The data consists of two classes of compounds, one of which produces mutagenic activity and one of which does not. The goal, from the point of view of classification, is to identify the mutation-causing molecules from their structure. There are 125 chemicals in the active class and 63 in the inactive class. The average number of nodes and the edges of these graphs is 17.93 and 19.79 respectively.

PTC [24]: PTC (Predictive Toxicology Challenge) dataset contains organic molecules marked on the basis of carcinogenicity on male and female mice and rats. The dataset consists of male mice (MM), male rate (MR), female mice (FM) and female rat (FR). The molecules are represented using graphs where atoms are represented by nodes and chemical bonds are represented by edges. The task here is to classify each graph into its corresponding class. There are 336 graphs in PTC-MM, 344 in PTC-MR, 349 in PTC-FM and 351 in PTC-FR.

AIDS [43]: In this dataset the molecular compounds are represented as graphs. the compounds are taken from AIDS Antiviral Screen Database of Active compounds. Atoms in the molecular compounds are represented as nodes and the covalent bonds are represented as edges. The dataset consists of two classes (active and inactive). The active class represents the molecules with activity against HIV and the other represents the molecules with no activity against HIV. Here we have to classify each graph into its corresponding class. There are 2000 graphs in the dataset in which 1600 are representing inactive elements and 400 representing active elements.

NCI1 [52]: This dataset consist of graphs representing two balanced subsets of chemical compounds. This dataset is screened for activity against cell lung cancer and ovarian cancer. There are 4110 graphs in NCI1. The average number of nodes of graphs is 29.87 and the average number of edges is 32.30.

FRANKENSTEIN [37]: Frankenstein is the modified version of the BURSI [26] dataset and is created by fusion with the MNIST dataset. There are 4337 molecules in this dataset with 2401 mutagens and 1936 nonmutagens.

[1] These datasets can be downloaded from https://ls11-www.cs.tu-dortmund.de/staff/morris/graphkerneldatasets

Tox 21 [29]: These graphs are derived from the data published by the National Center for Advancing Translational Sciences in the context of the Tox21 Data Challenge 2014 [2]. These datasets each contain more than 7000 graphs and the goal here is to asses the performance of the proposed method on larger datasets. In this paper we have selected two datasets, i.e., Tox-AHR and Tox-MMP.

In order to classify graphs in each of these datasets, we convert each graph into a normalised feature vector as described in Definition 3.4. In order to compare two graphs, we have used the first derivative of the entropy curve. For comparison purposes, we also construct a normalised feature vector, $EntV$, that uses the vertex-based Laplacian instead of the edge-based Laplacian. Further, we have also compared the proposed method with a number of alternative state-of-the-art methods, including graphlet kernel (GK) [40], the random walk kernel (RWK) [20], the shortest path kernel (SPK) [7], and ShapeDNA [42]. To estimate the classification accuracies of the proposed and alternate methods, we have applied 10-fold cross validation using a Support Vector Machine (SVM) with Gaussian kernel. The results are reported in Table 3.

Datasets	**Proposed**	**EntV**	**GK**	**RWK**	**SPK**	**ShapeDNA**
MUTAG	89.88 ± 2.54	84.56 ± 2.56	85.06 ± 2.76	86.20 ± 2.29	85.64 ± 2.22	87.87 ± 2.93
AIDS	98.51 ± 0.25	95.60 ± 0.33	96.45 ± 0.46	90.75 ± 0.45	99.45 ± 0.18	92.65 ± 0.51
NCI1	67.51 ± 0.92	64.96 ± 0.95	68.74 ± 0.63	61.78 ± 0.56	69.78 ± 0.93	68.35 ± 0.62
PTC-MR	61.55 ± 2.07	56.99 ± 2.12	59.77 ± 1.68	59.86 ± 1.97	60.14 ± 1.75	57.59 ± 1.95
PTC-MM	66.18 ± 2.24	63.65 ± 2.56	65.48 ± 1.62	62.86 ± 3.04	63.76 ± 2.51	65.48 ± 1.67
PTC-FR	68.57 ± 1.20	65.25 ± 1.61	68.57 ± 1.04	66.95 ± 1.96	66.39 ± 1.73	67.25 ± 1.11
PTC-FM	61.50 ± 2.22	59.34 ± 2.44	59.17 ± 2.60	59.89 ± 1.63	60.73 ± 1.39	60.72 ± 1.96
Frankenstein	67.93 ± 0.73	65.46 ± 0.89	65.52 ± 0.54	65.23 ± 0.48	67.86 ± 0.69	63.50 ± 0.75
Tox-AHR	88.87 ± 0.10	88.76 ± 0.10	88.76 ± 0.17	88.47 ± 0.16	88.96 ± 0.11	89.28 ± 0.10
Tox-MMP	85.91 ± 0.12	85.86 ± 0.14	85.27 ± 0.24	85.15 ± 0.22	85.88 ± 0.12	85.77 ± 0.26

Table 3. Experimental Results on different bioinformatics datasets and methods

Results show that the proposed method gives superior performance compared to alternative methods on most of the datasets. It is important to note that the proposed feature vector which is constructed instead using the *edge-based Laplacian* always outperforms an analogous feature vector which is constructed using the traditional *vertex-based Laplacian*. The difference between the classification accuracies of these two methods is also significant in most cases. This demonstrates the power of using the edge-based Laplacian over its vertex-based counterpart.

4.6 *Functional Brain Network Analysis Data*

In our final set of experiments with real-world datasets, we evaluate the performance of the proposed edge entropy feature vector on functional brain network analysis data [38] constructed from the whole brain functional magnetic resonance image (fMRI) atlas [8]. Here the raw data consists of voxel based BOLD (Blood-oxygen-level-dependent) images obtained sampled using fMRI over time. The fMRI brain volume is mapped into a network where nodes correspond to volumetric regions of interest (ROI). The edges indicate that the aggregate time varying BOLD signals for the two ROIs are strongly correlated. In order to discover relationships between ROIs, the mean values of the BOLD signal in each

[2] https://tripod.nih.gov/tox21/challenge/

ROI are recorded with respect to certain voxel time courses. By using Pearson correlations between two time courses, we can calculate correlation between two ROIs. Thus a graph is constructed by connecting ROIs whose correlations are higher than a threshold value [38]. This is a binary classification problem and the task here is to diagnose Attention Deficit Hyperactivity Disorder (ADHD) in the brain. We have used three different datasets collected from three different centres, i.e., Kennedy Krieger Institute (**KKI**), Oregon Health and Science University (**OHSU**), and Peking University (**PU**). There are 83 networks in the KKI dataset. The average number of nodes and edges in KKI are 26.96 and 48.42 respectively. The PU dataset has 85 networks. In this dataset, the average number of nodes and edges are 39.31 and 77.35 respectively. Finally, the OHSU dataset contains 79 networks, where average number of nodes and edges of the networks are 82.01 and 199.66 respectively.

As with bioinformatics datasets, for each dataset, we have converted each graph into feature vector using our proposed method. To classify those feature vectors, we have applied 10-fold cross validation using SVM with a Gaussian kernel and estimated the classification accuracies of the proposed method on all the three datasets. To compare the performance of the proposed method, similar experiments were performed with alternate methods. The results are reported in Table 4.

Datasets	**Proposed**	**EntV**	**GK**	**RWK**	**SPK**	**ShapeDNA**
KKI	56.81 ± 2.13	54.58 ± 2.24	46.94 ± 1.76	55.56 ± 2.05	50.83 ± 2.96	54.72 ± 2.53
PU	57.92 ± 2.63	39.03 ± 3.28	49.44 ± 3.76	57.64 ± 2.38	57.36 ± 2.59	56.14 ± 3.11
OHSU	68.75 ± 5.02	67.32 ± 2.62	64.46 ± 6.17	54.29 ± 4.4	54.46 ± 4.94	54.64 ± 5.83

Table 4. Experimental Results on different bioinformatics datasets and methods

Results reported in Table 4 show that the proposed method can classify brain networks with higher accuracy when compared to alternate methods. Note that all the other methods give poor performance on at least one of the brain network dataset. The shows the usefulness of the proposed method in classifying brain networks.

5. Conclusion and Future Work

In this paper we have presented a novel framework to characterise the structural complexity of a network represented by a quantum graph. Our definition of the entropy is based on the idea of a diffusion process defined on the edges of a graph. Based on the resulting diffusion process, we have defined a novel measure called the ENT_{EB} to gauge the structural content of a network. We have also demonstrated that the entropy computed at different time scales can be used as a tool to distinguish between different networks. Here we have constructed a feature vector whose components are composed of entropy values computed using the edge-based diffusion process at different time scales. To show the effectiveness of the proposed method, we have performed numerous experiments on both synthetic and real world data. Results show that the proposed method can be used to characterise networks with higher accuracy.

There are a number of directions in which the work reported here can be extended. Firstly, since the entropy computed here is based on edge-based information functional, the work can be easily extended to edge-weighted graphs or edge-labeled graphs. Secondly, since the proposed method based on heat diffusion process defined using the edge-based Laplacian of a graph, it would be interesting to define the entropy of a graph using the solutions of other partial differential equations such as edge-based wave equation. Finally, from an application perspective, it would be interesting to explore the applications of the entropy-based methods developed here in other domains such as 3D computer vision.

References

[1] ANAND, K. & BIANCONI, G. (2009) Entropy measures for networks: Toward an information theory of complex topologies. *Phys. Rev. E*, **80**, 045102.

[2] AZIZ, F., HANCOCK, E. R. & WILSON, R. C. (2016) Graph Entropy from Closed Walk and Cycle Functionals. in *Structural, Syntactic, and Statistical Pattern Recognition*, pp. 174–184.

[3] AZIZ, F., WILSON, R. & HANCOCK, E. (2012) Shape Signature using the Edge-Based Laplacian. *International Conference on Pattern Recognition*.

[4] AZIZ, F., WILSON, R. C. & HANCOCK, E. R. (2013) Backtrackless Walks on a Graph. *IEEE Transactions on Neural Networks and Learning Systems*, **24**(6), 977–989.

[5] AZIZ, F., WILSON, R. C. & HANCOCK, E. R. (2018) A wave packet signature for complex networks. *Journal of Complex Networks*, **7**(3), 346–374.

[6] BARABÁSI, A. & ALBERT, R. (1999) Emergence of Scaling in Random Networks. *Science*, pp. 509–512.

[7] BORGWARDT, K. M. & KRIEGEL, H. P. (2005) Shortest-path kernels on graphs. in *Fifth IEEE International Conference on Data Mining*, pp. 8–pp. IEEE.

[8] CRADDOCK, R. C., JAMES, G., HOLTZHEIMER III, P. E., HU, X. P. & MAYBERG, H. S. (2012) A whole brain fMRI atlas generated via spatially constrained spectral clustering. *Human Brain Mapping*, **33**(8), 1914–1928.

[9] DEHMER, M. (2008) Information processing in complex networks: Graph entropy and information functionals. *Applied Mathematics and Computation*, **201**, 82–94.

[10] DELAUNAY, B. (1934) Sur la sphére vide. *Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk*, pp. 793–800.

[11] DUTTA, A., LLADÓS, J., BUNKE, H. & PAL, U. (2018) Product graph-based higher order contextual similarities for inexact subgraph matching. *Pattern Recognition*, **76**, 596 – 611.

[12] ERDÕS, P., R. (1960) A.: On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, pp. 17–61.

[13] ESTRADA, E. (2000) Characterization of 3D molecular structure. *Chemical Physics Letters*, **319**(5), 713 – 718.

[14] ESTRADA, E., DE LA PEÑA, J. A. & HATANO, N. (2014) Walk entropies in graphs. *Linear Algebra and its Applications*, **443**, 235 – 244.

[15] ESTRADA, E. & HATANO, N. (2007) Statistical-mechanical approach to subgraph centrality in complex networks. *Chemical Physics Letters*, **439**, 247–251.

[16] ESTRADA, V. (2002) Characterization of the folding degree of proteins. *Bioinformatics*, **18**(5), 697–704.

[17] FERRAZ DE ARRUDA, H., NASCIMENTO SILVA, F., QUEIROZ MARINHO, V., RAPHAEL AMANCIO, D. & DA FONTOURA COSTA, L. (2017) Representation of texts as complex networks: a mesoscopic approach. *Journal of Complex Networks*, **6**(1), 125–144.

[18] FRIEDMAN, J. & TILLICH, J. P. (2004a) Calculus on Graphs. *CoRR*.

[19] FRIEDMAN, V. & TILLICH, J. P. (2004b) Wave equations for graphs and the edge based Laplacian. *Pacific Journal of Mathematics*, pp. 229–266.

[20] GÄRTNER, T., FLACH, P. & WROBEL, S. (2003) On graph kernels: Hardness results and efficient alternatives. in *Learning Theory and Kernel Machines*, pp. 129–143. Springer.

[21] GUTMAN, I. & ZHOU, B. (2006) Laplacian energy of a graph. *Linear Algebra and its Applications*, **414**(1), 29 – 37.

[22] HAN, L., ESCOLANO, F., HANCOCK, E. & WILSON, R. (2102) Graph characterizations from von Neumann entropy. *Pattern Recognition Letters*, pp. 1958–1967.

[23] HARRIS, C. & STEPHENS, M. (1988) A combined corner and edge detector. *In Fourth Alvey Vision Conference, Manchester, UK*, pp. 147–151.

[24] HELMA, C., KING, R. D., KRAMER, S. & SRINIVASAN, A. (2001) The predictive toxicology challenge 2000–2001. *Bioinformatics*, **17**(1), 107–108.

[25] HOLME, P. & KIM, B. J. (2002) Growing scale-free networks with tunable clustering. *Phys. Rev. E*, **65**, 026107.

[26] KAZIUS, J., MCGUIRE, R. & BURSI, R. (2005) Derivation and Validation of Toxicophores for Mutagenicity Prediction. *Journal of Medicinal Chemistry*, **48**(1), 312–320.

[27] KIM, J. & HASTAK, M. (2018) Social network analysis: Characteristics of online social networks after a disaster. *International Journal of Information Management*, **38**(1), 86–96.

[28] KÖRNER, J. (1973) Coding of an information source having ambiguous alphabet and the entropy of graphs. *6th Prague conference on information theory*.

[29] KRIEGE, N. M., JOHANSSON, F. D. & MORRIS, C. (2020) A survey on graph kernels. *Applied Network Science*, **5**(1), 6.

[30] KUCHMENT, P. (2008) Quantum graphs: an introduction and a brief survey Analysis on Graphs and its applications. *Proc. Symp. Pure Math. (Providence, RI: American Mathematical Society)*, pp. 291–314.

[31] KUHN, T., PERC, M. C. V. & HELBING, D. (2014) Inheritance Patterns in Citation Networks Reveal Scientific Memes. *Phys. Rev. X*, **4**, 041036.

[32] KUNEGIS, J. (2013) KONECT - The Koblenz Network Collection. *In Proc. Int. Conf. on World Wide Web Companion*, pp. 1343–1350.

[33] MORRIS, C., KRIEGE, N. M., KERSTING, K. & MUTZEL, P. (2016) Faster Kernels for Graphs with Continuous Attributes via Hashing. in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 1095–1100.

[34] NAYAR, S., NENE, S. & MURASE, H. (1996) Columbia object image library (coil 100). *Department of Comp. Science, Columbia University, Tech. Rep. CUCS-006-96*.

[35] NEWMAN, M. (2018) *Networks*. Oxford University Press.

[36] NEWMAN, M. & WATTS, D. (1999) Renormalization group analysis of the small-world network model. *Physics Letters A*, **263**(4), 341 – 346.

[37] ORSINI, F., FRASCONI, P. & DE RAEDT, L. (2015) Graph Invariant Kernels. in *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, p. 3756–3762. AAAI Press.

[38] PAN, S., WU, J., ZHU, X., LONG, G. & ZHANG, C. (2017) Task Sensitive Feature Exploration and Learning for Multitask Graph Classification. *IEEE Transactions on Cybernetics*, **47**(3), 744–758.

[39] PASSERINI, F. & SEVERINI, S. (2008) The von Neumann entropy of networks. *arXiv preprint arXiv:0812.2597*.

[40] PRŽULJ, N., CORNEIL, D. & JURISICA, I. (2004) Modeling Interactome: Scale-free or Geometric?. *Bioinformatics*, **20**(18), 3508–3515.

[41] REN, P., WILSON, R. C. & HANCOCK, E. R. (2011) Graph Characterization via Ihara Coefficients. *IEEE Transactions on Neural Networks*, **22**, 233–245.

[42] REUTER, M., WOLTER, F.-E. & PEINECKE, N. (2006) Laplace-Beltrami spectra as 'Shape-DNA' of surfaces and solids. *Computer-Aided Design*, **38**(4), 342 – 366.

[43] RIESEN, K. & BUNKE, H. (2008) IAM Graph Database Repository for Graph Based Pattern Recognition and Machine Learning. in *Structural, Syntactic, and Statistical Pattern Recognition*, pp. 287–297. Springer Berlin Heidelberg.

[44] SRINIVASAN, A., MUGGLETON, S. H., STERNBERG, M. J. E. & KING, R. D. (1996) Theories for mutagenicity: a study in first-order and feature-based induction. *Artificial Intelligence*, pp. 277–299.

[45] STEGER, A. & WORMALD, N. C. (1999) Generating Random Regular Graphs Quickly. *Combinatorics, Probability and Computing*, **8**(4), 377–396.

[46] SUN, J., OVSJANIKOV, M. & GUIBAS, L. (2009) A Concise and Provably Informative Multi-scale Signature Based on Heat Diffusion. in *Proceedings of the Symposium on Geometry Processing*, pp. 1383–1392.

[47] SVERCHKOV, Y. & CRAVEN, M. (2017) A review of active learning approaches to experimental design for uncovering biological networks. *PLOS Computational Biology*, **13**(6), 1–26.

[48] TRAVIESO, G., RUGGIERO, C. A., BRUNO, O. M. & DA F. COSTA, L. (2013) Predicting efficiency in master–slave grid computing systems. *Journal of Complex Networks*, **1**(1), 63–71.

[49] TSITSULIN, A., MOTTIN, D., KARRAS, P., BRONSTEIN, A. & MÜLLER, E. (2018) NetLSD: Hearing the Shape of a Graph. in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery &#38; Data Mining*, pp. 2347–2356. ACM.

[50] DEHMER, M. (2008) A Novel Method for Measuring the Structural Information Content of Networks. *Cybernetics and Systems*, **39**, 825–842.

[51] ESTRADA, E. (2010) Quantifying network heterogeneity. *Phys. Rev. E*, **82**, 066102.

[52] WALE, N., WATSON, I. A. & KARYPIS, G. (2008) Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowledge and Information Systems*, **14**(3), 347–375.

[53] WANG, J., WILSON, R. C. & HANCOCK, E. R. (2017) Spin statistics, partition functions and network entropy. *Journal of Complex Networks*, **5**(6), 858–883.

[54] WATTS, D. J. & STROGATZ, S. (1998) Collective dynamics of 'small-world' networks. *Nature*, pp. 440–442.

[55] WILSON, R. C., AZIZ, F. & HANCOCK, E. R. (2013) Eigenfunctions of the Edge-Based Laplacian on a Graph. *Linear Algebra and its Applications*, **438**(11), 4183–4189.

[56] XU, L., JIANG, X., BAI, L., XIAO, J. & LUO, B. (2018) A hybrid reproducing graph kernel based on information entropy. *Pattern Recognition*, **73**, 89 – 98.

[57] YE, C., COMIN, C. H., PERON, T. K. D., SILVA, F. N., RODRIGUES, F. A., COSTA, L. D. F., TORSELLO, A. & HANCOCK, E. R. (2015) Thermodynamic characterization of networks using graph polynomials. *Phys. Rev. E*, **92**, 032810.