Graph-based Identification and Authentication: A Stochastic Kronecker Approach

Shengmin Jin, *Student Member, IEEE,* Vir V. Phoha, *Senior Member, IEEE,* and Reza Zafarani, *Member, IEEE*

Abstract—A large body of research has focused on analyzing large networks and graphs. However, network and graph data is often anonymized for reasons such as protecting data privacy. Under such circumstances, it is difficult to verify the source of network data, which leads to questions such as: Given an anonymized graph, can we identify the network from which it is collected? Or if one claims the graph is sampled from a certain network, can we verify it? The intuitive approach is to check for subgraph isomophism. However, subgraph isomophism is NP-complete; hence, infeasible for most large networks. Inspired by biometrics studies, we address these challenges by formulating two new problems: *network identification* and *network authentication*. To tackle these problems, similar to research on human fingerprints, we introduce two versions of a *network identity*: (1) embedding-based identity and (2) distribution-based identity. We demonstrate the effectiveness of these network identities on twenty real-world networks. Using these identities, we propose two approaches for network identification. One method uses supervised learning and can achieve an identification accuracy of 84.4%, and the other, which is easier to implement, relies on distances between identities and achieves an accuracy rate of 70.8%. For network authentication, we propose two methods to build a network authentication system. The first is a supervised learner and provides a low false accept rate and the other method allows one to control the false reject rate with a reasonable false accept rate across networks. We demonstrate that network authentication can also be used for biometrics, authenticating users based on their touch data on phones and tablets. Our study can help identify or verify the source of network data, validate network-based research, and be used for network-based biometrics.

Index Terms—Network Identification, Network Authentication, Network Representation Learning, Network Embedding

1 Introduction

Networks are everywhere, from science and engineering (e.g. protein-protein interaction networks, technological networks) to our daily life (e.g. social networks, road networks), motivating research on graphs and networks. Networks have been used to study friendships in social networks [1], to gain insights into the meaning of biological networks [2], and many other phenomena. In spite of this remarkable progress, networks research has often been conducted on anonymized graphs, especially for social networks, as data privacy is critical. To protect the users' privacy while preserving network properties, anonymization techniques have been widely used before publishing social network data [3], [4]. To validate the authenticity of such anonymized graphs, it is natural to ask questions such as: Given a large graph G, can we verify that it is a Facebook graph but not collected from Twitter or a biological network? Can we identify the source of an anonymized network, i.e., its *network identity*? To answer these questions, the first natural solution that comes to mind is to check whether a network contains a subgraph that is isomorphic to G. The problem is called *subgraph isomorphism*, and is known to be NP-complete [5], so solving it is infeasible for most large networks. Hence, we need an alternative solution that is reasonably accurate and highly efficient.

Problem Formulation. To identify a person, two types of systems have been designed in biometrics literature: (1) *identification* systems and (2) *authentication* systems [6]. An identification system recognizes a subject without the subject claiming an identity, i.e., "Who am I?". It tries to match the subject with everyone enrolled in the system database and obtains the best match. On the other hand, an authentication system either rejects or accepts the submitted claim of identity, i.e., "Am I who I claim I am?". In spite of their differences, sometimes the terms authentication and identification are used interchangeably [6]. Inspired by biometrics research, we formulate two new problems:

1) **Network Identification**. Given a set of networks $N = \{N_1, N_2, ..., N_n\}$, and a subgraph G sampled from $N_i \in N$ using sampling strategy S, we want to identify G, i.e., the network N_i from which G is sampled.
2) **Network Authentication** (or *network identity-authentication*). Subgraph G is claimed to be sampled from a certain network N_i via sampling strategy S. The authentication system either accepts or rejects this claim.

Following the problem formulation, our first aim is to build an identity to represent a network, similar to how a fingerprint represents a person. We propose two ways to build a network identity:

1) **Embedding-based Identity**. Intuitively, one can represent a network using a feature vector or its graph embedding. Graph embedding methods aim to map a graph into a low-dimensional vector that preserves the network structure [7]. Hence, one can represent the identity of a network N_i with its embedding, and match the embedding of subgraph G with that of other networks.
2) **Distribution-based Identity**. One limitation of the embedding-based identity is that it is not unique, as graph embedding methods generally do not guarantee uniqueness for different networks. Hence, inspired by *ridge-based representation* [8] for fingerprints, we propose

distribution-based identity. The ridge-based representation is one of the most widely-used representations for fingerprints and it is based on the hypothesis that ridge structures (*minutiae*, e.g. ridge ending and ridge bifurcation) and their distributions are distinct across fingerprints. It inspires us to, instead of using one embedding, represent a network identity as the distribution of embedding values for subgraphs of a network, so that the identity is unique and can preserve subgraph information.

The Present Work. We introduce network identification and network authentication with the following contributions:

1) **Network Identity**. We introduce the *network identity* and two identity types: *embedding-based identity* and *distribution-based identity*. We prove that the embedding-based identity can capture graph structure information and/or other relationships between samples (subgraphs) and the source network. We demonstrate that the distribution-based identities are unique by showing that for real-world networks the similarity among such identities for various networks is generally low. Our distribution-based identities are visualizable in 3D and are easy to interpret; hence, we show examples on how the structural differences in networks are reflected in their identities. We compare the two types of identities in both identification/authentication problems.
2) **Network Identification**. We introduce two methods to predict the network from which a graph is sampled using the developed network identities. The first is a supervised learning method, which is highly accurate (84.4%). We also introduce an easier to implement method that relies on the distances between the sample embedding to the network identities, achieving a 70.8% accuracy.
3) **Network Authentication**. We propose two techniques to solve the network authentication problem: a *supervised splitter*, which has a low equal error rate, and a *Voronoi splitter*, which allows controlling the false reject with an acceptable false accept rate across networks.

The rest of the paper is organized as follows. In Section 2, we introduce two types of network identities, and connect the identities with the relationships between the sample subgraphs and source network. The data used in our experiments is summarized in Section 3. We discuss the uniqueness of identities and partial identity in Section 4. We propose methods to solve network identification in Section 5, and network authentication in Section 6. An application in biometrics is explored in Section 7 and the limitations of our work is discussed in Section 8. We review the related work in Section 9 and conclude in Section 10.

2 Network Identity

The first step to identify a graph is to build its identity. We propose two types of network identities: (1) embedding-based identity and (2) distribution-based identity.

2.1 Embedding-based Identity

In theory, any embedding method that can preserve network structural information and the similarity between samples (subgraphs) and the network from which they are sampled from can be used as an embedding-based identity. Here, we choose *Kronecker points* as the embedding method and prove its utility for both network authentication and identification.

Stochastic Kronecker Graphs and Kronecker Points. Stochastic Kronecker Graphs (SKG) [9] is a network model for large-scale graphs based on the *Kronecker product* $\otimes$ matrix operation. Starting from a small probability matrix $\Theta \in \mathbb{R}^{n\times n}$, known as the *Kronecker initiator matrix*, we can get a large probability matrix $\mathcal{P}$ with the k^{th} Kronecker power of Θ, i.e., $\mathcal{P} = \Theta^{\otimes k} = \underbrace{\Theta \otimes \Theta \cdots \otimes \Theta}_{k \text{ times}}$. Matrix $\mathcal{P}$ can be used to generate an adjacency matrix. When modeling a network with SKG, we aim to learn a Θ which is most likely to have generated its adjacency matrix $A \in \mathbb{R}^{n^k \times n^k}$, i.e., $P(A|\mathcal{P})$ is maximized. The KronFit algorithm can estimate the Kronecker initiator matrix in linear time using maximum likelihood over all the node correspondence permutations (for details refer to Ref. [9]). If one fits a 2×2 Kronecker initiator matrix $\Theta = \left[\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right]$ to an undirected graph, whose adjacency matrix is symmetric, the learned Kronecker initiator matrix will be symmetric, too, i.e., $b = c$. Hence, one can embed an undirected graph to a point (a, b, d) in the 3D space, and the point is denoted as the *Kronecker point* of a graph [10]. Kronecker initiator matrices are probability matrices, so values a, b and d are all between 0 and 1; hence, all possible graphs can be embedded in a $1 \times 1 \times 1$ cube.

Kronecker Points and Graph Structure. We can interpret the 2×2 initiator $\left[\begin{smallmatrix} a & b \\ b & d \end{smallmatrix}\right]$ of an undirected network as a recursive expansion of two groups of network nodes into subgroups [9]. Values a and d represent the proportion of edges within each of these two groups, and value b represents the proportion of edges between the two groups. Based on the order of these values (e.g., $a > b > d$ or $b > a > d$), one can obtain whether a network has a core-periphery, dual-core, or a random structure [10].

Kronecker Points and Graph Similarity. Graph kernels have been traditionally used to measure the similarity between two graphs [11]. Here, we prove that if Kronecker points of two graphs are more similar (or closer) in the 3D space, the graphs are expected to have higher similarity in terms of the random-walk graph kernel between them [12].

Theorem 2.1 (Kronecker Initiator and Graph Kernel)**.** *For graphs G_1 and G_2 generated by probability matrices $\mathcal{P}_1$ and $\mathcal{P}_2$, where $\mathcal{P}_1$ and $\mathcal{P}_2$ are the k^{th} Kronecker power of Kronecker initiator matrices Θ_1 and Θ_2, i.e., $\mathcal{P}_1 = \Theta_1^{\otimes k}, \mathcal{P}_2 = \Theta_2^{\otimes k}$, the expected random-walk graph kernel between G_1 and G_2 is lower bounded by the product of the ℓ_1-norm of Θ_1 and Θ_2, and the product of the sizes of G_1 and G_2:*

$$\mathbb{E}(\mathcal{K}(G_1, G_2)) \geq (|G_1||G_2| + |\lambda|(\|\Theta_1\|_1\|\Theta_2\|_1)^k)^{-1}, \quad (1)$$

where λ denotes the decay factor of the graph kernel.

Proof. Random-walk graph kernel performs random walks on both graphs and counts the number of matching walks, discounting longer walks. Random-walk graph kernel [12] is formulated as $\mathcal{K}(G_1, G_2) = \frac{1}{|G_1||G_2|}\boldsymbol{e}^T(I - \lambda A_\times)^{-1}\boldsymbol{e}$, where $\boldsymbol{e}$ denotes the all 1 vector, λ denotes the decay factor of the graph kernel, I is the identity matrix, and $A_\times$ is the

adjacency matrix of the direct product graph of G_1 and G_2, i.e., $A_\times = A_{G_1} \otimes A_{G_2}$. Hence, $I, A_\times \in \mathbb{R}^{|G_1||G_2|\times|G_1||G_2|}$,

$$\begin{aligned}\mathcal{K}(G_1,G_2) &= \frac{1}{|G_1||G_2|}\boldsymbol{e}^T(I-\lambda A_\times)^{-1}\boldsymbol{e}\\ &= \frac{1}{|G_1||G_2|}\|(I-\lambda A_\times)^{-1}\|_1\\ &\geq \frac{1}{|G_1||G_2|}\frac{\|I\|_1}{\|I-\lambda A_\times\|_1}\\ &= \frac{1}{\|I-\lambda A_\times\|_1}\\ &\geq \frac{1}{\|I\|_1+|\lambda|\|A_\times\|_1}.\end{aligned}$$

As $\|A_\times\|_1 = \|A_{G_1}\otimes A_{G_2}\|_1 = \|A_{G_1}\|_1\|A_{G_2}\|_1$ and $\mathbb{E}(\|A_{G_1}\|_1) = \|\Theta_1\|_1^k$ and $\mathbb{E}(\|A_{G_2}\|_1) = \|\Theta_2\|_1^k$, using Jensen's inequality, we get

$$\begin{aligned}\mathbb{E}(\mathcal{K}(G_1,G_2)) &\geq \mathbb{E}(\frac{1}{\|I\|_1+|\lambda|\|A_\times\|_1})\\ &\geq (|G_1||G_2|+|\lambda|(\|\Theta_1\|_1\|\Theta_2\|_1)^k)^{-1}.\end{aligned}$$ □

Corollary 2.1.1 (Kronecker Points and Graph Kernel)**.** *In Theorem 2.1, when Θ_1 and Θ_2 are 2×2 Kronecker initiator matrices, $\mathbb{E}(\mathcal{K}(G_1,G_2)) \geq (|G_1||G_2|+|\lambda|(\langle\Theta_1,\Theta_2\rangle_F+\frac{3}{2}(\|\Theta_1\|_2^2+\|\Theta_2\|_2^2))^k)^{-1}$, where $\langle\cdot,\cdot\rangle_F$ denotes Frobenius inner product.*

Proof. Assume $\Theta_1 = \begin{bmatrix}a_1 & b_1\\ c_1 & d_1\end{bmatrix}$ and $\Theta_2 = \begin{bmatrix}a_2 & b_2\\ c_2 & d_2\end{bmatrix}$. $\|\Theta_1\|_1\|\Theta_2\|_1 = (a_1+b_1+c_1+d_1)(a_2+b_2+c_2+d_2) = \sum_{x\in\{a,b,c,d\}} x_1x_2 + \sum_{\substack{x,y\in\{a,b,c,d\}\\ x\neq y}} x_1y_2 \leq \langle\Theta_1,\Theta_2\rangle_F + \sum_{\substack{x,y\in\{a,b,c,d\}\\ x\neq y}} \frac{x_1^2+y_2^2}{2} = \langle\Theta_1,\Theta_2\rangle_F + \frac{3}{2}(\|\Theta_1\|_2^2+\|\Theta_2\|_2^2)$. □

Corollary 2.1.1 indicates that two graphs are expected to be more similar if their Kronecker initiator matrices have larger inner products.

Kronecker Points of Sample Subgraphs. Next, we demonstrate that Kronecker points can preserve the relationships between sampled subgraphs and the network from which they are sampled. In Theorem 2.2, we prove that if a network and its subgraph are perfectly fitted by two Kronecker initiator matrices, then the euclidean distance between their Kronecker points is well bounded, corroborating previous empirical findings that Kronecker points of large sampled subgraphs are close to that of the whole network [10]. Theorem 2.3 gives the error bound of the fitting process using KRONFIT algorithm.

Theorem 2.2 (Kronecker Points of Samples)**.** *For network $G=(V,E), |V|=2^k$ generated by a Stochastic Kronecker graphs probability matrix $\mathcal{P}=\Theta^{\otimes k}$ and its subgraph G_s sampled using Random Node Sampling with sampling proportion p where $p>0.5$,[1] the expected ℓ_1-norm of the difference of their adjacency matrices is $\mathbb{E}(\|A_G-A_{G_s}\|_1) = (1-p^2)\|\Theta\|_1^k$.*

Proof. As G is generated by $\mathcal{P}$, $\mathbb{E}(\|A_G\|_1) = \|\mathcal{P}\|_1$. For random node sampling [14] with proportion p, we can consider sampling a subgraph from G as removing $|V|(1-p)$ rows and columns uniformly from $\mathcal{P}$ to get a sub-matrix $\mathcal{P}_s$ and using $\mathcal{P}_s$ to generate the subgraph. Therefore,

$$\begin{aligned}\mathbb{E}(\|A_G-A_{G_s}\|_1) &= \mathbb{E}(\|\mathcal{P}-\mathcal{P}_s\|_1)\\ &= \underbrace{2|V|(1-p)\frac{\|\Theta\|_1^k}{|V|}}_{\text{removed rows and columns}}\\ &\quad \underbrace{-|V|(1-p)\frac{\|\Theta\|_1^k}{|V|^2}}_{\text{removed diagonals}}\\ &\quad \underbrace{-2\binom{|V|(1-p)}{2}\frac{\|\Theta\|_1^k}{|V|^2}}_{\text{removed } \mathcal{P}_{ij} \text{ where } i\neq j}\\ &= (1-p^2)\|\Theta\|_1^k.\end{aligned}$$ □

Corollary 2.2.1. *If a network G and its subgraph G_s are perfectly fitted by two Kronecker initiator matrices Θ and Θ_s, $\|\Theta-\Theta_s\|_1 = \sqrt[k]{1-p^2}\|\Theta\|_1$.*

Corollary 2.2.1 bounds the distance between Kronecker points of a graph and its subgraph, assuming perfect fit. It also indicates that the Kronecker points of small subgraphs can be far away from the source network. However, a real-world graph can rarely be perfectly fit by Kronecker initiators, so Theorem 2.3 we will discuss the bounds on fitting error.

Theorem 2.3 (Error Bound on Fitting Real-World Graphs)**.** *For graph $G=(V,E)$ with $|V|=2^k$, fitting the most likely Kronecker initiator matrix Θ provides an upper bound on the expected error $\mathbb{E}(\|A_G-\mathcal{P}\|_1)$.*

Proof. Denote σ as a node mapping from A_G to $\mathcal{P}$ and $(A_G-\mathcal{P})_\sigma$ the difference between A_G and $\mathcal{P}$, given σ. Then,

$$\begin{aligned}&\|(A_G-\mathcal{P})_\sigma\|_1 = \sum_{(u,v)\in E}(1-\mathcal{P}[\sigma_u,\sigma_v]) + \sum_{(u,v)\notin E}\mathcal{P}[\sigma_u,\sigma_v]\\ &= |V|^2 - (\sum_{(u,v)\in E}\mathcal{P}[\sigma_u,\sigma_v] + \sum_{(u,v)\notin E}(1-\mathcal{P}[\sigma_u,\sigma_v]))\\ &\leq |V|^2 - |V|^2(\prod_{(u,v)\in E}\mathcal{P}[\sigma_u,\sigma_v]\prod_{(u,v)\notin E}(1-\mathcal{P}[\sigma_u,\sigma_v]))^{\frac{1}{|V|^2}}\\ &= |V|^2(1-(\prod_{(u,v)\in E}\mathcal{P}[\sigma_u,\sigma_v]\prod_{(u,v)\notin E}(1-\mathcal{P}[\sigma_u,\sigma_v]))^{\frac{1}{|V|^2}}).\end{aligned}$$

As $\prod_{(u,v)\in E}\mathcal{P}[\sigma_u,\sigma_v]\prod_{(u,v)\notin E}(1-\mathcal{P}[\sigma_u,\sigma_v])$ is the likelihood $P(G|\mathcal{P},\sigma)$ in Stochastic Kronecker Graphs [9], so

$$\begin{aligned}\mathbb{E}(\|A_G-\mathcal{P}\|_1) &= \sum_\sigma \|(A_G-\mathcal{P})_\sigma\|_1 P(\sigma)\\ &\leq |V|^2(1-\sum_\sigma P(G|\mathcal{P},\sigma)^{\frac{1}{|V|^2}}P(\sigma))\\ &= |V|^2(1-\mathbb{E}(P(G|\mathcal{P})^{\frac{1}{|V|^2}})).\end{aligned}$$

KRONFIT estimates the initiator matrix by maximizing $\mathbb{E}(P(G|\mathcal{P}))$, hence, KRONFIT provides an approximation on the upper bound of $\mathbb{E}(\|A_G-\mathcal{P}\|_1)$. □

Alternate Estimate of Kronecker Points. In some cases, the KRONFIT algorithm may lead to over/underestimation. When the number of nodes within a real-world network

1. The condition $p>0.5$ ensures that the size of the subgraph is greater than 2^{k-1}, and when performing the fitting, KRONFIT will add isolated nodes so that the number of nodes becomes 2^k [13].

is not a power of 2, KRONFIT will add isolated nodes so that the number of nodes becomes a power of 2 [13]. Adding isolated nodes may lead to underestimation of the parameters as it decreases the overall edge density and core strength of the groups. On the other hand, as the input to KRONFIT is a list of edges, when the network is extremely sparse and the graph size is small, KRONFIT can overestimate as it overlooks real isolated nodes within the network [10]. Therefore, in this paper, we use another estimator of Kronecker initiator matrix for comparison. Instead of maximizing the likelihood, method-of-moments estimator [15] minimizes the difference between the counts for edges, triangles, wedges and 3-stars of a real graph and the expected counts of the fitted Kronecker graph. Compared with KRONFIT, it gets closer to the counts of these local structures. However, our experiments show that in general the Kronecker point estimated by the method-of-moments estimator show less classification power on graphs. In the rest of the paper, by default we refer to Kronecker points estimated by KRONFIT, and we will explicitly mention it if we are using method-of-moments estimator.

2.2 Distribution-based Identity

Here, we aim to represent a network identity with the distribution of embedding values for subgraphs of a network. We construct the distribution-based identity based on recent advancement in network representation:

Network Shapes. Network shapes represent a network using a 3D shape [10]. Building a network shape involves three steps: (1) *Sampling many subgraphs from the network.* For a network shape to represent the distribution of embedding values for subgraphs of the network, we should first sample many subgraphs. Theoretically, any sampling method can work; (2) *Mapping the sampled subgraphs to 3D points using a graph embedding method.* Preferably, an embedding method that can well capture graph properties should be used, so that the distribution of embedding values are closely related to the network properties. Given such an embedding method, one can represent a network and its subgraphs sampled in Step 1 as a set of 3D points; and (3) *Fitting a 3D shape to a set of 3D points obtained in Step 2*. This can be done by fitting various shapes, e.g., spheres/cubes.

Hence, by using different algorithms for these three steps, we can build different types of network shapes. Here, we build a network shape for each network and use it as its distribution-based identity: For step (1), we use *Random Node Sampling* (we have proved its utility in Theorem 2.2) to sample subgraphs from the network by varying the proportion of nodes from 10% to 100% with step size $s = 10\%$. For each proportion, except for 100%, which represents the whole network, we generate $t = 20$ independently sampled subgraphs; For step (2), for each sample (and the whole network), we embeds it to a Kronecker point in the 3D space. In total, we generate $20 \times 9 + 1 = 181$ Kronecker points for each network; For step (3), we fit all the Kronecker points to a 3D shape by computing their convex hull, which compared to other methods, is very compact and effective [16]. The convex hull is used as the distribution-based identity. The time complexity to compute the convex hull is $\mathcal{O}(\frac{t}{s}(n+m))$, linear in the number of nodes n and edges m. As the network shape is visualizable in 3D space, we can plot a network identity. Figure 1 shows the network identity for YouTube (the data is detailed in Section 3).

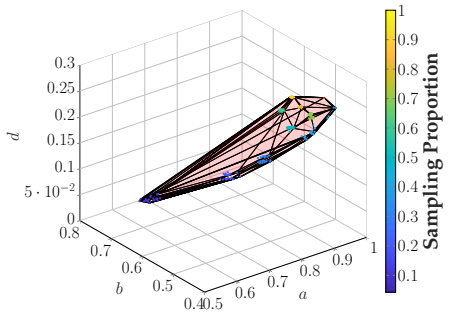


Fig. 1: Distribution-based Identity for YouTube

3 DATA DESCRIPTION

For our experiments, we use twenty real-world networks from four general network categories: social networks, collaboration networks, road networks, and biological networks. The data statistics are in presented in Table 1.

Social Networks: In total, we have eight social networks.

1) *Brightkite* [17]: was a location-based social networking site where users shared their locations by checking-in.
2) *Flixster* [18]: a social network allowing users to buy, rent, or watch movies, share ratings, and find new movies.
3) *Gowalla* [17]: similar to Brightkite, was a location-based social networking site where users shared their locations.
4) *Hyves* [18]: the most popular social networking site in the Netherlands with mainly Dutch visitors. It competes with sites such as Facebook and MySpace in that country.
5) *Livejournal* [19]: a social network where users can keep a blog or journal. Users can form friendship or follow others. Here, edges represent friendships (undirected).
6) *MySpace* [19]: a social network having a significant influence on pop culture and music.
7) *Orkut* [17]: was a social networking website owned and operated by Google, shutdown in 2014.
8) *YouTube* [17]: a video-sharing site with a social network.

TABLE 1: Dataset Statistics

Type	Network	$\|V\| = n$	$\|E\| = m$
Social Networks	Brightkite [17]	58,228	214,078
	Flixster [18]	2,523,386	7,918,801
	Gowalla [17]	196,591	950,327
	Hyves [18]	1,402,673	2,777,419
	Livejournal [19]	3,017,286	85,654,976
	MySpace [19]	854,498	5,635,296
	Orkut [17]	3,072,441	117,185,083
	YouTube [17]	1,134,890	2,987,624
Collaboration Networks	Astro-Ph [17]	18,772	198,050
	Cond-Mat [17]	23,133	93,439
	Gr-Qc [17]	5,242	14,484
	Hep-Th [17]	9,877	25,973
Road Networks	Road-BEL [20]	1,441,295	1,549,970
	Road-CA [17]	1,965,206	2,766,607
	Road-PA [17]	1,088,092	1,541,898
	Road-TX [17]	1,379,917	1,921,660
Biological Networks	Bio-Dmela [20]	7,393	25,569
	Bio-Grid-Human [20]	9,527	62,364
	Bio-Grid-Yeast [20]	5,870	313,890
	Human-Brain [20]	177,600	15,669,036

TABLE 2: Distribution-based Identity Similarity

Types	Network	Brightkite	Flixster	Gowalla	Hyves	Livejournal	MySpace	Orkut	YouTube	Astro-Ph	Cond-Mat	Gr-Qc	Hep-Th	Road-BEL	Road-CA	Road-PA	Road-TX	Bio-Dmela	Bio-Grid-Human	Bio-Grid-Yeast	Human-Brain
Social Networks	Brightkite	1	0	0.04	0	0	0	0	0	0	0	0	0	0	0	0	0	0.09	0.18	0	0
	Flixster	0	1	0.12	0	0	0.05	0	0.22	0.07	0	0	0.01	0	0	0	0	0	0	0	0
	Gowalla	0.04	0.12	1	0	0.01	0.01	0	0.04	0.12	0.01	0.01	0.02	0	0	0	0	0	0.01	0	0
	Hyves	0	0	0	1	0	0.01	0	0	0	0.03	0.02	0.04	0	0	0	0	0	0	0	0
	Livejournal	0	0	0.01	0	1	0	0	0	0.04	0	0	0	0	0	0	0	0	0	0	0
	MySpace	0	0.05	0.01	0.01	0	1	0	0.07	0.04	0.03	0.02	0.03	0	0	0	0	0	0	0	0
	Orkut	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	YouTube	0	0.22	0.04	0	0	0.07	0	1	0.05	0.01	0.01	0.02	0	0	0	0	0	0	0	0
Collaboration Networks	Astro-Ph	0	0.07	0.12	0	0.04	0.04	0	0.05	1	0.08	0.05	0.05	0	0	0	0	0	0	0	0
	Cond-Mat	0	0	0.01	0.03	0	0.03	0	0.01	0.08	1	0.45	0.57	0	0	0	0	0	0	0	0
	Gr-Qc	0	0	0.01	0.02	0	0.02	0	0.01	0.05	0.45	1	0.43	0	0	0	0	0	0	0	0
	Hep-Th	0	0.01	0.02	0.04	0	0.03	0	0.02	0.05	0.57	0.43	1	0	0	0	0	0	0	0	0
Road Networks	Road-BEL	0	0	0	0	0	0	0	0	0	0	0	0	1	0.23	0	0.13	0	0	0	0
	Road-CA	0	0	0	0	0	0	0	0	0	0	0	0	0.23	1	0.22	0.48	0	0	0	0
	Road-PA	0	0	0	0	0	0	0	0	0	0	0	0	0	0.22	1	0.25	0	0	0	0
	Road-TX	0	0	0	0	0	0	0	0	0	0	0	0	0.13	0.48	0.25	1	0	0	0	0
Biological Networks	Bio-Dmela	0.09	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.49	0	0
	Bio-Grid-Human	0.18	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0.49	1	0	0
	Bio-Grid-Yeast	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.02
	Human-Brain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.02	1

Collaboration Networks: We include four collaboration networks from arXiv.org, capturing scientific collaborations between authors with various scientific interests. In a collaboration network, an undirected edge between nodes i and j exists, if authors i and j have co-authored at least one paper.

9) *Astro-Ph* [17]: Astro physics.
10) *Cond-Mat* [17]: Condense matter physics.
11) *Gr-Qc* [17]: General relativity and quantum cosmology.
12) *Hep-Th* [17]: High energy physics theory.

Road Networks: We include four road networks. In road networks, nodes are intersections/endpoints and undirected edges are the roads connecting these intersections/road endpoints.

13) *Road-BEL* [17]: Belgium's OpenStreetMap road network.
14) *Road-CA* [17]: the road network of California.
15) *Road-PA* [17]: the road network of Pennsylvania.
16) *Road-TX* [17]: the road network of Texas.

Biological Networks: We include four biological networks.

17) *Bio-Dmela*[20]:protein-protein interaction (PPI) network.
18) *Bio-Grid-Human* [20]: a PPI network.
19) *Bio-Grid-Yeast* [20]: a PPI network.
20) *Human-Brain* [20]: the network of human brain.

4 Uniqueness and Partial Network Identity

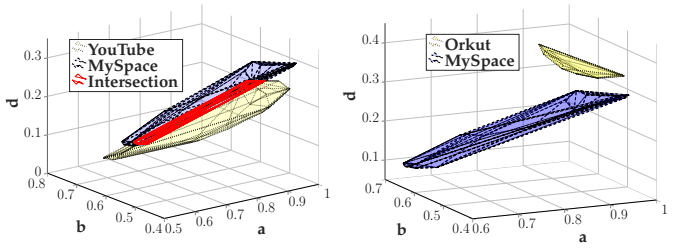
4.1 Uniqueness of Network Identity

Uniqueness is a basic requirement for an identity. We have mentioned that generally graph embedding does not guarantee the uniqueness, which is also true for Kronecker points. Hence, we check whether distribution-based identity can capture the distinctiveness of networks. We define the distribution-based identity similarity and investigate the similarity between identities of different networks.

Distribution-based Identity Similarity. To view how similar two distribution-based identities are, let us take a look at an example first. Figure 2 provides two pairs of distribution-based identities, i.e., YouTube vs. MySpace and Orkut vs. MySpace. We observe that distribution-based identities (1) have different volume, and e.g. the identity of MySpace is larger than that of Orkut; (2) may or may not have overlap. e.g. YouTube and MySpace have overlap, while Orkut and MySpace have no overlap. Looking at the Kronecker points that form the identities, we notice that network identities can capture network properties. For example, YouTube, MySpace and Orkut are all social networks, and the majority of their identities are located in the area $a > b > d$. When $a > b > d$ in a Kronecker point, the fitted network exhibits a *core-periphery* structure [9], [10], where a represents the strength of the core of the network and a small d indicates a sparse periphery. The result is in accordance with that social networks exhibit a core-periphery structure [9]. Furthermore, we notice that compared to the other two networks, Orkut network and its subgraphs have larger values of a and d but smaller values of b. It indicates that Orkut has a very dense core group, a periphery group denser than that of others, but the connections between these two groups are sparse. Based on the observations, we define the similarity between identities using Jaccard Index:

Fig. 2: Two Pairs of Distribution-based Identities

(a) YouTube/MySpace (b) Orkut/MySpace



$$\text{similarity}(A, B) = \frac{\text{volume}(\text{ID}_A \cap \text{ID}_B)}{\text{volume}(\text{ID}_A \cup \text{ID}_B)}, \quad (2)$$

where volume is the volume of a distribution-based identity, and ID_A and ID_B represent identities of networks A and B, respectively. It is easy to find that $\text{volume}(\text{ID}_A \cup \text{ID}_B) = \text{volume}(\text{ID}_A) + \text{volume}(\text{ID}_B) - \text{volume}(\text{ID}_A \cap \text{ID}_B)$, and $\text{volume}(\text{ID}_A \cap \text{ID}_B)$ is easy to calculate as intersection of convex sets is convex. Table 2 lists the similarity between all pairs of the identities (the results have been rounded to two digits). We observe that (1) similarity between most identities (i.e., shapes) is small, i.e., below 0.1; (2) networks from different categories in general have very low similarity. Road networks and biological networks are not similar to networks from other categories, while social networks and collaboration networks have some similarity; (3) within the same category, some similarity exists. For example, the similarity between YouTube and Flixster is 0.22, road networks are relatively similar to each other, and three collaboration networks are also relatively similar. In general, the highest similarity is 0.57, which does not violate the uniqueness of the network identity across different networks. Note that we do not claim the absolute uniqueness of the distribution-based identity as it is built using graph sampling, but we assume that the distribution of embedding values for subgraphs can capture the distinctness of the network identities. Moreover, previous studies [10] have shown that points representing samples of the same proportion exhibit a cluster-

TABLE 3: Distribution-based Identity Similarity (using method-of-moments estimator)

Types	Network	Brightkite	Flixster	Gowalla	Hyves	Livejournal	MySpace	Orkut	YouTube	Astro-Ph	Cond-Mat	Gr-Qc	Hep-Th	Road-BEL	Road-CA	Road-PA	Road-TX	Bio-Dmela	Bio-Grid-Human	Bio-Grid-Yeast	Human-Brain
Social Networks	Brightkite	1	0.06	0.03	0.02	0	0.03	0.13	0	0	0.02	0.03	0	0	0	0	0	0.06	0.28	0	0
	Flixster	0.06	1	0	0.07	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0.04	0	0
	Gowalla	0.03	0	1	0	0.01	0.21	0.06	0	0	0	0	0	0	0	0	0	0	0	0	0
	Hyves	0.02	0.07	0.01	1	0	0.03	0	0	0	0	0	0	0	0	0	0	0.01	0.03	0	0
	Livejournal	0	0	0.01	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MySpace	0.03	0	0.21	0.03	0	1	0.03	0	0	0	0	0	0	0	0	0	0	0	0	0
	Orkut	0.13	0	0.06	0	0	0.03	1	0	0	0	0	0	0	0	0	0	0	0.02	0	0
	YouTube	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Collaboration Networks	Astro-Ph	0	0	0	0	0	0	0	0	1	0.02	0.03	0	0	0	0	0	0.06	0.02	0.01	0
	Cond-Mat	0.02	0	0	0	0	0	0	0	0.02	1	0.48	0.05	0	0	0	0	0.49	0.15	0	0
	Gr-Qc	0.03	0	0	0	0	0	0	0	0.03	0.48	1	0.12	0	0	0	0	0.49	0.13	0	0
	Hep-Th	0	0	0	0	0	0	0	0	0	0.05	0.12	1	0.16	0.20	0.03	0.13	0.03	0	0	0
Road Networks	Road-BEL	0	0	0	0	0	0	0	0	0	0	0	0.16	1	0.77	0.08	0.78	0	0	0	0
	Road-CA	0	0	0	0	0	0	0	0	0	0	0	0.20	0.77	1	0.15	0.75	0	0	0	0
	Road-PA	0	0	0	0	0	0	0	0	0	0	0	0.03	0.08	0.15	1	0.13	0	0	0	0
	Road-TX	0	0	0	0	0	0	0	0	0	0	0	0.13	0.78	0.75	0.13	1	0	0	0	0
Biological Networks	Bio-Dmela	0.06	0.01	0	0.01	0	0	0	0	0.06	0.49	0.49	0.03	0	0	0	0	1	0.25	0	0
	Bio-Grid-Human	0.28	0.04	0	0.03	0	0	0.02	0	0.02	0.15	0.13	0	0	0	0	0	0.25	1	0	0
	Bio-Grid-Yeast	0	0	0	0	0	0	0	0	0.01	0	0	0	0	0	0	0	0	0	1	0.09
	Human-Brain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.09	1

Fig. 3: Distribution-based Identity Change with t

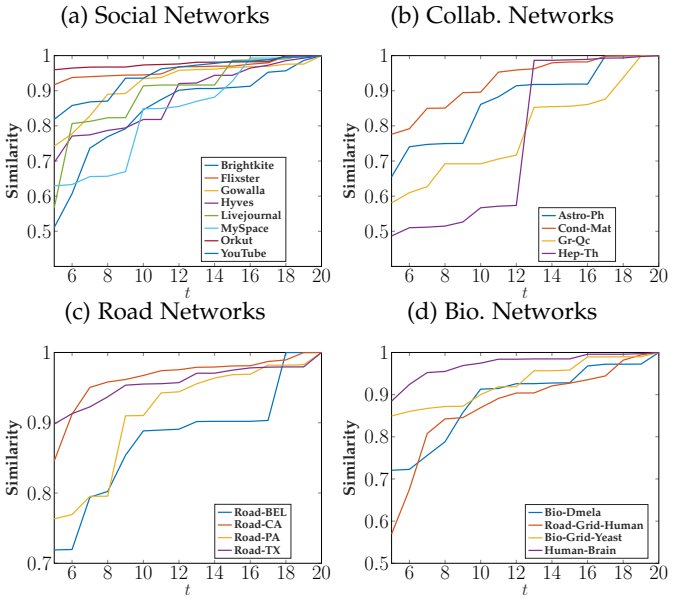
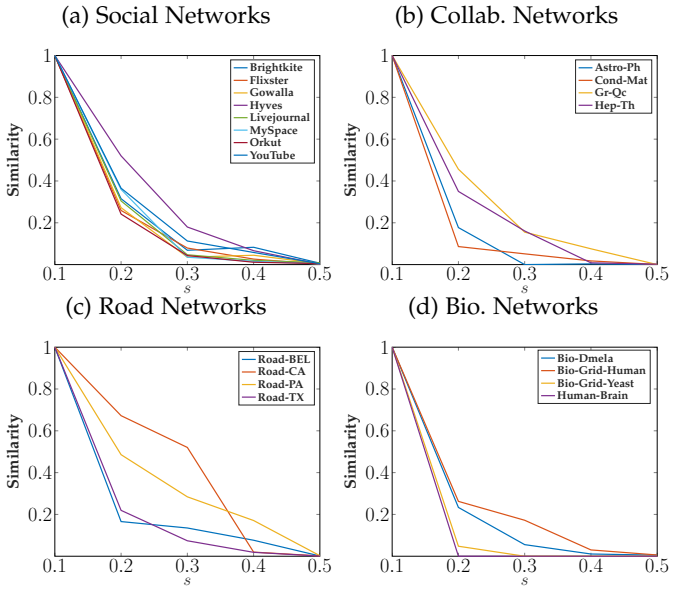


Fig. 4: Distribution-based Identity Change with s



ing phenomenon, indicating the stability of a distribution-based identity to some extent, which we also observe in our experiments (see Sections 5 and 6). When we estimate Kronecker points by using the method-of-moments estimator, we find similar patterns but the similarity between the network identities are generally higher, see Table 3 .

4.2 Partial Distribution-based Network Identity

Theoretically, one can sample all the possible subgraphs from a network to build a distribution-based identity that represents a "complete" network identity (similar to how one can have a high resolution fingerprint scan). However, this violates our idea for efficiency. Let us assume the network identity we have constructed is a practically "complete" network identity. A few questions comes up: How sensitive is a network identity to the number of sample points taken? Due to the definition of convex hull, if we take a subset of Kronecker points of the complete network identity to build a partial network identity, the partial network identity should also be a subset of the complete network identity. In other words, the complete network identity can shrink to a partial network identity. How different are the complete network identity and a partial one? To answer these questions, we study the partial network identity by varying the sampling step size s and the number of independent samples for each proportion t, and check the similarity between the partial identities and the complete one. We first fix $s = 10\%$ and vary t from 5 to 20, and Figure 3 indicates that the distribution-based identity is not sensitive to t generally as (1) for the smallest $t = 5$, the similarity is over 50%; (2) for most networks, by sampling 13 to 14 subgraphs for each proportion, we can create a partial network identity which is 90% similar to the complete network identity. This means taking fewer samples of the same proportion will not change the identity much and it follows the observation of [10] that the Kronecker points representing samples of the same proportion have the clustering phenomenon. Next, we fix $t = 20$ and vary the step size s from 10% to 50%. Figure 4 shows that the network identity is more sensitive to s as (1) the similarity drops quickly with the increase of sampling step size, and (2) the similarity drops to 0 as the volume turns to 0 when the identity is degraded to the 2D space. It shows that by setting s smaller and taking samplings of more different sizes will make the network identity stabler. For the partial identities using the method-of-moments estimator, the patterns are similar but in general the partial identities are more sensitive to the change of t and s. In Section 5 and 6, we will discuss the performance of the partial identity for the network identification/authentication problems.

5 Network Identification

5.1 Experimental Setup

From each network, we sample many subgraphs representing graphs G which are to be identified/authenticated.

We vary the sampling proportion from 10% to 99% and sample using random node sampling. For each proportion, we sample two subgraphs. Hence, for each network we have $90 \times 2 = 180$ subgraphs, and for twenty networks, we have $180 \times 20 = 3,600$ samples to be identified/authenticated.

5.2 Identification with Embedding-based Identity

To use the embedding-based identity for identification, we embed both G and all other N_i as Kronecker points. We consider the identification problem in the following way: Given the n (=20) identities of N_i's, we split the whole embedding space, the $1 \times 1 \times 1$ cube, into n regions, so that each region represents the embedding space for the samples of a certain network. In our work, we propose two splitters.

Voronoi Splitter. It calculates the Euclidean distance between the Kronecker point of a graph G and that of all other networks (N_i's) and reports the closest N_i as the identified network. This is equivalent to building a *Voronoi diagram* [21] for the set of Kronecker points of all N_i's, where the *Voronoi cell* for N_j denotes the set of graphs identified as N_j.

Supervised Splitter. Instead of reporting the closest N_i, for each sample G, we use the 20 distances (from a sample to each N_i) as features, and the name of the networks as the class label, to train a multiclass classification model. In this experiment, we use 10-fold cross validation, and decision tree, linear SVM, k-NN and bagged trees as our classifiers.

We provide four baselines for comparison.

1) **Top Eigenvalues**. Top eigenvalues have been used to study graph similarity [22]. We compute the top 5 eigenvalues of each sample as features for classification. The time complexity of the method is $\mathcal{O}(n^2)$, where n is the number of nodes.
2) **Truncated Spectral Moments**. The spectral moments of the random walk transition matrix of a network have been proven to be closely related to the network structure and various network properties [23]. Therefore, we compute the truncated (first four) spectral moments of each sample as features for classification. We use the APPROXSPECTRALMOMENT algorithm proposed by [24] to compute the accurate estimates of the low-order moments. The algorithm estimates the moments by simulating many random walks and computes the proportion of closed walks. To compute the ℓ-th spectral moment by simulating s random walks, it takes $O(s\ell)$ time.
3) **Graph2Vec**. GRAPH2VEC is a graph embedding technique, which views a graph as a document and the rooted subgraphs around each node as words. It extends document embedding neural networks to embed a graph as a vector [25].
4) **Random Prediction**. A simple *random prediction*, so the accuracy will be $1/n$ where n is the number of networks.

We evaluate the methods for all networks and within each network category and report the results in Table 4. For supervised splitter, we report the result of the best classifier, as the prediction turns out to be not sensitive to the choice of learning algorithm. Table 4 illustrates that (1) both Voronoi splitter and Supervised Splitter outperform the random prediction; (2) Voronoi splitter performs not as good as the other baselines, which is not surprising, as Theorem 2.2 has shown that when the sampling proportion p is small, the Kronecker point of the sample can be far away from that of the whole network; (3) Supervised Splitter performs best and achieves an overall accuracy of 84.0%. It is slightly better than Truncated Spectral Moments and GRAPH2VEC, and significantly better than the other methods; and (4) the performance on road networks is not as good as other categories, while GRAPH2VEC performs relatively stable across different categories. Comparing both methods, we find that (1) Voronoi Splitter is simple and does not need a training process, but it can make mistakes, especially on smaller samples; (2) Supervised Splitter performs better as it learns from the distances from the samples to different networks, making more informed decisions. Table 5 lists the result of using the method-of-moments estimator. The result is not as good as using KRONFIT, but is still comparable with the baseline using top eigenvalues.

TABLE 4: Network Identification Accuracy with Embedding-based Identity

Type	Voronoi Splitter	Supervised Splitter	Baselines			
			Top Eigenvalues	Truncated Spectral Moments	Graph2Vec	Random Prediction ($1/n$)
All Networks	40.2%	**84.0%**	62.4%	82.0%	81.7%	5%
Social Networks	50.3%	94.2%	74.8%	95.5%	83.7%	12.5%
Collaboration Networks	58.2%	78.8%	70.9%	94.8%	97.4%	25%
Road Networks	32.9%	67.0%	44.9%	51.2%	86.3%	25%
Biological Networks	66.5%	98.3%	90.4%	99.7%	89.9%	25%

TABLE 5: Network Identification Accuracy with Embedding-based Identity (method-of-moments estimator)

Type	Voronoi Splitter	Supervised Splitter	Baselines			
			Top Eigenvalues	Truncated Spectral Moments	Graph2Vec	Random Prediction ($1/n$)
All Networks	37.7%	61.6%	62.4%	82.0%	81.7%	5%
Social Networks	53.5%	67.3%	74.8%	95.5%	83.7%	12.5%
Collaboration Networks	39.9%	75.3%	70.9%	94.8%	97.4%	25%
Road Networks	24.0%	35.1%	44.9%	51.2%	86.3%	25%
Biological Networks	69.4%	88.1%	90.4%	99.7%	89.9%	25%

5.3 Identification with Distribution-based Identity

To use the distribution-based identity for identification, we follow a roadmap similar to that of the embedding-based method. The difference is that the network identity N_i is represented as a 3D shape. Therefore, we need to define the distance between a 3D point and a 3D shape. Considering definitions of the distance between two sets of points and geometrical properties of a convex polyhedron, we make the following three Euclidean distances as candidates:

1) $\mathsf{d}_{\mathsf{shortest}}$. $\mathsf{d}_{\mathsf{shortest}}$ is defined based on the shortest distance between two points from set A and B respectively:

$$d(A, B) = \inf\{d(x, y) | x \in A, y \in B\}. \quad (3)$$

In our case, it refers to the distance from a point to the closest point on the surface (all the facets) of the shape if the point is outside the shape, otherwise it is 0.

2) $\mathsf{d}_{\mathsf{Hausdorff}}$. Hausdorff distance is used to measure how far two sets A and B are in a metric space:

$$d_H(A, B) = \max\{\sup_{a \in A} \inf_{b \in B} d(a, b), \sup_{b \in B} \inf_{a \in A} d(a, b)\}. \quad (4)$$

It is the largest of the distances from a point in one set to the closest point in the other and is commonly used in computer vision research [26]. In our case, $\mathsf{d}_{\mathsf{Hausdorff}}$ refers

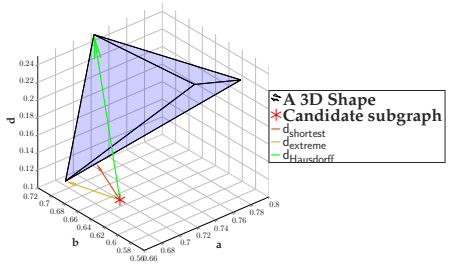


Fig. 5: Three Distances between a 3D point and a 3D Shape.

TABLE 6: 90th Percentile of the Distance Distribution

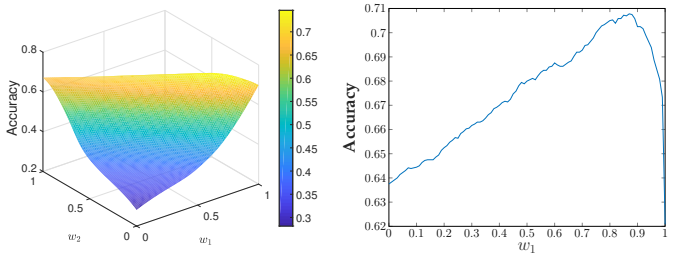
Type	Network	$d_{shortest}$	$d_{extreme}$	$d_{Hausdorff}$
Social Networks	Brightkite	0.0139	0.0861	0.6017
	Flixster	0.0149	0.0344	0.3476
	Gowalla	0.0257	0.0494	0.5898
	Hyves	0.0149	0.0331	0.5218
	Livejournal	0.0069	0.0247	0.1547
	MySpace	0.0130	0.0325	0.4399
	Orkut	0.0055	0.0161	0.1713
	YouTube	0.0138	0.0394	0.4687
Collaboration Networks	Astro-Ph	0.0196	0.0572	0.5938
	Cond-Mat	0.0189	0.0670	1.0400
	Gr-Qc	0.0143	0.0818	1.1514
	Hep-Th	0.0147	0.0394	1.0730
Road Networks	Road-BEL	0.0306	0.0598	0.8940
	Road-CA	0.0182	0.0595	0.7866
	Road-PA	0.0297	0.0917	0.8327
	Road-TX	0.0220	0.0812	0.7651
Biological Networks	Bio-Dmela	0.0087	0.0413	0.7175
	Bio-Grid-Human	0.0127	0.0489	0.5932
	Bio-Grid-Yeast	0.0040	0.0483	0.2489
	Human-Brain	0.0063	0.0248	0.1341

to the distance from a point to the farthest boundary point (i.e., extreme points) of the shape.

3) $d_{extreme}$. As all of the boundary points of a network shape are some of the Kronecker points of samples used for generating the shape, we also use the distance from a point to the closest boundary point of the shape.

Fig. 5 is a simple example to illustrate these three distances. For each network and the test samples drawn from it, we list the 90th percentile of the distances distribution in Table 6 and 7. Based on the definitions, we know that $d_{shortest} \leq d_{extreme} \leq d_{Hausdorff}$. From the table, we observe that most of the Kronecker points of the subgraphs are around the surface and the boundary of the network shape of the source network, and for most networks $d_{Hausdorff}$ is large, especially for collaboration networks, which indicates that different subgraphs of the same network can be far away from each other.

Next, we use the three distances with the two splitters we used in the last section for identification. To make it intuitively clear, we provide an example in Figure 7. We report the result in Table 8. We find that for Voronoi Splitter, compared with the embedding-based identity, the distribution-based identity with $d_{shortest}$ and $d_{extreme}$ improve a lot on performance and can beat both Top Eigenvalues and Random Prediction, as the distribution-based identity can preserve subgraph information. It is not surprising that $d_{Hausdorff}$ performs not good as it can be explained by our observation and discussion of the 90 percentile of the distance distribution. Based on these observations, we consider using a combination of these three distances for the identification. We use the weighted average $d_{weighted} = w_1 \times d_{shortest} +$



(a) $w_1 + w_2 + w_3 = 1$ (b) $w_1 + w_2 = 1, w_3 = 0$

Fig. 6: Accuracy with Weighted Distance $d_{weighted}$

TABLE 7: 90th Percentile of the Distance Distribution (using method-of-moments estimator)

Type	Network	$d_{shortest}$	$d_{extreme}$	$d_{Hausdorff}$
Social Networks	Brightkite	0.0143	0.0480	0.3896
	Flixster	0.0104	0.0308	0.5672
	Gowalla	0	0.0510	0.2606
	Hyves	0	0.0307	0.5497
	Livejournal	0	0.0459	0.0957
	MySpace	0	0.0324	0.1788
	Orkut	0.0171	0.0372	0.1825
	YouTube	0	0.0272	0.1253
Collaboration Networks	Astro-Ph	0.0176	0.0501	0.2997
	Cond-Mat	0.0357	0.0695	0.8505
	Gr-Qc	0.0263	0.0790	0.9066
	Hep-Th	0.0375	0.1074	1.1391
Road Networks	Road-BEL	0.0071	0.3314	1.6682
	Road-CA	0.0295	0.5531	1.6576
	Road-PA	0.0299	0.1365	1.6636
	Road-TX	0.0502	0.5129	1.6470
Biological Networks	Bio-Dmela	0.0208	0.0678	0.7309
	Bio-Grid-Human	0.0215	0.0760	0.5702
	Bio-Grid-Yeast	0.0050	0.0806	0.4375
	Human-Brain	0.0115	0.0321	0.1902

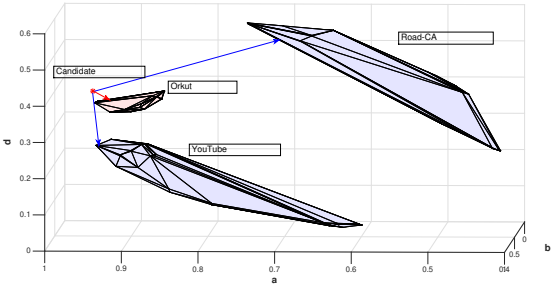


Fig. 7: **An example of the two splitters**. Here, we have three distribution-based identities in our database: Orkut, YouTube, and road network of California. Our goal is to identify a candidate graph, which is the red point. To identify, we check the distances between the red point and the three 3D shapes. For Voronoi splitter, we pick the closest shape, in this case Orkut, as its identity. For supervised splitter, we use distances to all network identities as features and the network name as the label to train a classifier.

TABLE 8: Network Identification Accuracy with Distribution-based Identity

Type	Voronoi Splitter				Supervised Splitter	Baselines			
	$d_{shortest}$	$d_{extreme}$	$d_{hausdorff}$	$d_{weighted}$		Top Eigenvalues	Truncated Spectral Moments	Graph2Vec	Random Prediction $(1/n)$
All Networks	61.6%	63.8%	16.8%	70.8%	**84.4%**	62.4%	82.0%	81.7%	5%
Social Networks	81.3%	81.4%	25.7%	86.7%	96.4%	74.8%	95.5%	83.7%	12.5%
Collaboration Networks	65.7%	63.7%	25%	75.0%	84.2%	70.9%	94.8%	97.4%	25%
Road Networks	48.8%	46.7%	35%	52.4%	76.8%	44.9%	51.2%	86.3%	25%
Biological Networks	70.8%	63.1%	34.6%	76.3%	80.4%	90.4%	99.7%	89.9%	25%

TABLE 9: Network Identification Accuracy with Distribution-based Identity (method-of-moments estimator)

Type	Voronoi Splitter				Supervised Splitter	Baselines			
	$d_{shortest}$	$d_{extreme}$	$d_{hausdorff}$	$d_{weighted}$		Top Eigenvalues	Truncated Spectral Moments	Graph2Vec	Random Prediction ($1/n$)
All Networks	45.9%	39.3%	20.2%	51.5%	64.1%	62.4%	82.0%	81.7%	5%
Social Networks	52.5%	41.5%	28.1%	53.7%	68.5%	74.8%	95.5%	83.7%	12.5%
Collaboration Networks	51.4%	52.5%	31.9%	67.8%	80.1%	70.9%	94.8%	97.4%	25%
Road Networks	36.7%	26.8%	23.3%	32.6%	43.9%	44.9%	51.2%	86.3%	25%
Biological Networks	77.5%	76.7%	25%	88.6%	89.6%	90.4%	99.7%	89.9%	25%

$w_2 \times \mathsf{d}_{\mathsf{extreme}} + w_3 \times \mathsf{d}_{\mathsf{Hausdorff}}$, where $w_1 + w_2 + w_3 = 1$. To get the best weights, one can use supervised learning for learning the weights. Here, for simplicity we do grid search on the feasible weights w_1, w_2, w_3 and plot the accuracy change in Figure 6a. The plot shows that the accuracy is high when $w_1 + w_2 \approx 1$ and it drops as w_3 increases. The best accuracy is 70.8% with $w_1 = 0.87, w_2 = 0.13, w_3 = 0$. Figure 6b provides the accuracy change when w_3 is set to 0, i.e., $w_1 + w_2 = 1$. We find the accuracy increases quickly when w_1 increases from 0 to 0.7 and drops quickly when w_1 is greater than 0.9. Based on the observations, we set $\mathsf{d}_{\mathsf{weighted}} = 0.87 \times \mathsf{d}_{\mathsf{shortest}} + 0.13 \times \mathsf{d}_{\mathsf{extreme}}$, and in general $\mathsf{d}_{\mathsf{weighted}}$ performs best among these distances.

For Supervised Splitter, we use $\mathsf{d}_{\mathsf{shortest}}$, $\mathsf{d}_{\mathsf{extreme}}$ and $\mathsf{d}_{\mathsf{Hausdorff}}$ as features. Each graph G has $3 \times 20 = 60$ features for all networks and we use the name of the networks as the class labels. Table 8 shows that compared with the Embedding-based identity, the performance improves a little and it reaches an overall accuracy 84.4%.

We conduct the same experiments by using the method-of-moments estimator, and Table 9 indicates that it performs slightly better than the Embedding-based identity, but not as good as using KRONFIT.

Identification with Partial Network Identity. As discussed in Section 4, partial distribution-based network identity can be constructed similar to the complete network identity by taking fewer sample subgraphs. We investigate how effective partial distribution-based identities are in the network identification task. Based on the previous study on the similarity of partial network identity and complete network identity, we speculate that the network identification accuracy is more sensitive to the change of the sampling step size s. Figure 8a and 8b illustrate the accuracy change of Voronoi splitter (using $\mathsf{d}_{\mathsf{weighted}}$) and Supervised Splitter respectively with different s and t configuration. In general, the accuracy does not change with the number of samples t for each proportion and it slightly drops with the increase in sampling step size s.

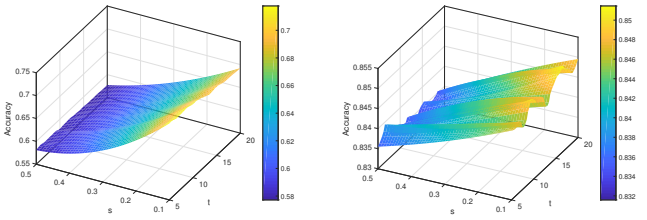
6 Network Authentication

For network authentication, given the distance from the identity of G to that of a network N_i, we aim to accept or reject the claim that G is sampled from N_i.

6.1 Authentication

Different from network identification, for network authentication, we need to split the whole embedding space into two regions: the *accept* and *reject* regions. We also propose two methods: a Voronoi splitter and a supervised splitter.

Voronoi Splitter. For the embedding-based identity, we use the r-percentile of the distances from the Kronecker points of samples to that of the source network as a threshold.



(a) Voronoi Splitter (b) Supervised Splitter

Fig. 8: **Prediction Performance with Partial Identity**. (1) The Supervised Splitter is robust to the change of both t and s. The accuracy does not change with the number of samples t for each proportion and it slightly drops with the increase in sampling step size s from 85% to 83%. (2) Similar patterns are observed for Voronoi Splitter. Differently, the accuracy decreases more with the increase of s, from 70% to 58%.

If the distance between identities of G and N_i is less than threshold d, we accept the claim; otherwise, we reject it. An advantage of this method is that we can control the false reject rate (FRR) of the authentication system, e.g., in one experiment, we set r = 90, so FRR is fixed at 10%. It allows one to have a geometric interpretation of this splitter. That is, we create a ball centered at the Kronecker point of the network with a diameter equal to $2 \times d$. Everything inside the ball (the boundary included) will be accepted and everything outside is rejected. For the distribution-based identity, we know from the distribution of $\mathsf{d}_{\mathsf{shortest}}$, $\mathsf{d}_{\mathsf{extreme}}$ and $\mathsf{d}_{\mathsf{Hausdorff}}$ for samples of each network that most points are around the surface of the network shape; hence, we can use the r-percentile of the distances to the surfaces as the threshold. Similarly, one can interpret the splitter as creating a band around the surface of the distribution-based identity with a diameter equal to $2 \times d$, accepting everything inside the band and rejecting everything outside. Table 10 shows that the method does not work well with embedding-based identity, but performs well with distribution-based identity. The false accept rate (FAR) varies from 0% to more than 20%, and for most networks it is below 10%. When we use method-of-moments estimator, the result does not change much, see Table 11. Moreover, we vary r when we use the distribution-based identity and plot the change of average FAR and FRR across networks in Figure 9, and it turns out that $r = 90$ leads to the equal error rate.

Supervised Splitter. For distribution-based identity, we use $\mathsf{d}_{\mathsf{shortest}}$, $\mathsf{d}_{\mathsf{extreme}}$ and $\mathsf{d}_{\mathsf{Hausdorff}}$ between identities of G and

TABLE 10: Authentication with Voronoi Splitter ($r = 90$)

Type	Networks	Embedding-based			Distribution-based ($\mathsf{d}_{\mathsf{shortest}}$)		
		Accuracy	AUC	FAR	Accuracy	AUC	FAR
Social Networks	Brightkite	39.83%	0.61	62.51%	97.81%	0.94	1.73%
	Flixster	61.67%	0.67	38.92%	90.77%	0.90	9.18%
	Gowalla	48.72%	0.67	53.27%	78.52%	0.84	22.16%
	Hyves	39.50%	0.54	62.16%	99.04%	0.95	0.42%
	Livejournal	84.58%	0.62	11.81%	98.52%	0.95	0.98%
	MySpace	55.72%	0.70	45.82%	94.32%	0.92	5.42%
	Orkut	85.31%	0.45	10.20%	99.44%	0.95	0.00%
	YouTube	48.08%	0.64	53.71%	91.57%	0.91	8.33%
Collaboration Networks	Astro-Ph	40.08%	0.59	40.08%	77.50%	0.83	23.24%
	Cond-Mat	35.00%	0.56	67.37%	78.77%	0.84	21.90%
	Gr-Qc	5.08%	0.50	99.91%	79.58%	0.84	21.04%
	Hep-Th	21.17%	0.47	81.67%	84.54%	0.87	15.78%
Road Networks	Road-BEL	5.03%	0.48	99.80%	88.33%	0.89	11.75%
	Road-CA	16.67%	0.41	86.02%	90.70%	0.90	9.28%
	Road-PA	7.42%	0.43	96.49%	89.20%	0.90	10.85%
	Road-TX	8.06%	0.47	95.79%	89.97%	0.90	10.03%
Biological Networks	Bio-Dmela	37.97%	0.46	62.89%	97.96%	0.94	1.57%
	Bio-Grid-Human	36.58%	0.54	65.32%	92.80%	0.91	7.05%
	Bio-Grid-Yeast	88.31%	0.84	11.17%	98.18%	0.94	1.34%
	Human-Brain	95.19%	0.93	4.53%	98.95%	0.95	0.52%

TABLE 11: Authentication with Voronoi Splitter (moment based $r = 90$)

Type	Networks	Embedding-based			Distribution-based ($d_{shortest}$)		
		Accuracy	AUC	FAR	Accuracy	AUC	FAR
Social Networks	Brightkite	87.00%	0.46	8.42%	89.81%	0.90	10.20%
	Flixster	89.14%	0.47	6.17%	98.39%	0.94	1.17%
	Gowalla	90.22%	0.47	5.03%	82.44%	0.88	18.19%
	Hyves	93.81%	0.49	1.26%	92.78%	0.93	7.19%
	Livejournal	91.75%	0.48	3.42%	95.56%	0.96	4.47%
	MySpace	91.53%	0.48	3.65%	84.80%	0.90	15.82%
	Orkut	88.11%	0.48	7.46%	94.11%	0.92	5.67%
	YouTube	94.78%	0.50	0.23%	90.86%	0.93	9.33%
Collaboration Networks	Astro-Ph	88.56%	0.48	6.96%	89.86%	0.90	10.15%
	Cond-Mat	25.97%	0.60	77.78%	84.97%	0.87	15.29%
	Gr-Qc	26.44%	0.55	76.70%	84.03%	0.87	16.29%
	Hep-Th	22.75%	0.59	81.23%	94.47%	0.92	5.29%
Road Networks	Road-BEL	9.28%	0.32	93.30%	89.14%	0.90	10.91%
	Road-CA	5.03%	0.50	99.94%	86.58%	0.88	13.60%
	Road-PA	7.44%	0.37	95.79%	86.81%	0.88	13.36%
	Road-TX	5.00%	0.50	100.00%	86.81%	0.88	13.36%
Biological Networks	Bio-Dmela	31.28%	0.52	70.99%	86.47%	0.88	13.71%
	Bio-Grid-Human	41.22%	0.61	60.99%	84.19%	0.87	16.11%
	Bio-Grid-Yeast	43.67%	0.70	59.30%	93.47%	0.92	6.35%
	Human-Brain	97.81%	0.94	1.78%	99.14%	0.95	0.38%

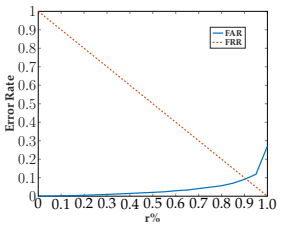


Fig. 9: **Equal Error Rate**. (1) With the increase of r (decrease of FRR), FAR increases slowly. (2) When r is 100, the FRR is around 25%. It shows that $d_{surface}$ is a good indicator as the majority of the samples from other networks are far away. (3) when r is 90, the FAR and FRR are equal, which leads to the equal error rate.

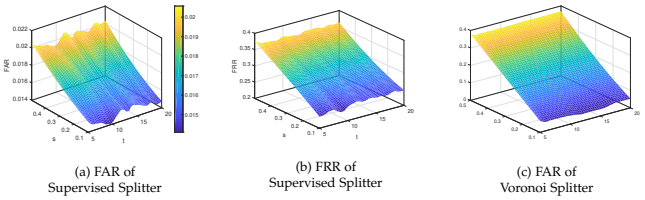
N_i as three features, and whether G is sampled from N_i as a binary label. We train a supervised learning classifier with 10-fold cross validation for each network. For the embedding-based identity, we use the distance between the Kronecker points of G and N_i as the only feature. We report Equal Error Rate (EER), at which the false accept rate (FAR) is equal to the false reject rate (FRR), in Table 12. The results show that classifiers using both identities have a low EER indicating a reasonable performance, especially for using distribution-based identity. Comparing the two splitters, one can see a trade-off between the FAR and the FRR.

Authentication with Partial Network Identity. Next, we use partial distribution-based network identity for network authentication. Figures 10a and 10b illustrate the change of the average FAR and FRR with the partial network identities for Supervised Splitter, and we notice that both FAR and FRR increase with the sampling step size s slowly and they do not change much with t. For Voronoi Splitter, Figure 10c shows the FAR quickly increases with the increase in sampling step size s. This can be explained by the fact that with the increase of s, the partial network identity rapidly shrinks as its similarity to the complete network identity drops fast, which in turn leads to the 90th percentile d becoming large. A large threshold d will accept more false samples. In this case, we need to find the equal error rate for the partial network identity to strike a balance between

TABLE 12: Authentication with Supervised Splitter

Classifier	Embedding-based	Distribution-based
Decision Tree	0.21 (0.02)	0.07 (0.01)
k-NN	0.21 (0.02)	0.09 (0.01)
SVM	0.28 (0.02)	0.10 (0.01)

Note: Mean and standard deviation of the EERs across networks.



(a) FAR of Supervised Splitter (b) FRR of Supervised Splitter (c) FAR of Voronoi Splitter

Fig. 10: **Authentication Performance with Partial Identity**. (a-b) For Supervised Splitter, both FAR and FRR increase with the sampling step size s slowly and they do not change much with t. FAR is generally lower than 2% and FRR increases from 20% to over 35%; (c) For Voronoi Splitter, we use the distances to the surfaces and set the percentile $r = 90$, so FRR is fixed at 10%. Similarly, we find the splitter is not sensitive to t, but the FAR quickly increases with the increase in sampling step size s.

FAR and FRR.

7 Application to Biometrics

As we have shown, our methods can be used for network identification and network authentication. Here, we explore whether they can help with real-world biometrics. Here, we use two touch-based biometrics dataset to authenticate user identity: (1) *La. Tech Touch Dataset.* This dataset was collected by Abdul Serwadda and his collegues [27]. The dataset contains touch data for 138 users on a phone screen. The subjects were students, faculty or staff at Louisiana Tech University; (2) *SU Touch Dataset.* The second dataset is collected by us. It includes the touch data on both a phone screen and a tablet screen from 116 students at Syracuse University. For both datasets, users were asked to answer multiple questions through Android applications. Users had to scroll/swipe back and forth to find the answers to the questions, and for each stroke representing the path taken by a user's finger during scrolling, the applications recorded points touched by the user's finger (the x and y coordinates) every 15 milliseconds. We aim to authenticate a user's identity based on her overall swipe/scroll patterns on the screen. Therefore, we represent the whole phone/tablet screen as a grid network. We set the whole screen as a grid network with each node covering a square area. For the La. Tech dataset, the layout of the phone is 480×800, we set the node as a 4×4 square area. For each user, we create an undirected simple graph, a *swipe graph*. For each two consecutive recorded points along all the paths (strokes) of the user, we connect the two nodes (areas) where the two points are located. For each user swipe graph, we construct the distribution-based network identity and sample 180 subgraphs for testing, and we use the supervised splitter for authentication. Serwaddaet et al. [27] proposes a stroke-based authentication by using 28 features of each stroke, including velocity and acceleration along the stroke, to authenticate user identity. Table 13 lists the EERs of the two methods, and the network authentication method has comparable mean EERs and smaller standard deviation indicating that the method is stable across users. For the SU dataset, two types of phones are used: Samsung Galaxy S6 whose layout is 1440×2560, and we set the node as a 12×12 square area; HTC-One whose layout is 1080×1920,

TABLE 13: User Authentication Performance

Classifier	Network Authentication	Stroke Authentication	
		Horizontal	Vertical
Decision Tree	0.30 (0.04)	0.31 (0.29)	0.38 (0.28)
k-NN	0.39 (0.11)	0.17 (0.14)	0.27 (0.15)
SVM	0.28 (0.04)	0.16 (0.14)	0.18 (0.14)

Mean and standard deviation of the EERs across the population. Stroke authentication separates horizontal and vertical stroke.

TABLE 14: User Authentication Performance

Classifier	Phone	Tablet
Decision Tree	0.41 (0.01)	0.43 (0.05)
k-NN	0.40 (0.01)	0.42 (0.04)
SVM	0.42 (0.01)	0.45 (0.04)

Mean and standard deviation of the EERs across the population.

and we set the node size 9×9. We set these node size to make the generated grid networks having similar size as the networks in the Latech dataset. Also, the layout of the tablet is 1536×2048 and we set the node as a 13×13 square area. Table 14 shows the EERs of the SU dataset are around 0.4 for both phone and tablet.

8 Limitations

We have demonstrated the effectiveness of using network identities for network identification and network authentication. However, based on the problem settings, there are a few assumptions and limitations of the method: (1) The networks are not isomorphic, i.e., N_i and N_j are isomorphic $\implies i = j$. If two networks are isomorphic, they are basically the same graph after anonymization, and there is no way to distinguish them; and (2) Subgraph G is not too small to lose its identity. Consider a small subgraph such as a triad $\triangle$, which can be found in most networks, and it does not make much sense to verify its identity; (3) While networks can be sampled using different sampling strategies, here we assume subgraphs are sampled by random node sampling, and we create distribution-based network identities using the same sampling strategy; (4) We assume that we have the whole networks, so we can build ground-truth network identities in the system; and (5) We do not consider fraud in the identity collection process.

9 Related Work

Additionally, our work has links to the following areas:

I. Subgraph Isomorphism. Subgraph isomorphism problem has been long studied in graph theory. As solving subgraph isomorphism leads to the maximum clique problem and testing the existence of a Hamiltonian cycle in a graph, it is an NP-complete problem [5]. Under certain conditions (e.g. the subgraph is a planar graph), the complexity of the problem can be reduced to linear time [28]. However, the conditions are hard to fulfill for large networks, which is the main motivation for us to investigate identification/authentication methods. As we mentioned before, the time complexity to compute the network identity (both embedding-based and distribution-based) is linear in the number of nodes n and edges m of the network.

II. Graph Embedding. Theoretically, any (sub)graph embedding method [29], [30], [31] can be used for embedding-based identity and distribution-based identity if it can preserve subgraphs information and/or capture other relationships between subgraphs and the network. Our study uses Kronecker points as an example, as we mathematically prove that Kronecker points meet these requirements.

III. Identification and Authentication. In biometrics, many human physiological or behavioral characteristic are used to facilitate identification, such as facial information [32], iris [33] and gait [34]. Theoretically, if one converts these characteristic to network data, our graph-based method can be applied leading to graph-based biometrics.

IV. Network Categorization and Network Classification. Network categorization aims to predict the category (or domain) of a network, and most network classification studies have been focused on the classification of graphs within a particular category such as molecular graphs [35], [36]. C. James et al. [37] use 12 graph features, such as density, number of triangle, and the like to predict the category of a network with a high accuracy. Different from them, we are trying to identify whether a graph is sampled from a network not a general category. However, these methods may help us improve our identification systems introduced here by conducting network categorization as a preprocessing step.

10 Conclusions

To the best of our knowledge, we are the first one to formulate the network identification and network authentication problems. In this study, we propose and compare two types of network identities, and we demonstrate their utility in both problems. The embedding-based identity is easy to construct, but the distribution-based identity performs better with simple methods. For network identification, we propose two approaches to predict the network from which a graph is sampled. The supervised learning method is highly accurate, and a simple method that uses only one Euclidean distance has a reasonable accuracy. For network authentication, we show that the supervised method provides a low equal error rate, and the Voronoi method enables controlling the false reject rate, while attaining a reasonable false accept rate across networks. We show that our graph-based methods can also be used for biometrics, authenticating users based on their touch data on devices.

Acknowledgment

This research is in part supported by the NSF CAREER IIS-1942929.

References

[1] L. Backstrom and J. Leskovec, "Supervised random walks: predicting and recommending links in social networks," in *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 2011, pp. 635–644.

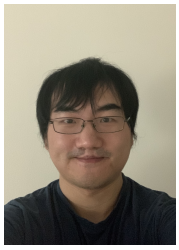
[2] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems*, 2017, pp. 1025–1035.

[3] B. Zhou, J. Pei, and W. Luk, "A brief survey on anonymization techniques for privacy preserving publishing of social network data," *ACM Sigkdd Explorations Newsletter*, vol. 10, no. 2, pp. 12–22, 2008.

[4] L. Backstrom, C. Dwork, and J. Kleinberg, "Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography," in *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 181–190.

[5] S. A. Cook, "The complexity of theorem-proving procedures," in *Proceedings of STOC*. ACM, 1971, pp. 151–158.

[6] A. K. Jain, L. Hong, S. Pankanti, and R. Bolle, "An identity-authentication system using fingerprints," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1365–1388, 1997.
[7] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *arXiv preprint arXiv:1709.05584*, 2017.
[8] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of fingerprint recognition*. Springer Science & Business Media, 2009.
[9] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, "Kronecker graphs: An approach to modeling networks," *JMLR*, vol. 11, no. Feb, pp. 985–1042, 2010.
[10] S. Jin and R. Zafarani, "Representing networks with 3d shapes," in *2018 IEEE ICDM*. IEEE, 2018, pp. 177–186.
[11] S. V. N. Vishwanathan, N. N. Schraudolph, R. Kondor, and K. M. Borgwardt, "Graph kernels," *Journal of Machine Learning Research*, vol. 11, no. Apr, pp. 1201–1242, 2010.
[12] T. Gärtner, P. Flach, and S. Wrobel, "On graph kernels: Hardness results and efficient alternatives," in *Learning theory and kernel machines*. Springer, 2003, pp. 129–143.
[13] J. Leskovec and C. Faloutsos, "Scalable modeling of real graphs using kronecker multiplication," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 497–504.
[14] ——, "Sampling from large graphs," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 631–636.
[15] D. F. Gleich and A. B. Owen, "Moment-based estimation of stochastic kronecker graph parameters," *Internet Mathematics*, vol. 8, no. 3, pp. 232–256, 2012.
[16] S. Jin, R. Wituszynski, M. Caiello-Gingold, and R. Zafarani, "Web-shapes: Network visualization with 3d shapes," in *Proceedings of the 13th International Conference on Web Search and Data Mining*, 2020, pp. 837–840.
[17] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data, Jun. 2014.
[18] R. Zafarani and H. Liu, "Social computing data repository at ASU," 2009. [Online]. Available: http://socialcomputing.asu.edu
[19] Y. Zhang, J. Tang, Z. Yang, J. Pei, and P. S. Yu, "Cosnet: Connecting heterogeneous social networks with local and global consistency," in *Proceedings of the SIGKDD*. ACM, 2015, pp. 1485–1494.
[20] R. Rossi and N. Ahmed, "The network data repository with interactive graph analytics and visualization." in *AAAI*, vol. 15, 2015, pp. 4292–4293.
[21] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial tessellations: concepts and applications of Voronoi diagrams*. John Wiley & Sons, 2009, vol. 501.
[22] D. Koutra, A. Parikh, A. Ramdas, and J. Xiang, "Algorithms for graph similarity and subgraph matching," in *Proc. Ecol. Inference Conf*, 2011.
[23] S. Jin and R. Zafarani, "The spectral zoo of networks: Embedding and visualizing networks with spectral moments," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 1426–1434.
[24] D. Cohen-Steiner, W. Kong, C. Sohler, and G. Valiant, "Approximating the spectrum of a graph," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 1263–1271.
[25] A. Narayanan, M. Chandramohan, R. Venkatesan, L. Chen, Y. Liu, and S. Jaiswal, "graph2vec: Learning distributed representations of graphs," *arXiv preprint arXiv:1707.05005*, 2017.
[26] D. P. Huttenlocher, W. J. Rucklidge, and G. A. Klanderman, "Comparing images using the hausdorff distance under translation," in *Proceedings of CVPR*. IEEE, 1992, pp. 654–656.
[27] A. Serwadda, V. V. Phoha, and Z. Wang, "Which verifiers work?: A benchmark evaluation of touch-based authentication algorithms," in *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*. IEEE, 2013, pp. 1–8.
[28] J. R. Ullmann, "Bit-vector algorithms for binary constraint satisfaction and subgraph isomorphism," *Journal of Experimental Algorithmics (JEA)*, vol. 15, pp. 1–6, 2010.
[29] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in *Advances in neural information processing systems*, 2015, pp. 2224–2232.
[30] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in Neural Information Processing Systems*, 2016, pp. 3844–3852.
[31] B. Adhikari, Y. Zhang, N. Ramakrishnan, and B. A. Prakash, "Sub2vec: Feature learning for subgraphs," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2018, pp. 170–182.
[32] A. K. Jain and S. Z. Li, *Handbook of face recognition*. Springer, 2011.
[33] W. W. Boles and B. Boashash, "A human identification technique using images of the iris and wavelet transform," *IEEE transactions on signal processing*, vol. 46, no. 4, pp. 1185–1188, 1998.
[34] J. Mantyjarvi, M. Lindholm, E. Vildjiounaite, S.-M. Makela, and H. Ailisto, "Identifying users of portable devices from gait pattern with accelerometers," in *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, vol. 2. IEEE, 2005, pp. ii–973.
[35] G. Li, M. Semerci, B. Yener, and M. J. Zaki, "Effective graph classification based on topological and label attributes," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 5, no. 4, pp. 265–283, 2012.
[36] L. Ralaivola, S. J. Swamidass, H. Saigo, and P. Baldi, "Graph kernels for chemical informatics," *Neural networks*, vol. 18, no. 8, pp. 1093–1110, 2005.
[37] J. P. Canning, E. E. Ingram, S. Nowak-Wolff, A. M. Ortiz, N. K. Ahmed, R. A. Rossi, K. R. Schmitt, and S. Soundarajan, "Predicting graph categories from structural properties," *arXiv preprint arXiv:1805.02682*, 2018.



Shengmin Jin Shengmin Jin (S'18) received the B.S. degree in Information and Computing Sciences from Fudan University, Shanghai, China, in 2010 and the M.S. degree in Computer Science from Syracuse University in 2016. He is currently pursuing the Ph.D. degree in the Department of Electrical Engineering and Computer Science at Syracuse University. His research interest includes large scale graph mining and graph representation. His work has been published in data mining conferences including KDD, WSDM, ICDM, and CIKM. Mr. Jin was the recipient of the 2016 Syracuse outstanding achievement award in graduate study.

Vir V. Phoha Vir V. Phoha (M'96 – SM'02) received the Ph.D. degree in computer science from Texas Tech University, Lubbock, in 1992. He is currently a Professor of Electrical Engineering and Computer Science in the College of Engineering and Computer Science at Syracuse University. His research interest includes attack-averse authentication, optimized attack formulation, machine learning, anomaly detection, spatial-temporal pattern detection and event recognition, and knowledge discovery and analysis. Professor Phoha is an ACM Distinguished Scientist and is a Fellow of AAAS and a Fellow of SDPS.

Reza Zafarani Reza Zafarani received his PhD in computer science from Arizona State University in 2015. He is currently an Assistant Professor at the department of electrical engineering and computer science at Syracuse University. His research interests are in Data Mining, Machine Learning, Social Media Mining, and Social Network Analysis. His research has been published at major academic venues, and highlighted in various scientific outlets. He is the principal author of "Social Media Mining: An Introduction", a textbook by Cambridge University Press and the associate editor for SIGKDD Explorations and Frontiers in communication. He is the winner of the President's Award for Innovation and outstanding teaching award at Arizona State University.