Network Identification and Authentication

Shengmin Jin
Data Lab, EECS Department
Syracuse University
shengmin@data.syr.edu

Vir V. Phoha
EECS Department
Syracuse University
vvphoha@syr.edu

Reza Zafarani
Data Lab, EECS Department
Syracuse University
reza@data.syr.edu

***Abstract*—Research on networks is commonly performed using anonymized network data for various 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 isomorphism. However, subgraph isomorphism 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 various real-world networks. Using these identities, we propose two approaches for network identification. One method uses supervised learning and can achieve an identification accuracy rate of 94.7%, and the other, which is easier to implement, relies on distances between identities and achieves an accuracy rate of 85.5%. 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. 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**

I. Introduction

Networks are all around us, in science (e.g. biological networks), engineering (e.g. power grids), and our daily life (e.g. communication networks), motivating research on networks. Research on networks is commonly conducted on anonymized graphs for many reasons such as privacy protection. For example, to protect the privacy of users while preserving network properties, anonymization methods are often used before social network data is published [1]. To validate the authenticity of such anonymized graphs, one may ask questions such as: Given a large graph G, can we verify that it is a Twitter graph but not collected from Facebook or a biological network? Can we identify the source of the anonymized graph, i.e., its network identity? To answer these questions, a natural solution is to check if a network (e.g., Twitter) has a subgraph isomorphic to anonymized G. This requires solving *subgraph isomorphism*, which is NP-complete [2], so this solution is infeasible for most large networks. Hence, we need an alternative with reasonable accuracy and higher efficiency.

Problem Formulation. In biometrics, there are two types of systems to identify a person: (1) identification systems and (2) authentication systems [3]. An identification system identifies a subject without the subject's claims of her identity. It tries to match the subject with everyone enrolled in the system database and gets the best match. On the other hand, an authentication system either rejects or accepts the submitted claim of identity. In spite of their differences, sometimes the terms authentication and identification are used interchangeably [3]. Inspired by biometrics, 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*). G is claimed to be a subgraph sampled from a certain network N_i using sampling strategy S. The authentication system either accepts or rejects this claim.

In both problem settings, there are a few assumptions: (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.

Following the problem formulation, we first aim to build an identity for a network, similar to how a fingerprint represents a person. In this paper, we propose two approaches to build a network identity: **I. Embedding-based Identity**. Intuitively, one can represent a network using a feature vector or its graph embedding. Graph embedding techniques aim to map a graph into a low-dimensional vector and efficiently preserve the network structure. Therefore, one can represent the identity of a network N_i with its embedding, and match the embedding of subgraph G with the network identities of others. **II. Distribution-based Identity**. One limitation of the embedding-based identity is that it is not unique, as generally graph embedding approaches do not guarantee uniqueness for different networks. Therefore, inspired by the design of the *ridge-based representation* [4] 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 common hypothesis that the local ridge

structures (minutiae, e.g. ridge ending and ridge bifurcation) and their distributions can capture the distinctiveness of 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 can preserve uniqueness and subgraph information.

The Present Work. In this paper, we introduce network identification and authentication with the following contributions:

1. Network Identity. We introduce a network identity and two identity types: *embedding-based identity* and *distribution-based identity*. We demonstrate the uniqueness of the distribution-based identity by showing that for real-world networks the similarity of such identities of various networks is generally low. We show examples on how the structural differences in networks are reflected in their identities.

2. Network Identification. With the network identities, we provide two methods to predict the network from which a graph is sampled. The supervised learning method shows a high accuracy (94.7%). We also provide an easier to implement method which relies on the distances between the embeddings to the network identity, achieving an 85.5% accuracy.

3. Network Authentication. We propose two methods to solve the problem: a *supervised splitter*, which has a low equal error rate, and a *Voronoi splitter*, which allows controlling the false reject rate with an acceptable false accept rate across networks.

The paper is organized as follows. In Section II, we introduce two types of network identities. Our experimental data is detailed in Section III. We discuss the uniqueness of identities and *partial identity* in Section IV. We propose solutions to network identification in Section V, and network authentication in Section VI. We conclude in Section VII.

II. Network Identity

To identify a graph, the first step is to build an identity for each network. Here, we propose two types of identities: (A) embedding-based identity and (B) distribution-based identity.

A. Embedding-based Identity

Theoretically, any embedding method that can preserve structural network information and capture the similarity and/or other relationships between samples (subgraphs) and the network can be used as an embedding-based identity. Here, we choose *Kronecker points* as the embedding method and show its utility for both network authentication/identification.

Stochastic Kronecker Graphs and Kronecker Points. Stochastic Kronecker Graphs [5] 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*, one can get a large probability matrix $\mathcal{P}$ with the k^{th} Kronecker power of Θ, i.e., $\mathcal{P} = \Theta^{\otimes k} = \underbrace{\Theta \otimes \Theta \cdots \otimes \Theta}_{k \text{ times}}$, and $\mathcal{P}$ can be used to generate an adjacency matrix. When modeling a network using Stochastic Kronecker graphs, we aim to learn Θ which is most likely to have generated the adjacency matrix $A \in \mathbb{R}^{n^k \times n^k}$ of the network which we are modeling, i.e., $P(A|\mathcal{P})$ is maximized. The KronFit algorithm can estimate the Kronecker initiator matrix for a real-world graph in linear time using maximum likelihood (for details refer to Ref. [5]). 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 [6]. 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. One 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 [5]. Values a and d represent the proportion of edges within each of the groups, and value b represents the proportion of edges between the two groups. Based on the values order (e.g., $a > b > d$ or $b > a > d$), one can obtain whether a network has a core-periphery, dual-core or random structures [6].

B. 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 a recent advancement in network representation:

Network Shapes. Network shapes aim to represent a network using 3D shapes [6]. The framework to build a network shape includes three steps: (1) *Sample subgraphs from the network via a sampling method.* For a network shape to represent the distribution of embedding values for subgraphs of the network, one should sample many subgraphs first. Theoretically, any sampling method can work. (2) *Use a graph embedding method that can map a graph to a 3D point.* The goal is to use an embedding method that can capture the properties of the graph within its embeddings, so that the distribution of embedding values can be closely related to the network properties. Given such a method, one can represent a network and its subgraphs obtained from Step 1 as a set of 3D points. (3) *Fit 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.

With the framework to build a network shape, one can have her own algorithm to build a concrete shape. Here, we build a network shape for each network as its distribution-based identity: (1) We utilizes *Random Node Sampling* to sample subgraphs from the network by varying the proportion of nodes from 0% 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; (2) For each sample (and the whole network), we embed it to a Kronecker point in the 3D space. In total, we generate $20 \times 9 + 1 = 181$ Kronecker points for each network; (3) We fit a 3D shape to all the Kronecker points by computing their convex hull, which 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, we can plot a network identity. Figure 1 shows the identity for YouTube (detailed in Section III).

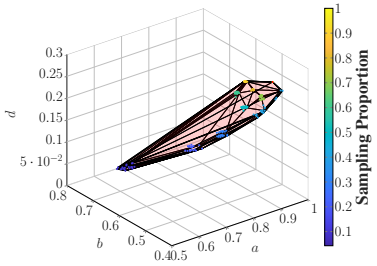
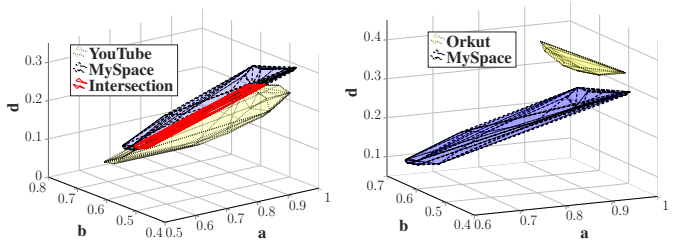


Fig. 1: Distribution-based Identity for YouTube



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

Fig. 2: Two Pairs of Distribution-based Identities

III. Data Description

For our experiments, we use ten real-world networks from four general network categories: social networks, collaboration networks, road networks, and biological networks. We include four social networks: *Hyves* [7], *MySpace* [8], *Orkut* [9], and *YouTube* [9]; two collaboration networks: *Astro-Ph* [9] and *Cond-Mat* [9]; two road networks: *Road-CA* [9] and *Road-PA* [9]; and two biological networks: *Bio-Grid-Yeast* [10] and *Bio-Dmela* [10]. The data statistics are in Table I.

IV. Uniqueness and Partial Network Identity

A. Uniqueness of Network Identity

An identity is to be unique. As discussed, graph embedding generally does not guarantee 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 [5], [6], 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 [5]. 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:

$$\text{similarity}(A, B) = \frac{\text{volume}(\text{ID}_A \cap \text{ID}_B)}{\text{volume}(\text{ID}_A \cup \text{ID}_B)}, \quad (1)$$

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 II lists the similarity between all pairs of the identities. 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. In general, the highest similarity is 0.22, which does not violate the uniqueness of the network identity across different networks.

TABLE I: Dataset Statistics

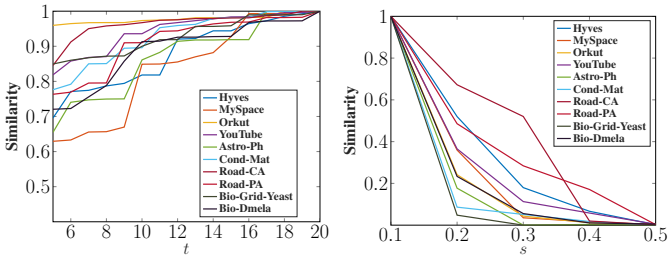
Type	Network	$\|V\| = n$	$\|E\| = m$
Social Networks	Hyves [7]	1,402,673	2,777,419
	MySpace [8]	854,498	5,635,296
	Orkut [9]	3,072,441	117,185,083
	YouTube [9]	1,134,890	2,987,624
Collaboration Networks	Astro-Ph [9]	18,772	198,050
	Cond-Mat [9]	23,133	93,439
Road Networks	Road-CA [9]	1,965,206	2,766,607
	Road-PA [9]	1,088,092	1,541,898
Biological Networks	Bio-Dmela [10]	7,393	25,569
	Bio-Grid-Yeast [10]	5,870	313,890

TABLE II: Distribution-based Identity Similarity

Type	Network	Hyves	MySpace	Orkut	YouTube	Astro-Ph	Cond-Mat	Road-CA	Road-PA	Bio-Dmela	Bio-Grid-Yeast
Social Networks	Hyves	1	0.01	0	0	0	0.03	0	0	0	0
	MySpace	0.01	1	0	0.07	0.04	0.03	0	0	0	0
	Orkut	0	0	1	0	0	0	0	0	0	0
	YouTube	0	0.07	0	1	0.05	0.01	0	0	0	0
Collaboration Networks	Astro-Ph	0	0.04	0	0.05	1	0.08	0	0	0	0
	Cond-Mat	0.03	0.03	0	0.01	0.08	1	0	0	0	0
Road Networks	Road-CA	0	0	0	0	0	0	1	0.22	0	0
	Road-PA	0	0	0	0	0	0	0.22	1	0	0
Biological Networks	Bio-Dmela	0	0	0	0	0	0	0	0	1	0
	Bio-Grid-Yeast	0	0	0	0	0	0	0	0	0	1

B. 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



(a) Change with t (b) Change with s

Fig. 3: Distribution-based Identity Change with t and s

identity we have constructed is a practically "complete" network identity. A few questions come 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 3a 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. Next, we fix $t = 20$ and vary the step size s from 10% to 50%. Figure 3b 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. In Section V and VI, we will discuss the performance of the partial identity for the network identification/authentication problems.

V. Network Identification

A. 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 ten networks, we have $180 \times 10 = 1,800$ samples to be identified/authenticated.

B. 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 (=10) 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* [11] 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 10 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.

TABLE III: Identification Accuracy with Embedding-based Identity

Type	Voronoi Splitter	Supervised Splitter	Baselines	
			Top Eigenvalues	Random Prediction $(1/n)$
All Networks	50.7%	**93.3%**	70.5%	10%
Social Networks	61.4%	97.5%	78.5%	25%
Collaboration Networks	82.8%	87.2%	83.1%	50%
Road Networks	46.7%	78.3%	55.3%	50%
Biological Networks	94.2%	100%	99.7%	50%

We provide two baselines for comparison.

1) **Top Eigenvalues**. Top eigenvalues have been used to study graph similarity. We compute the top 5 eigenvalues of each sample and use as features for classification.
2) **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 III. 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 III illustrates that (1) both Voronoi splitter and Supervised Splitter outperform the random prediction; (2) Voronoi splitter performs not as good as the Top Eigenvalues; (3) Supervised Splitter performs best and achieves an overall accuracy of 93.3% ; and (4) the performance on road networks is not as good as other 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.

C. Identification with Distribution-based Identity

To use the distribution-based identity for identification, we follow the similar roadmap as the embedding-based method. The difference is that now 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:
$$d(A, B) = \inf\{d(x, y) | x \in A, y \in B\}. \quad (2)$$
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:
$$d_H(A, B) = \max\{\sup_{a \in A} \inf_{b \in B} d(a, b), \sup_{b \in B} \inf_{a \in A} d(a, b)\}. \quad (3)$$
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 [12]. In our case, $\mathsf{d}_{\mathsf{Hausdorff}}$ refers to the distance from a point to the farthest boundary point (i.e., extreme points) of the shape.

TABLE IV: 90th Percentile of the Distance Distribution

Type	Network	$d_{shortest}$	$d_{extreme}$	$d_{Hausdorff}$
Social Networks	Hyves	0.0149	0.0331	0.5218
	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
Road Networks	Road-CA	0.0182	0.0595	0.7866
	Road-PA	0.0297	0.0917	0.8327
Biological Networks	Bio-Dmela	0.0087	0.0413	0.7175
	Bio-Grid-Yeast	0.0040	0.0483	0.2489

TABLE V: Identification Accuracy with Distribution-based Identity

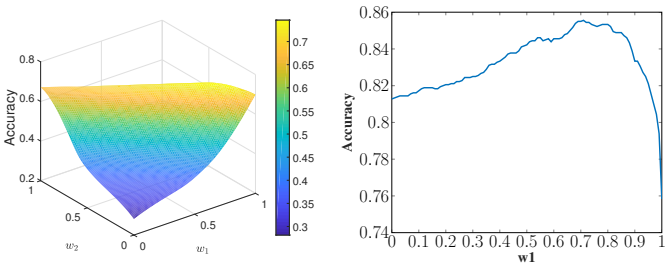
Type	Voronoi Splitter				Supervised Splitter	Baselines	
	$d_{shortest}$	$d_{extreme}$	$d_{hausdorff}$	$d_{weighted}$		Top Eigenvalues	Random Prediction $(1/n)$
All Networks	78.9%	81.3%	24.7%	**85.5%**	**94.7%**	70.5%	10%
Social Networks	93.5%	93.1%	35.3%	94.4%	98.9%	78.5%	25%
Collaboration Networks	86.7%	91.1%	50%	95.3%	98.8%	83.1%	50%
Road Networks	66.4%	56.9%	60%	61.9%	79.7%	55.3%	50%
Biological Networks	100%	100%	57.5%	100%	100%	99.7%	50%

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.

For each network and the test samples drawn from it, we list the 90th percentile of the distances distribution in Table IV. Based on the definitions, we know that $d_{shortest} \leq d_{extreme} \leq d_{Hausdorff}$. 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, indicating that different subgraphs of the same network can be far away from each other.

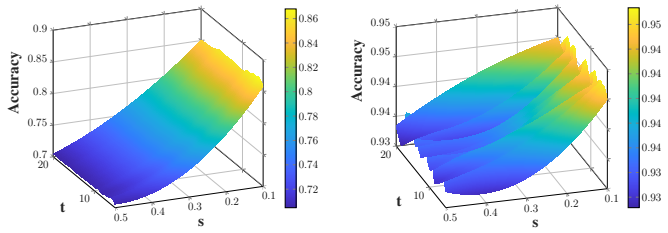
Next, we use the three distances with the two splitters we used in the last section for identification, and report the result in Table V. Compared with the embedding-based identity, the Voronoi Splitter using the distribution-based identity with $d_{shortest}$ and $d_{extreme}$ performs significantly better and can outperform both baselines. We hypothesize that this is due to the distribution-based identity better capturing subgraph information. It is not surprising that $d_{Hausdorff}$ does not perform well as it can be explained by our observation and discussion of the 90th 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 $d_{weighted} = w_1 \times d_{shortest} + w_2 \times d_{extreme} + w_3 \times d_{Hausdorff}$, where $w_1 + w_2 + w_3 = 1$. We do grid search on the feasible weights w_1, w_2, w_3 and plot the accuracy change in Figure 4a. 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 85.5% with $w_1 = 0.71, w_2 = 0.29, w_3 = 0$. Figure 4b 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 $d_{weighted} = 0.71 \times d_{extreme} + 0.29 \times d_{extreme}$, and in general $d_{weighted}$ performs best among these distances.

For Supervised Splitter, we use $d_{shortest}$, $d_{extreme}$ and $d_{Hausdorff}$ as features. Each graph G has $3 \times 10 = 30$ features for all networks and we use the name of the networks as the class labels. Table V shows that compared with the Embedding-based identity, the performance improves and reaches an overall accuracy rate of 94.7%.



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

Fig. 4: Accuracy with Weighted Distance $d_{weighted}$



(a) Voronoi Splitter (b) Supervised Splitter

Fig. 5: Prediction Performance with Partial Identity

Identification with Partial Network Identity. As discussed in Section IV, 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 our previous study on the similarity of partial network identity to complete network identity, we speculate that the network identification accuracy is more sensitive to the change of the sampling step size s. Figure 5a and 5b plot the accuracy change of Voronoi splitter (using $d_{weighted}$) and Supervised Splitter respectively with different s and t configurations. We observe that (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 slightly drops with the increase in sampling step size s from 94% to 93%. (2) Similar patterns are observed for Voronoi Splitter. Differently, the accuracy decreases more and faster with the increase of s, from 85% to 70%.

VI. 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.

A. 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. 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,

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

Type	Networks	Embedding-based			Distribution-based ($\mathsf{d}_{\mathsf{shortest}}$)		
		Accuracy	AUC	FAR	Accuracy	AUC	FAR
Social Networks	Hyves	34.33%	0.59	71.73%	98.38%	0.95	0.68%
	MySpace	39.06%	0.64	67.22%	92.38%	0.91	7.47%
	Orkut	34.89%	0.64	72.35%	99.61%	0.98	0.00%
	YouTube	41.05%	0.64	64.57%	95.00%	0.82	1.73%
Collaboration Networks	Astro-Ph	40.89%	0.62	64.38%	85.17%	0.86	15.00%
	Cond-Mat	34.28%	0.54	70.56%	77.94%	0.83	23.40%
Road Networks	Road-CA	86.00%	0.80	12.65%	91.89%	0.91	8.09%
	Road-PA	72.50%	0.63	25.06%	89.67%	0.86	9.44%
Biological Networks	Bio-Dmela	35.83%	0.60	70.19%	99.44%	0.83	0.12%
	Bio-Grid-Yeast	44.17%	0.68	62.04%	99.89%	0.99	0%

TABLE VII: Authentication with Supervised Splitter

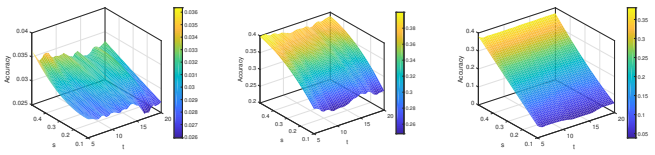
Classifier	Embedding-based	Distribution-based
Decision Tree	0.18 (0.08)	0.08 (0.07)
k-NN	0.22 (0.13)	0.16 (0.19)
SVM	0.27 (0.15)	0.08 (0.08)

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

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 VI shows that the method does not work well with embedding-based identity, but performs well with distribution-based identity. The FAR varies from 0% to more than 20%, and for most networks it is below 10%. Moreover, we vary r when using the distribution-based identity and find that when r is 90, the FAR and FRR are equal, which 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 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 VII. The results show that classifiers using both identities have a low EER indicating a reasonable performance. 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 6a and 6b illustrate the change of the average FAR and FRR for Supervised Splitter with the partial network identities with various s and t values. We notice that both FAR and FRR slowly increase with the sampling step size s and do not change much with t. FAR is generally lower than 4% and FRR increases from 20% to 40%. For Voronoi Splitter, we use the distances to the surfaces and set the percentile $r = 90$, so FRR is fixed at 10%. Figure 6c shows the change in FAR. Similarly, we find the splitter is not sensitive



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

Fig. 6: Authentication Performance with Partial Identity

to t. However, 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 FAR and FRR.

VII. Conclusions

We introduce the network identification and network authentication problems. We propose and compare two types of network identities, and 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.

References

[1] 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.

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

[3] 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.

[4] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of fingerprint recognition*. Springer Science & Business Media, 2009.

[5] 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.

[6] S. Jin and R. Zafarani, "Representing networks with 3d shapes," in *2018 IEEE ICDM*. IEEE, 2018, pp. 177–186.

[7] R. Zafarani and H. Liu, "Social computing data repository at ASU," 2009. [Online]. Available: http://socialcomputing.asu.edu

[8] 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.

[9] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data, Jun. 2014.

[10] R. Rossi and N. Ahmed, "The network data repository with interactive graph analytics and visualization." in *AAAI*, vol. 15, 2015, pp. 4292–4293.

[11] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial tessellations: concepts and applications of Voronoi diagrams*. John Wiley & Sons, 2009, vol. 501.

[12] 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.