Separating Terrorist-Like Topological Signatures Embedded in Benign Networks

Kshiteesh Hegde
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, NY
hegdek2@rpi.edu

Malik Magdon-Ismail
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, NY
magdon@rpi.edu

***Abstract*—We study the problem of identifying topologically adversarial nodes in real networks using a graph classification methodology. To test our approach, we implant nodes from a terrorist network in a wide variety of benign real networks to create a topologically heterogeneous network. We capture local information using structured image embeddings of adjacency matrices and identify the terrorist nodes. We recover up to 85% of the terrorist-like nodes implanted in friendly host networks while keeping the false positive rate as low as 1%.**

***Index Terms*—graph classification, terrorist network, deep learning, network signature**

I. Introduction

Identifying rogue actors in an otherwise friendly scenario is one of the primary concerns for the military in the US and across the world. A group of nodes in a Facebook network conspiring to perform a terrorist attack is a prime example of such a problem. In this work, we provide an approach to this problem. We ask the question: Can one identify malicious nodes in a real network? We use an approach based on graph classification algorithms presented in [1] and [2].

To test our approach, we construct a hybrid terrorist-X network, where X represents the benign network. We start with a real world network (see Section III-A) like e-commerce, citation, social networks etc. We collect subgraphs of different sizes using random walks from various parts of the friendly host network and do the same for a terrorist network from [3]. The model, which is trained on pure networks, outputs a set of labels which serve to classify a node as friendly or terrorist. The outline of our approach to the problem is shown in Figure 1.

The input to our graph classification algorithm are network signatures as depicted in Figure 1 and 2. The network signatures shown in Figure 2 as introduced in [1] are the structured adjacency matrices of subgraphs. In the image, a black pixel at position (i, j) denotes an edge between nodes i and j. It is structured according to the ordering scheme presented in [1] and is presented in more detail in Section III-C. They are a powerful representation [2] of networks since they are agnostic to the type of the network. They can be applied to a wide variety of networks including, but not limited to, social, information, transportation and even terrorist networks. One of the applications of this representation is subnetwork classification. They are good for machine learning algorithms and have an intuitive visual representation.

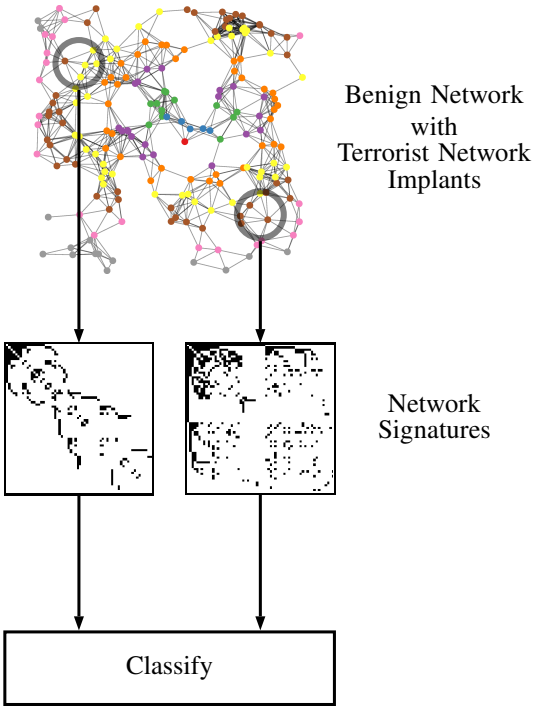
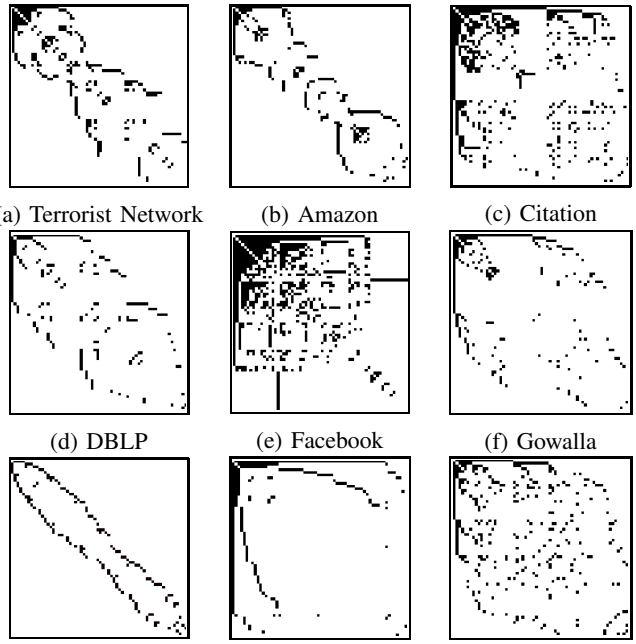


Fig. 1: Workflow of this study. Image embeddings of local adjacency matrices are the input to a classifier, which produces node labels.

A signature of a network reflects its function. For example, the function of a road network is transportation. The functions of a transportation network include having the ability to connect local places in a city as well as distant cities via highways. It also needs to manage traffic during rush hours. It need not have connections from every place to every other place but the it has a large cut. The networks that evolve to support a transportation function are grid-like. That is the



(a) Terrorist Network (b) Amazon (c) Citation

(d) DBLP (e) Facebook (f) Gowalla

(g) Road Network (h) Web (i) Wikipedia Network

Fig. 2: Signatures of all networks to demonstrate the structured image embedding feature.

signature of a road network (Figure 2g). The connections in a network evolve in a particular way to support a particular function and develop a signature. Similarly, different networks with different functions evolve in different ways and develop different signatures.

In this work, we use the signatures to identify the terrorist nodes in real networks. We use models previously trained on these signatures (see Section III) from the untouched friendly setting to obtain a set of predicted labels for each node. We then use linear programming to arrive at the optimal weights for labels and construct a final set of predicted labels for each node.

Our Contribution

We show that a method for node classification based on image classification is highly intuitive and simple while being tremendously successful at recovering terrorist nodes in real world networks up to a rate of 85% while keeping the false positive rate as low as 1%.

II. Related Work

We study the problem of identifying topologically terrorist-like nodes in real networks by using image classification to capture local structure.

The idea of using the image embedding of the adjacency matrix as a feature was first introduced in [1]. Based on this idea, authors in [2] showed with great success that parent networks of tiny subgraphs (as small as 8 nodes) can be identified. They also used Caffe [4] to show that the structured image embedding features can be used for classification in a transfer learning setting. In this work, we use the idea to identify terrorist-like behavior in real networks.

There has been little work on this problem. The authors in [5] study the structure of a terrorist network and do a case study but do not address the problem of classification. A classification study on simulated networks was done by [6]. But they do not provide any guarantees as to how their proposed method will work on real world networks.

However, there has been a lot of work in the area of graph classification in general. Authors in [7] perform semi-supervised feature selection by searching for optimal subgraph features. They define a metric that governs how features are selected. There is also the idea of using pattern recognition along with feature selection where the idea is that graphs from the same class should have similar attributes [8]. Spatial distribution of subgraphs is used as features in [9]. In a similar vein, [10] introduces a pattern exploration scheme that looks for co-occurring features in subgraphs to perform binary classification. It is unclear how multi-class classification can be achieved (if at all) using this approach. In [11], the authors talk about extracting important features in a multi-label setting. They assume that the given data is already labeled (multiple times) and the task is to choose the correct label from the set. All the above methods require construction of features that are dependent on the given data. This can be non-trivial in cases where one has to deal with a diverse set of data as is the case in this study. Developing a one size fits all kind of a set of features is near impossible. In case of pattern recognition, if a new pattern or set of patterns emerge only in the test set, then the chances of catching them drastically decreases.

Many graph kernels based on walks, subtrees, cycles, shortest paths etc. have been proposed [12]–[17]. The kernel function computes the similarity between two graphs and then a classifier such as SVM is used for classification. As evidenced by the abundance of different types of kernel functions, it is difficult to come up with a kernel that ticks all the boxes for a given classification problem. The size and domain of the network, complexity of the kernel function all affect the decision of choosing the *right* kernel. So, kernel methods are also affected by the same problems as feature selection methods.

All of the above mentioned literature assumes a friendly setting where one network contains only one type of network. They are of little use when topologically different nodes are embedded in real networks and the task is to identify them.

III. Data and Methodology

A. Data

We used 9 real world networks to construct our heterogenous network. The networks are from a diverse set of domains like e-commerce, social, web, roads etc. Table I provides the number of nodes and edges in each of the individual networks.

B. Implanting Terrorist Subnetworks into Real Networks

Our first task is to construct a testbed using real networks. Each of the real world networks in Table I behaves differently

TABLE I: The homogenous networks used in this study

Dataset	# Nodes	# Edges
Citation [18], [19]	34,546	421,578
Facebook [20]	4039	88,234
Road Network [21]	1,088,092	1,541,898
Web [21]	875,713	5,105,039
Wikipedia [22], [23]	4,604	119,882
Amazon [24]	334,863	925,872
DBLP [25]	317,080	1,049,866
Terrorist Net. [3]	271	756
Gowalla [26]	196,591	950,327

and has a different individual signature [2]. For every network, we take several snapshots of it and splice them together with several snapshots of the terrorist network to obtain a "contaminated" version of each of the networks.

First, we extract 10,000 subgraphs with 8, 16, 32 and 64 nodes from each of the networks shown in Table I (except Terrorist Net.) yielding 8 real world host networks. These are the benign networks that we will introduce terrorist networks nodes to. We create 4 versions of the terrorist network by splicing together 500, 1000, 1500 and 2000 subgraphs of sizes 8, 16, 32 and 64 nodes. Now, these 4 terrorist networks are spliced with each of the 8 friendly host networks to create terrorist-affected host networks with different levels of contamination. Thus, we now have 4 versions of every host-terrorist network combination. Note that at this stage there are no edges between host nodes and terrorist nodes. So, we choose two subgraphs at random insert an edge between two random nodes from the subgraphs. This edge-introduction process is repeated 5 $\times$ the number of subgraphs (10,000 + either 500, 1000, 1500 or 2000) resulting in connected networks. This process in illustrated in Figure 3.

C. Graph Image Embeddings

We briefly describe the process of converting adjacency matrices to lossless image features [1] here. The adjacency matrices can be visualized as images by simply treating 1s as black pixels and 0s as white pixels. However, the same adjacency matrix can be mapped to different images by permuting the rows. Using the image from a random permutation of the rows as input directly to a classifier such as a Convolutional Neural Network (CNN) results in very poor results. It is necessary to first re-order the nodes in a canonized form. We use the ordering scheme shown in [1], to make sure that all permutations of a given adjacency matrix map to the same structured image making it permutation invariant. When these structured images are fed to a CNN, classification performance is significantly improved. Neural networks show tremendous accuracy when it comes to recognizing real world images [4]. As shown in [2], they do very well with homogenous networks as well. Different subgraphs from the same network are different at the microscopic level but are similar on a macroscopic level. Figure 4 is a visualization of the process of extracting network signatures.

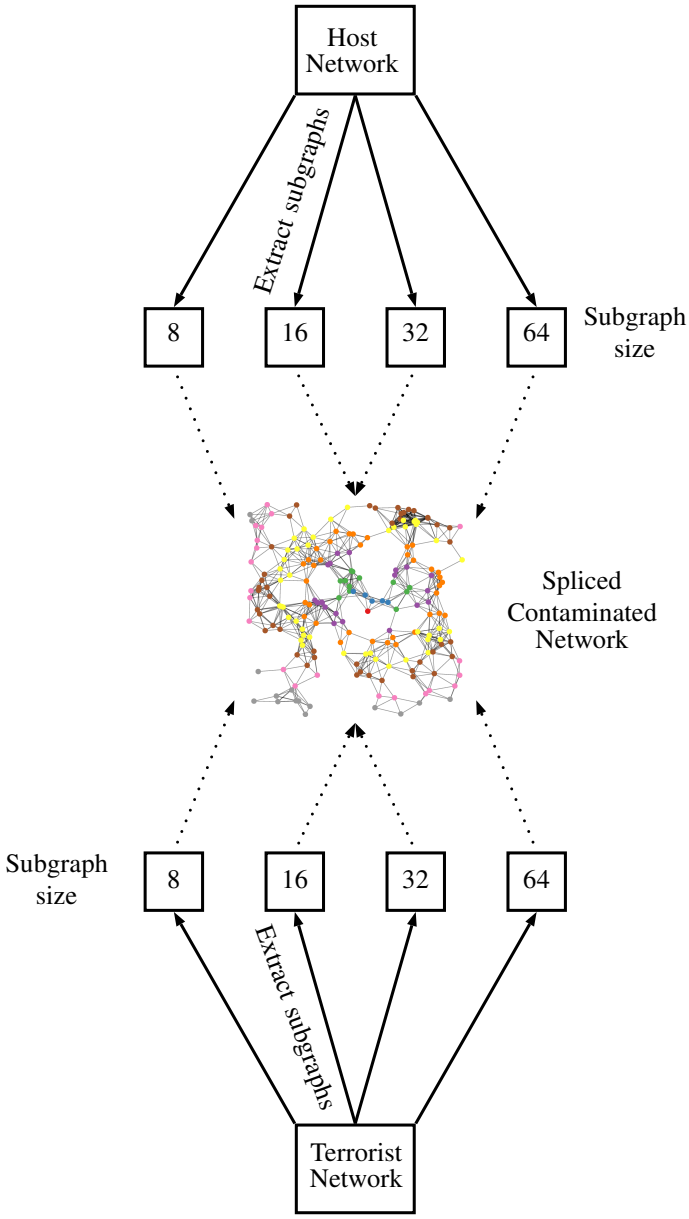


Fig. 3: Implanting terrorist network signatures into real benign networks by splicing subgraphs together.

D. Deep Learning Models

In [2], authors test several classifiers with the task of discriminating between the real world networks mentioned in Table I. CNN performs best with about 86% accuracy. In this work, we trained multiple CNN models for each of the possible 8 pairs of networks. For example, the model used to detect terrorist nodes in a Road - Terrorist network setting, would have been trained only on *pure* road network and *pure* terrorist network. We make this choice since it is reasonable to assume that the user that is using our model to detect terrorist nodes has access to the pure version of the host network and the model has learned the signature of the benign host network. If trained in the heterogenous setting, the model will

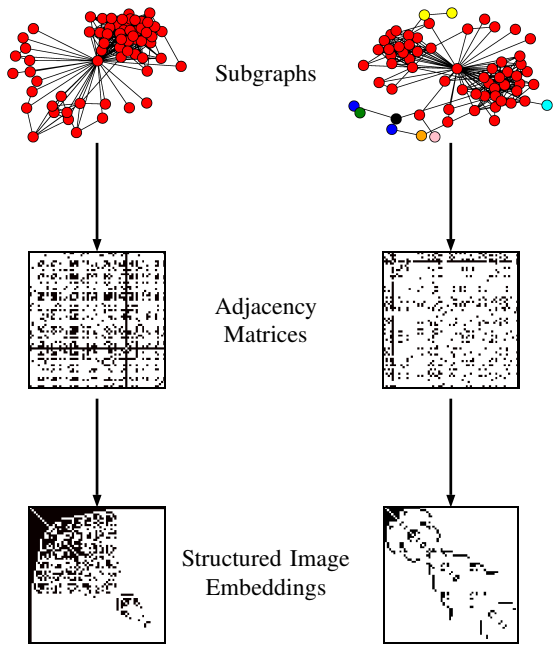


Fig. 4: Obtaining structured image embeddings of graphs

fail to effectively learn network signatures because of lack of consistent local structure.

IV. Experiments and Results

We perform random walks from every node in the contaminated version of each of the real world networks (except Terrorist Net.) in Table I. We construct the structured image embedding of this random walk as described in Section III. Now, the label predicted by the model is assigned to the starting node of the random walk and to all the nodes in the subgraph. The process is repeated with different subgraph sizes (8, 16, 32 and 64).

We weight the labels received from each of the different subgraph sizes using weights calculated in a heterogeneous setting with all 9 (including the terrorist network) networks involved. These weights were arrived at using a linear program. Since the weights apply to the size of subgraph and are independent of the type of the network, they can be used in this setting for simplicity.

Now, each node will have a weight associated to each of the two possible labels. For example, a road network node could be assigned the following weights: (road: 0.89, terrorist net.: 0.11). We then compute different metrics (which are explained below) for each of the networks to measure our model's performance. We only show the best value for each of the metrics for brevity. We observed that the metrics were highly consistent irrespective of the amount of terrorist nodes in the network. We now briefly describe the following metrics before discussing Table II.

- True Positive (TP): Terrorist node is correctly classified as terrorist.

	TPR (Sensitivity)	TNR (Specificity)	False Positive Rate
Road Net.	0.85	0.98	0.01
Web	0.72	0.93	0.03
Wikipedia	0.71	0.97	0.03
Facebook	0.69	0.89	0.07
Citation	0.50	0.96	0.04
DBLP	0.43	0.96	0.03
Amazon	0.42	0.97	0.03
Gowalla	0.42	0.94	0.09

TABLE II: Various Metrics for each of the host networks

- True Negative (TN): Benign node is correctly classified as benign.
- False Negative (FN): Terrorist node is incorrectly classified as benign.
- False Positive (FP): Benign node is incorrectly classified as terrorist.
- True positive rate (TPR) = TP/(TP + FN)
- True negative rate (TNR) = TN/(FP + TN)
- False positive rate (FPR) = FP/(FP + TN)

TPR measure the proportion of terrorist nodes that were correctly classified as such. TNR measure the proportion of host nodes that were correctly ruled out as terrorist. Finally, FPR measures the amount of false hits, i.e., the amount of host nodes that were falsely judged to be terrorist.

The networks are arranged in the decreasing order of TPR in Table II. We note that the road network has the best TPR since it has the strongest signature and hence the most structured. It is interesting to see that Facebook, which is the closest to the terrorist network in terms of both being a network of people, performs third best.

We note that, to mimic real world scenario, we have introduced very few terrorist nodes relative to the number of host nodes while constructing our hybrid terrorist-X networks. This makes the problem much harder, because even if the random walk starts at a terrorist node, it could easily meander in to benign territory.

V. Conclusion and Future Work

In summary, we successfully used a new way to identify terrorist like behavior in topologically heterogeneous networks using structured image embeddings. We showed that this technique is widely applicable since our simple and easy to understand model performs well on a variety of real networks. A future direction would be to train the model on multiple kinds of terrorist networks to increase its capability of recognizing threats to security.

Acknowledgment

This research was supported by the Army Research Laboratory under Cooperative Agreement W911NF-09-2-0053 (the ARL-NSCTA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and

distribute reprints for government purposes notwithstanding any copyright notation here on.

REFERENCES

[1] K. Wu, P. Watters, and M. Magdon-Ismail, "Network classification using adjacency matrix embeddings and deep learning," in *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, Aug 2016, pp. 299–306.

[2] K. Hegde, M. Magdon-Ismail, R. Ramanathan, and B. Thapa, "Network signatures from image representation of adjacency matrices: Deep/transfer learning for subgraph classification," *arXiv:1804.06275*, 2018.

[3] JJATT, "http://doitapps.jjay.cuny.edu/jjatt/data.php," 2009.

[4] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[5] R. M. Medina, "Social network analysis: a case study of the islamist terrorist network," *Security Journal*, vol. 27, no. 1, pp. 97–121, 2014.

[6] T. R. Coffman and S. E. Marcus, "Pattern classification in social network analysis: A case study," in *Aerospace conference, 2004. proceedings. 2004 ieee*, vol. 5. IEEE, 2004, pp. 3162–3175.

[7] X. Kong and P. S. Yu, "Semi-supervised feature selection for graph classification," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '10. New York, NY, USA: ACM, 2010, pp. 793–802.

[8] 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.

[9] H. Fei and J. Huan, "Structure feature selection for graph classification," in *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, ser. CIKM '08. New York, NY, USA: ACM, 2008, pp. 991–1000. [Online]. Available: http://doi.acm.org/10.1145/1458082.1458212

[10] N. Jin, C. Young, and W. Wang, "Graph classification based on pattern co-occurrence," in *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, ser. CIKM '09. New York, NY, USA: ACM, 2009, pp. 573–582.

[11] X. Kong and S. Y. Philip, "Multi-label feature selection for graph classification," in *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 2010, pp. 274–283.

[12] H. Kashima and A. Inokuchi, "Kernels for graph classification," *International Workshop on Active Mining*, 2002.

[13] H. Kashima, K. Tsuda, and A. Inokuchi, "Marginalized kernels between labeled graphs," *ICML*, 2003.

[14] T. Kudo, E. Maeda, and Y. Matsumoto, "An application of boosting to graph classification," *NIPS*, 2004.

[15] K. M. Borgwardt and H.-P. Kriegel, "Shortest-path kernels on graphs," in *Data Mining, Fifth IEEE International Conference on*. IEEE, 2005, pp. 8–pp.

[16] 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.

[17] K. Riesen and H. Bunke, "Graph classification based on vector space embedding," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 06, pp. 1053–1081, 2009.

[18] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over time: Densification laws, shrinking diameters and possible explanations," *KDD*, 2005.

[19] J. Gehrke, P. Ginsparg, and J. Kleinberg, "Overview of the 2003 KDD cup," *SIGKDD Explor. Newsl.*, 2003.

[20] J. Leskovec and J. J. Mcauley, "Learning to discover social circles in ego networks," *NIPS*, 2012.

[21] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, "Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters," *Internet Mathematics*, 2009.

[22] R. West and J. Leskovec, "Human wayfinding in information networks," *WWW*, 2012.

[23] R. West, J. Pineau, and D. Precup, "Wikispeedia: An online game for inferring semantic distances between concepts," *IJCAI*, 2009.

[24] J. Leskovec, L. A. Adamic, and B. A. Huberman, "The dynamics of viral marketing," *TWEB*, 2007.

[25] J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth," *ICDM*, 2012.

[26] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," *KDD*, 2011.