SAND2019-6578C

# A Unique Similarity Metric for Anomaly Detection in Social Networks

## Abstract

*The ability to detect changes between similar networks can be viewed as an anomaly detection problem. This paper presents a network similarity metric that is sensitive enough to detect subtle changes in link strength or network structure without early saturation when dramatic changes must be detected. The algorithm achieves this level of fidelity by combining multiple network analysis algorithms in an efficient manner.*

## 1. Introduction

Graphs are used to represent a variety of network phenomena, such as social interactions in humans [1] and animals [2], web searches [3], traffic [4], semantic analysis [5], brain function [6], and molecular physics [7]. Often these networks are observed to evolve over time, and therefore it is of interest to detect changes in the network structure. In order to detect differences in a network at different points in time, it is necessary to quantify the amount of similarity between two networks. Several graph similarity metrics have been proposed in the literature, each of which leverages different network properties.

There has been a great deal of work regarding graph similarity and how to measure it, but relatively little attention has been paid to methods for continuously monitoring these changes for anomalous behavior. In dynamic networks one would expect a certain amount of natural shifting in nodes and edges over time, which may be characterized as noise. However, a substantial change in the network topology would indicate anomalous activity. By quantifying and tracking the similarity between instances of the network over time, anomaly detection methods could be applied to understand when a network's baseline level of shifting has been punctuated by an event of interest. Our contribution is twofold: 1) We present a new network similarity metric which simultaneously and directly incorporates: node existence, edge existence, edge weight, and higher-order structure and 2) we propose using control charts of the similarity metric to detect anomalies in a network over time.

The literature examining and quantifying graph similarity is extensive. Graph isomorphism, in which two graphs with the same number of nodes are connected in an identical way, is one way of determining whether two graphs are similar [8]. Graph isomorphism is a combinatorial optimization problem that is difficult to solve, and many procedures have been proposed for doing so [9-11]. In addition to algorithms that determine whether two graphs are isomorphic there is also a similarity measure based upon data fusion of isomorphic and nonisomorphic subgraphs [12]. Research on common graphs seeks to identify the maximum or minimum common isomorphic subgraph or supergraph, respectively [13-14].

While graph isomorphism determines whether two graphs are identical, a graph edit distance approach allows one to quantify the similarity between graphs at a higher resolution. Graph edit distance is the smallest cost needed to transform one graph to another via edit operations (i.e., addition and deletion of nodes and edges) [3, 15-17]. However, calculating graph edit distance is computationally expensive, and there is no effective, general method since calculating cost is application-dependent. One study [18] proposes using the Levenshtein distance (i.e., string edit distance) in combination with a canonical labeling system as an alternative. Related to the idea of graph edit distance, another study [1] characterizes the similarity between graphs as a graph differential tuple, consisting of the set of added and removed nodes and edges as well as the set of modified edge weights. They propose several metrics combining information from these three forms of graph edits, and they propose a separate metric based solely upon the difference in edge weights. However, the authors do not combine the graph edits with changes in edge weights for a single, comprehensive metric. Other researchers have leveraged aggregate measures of graph topology using a statistical framework [19-20]. By defining and quantifying network structural properties such as betweenness and degree distribution, similar graphs can be compared, and their structural differences can be described in an interpretable way. Other researchers [21], for example, employ a forgetting mechanism in their calculation of such structural measures to characterize structure in an evolving network. Spectral graph theory has emerged as another option for

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

assessing graph similarity. These methods calculate the distance between the eigen decomposition of the two graphs' matrices [22-23]. Another study [24] proposes the DeltaCon similarity metric which compares pair-wise node affinities of two graphs, although the metric is heavily influenced by which characterization of node affinity is utilized.

Several attempts have been made to combine approaches. One study [6] propose an algorithm that incorporates graph edit distance, string edit distance, and physical location to compare graphs of brain connectivity. Another study [25] fuse graph edit distance and maximum common subgraph into a graph distance metric in an anomaly detection setting for detecting changes on the attack surface of dynamic computer networks. Other researchers [3] also consider the problem of anomaly detection on consecutive graphs in time. They propose sequence and signature similarity measures, adapted from document and vector similarity methods, to detect anomalies on web graphs.

## 2. Formulation

The similarity metric is composed both of simple network metric calculations as well as a small optimization for determining the node matches between clusters.

### 2.1. Similarity calculation

Assume we have two similar networks A and B which each have nodes with unique numeric IDs. Nodes are considered to match when their numeric IDs match. Two links are considered to match when they connect the same two nodes with the same IDs. Note that our implementation assumes bidirectional links for simplicity, but the calculation is still valid when used with directed networks. The similarity between the two networks is calculated as the weighted sum of the four components defined below and can range in value from zero to one.

Link Strength Similarity provides a measure of the link strength similarity between networks and is the sum of the absolute strength differences normalized by the sum of the strengths across both networks. If there is a link in one network that does not exist in the other, then the "missing" link has a strength value of zero. If all link strengths match, this component has value zero. If there are no matching links, then this component has value one.

$$L_S = \frac{\sum_{i=1}^{N} \left| S_i^A - S_i^B \right|}{\sum_{i=1}^{N} [S_i^A + S_i^B]} \quad (1)$$

In this equation, $S_i^A$ is the strength of link $i$ from network $A$, $S_i^B$ is the strength of link $i$ from network $B$, and $N$ is the maximum number of links between both networks.

Matching Link Ratio is the ratio of the number of matching links between networks to the total number of unique links across both networks. Two links are a match if they connect two nodes with the same IDs. If all links match, this component has value one. If no links match, this component has value zero.

$$L_M = \frac{\textit{Number matching links between networks}}{\textit{Total number unique links in both networks}} \quad (2)$$

Matching Node Ratio is the ratio of the number of matching nodes (nodes with equal-valued IDs) between networks to the total number of unique nodes in both networks. If all nodes match, this component has value one. If none of the nodes match, this component has value zero.

$$N_M = \frac{\textit{Number matching nodes between networks}}{\textit{Total number unique nodes in both networks}} \quad (3)$$

Matching Cluster Ratio is the ratio of the number of matching node-cluster labels to the total number of unique nodes across both networks. It is assumed that there are an equal number of clusters in each network. If all nodes have matching cluster labels, this component has value one. If none of the nodes have matching cluster labels, this component has value zero.

$$N_C = \frac{\textit{Number matching node cluster labels between networks}}{\textit{Total number unique nodes in both networks}} \quad (4)$$

Total Similarity, *S*, is the weighted sum of these components:

$$S = 0.25*(1-L_S) + 0.25*L_M + 0.25*N_M + 0.25*N_C \quad (5)$$

Note that the components could be weighted differently based on the application so long as the similarity still has a range from 0 to 1. For example, if the nodes in both network are guaranteed to match (resulting in a constant value of 1 for $N_M$), the coefficient of $N_M$ could be changed to zero and the coefficients for all other components could be changed to 1/3.

## 2.2. Node cluster mapping optimization

In order to calculate the matching cluster ratio component of the similarity, the best cluster mapping between the two networks must be determined. To find the best cluster mapping for a large number of clusters, it becomes too expensive to examine all $C!$ permutations, where $C$ is the number of clusters in each network (assumed to be the same). The goal is to find the mapping with the greatest number of overlapping nodes across all cluster-cluster assignments between networks $A$ and $B$. To find this mapping, a population-based metaheuristic is employed.

A selection of initial solutions is generated by randomly assigning clusters in network $A$ to clusters in network $B$ and calculating the node overlap. The set of random solutions is augmented with a set of greedy solutions which attempt to assign the highest overlapping clusters to each other in a systematic fashion. The overlap counts for each cluster in network $A$ to the highest overlapping cluster in network $B$ can be summed to give the theoretical maximum overlap count for the best mapping solution. A population-based local search technique is used to gradually improve the collection of initial solutions generated.

## 2.3. Node clustering optimization

To compare the clustering of nodes in each network, we must first determine the node-cluster assignments. The standard Louvain algorithm [26] determines the optimal clustering for a collection of linked nodes by maximizing the modularity. The final number of clusters is determined by the algorithm and cannot be a fixed value. Our modified version of the algorithm allows the number of clusters to have a fixed upper bound and is a population-based metaheuristic using modularity as the objective value. The optimization is broken down into two main pieces: initialization followed by solution tuning via local search. The initialization procedure essentially follows the first phase of the Louvain Method by separating all nodes into separate communities then finding the best community for each node by calculating the relative gain in modularity. If the number of remaining communities is still greater than the maximum allowed number of clusters, the communities are optimally combined until there are no more than the allowed number remaining. The resulting solution is augmented by creating a collection of solutions with random node-cluster assignments. All solutions are then tuned by executing a local search of moving single nodes or fully-connected groups of nodes to new communities in an effort to maximize the modularity. The solution with the highest modularity is then retained as the final solution.

# 3. Experiments

In order to determine the efficacy of the algorithm for detecting changes to a network, a simple network consisting of 25 nodes in five distinct clusters is constructed as shown in figure 1.

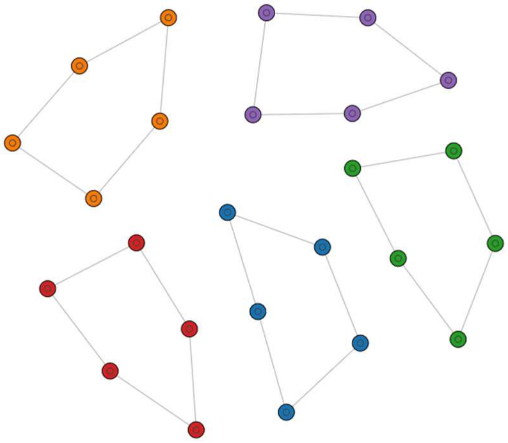

**Figure 1. Simple 25-node, 5-cluster network**

The nodes are minimally connected within each cluster so that overall network modularity is 0.8 with no cross-cluster links. Two experiments are conducted: one to validate the efficacy of the similarity metric when comparing a static network to a randomly modified version of the same network and another which demonstrates the effects on similarity of a gradually changing social network.

## 3.1. Random network modification experiment

In this experiment, a series of modifications is applied to a second, initially identical network (identified as “network 2”) to slowly modify it in specific ways to determine the resultant similarity value. Over a period of 288 iterations, the following modifications are applied to alter the network. Firstly, at each iteration, a random increment or decrement operation is selected at random (50% chance of each). If the action is to decrement, the link strength is decremented by a random amount between one and 15. If the strength goes below zero, the link is removed. If the action is to increment, two nodes are selected at random. If a link currently exists between them, the strength is incremented by a random amount between one and 15. If a link does not exist, a new one is created with strength between one and 15. Secondly, every 30 iterations a node is dropped from the network, meaning that all connected links are removed. This missing node is not included in any of the node-matching or clustering ratio calculations for the similarity metric, so this action will eventually force network 2 to be completely different from network 1 (similarity is zero since there are no common nodes or links).

Since each random change may result in either an increase or a decrease in similarity, the sampled changes are sorted by decreasing similarity value in the figure 2 plots below. These plots demonstrate that the metric can span its full range depending on the changes made to the network.

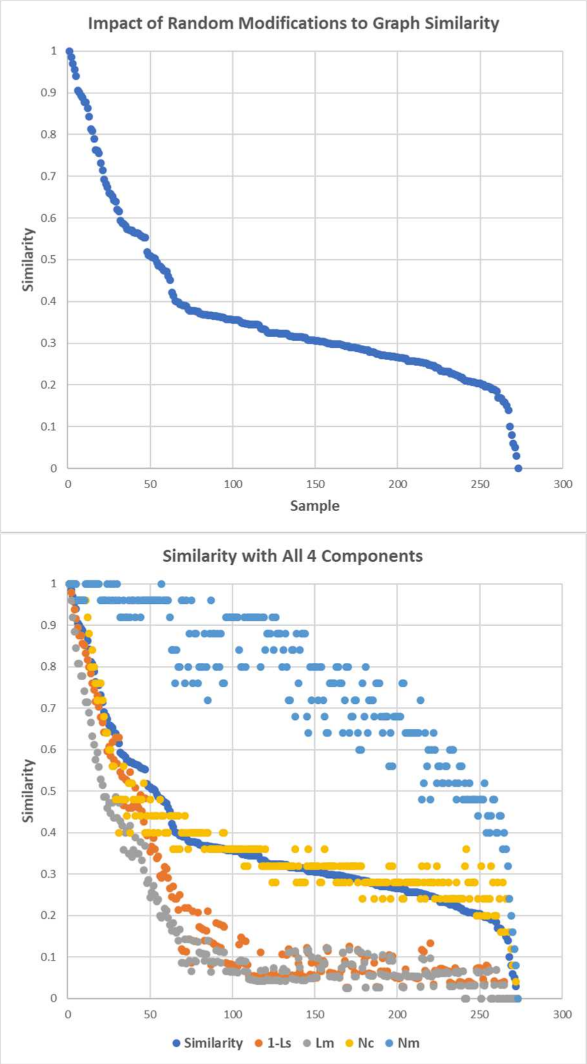

**Figure 2. Impact of random network modifications to similarity metric and associated components**

During the experiment, intermediate networks at similarity values close to 0.75 and 0.50 are captured and examined to see how they differ from the original network. Table 1 summarizes the differences in the node clustering.

**Table 1. Impact of random network modifications to network 2 node clustering**

| Cluster | Seed Network Nodes | 0.76 Network Nodes | 0.50 Network Nodes |
|---|---|---|---|
| c1 | 1-5 | **1-3**, 24 | **1**, 6, 10, 16 |
| c2 | 6-10 | **6-10**, 14-15 | **7-9**, 15 |
| c3 | 11-15 | 4-5, **12-13** | 4-5, **12, 14**, 19, 21-22, 25 |
| c4 | 16-20 | **16-20** | 3, 11, **17-18, 20**, 23 |
| c5 | 21-25 | 11, **21-23, 25** | 2, **24** |

The structural clustering difference can also be seen in figure 3 (split into figures 3a and 3b) where the interior of each node has the original cluster color and the exterior of each node has the new clustering assignment. The width of each link is proportional to its strength and the size of each node is proportional to the strength of the links connected to it.

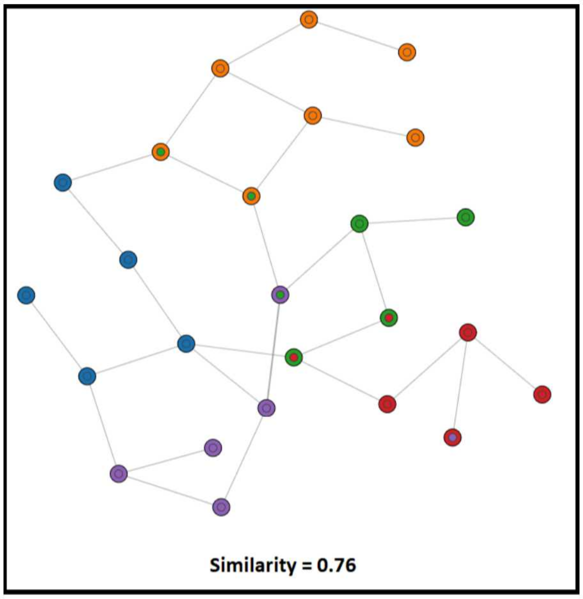

**Figure 3a. Network 2 at similarity value 0.76**

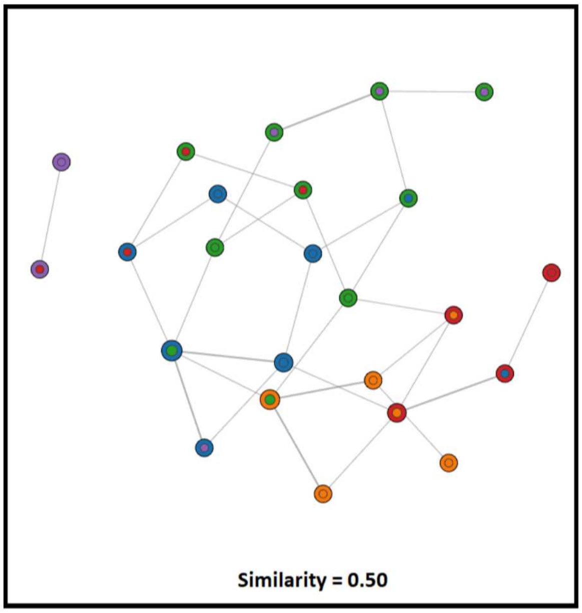

**Figure 3b. Network 2 at similarity value 0.50**

## 3.2. Gradual change of a social network

For this experiment, the same node clusters shown in figure 1 are preserved as the underlying "ground truth" clustering and each node is assumed to be a person in a social network. From this seed network, all possible in-cluster and cross-cluster links are defined. A series of networks are then generated where the link structure is maintained but the level of weekly communication is modeled by a Poisson process with mean two. The sparsest matrix has 60% in-cluster communication with no cross-cluster communication. The densest network has 100% of the in-cluster links with 40% of the cross-cluster links. Note that all links (in- and cross-cluster) are a progressively larger superset of the 60-0 network links as would be (somewhat) expected in a gradually changing network (i.e., new relationships are formed but old relationships are not lost). For example, the 70-30 network contains 70% of the possible in-cluster links and 30% of the possible cross-cluster links and holds all of the same links as the 60-20 network in addition to several new links. The 70-30 in-cluster links include all 25 of the 60-20 network in-cluster links as well as 5 new links and also include all 50 of the 60-20 cross-cluster links with 25 additional links.

Table 2 summarizes the number of links for each percentage combination based on a total of 300 possible links where 50 are in-cluster and 250 are cross-cluster. 25 different networks are generated with link percentages ranging from 60-0 (in-cluster/cross-cluster percentage) to 100-40.

**Table 2. Number of links for each permutation of in-cluster and cross-cluster links**

| In-Cluster % | In-Cluster Link Count | Cross-Cluster % | Cross-Cluster Link Count |
|---|---|---|---|
| 100 | 50 | 40 | 100 |
| 90 | 45 | 30 | 75 |
| 80 | 40 | 20 | 50 |
| 70 | 35 | **10** | 25 |
| 60 | 30 | 0 | 0 |

In order to do a comparison to an "average", we chose the 70-10 permutation as a representative network and set the strength of each link equal to the average of 100 draws from the Poisson process described above. For each of the 25 network permutations, a series of 100 random instances is generated where the link structure is preserved but the link strength is generated from the same Poisson process. If the strength of a link is zero for a draw, it is considered to be nonexistent with respect to the similarity calculation. Similarly, if a node has no associated non-zero links it is also considered to be nonexistent with respect to the metric calculation. The similarity metric is calculated for each randomly generated instance to create a virtual time series of length 2500 across all instances, grouped by network permutation.

Since the overall goal is to create a control chart which detects when the network has significantly changed from the reference point, we performed a distribution fit on the 70-10 network random draws. We compared the fits of three possible distributions: truncated normal, beta, and a mixture of normals. The best fit was selected using a voting procedure based on three model fit criteria: AIC (Akaike Information Criterion), BIC (Bayesian Information Criterion), and the Kolmogorov-Smirnov test. For the 70-10 similarity dataset, the best fit is a truncated normal with mean 0.82653 and standard deviation 0.034531. Figure 4 is the smoothed histogram of similarity values (black dotted line) overlaid with the various distribution fits.

Once the underlying distribution is selected, the control chart limits can be established using the techniques described in [27]. This involves picking a "running length", the average number of samples from the in-control distribution before a sample is observed that appears to be out-of-control. In our case we selected 75. Next, we calculate the probability limits for the out-of-control limits. We use α = 1/(running length) = 1/75 and probability limits = (α/2, 1-α/2) = (1/150, 149/150). Then, using a numerical approximation to the inverse truncated normal distribution, we find the similarity values corresponding to the probability limits and set as the control chart limits.

Figure 5 illustrates the application of these control limits to the virtual time series. The similarity values from the 70-10 series is nearly fully encapsulated by the control limits whereas even the small network change to 70-20 results in the bulk of the samples being outside of the control limits. Also of note is that the XX-0 networks go down in similarity because they do not have any of the cross-cluster links present in the 70-10 network structure.

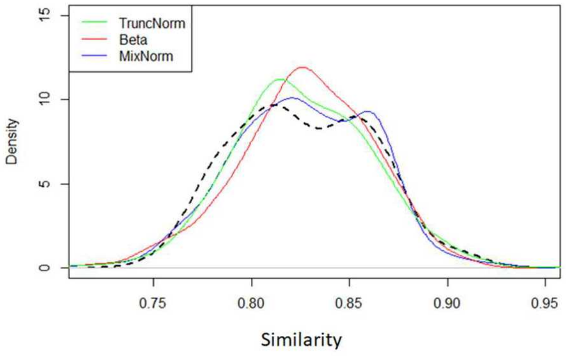

**Figure 4. Selecting the best distribution fit for the 70-10 network similarity values (black dotted line)**

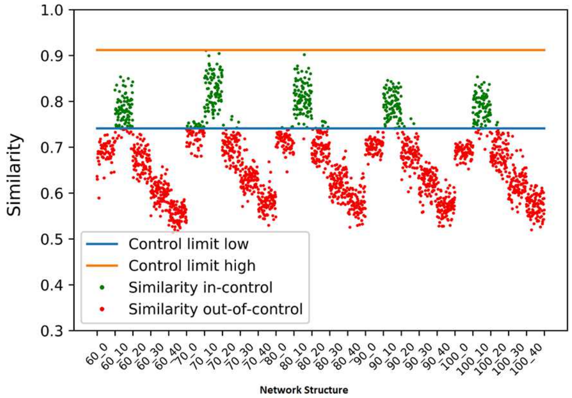

**Figure 5. Application of control chart limits to the full similarity of the simulated network data**

When comparing our similarity metric to that proposed by [1] the main difference is our inclusion of the matching cluster ratio. This added information is structural and not readily apparent by doing basic comparisons between links and nodes. To illustrate the value of this addition, we pulled out the matching cluster ratio contribution to the similarity metric and renormalized so that the new metric has value ranging from zero to one (this is equivalent to setting the weights on all components in equation 5 to be 1/3 except for $N_m$ which is given a weight of zero). We found that a beta distribution with shape 1 parameter 181.41 and shape 2 parameter 27.61 had the best fit, and we used this to create control chart limits for the modified similarity metric. The results of applying the control chart are shown in figure 6.

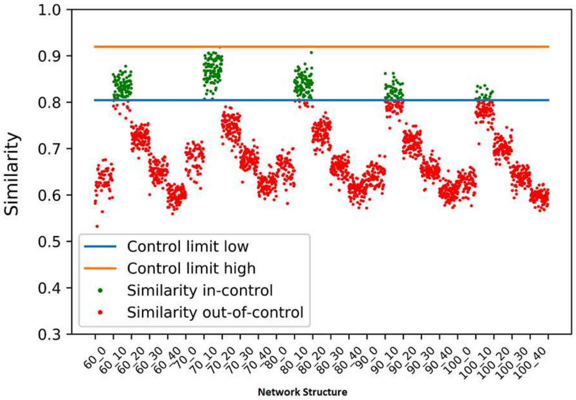

**Figure 6. Application of control chart limits to modified similarity metric with no clustering component**

There are few items to note when comparing full similarity metric (figure 5) to similarity with no clustering information (figure 6). Firstly, the XX_0 networks in figure 6 have a much greater deviation from the XX_10 network similarity values as compared to figure 5 even though their clustering is quite similar. The histogram for 90_0 in figure 7 illustrates how that network clustering structure is actually quite similar to the 70_10 with a fixed cluster overlap of 88% (22 out of 25 nodes matching). Secondly, the 90_10 and 100_10 networks in figure 6 have a much larger percentage of out-of-control samples than their counterparts in figure 5, which shows the effect of eliminating the consideration of clustering from the similarity metric. Though these networks exhibit similar clustering to the 70_10 network, not including clustering causes them to appear out-of-control more often than one would intuitively expect. Finally, for the 60_40 network, it is expected that the similarity should be fairly low, as shown in figure 6, given that the cluster overlap is typically in the 40 – 45% range as show in the figure 7 histogram. The fact that the similarity is higher in figure 5 is also expected, since it does not include the cluster overlap metric, which demonstrates why it’s important to include that component.

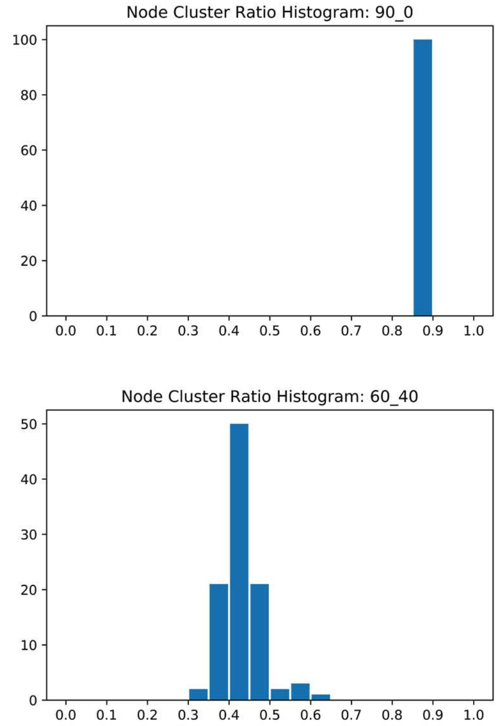

**Figure 7. Histograms of the node cluster ratio component for the 90-0 and 60-40 network structures**

## 4. Conclusion

Having a capability that can detect significant changes within a network without being overly reactive to smaller changes is a difficult balance to strike, especially with a single metric. Our graph similarity metric appears to have achieved this goal by aggregating several graph analysis techniques in a unique manner to produce a value that is easy to interpret and decompose if necessary. By using our metric in conjunction with a control chart we demonstrated that networks which differ in small ways will be recognized as being similar but that significant (though not hugely dramatic) changes will still be detected. This temporal anomaly detection could be useful in determining if organized crime networks have changed in some meaningful way. Though our case study focused on social network analysis, the metric could be equally valuable to areas such as chemical graph theory and semantic analysis where the clustering of similar nodes has as much or more meaning than the simple presence or absence of particular nodes and arcs.

One extension to this model would be to explore how the component weights could be optimally adjusted for specific applications. For example, if changes in the clustering structure of the network are more important than the presence or absence of weak links, the matching cluster ratio could have a higher relative weighting with respect to the other components.

## 5. Acknowledgements

Special thanks to Drew Levin of Sandia National Laboratories for acting as a sounding board when designing the experiments and with his help creating the network graphics. This work was supported by the Laboratory Directed Research and Development program at Sandia National Laboratories, a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525. SAND Number: SAND2019-xxxx C.

## 6. References

[1] R. Michalski, P. Brodka, P. Kazienko, and K. Juszczyszyn, "Quantifying Social Network Dynamics", In Proceedings of the 4th International Conference on Computational Aspects of Social Networks, Sao Carlos, Brazil, November 21-23, 2012, 69-74.

[2] E.A. Hobson, M.L. Avery, and T.F. Wright, "An Analytical Framework for Quantifying and Testing Patterns of Temporal Dynamics in Social Networks", Animal Behavior 85, 2013, 83-96.

[3] P. Papadimitriou, A. Dasdan, and H. Garcia-Molina, "Web Graph Similarity for Anomaly Detection", Journal of Internet Server Applications 1, 2010, 19-30.

[4] Ramdlani, G.A. Putri Saptawati, and Y. Asnar, "Graph Analysis on ATCS Data in Road Network for Congestion Detection", International Conference on Data and Software Engineering, Palembang, Indonesia, November 1-7, 2017.

[5] P. Drieger, "Semantic Network Analysis as a Method for Visual Text Analytics", Procedia – Social and Behavioral Sciences 79, 2013, 4-17.

[6] Mheich, M. Hassan, V. Gripon, M. Khalil, C. Berrou, O. Dufor, and F. Wendling, "A Novel Algorithm for Measuring Graph Similarity: Application to Brain Networks", 7th Annual International IEEE EMBS Conference on Neural Engineering, Montpellier, France, April 22-24, 2015, 1068-1071.

[7] M. Hernandez, A. Zaribafiyan, M. Aramon, and M. Naghibi, “Quantum Approaches to Graph Similarity”, 1QB Information Technologies, 2016.

[8] G. Chartrand, Introductory Graph Theory, Dover, New York, 1985.

[9] D.G. Corneil and C.C. Gotlieb, “An Efficient Algorithm for Graph Isomorphism”, Journal of the Association for Computing Machinery 17(1), 1970, 51-64.

[10] M. Pelillo, “Replicator Equations, Maximal Cliques, and Graph Isomorphism”, Neural Computation 11(8), 1999, 1933-1955.

[11] J.R. Ullman, “An Algorithm for Subgraph Isomorphism”, Journal of the Association for Computing Machinery 23(1), 1976, 31-42.

[12] I.L. Ruiz, M. U. Cuadrado, and M.A. Gomez-Nieto, “New Graph Similarity Measurements Based on Isomorphic and Nonisomophic Data Fusion and their Use in the Prediction of the Pharmacological Behavior of Drugs”, International Journal of Pharmacological and Pharmaceutical Sciences 1(2), 2007, 47-51.

[13] M.L. Fernandez and G. Valiente, “A Graph Distance Metric Combining Maximum Common Subgraph and Minimum Common Supergraph”, Pattern Recognition Letters 22(6-7), 2001, 753-758.

[14] H. Bunke, X. Jiang, and A. Kandel, “On the Minimum Common Supergraph of Two Graphs”, Computing 65(1), 2000, 13-25.

[15] H. Bunke, “Error Correcting Graph Matching: On the Influence of the Underlying Cost Function”, IEEE Transactions on Pattern Analysis and Machine Intelligence 21(9), 1999, 917-922.

[16] B.T. Messmer and H. Bunke, “A New Algorithm for Error-Tolerant Subgraph Isomorphism Detection”, IEEE Transactions on Pattern Analysis and Machine Intelligence 20(5), 1998, 493-504.

[17] X. Gao, B. Xiao, D. Tao, and X. Li, “A Survey of Graph Edit Distance”, Pattern Analysis and Applications 13, 2010, 113-129.

[18] B. Cao, Y. Li, and J. Yin, “Measuring Similarity Between Graphs Based on the Levenshtein Distance”, Applied Mathematics and Information Sciences 7(1L), 2013, 169-175.

[19] R. Albert and A.L. Barabasi, “Statistical Mechanics of Complex Networks”, Reviews of Modern Physics 74(1), 2002, 47-97.

[20] S. Dill, R. Kumar, K.S. McCurley, S. Rajagopalan, D. Sivakumar, and A. Tomkins, “Self-Similarity in the Web”, ACM Transactions on Internet Technology 2(3), 2002, 205-223.

[21] R. Michalski, T. Kajdanowicz, P. Brodka, and P. Kazienko, “Seed Selection for Spread of Influence in Social Networks: Temporal vs. Static Approach”, New Generation Computing 32, 2014, 213-235.

[22] E. Lagunas, A.G. Marques, S. Chatzinotas, and B. Ottersten, “Graph Similarity Based on Graph Fourier Distances”, 26th European Signal Processing Conference, Rome, Italy, September 2018.

[23] R.C. Wilson and P. Zhu, “A Study of Graph Spectra for Comparing Graphs and Trees”, Pattern Recognition 41, 2008, 2833-2841.

[24] D. Koutra, J.T. Vogelstein, and C. Faloutsos“DeltaCon: A Principled Mass-Graph Similarity Function”, SIAM International Conference on Data Mining, Austin, Texas, May 2013.

[25] G.S. Bopche, and B.M. Mehtre, “Graph Similarity Metrics for Assessing Temporal Changes in Attack Surface of Dynamic Networks”, Computers and Security 64(C), 2017, 16-43.

[26] V.D. Blondel, J.L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast Unfolding of Communities in Large Networks”, In Journal of Statistical Mechanics: Theory and Experiment (10), 2008, P10008.

[27] D.C. Montgomery, Introduction to Statistical Quality Control. 6th ed., John Wiley and Sons, Inc., Hoboken, New Jersey, 2009.