Compliance Graph Analysis Using Network Science and Structure Variations

Noah L. Schrick
Tandy School of Computer Science
The University of Tulsa
Tulsa, USA
noah-schrick@utulsa.edu

Peter J. Hawrylak
Tandy School of Computer Science
The University of Tulsa
Tulsa, USA
peter-hawrylak@utulsa.edu

***Abstract*—**

***Index Terms*—Attack Graph; Compliance Graph; Cybersecurity; Compliance and Regulation; Network Theory; Centrality;**

I. Introduction

A. Compliance Graphs

Compliance graphs are an alternate form of attack graphs, utilized specifically for examining compliance and regulation statuses of systems. Like attack graphs, compliance graphs can be used to determine all ways that systems may fall out of compliance or violate regulations, or highlight the ways in which violations are already present. These graphs are notably useful for cyber-physical systems due to the increased need for compliance. As the authors of [1], [2], and [3] discuss, cyber-physical systems have seen greater usage, especially in areas such as critical infrastructure and Internet of Things.

The semantics of compliance graphs are similar to that of attack graphs, but with a few differences regarding the information at each state. While security and compliance statuses are related, the information that is analyzed in compliance graphs is focused less on certain security properties, but is expanded to also examine administrative policies and properties of systems. Since compliance and regulation is broad and can vary by industry and application, the information to analyze can range from safety regulations, maintenance compliance, or any other regulatory compliance, including internal company standards. However, the graph structure of compliance graphs is identical to that of attack graphs, where edges represent a modification to the systems, and nodes represent all current information in the system.

Compliance graphs begin with a root node that contains all the current information of the system or set of systems. From this initial root state, all assets in the system are examined to see if any single modification can be made, where a modification can include a change in system policy or security settings. If a modification can be made, an edge is drawn from the previous state to a new state that includes all of the previous state's information, but now reflects the change in the system. This edge is labeled to reflect which change was made to the system. This process is exhaustively repeated, where all system properties are examined, all modification options are fully enumerated, all permutations are examined, and all changes to a system are encoded into their own independent states, where these states are then individually analyzed through the process.

B. Difficulties of Attack and Compliance Graph Analysis

Compliance graphs, like attack graphs, are directed acyclic graphs, and analysis of directed graphs is notably more involved compared to their undirected counterparts. The primary contributor to the increased difficulty is due to the asymmetric adjacency matrix present in directed graphs. With undirected graphs, simplifications can be made in the analysis process both computationally and conceptually. Since the "in" degrees are equal to the "out" degrees, less work is required both in terms of parsing the adjacency matrix, but also in terms of determining importance of nodes. The author of [4] discusses that common analysis techniques such as eigenvector centrality is often unapplicable to directed acyclic graphs. As the author of [5] discusses, the difficulty of directed graphs also extends to the graph Laplacian, where the definition for asymmetric adjacency matrices is not uniquely defined, and is based on either row or column sums computing to zero, but both cannot. The author of [5] continues to discuss that directed graphs lead to complex eigenvalues, and can lead to adjacency matrices that are unable to be diagonalized. These challenges require different approaches for typical clustering or centrality measures.

II. Related Works

The author of [6] presents three centrality measures that were applied to various attack graphs. The centrality measures implemented were Katz, K-path Edge, and Adapted PageRank.

TABLE I: Network Properties for the Three Networks Utilized

Network	Nodes	Edges	Connectivity (%)
Car	2491	12968	0.209
HIPAA	2321	8063	0.150
PCI DSS	61	163	4.381

Each of these centrality measures are applicable to the directed format of attack graphs, and conclusions were drawn regarding patching schemes for preventing exploits. As an approach for avoiding complex eigenvalues, the authors of [7] present work examining directed, undirected, and mixed graphs using its Hermitian adjacency matrix. Other works, such as that discussed by the author of [5], include mathematical manipulation of directed graph spectra (originally presented by the author of [8]) with Schur's Theorem to bound eigenvalues and allow for explicit computation, which can then be used for additional analysis metrics.

III. Experimental Networks

The work conducted in this approach utilized three compliance graphs, with their properties displayed in Table I. Connectivity in this table refers to the mean degree, divided by the number of nodes in the network, multiplied by 100 to return a percentage. Network 1 is a vehicle maintenance network. This network has one car asset that is deemed "brand new", and has zero mileage. This network is examined at its current state, and progresses through time with time steps of 1 month, up to 12 months total. At each time step the car gains mileage and increases its age property, and is reexamined to evaluate its standing in regards to its vehicular regulatory maintenance schedule. Network 2 is an artificial company network that is attempting to maintain HIPAA compliance [9]. This network examines its standing in relation to security properties that are required per HIPAA guidelines, as well as employee cooperation to training and administrative policies. This network is also progressed through time to illustrate the company's standing in relation to yearly audits and trainings that must be followed. Employees are also added and removed through the network at set points during the time progression process. Network 3 is another artificial company network. This company is attempting to maintain PCI DSS compliance [10]. This network generation was static and did not progress through time. This network examined the company and its current state, and examined a list of changes that could occur. These changes were primarily tied to security properties such as physical break-ins on the property, disabling firewalls, leaving default system settings, and encryption expiration.

IV. Centralities and their Contextualizations to Compliance Graphs

A. Introduction

The author of [11] provides a survey of centrality measures, and discusses how various centrality measures have been implemented and brought forth in order to determine node importance in networks. By determining the importance of nodes, various conclusions can be drawn regarding the network. In the case of compliance graphs, conclusions can be drawn regarding the prioritization of patching or correction schemes. If one node is known to lead to the creation of many other nodes, it may be said that a patch is imperative to prevent further opportunities for compliance violation. This work discusses five centrality measures, and discusses their application to compliance graphs.

B. Degree

Degree centrality is a trivial, localized measure of node importance based on the number of edges that a node has. In an undirected graph, the degree centrality is predicated solely on the number of edges. However, in the case of a directed graph, a distinction is drawn with a degree centrality oriented on the number of edges coming into a node, and another measure focused on the number of edges leaving a node. Both of these cases provide useful information for compliance graphs. When a node has a large number of other nodes it directs to, this node may be prioritized since it creates further opportunity for violation. When a node has a large number of edges pointing to it, this node may be prioritized since the probability that systems may enter this state is higher due to the increased number of possibilities that a system change could lead to this state.

C. Betweenness

Betweenness centrality ranks node importance based on its ability to transfer information in a network. For all pairs of nodes in a network, a shortest path is determined. A node that is in this shortest path is considered to have importance. The total betweenness centrality is based on the number of shortest paths that pass through a given node. For compliance graphs, the shortest paths are useful to identify the quickest way (least number of steps) that systems may fall out of compliance. By prioritizing the nodes that fall in the highest number of shortest paths, correction schemes can be employed to prolong or prevent systems from falling out of compliance.

Betweenness centrality is given in Equation 1, where *i* and *j* are two different, individual nodes in the network, σ_{ij} is the total number of shortest paths from *i* to *j*, and $\sigma_{ij}(v)$ is the number of shortest paths that include a node *v*.

$$\sum_{i \neq i \neq v} \frac{\sigma_{ij}(v)}{\sigma_{ij}} \tag{1}$$

D. Katz

Katz centrality was first introduced by the author of [12], and measures the importance of nodes through all paths in a network. Katz centrality varies in that its centrality measure is not limited to solely the shortest path between any two given nodes. The original work by the author defines Katz as seen in Equation 2, where *i* and *j* are nodes in the network, *n* is the total number of nodes in the network, *A* is the adjacency matrix, and α is an attenuation factor and has a value between 0 and 1. A value of 1 is assigned if node *i* is connected to node *j*.

$$C_{\text{Katz}}(i) = \sum_{k=1}^{\infty} \sum_{j=1}^{n} \alpha^k (A^k)_{ji} \tag{2}$$

Later works have expanded on the original Katz to include a β vector that allows for additional scaling in the instance that prior knowledge of the network exists. The modified equation can be seen in Equation 3.

$$\vec{x} = (I - \alpha A)^{-1} \vec{\beta} \tag{3}$$

For compliance graphs, Katz centrality represents the total number of paths that exist from a given node to any other downstream nodes, and is scaled based on the attenuation factor as well as the prior knowledge vector β. When the Katz centrality of a given node is high, prioritizing a correction scheme for the node would be useful to prevent opportunity of future compliance violations that may be many steps ahead, but still reachable from the current state.

E. K-Path Edge

K-path edge centrality, as discussed by the authors of [13], is predicated on information passing through a network as a means of generalizing k-path centrality. With K-path edge centrality, importance is based on the edges of the network. One difference from betweenness centrality, is that as discussed in Section IV-C, betweenness centrality is global and counts all nodes in a shortest path. K-path edge centrality is localized, and is constrained by *k* steps from a given node. Equation 4 displays the centrality measure for K-path edge centrality, where *m* is a given edge in the network, *N* is the total number of nodes in the network, $\delta_n^{(K)}$ is the number of K-paths from node *n*, and $\delta_n^{(K)}(m)$ is the number of K-paths from node *n* that include edge *m*.

$$L^{(K)}(m) = \sum_{n=1}^{N} \frac{\delta_n^{(K)}(m)}{\delta_n^{(K)}} \tag{4}$$

For compliance graphs, K-path edge centrality is useful to identify a short chain of changes that may result in a compliance violation. If a node has a high K-path edge centrality and it is likely that the system will be put into that node, then a series of changes could occur that could then put the system in a different states. Prioritizing nodes that have a high K-path edge centrality could be useful in deterring a short chain of changes that could cripple the system further. It is also useful to prevent states where the system is near a compliance violation.

F. Adapted Page Rank

The original PageRank algorithm was first designed by the authors of [14] for the Google prototype for ranking web pages. The authors of [15] later introduced an Adapated PageRank algorithm that was designed to measure both the number and quality of connections specifically for an urban network. Equation 5 displays the PageRank algorithm, where γ is a damping factor with a value between 0 and 1, *n* is the total number of nodes in the network, *A* is the adjacency matrix of the network, *i* and *j* represent the row and column of the adjacency matrix, *x* is a given node in the network, and *k* is the row sum out degree. Since the Adapted PageRank algorithm measures the quality of connections, there is increased application to directed networks such as compliance graphs. As seen in Equation 5, the k_j term is a penalizing factor. Importance is based on the in degree of a node, with a penalty for the out degree. If many nodes point to a given node, then that node is considered important due to its accessibility.

$$x_i = \frac{1-\gamma}{n} + \gamma \sum_{j=1}^{n} \frac{A_{ij}}{k_j} x_j \tag{5}$$

The adapted PageRank algorithm includes additional data that may be present in an urban network, such as geographical position, resource availability, and proximity to facilities. This data is user-defined, and may not be present in the network. Equation 6 displays the Adapated PageRank algorithm in matrix form where *D* is the user-defined data matrix, *I* is the identity matrix, and $\mathbb{1}$ is a column matrix comprised of 1s.

$$(I - \gamma AD)\vec{x} = \frac{1-\gamma}{n}\mathbb{1} \tag{6}$$

For compliance graphs, the Adapted Page Rank algorithm is useful for a few reasons. First, it is able to include user-

defined data regarding the network. This could include scaling certain nodes to have greater weight, such as those known to be a compromised state. Second, since nodes are penalized for pointing to other nodes, this algorithm is useful for determining nodes that are likely to be visited. If a state has a greater in-degree, it may require greater prioritization since the system has a higher likelihood of falling into this state.

V. Transitive Closure

A. Introduction and Application

Transitive closure represents a transitive relation on a given binary set, and can be used to determine reachability of a given network. Figure 1 displays an example output when performing transitive closure. In context of compliance graphs, it is useful to consider that an adversary (whether an internal or external malicious actor, poor policy execution by an organization, accidental misuse, or any other adversarial occurrence) could have no time constraints. That is, for any given state of the system or set of systems, an adversarial act could have "infinite" time to perform a series of actions. If no prior knowledge is known about the network, it can be assumed that all changes performed on the systems are equally likely. In practice, specifying a probability that a change can occur has been performed through a Markov Decision Process, such as that seen by the authors of [16] and [17]. When under these assumptions, it is useful to then consider which nodes are important, assuming they have 1-step reachability to any downstream node they may have a transitive connection to. This work identified a transitive closure for all networks described in Section III, and this transitive closure was then analyzed through the five centrality methods discussed in Section IV. Results and a discussion of the results can be seen in Section VII.

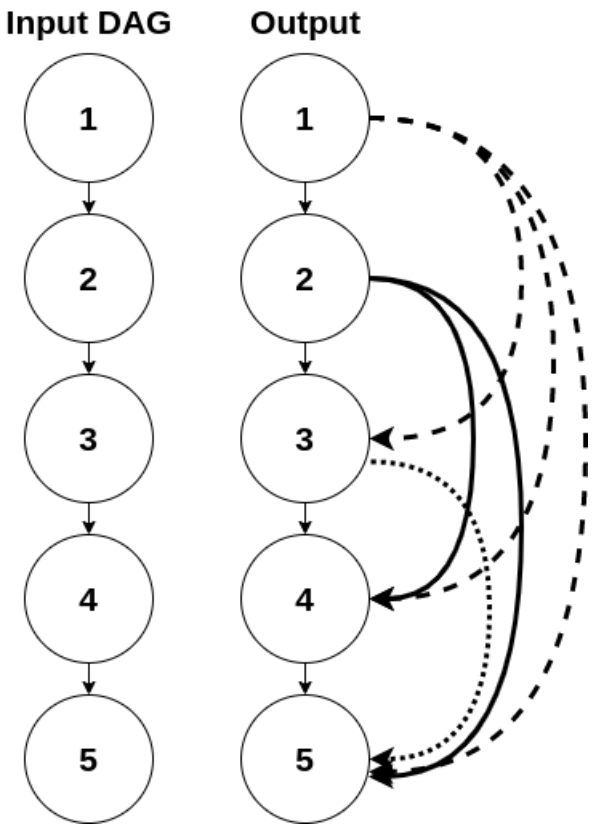

Fig. 1: Example of Transitive Closure

VI. Dominant Tree

A. Introduction and Application

Dominance, as initially introduced by the author of [18] in terms of flow, is defined as a node that is in every path to another node. If a node i is a destination node, and every path to i from a source node includes node j, then node j is said to dominate node i. Figure 2 displays an example starting network. With node 1 as the source node, it is evident that node 2 immediately dominates nodes 3, 4, 5, and 6, since all messages from node 1 must pass through node 2. By definition, each node must also dominate itself, so node 2 also dominates node 2.

Following the properties of dominance, a dominator tree can be derived. In a dominator tree, each node has children that it immediately dominates. Immediate dominance is referred to nodes that strictly dominate a given node, but do not strictly dominate any other node that may strictly dominate a node. Figure 3 displays the dominant tree of the network seen in Figure 2.

Dominant trees do alter the structure of compliance graphs, and lead to leaf nodes and branches that do not exist in the original network. As a result, some nodes that have directed edges to other nodes may be moved to a position where the edge no longer points to the original nodes. However, in dominant trees, all node parents dominate their children. In this format, the information flow is guided predominantly by the upstream nodes, and all parents in the dominant tree exist as upstream nodes in the original compliance graph. While some downstream nodes may be altered, the importance of nodes can be reexamined in the dominant tree to see how importance differs when information flow is refined. To this end, dominant trees were identified for all networks described in Section III, and these dominant trees were then analyzed through the five centrality methods discussed in Section IV. Results and a discussion of the results can be seen in Section VII.

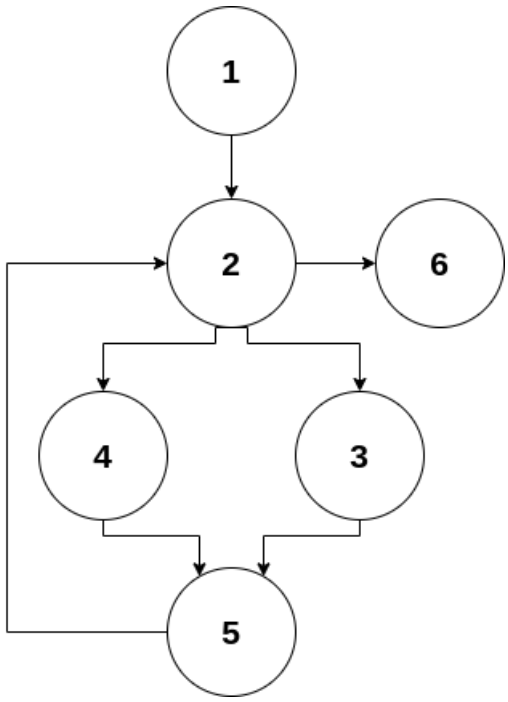

Fig. 2: Example Network for Illustrating Dominance

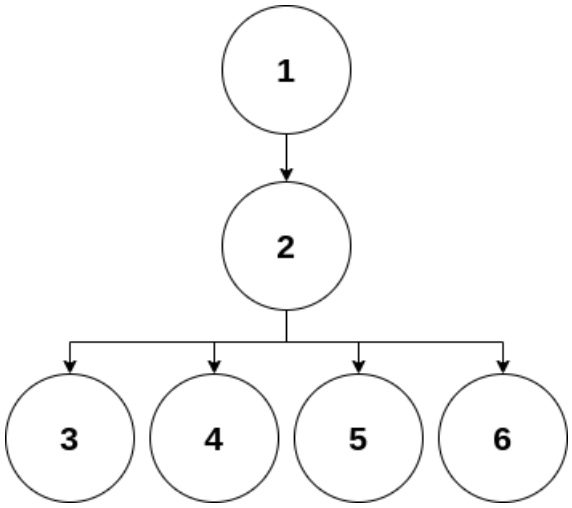

Fig. 3: Dominant Tree Derived from the Network Displayed in Figure 2

TABLE II: Top 15 Nodes with Degree Centrality

Base		Transitive Closure		Dominant Tree	
Node	**% Value**	**Node**	**% Value**	**Node**	**% Value**
314	0.08	0	0.65	1	50.04
346	0.08	1	0.65	3	37.51
362	0.08	3	0.65	7	6.27
370	0.08	7	0.65	42	4.62
374	0.08	15	0.64	314	1.24
376	0.08	27	0.64	0	0.04
377	0.08	42	0.64	15	0.04
378	0.08	60	0.62	27	0.04
379	0.08	87	0.60	60	0.04
380	0.08	130	0.59	87	0.04
381	0.08	187	0.57	130	0.04
382	0.08	250	0.55	187	0.04
398	0.07	314	0.54	250	0.04
406	0.07	2	0.33	2	0.00
410	0.07	4	0.33	4	0.00

TABLE III: Top 15 Nodes with Katz Centrality

Base		Transitive Closure		Dominant Tree	
Node	**% Value**	**Node**	**% Value**	**Node**	**% Value**
314	0.13	0	9.09	1	4.88
377	0.10	1	8.26	3	3.47
346	0.10	3	7.39	7	0.60
376	0.10	7	6.34	0	0.52
374	0.10	15	5.27	42	0.45
378	0.10	27	4.37	314	0.15
380	0.10	42	3.61	27	0.08
381	0.10	60	2.71	250	0.05
382	0.10	87	2.02	15	0.04
262	0.10	130	1.48	187	0.04
370	0.10	2	1.31	130	0.04
379	0.10	4	1.19	87	0.04
418	0.08	5	1.19	60	0.04
459	0.08	6	1.19	2	0.04
467	0.08	187	1.08	4	0.04

VII. Results and Result Analysis

A. Results

In this section, only results for the car network are displayed for brevity. These results can be seen in Tables II through VI. The results for the HIPAA and PCI DSS networks can be found in the Supplementary Material included with this work.

B. Result Analysis

When viewing the results of the car networks, unsurprisingly, each centrality method ranks nodes in a different order. These differences in rankings can be used

TABLE IV: Top 15 Nodes with K-path Edge Centrality

Base		Transitive Closure		Dominant Tree	
Node	**% Value**	**Node**	**% Value**	**Node**	**% Value**
314	0.25	0	0.65	1	37.54
346	0.19	1	0.39	0	35.05
362	0.19	3	0.65	3	17.53
370	0.19	7	0.65	7	2.54
374	0.19	15	0.64	15	1.88
376	0.19	27	0.64	27	1.88
377	0.19	42	0.64	42	1.88
378	0.19	60	0.62	187	0.53
379	0.19	87	0.60	250	0.51
380	0.19	130	0.59	314	0.50
381	0.19	187	0.57	60	0.05
382	0.19	250	0.55	86	0.05
398	0.14	314	0.54	130	0.05
406	0.14	2	0.33	2	0.00
410	0.14	4	0.33	4	0.00

TABLE V: Top 15 Nodes with PageRank Centrality

Base		Transitive Closure		Dominant Tree	
Node	**% Value**	**Node**	**% Value**	**Node**	**% Value**
2490	8.27	2490	19.92	314	0.17
1004	1.06	2479	1.58	250	0.15
1467	0.97	2480	1.58	187	0.13
2479	0.95	2481	1.58	130	0.10
2480	0.95	2482	1.58	42	0.10
2481	0.95	2483	1.58	87	0.07
2482	0.95	2484	1.4	27	0.07
2483	0.95	2485	1.4	1	0.07
667	0.92	2486	1.39	378	0.04
2484	0.83	2487	1.39	379	0.04
2485	0.83	2488	1.39	380	0.04
2486	0.83	2489	1.39	381	0.04
2487	0.83	2424	0.29	382	0.04
2488	0.83	2425	0.29	470	0.04
2489	0.83	2426	0.29	471	0.04

based on additional metrics, such as severity, cost, or disturbance of systems, to identify correction schemes best suited for a given environment. However, degree centrality and K-path edge centrality rankings for the top 15 were identical for the car network. This also extends to the HIPAA network, as seen in the included results in the Supplementary Material, but does not extend to the PCI DSS network. The value for k in K-path edge centrality was set to 3. With a relatively small k value in comparison to the overall size of the car and HIPAA networks, coupled with the high degree count of the top 15 nodes ranked with degree centrality, it is likely that the high degree count correlates to the K-path edge centrality scoring. This reasoning extends to the PCI DSS network, where the network is substantially smaller and there is a greater percentage of connectivity.

TABLE VI: Top 15 Nodes with Betweenness Centrality

Base		Transitive Closure		Dominant Tree	
Node	**% Value**	**Node**	**% Value**	**Node**	**% Value**
42	0.43	0	0	1	24.71
27	0.40	1	0	3	24.68
60	0.39	2	0	7	9.20
87	0.36	3	0	42	9.00
15	0.36	4	0	27	7.55
130	0.33	5	0	15	6.08
7	0.31	6	0	314	3.69
187	0.29	7	0	250	3.49
50	0.28	8	0	187	3.28
70	0.28	9	0	130	3.04
104	0.27	10	0	87	2.78
156	0.26	11	0	60	2.50
1467	0.25	12	0	0	0.00
250	0.25	13	0	2	0.00
115	0.25	14	0	4	0.00

Comparing the transitive closure format of compliance graphs, the associated centrality rankings greatly vary from their original compliance graph rankings. As expected however, the root or a leaf node has the highest centrality value. Since the root node can reach all nodes, and a leaf node can be reached by all nodes, these two nodes are expectedly ranked high. What is unexpected, however, is that the top 15 rankings are not comprised of the most upstream 15 nodes or the 15 most downstream nodes. While rankings do tend to be higher for more upstream for K-path edge, Katz, and degree centralities, node IDs in the 100s, 200s, and 300s ("midstream" nodes) all make appearances. Betweenness centrality for the transitive closure representation yielded no valuable insight, since shortest paths to a node from any given node is reachable in 1 step.

For the dominant tree representation, it was initially hypothesized that nodes ranked highly in the original compliance graph's betweenness centrality or Katz centrality measures would closely relate to the dominant tree results. However, the dominant tree rankings also vary greatly from the original compliance graph's rankings. Even nodes that saw no appearances in the top 15 of the base compliance graph or transitive closure representation made appearances in the dominant tree results. Since the dominant tree format does favor the upstream nodes due to a lesser reordering effect caused by dominance, the PageRank ordering were not predominantly downstream nodes, but mostly node IDs in the 300s.

VIII. Conclusions and Future Work

A. Conclusions

Each centrality measure implemented in this work provides various information that is useful for identifying correction schemes based on a network science approach. The results from the centrality methods differ, and each network can determine which rankings should be preferred based on prior knowledge of the network and the overhead of implementing correction measures. In addition, transitive closure representations and dominant trees were derived from the original compliance graphs, and unique rankings were identified. Transitive closure rankings are useful for determining which nodes are most important when an adversarial action can be considered to have infinite time and resources to perform changes to the original system. Dominant tree rankings are useful for determining which nodes are most important from an information flow perspective, where adversarial actions must pass though a series of nodes to reach any other node in the network. By applying correction schemes to the bottlenecks of the network, it may be possible to eliminate branches of the dominant tree entirely, leading to a removal of nodes in the original compliance graph.

B. Future Work

Based on the results of this work, there is ample room to continue investigation of centrality methods for compliance graphs. With three compliance graphs generated for three different networks along with various node importance rankings, it would be useful to artificially implement correction schemes based on the rankings to see their effects on the compliance graph. Likewise, using a user-defined data matrix in centrality methods like PageRank, further research could examine how node importance varies based on user-defined metrics. Edge weights could also be assigned to the original compliance graphs to represent the probability that a given change in the network could occur. Edge weights would be reflected in the adjacency matrices of the graphs, and centrality methods could be reexamined to determine node importance when state transition probabilities are given. Transitive closures and dominant trees derived from the compliance graphs present a new approach for examining compliance graphs. Further research can be conducted to determine the effects of correction schemes when employed on nodes ranked highly in their respective centrality measures in these formats.

References

[1] J. Hale, P. Hawrylak, and M. Papa, "Compliance Method for a Cyber-Physical System." U.S. Patent Number 9,471,789, Oct. 18, 2016.

[2] N. Baloyi and P. Kotzé, "Guidelines for Data Privacy Compliance: A Focus on Cyberphysical Systems and Internet of Things," in *SAICSIT '19: Proceedings of the South African Institute of Computer Scientists and Information Technologists 2019*, (Skukuza South Africa), Association for Computing Machinery, 2019.

[3] E. Allman, "Complying with Compliance: Blowing it off is not an option.," *ACM Queue*, vol. 4, no. 7, 2006.

[4] M. Newman, *Networks: An Introduction*. Oxford University Press, 2010.

[5] P. V. Mieghem, "Directed graphs and mysterious complex eigenvalues," 2018.

[6] M. Li, P. Hawrylak, and J. Hale, "Strategies for practical hybrid attack graph generation and analysis," *Digital Threats*, oct 2021. Just Accepted.

[7] K. Guo and B. Mohar, "Hermitian adjacency matrix of digraphs and mixed graphs," *Journal of Graph Theory*, vol. 85, 2017.

[8] R. A. Brualdi, "Spectra of digraphs," *Linear Algebra and its Applications*, vol. 432, pp. 2181–2213, 2010.

[9] "Health Insurance Portability and Accountability Act of 1996." Pub. L. No. 104-191. 1996 [Online]. Available: https://www.govinfo.gov/content/pkg/PLAW-104publ191/html/PLAW-104publ191.htm.

[10] P. S. S. Council, "Payment Card Industry (PCI) Data Security Standard," May 2018. Available: https://www.pcisecuritystandards.org/documents/PCI_DSS_v3-2-1.pdf.

[11] M. Ashtiani, A. Salehzadeh-Yazdi, Z. Razaghi-Moghadam, H. Hennig, O. Wolkenhauer, M. Mirzaie, and M. Jafari, "A systematic survey of centrality measures for protein-protein interaction networks," *BMC systems biology*, vol. 12, p. 80, July 2018.

[12] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, pp. 39–43, March 1953.

[13] P. D. Meo, E. Ferrara, G. Fiumara, and A. Ricciardello, "A novel measure of edge centrality in social networks," *Knowledge-Based Systems*, vol. 30, pp. 136–150, jun 2012.

[14] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1, pp. 107–117, 1998. Proceedings of the Seventh International World Wide Web Conference.

[15] T. Agryzkov, J. L. Oliver, L. Tortosa, and J.-F. Vicent, "An algorithm for ranking the nodes of an urban network based on the concept of pagerank vector," *Appl. Math. Comput.*, vol. 219, pp. 2186–2193, 2012.

[16] M. Li, P. Hawrylak, and J. Hale, "Combining OpenCL and MPI to support heterogeneous computing on a cluster," *ACM International Conference Proceeding Series*, 2019.

[17] K. Zeng, "Cyber Attack Analysis Based on Markov Process Model," 2017.

[18] R. T. Prosser, "Applications of boolean matrices to the analysis of flow diagrams," in *Papers Presented at the December 1-3, 1959, Eastern Joint IRE-AIEE-ACM Computer Conference*, IRE-AIEE-ACM '59 (Eastern), (New York, NY, USA), p. 133–138, Association for Computing Machinery, 1959.