Compliance Graph Analysis Using Network Theory Principles

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. Attack Graphs

To address the rising risks of computing and threats to cybersecurity, vulnerability analysis modeling is a technique employed by experts to identify weak points in a system or set of systems. One such modeling approach is to represent the system or set of systems through graphical means, with system information encoded into the nodes and edges of the graph. This modeling approach was first utilized in the 1990s in a format called attack trees, and can be seen through the works of the authors of [1] and [2]. These attack trees would later be expanded into attack graphs.

Attack 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 is typically 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 attack 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. 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 also be used to determine all ways that systems may fall out of compliance or violate regulations. These graphs are notably useful for cyber-physical systems due to the increased need for compliance. As the authors of [3], [4], and [5] 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, and 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. 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.

C. Difficulties of Attack and Compliance Graph Analysis

Attack and compliance graphs are directed acyclic graphs, and analysis of directed graphs is not as simple as 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 [6] discusses that common analysis techniques such as eigenvector centrality is often unapplicable to directed acyclic graphs. As the author of [7] 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 [7] 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 [**?**] presents three centrality measures that were applied to various attack graphs. The centrality measures implemented were Katz, K-path Edge, and Adapted PageRank. Each of these centrality measures are applicable to the directed format of attack graphs, and conclusions can be drawn regarding patching schemes for preventing exploits. As an approach for avoiding complex eigenvalues, the authors of [9] present work examining directed, undirected, and mixed graphs using its Hermitian adjacency matrix. Other works, such as that discussed by the author of [7] include mathematical manipulation of directed graph spectra (originally presented by the author of [10]) 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 get the number in a percentage form. 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 [11]. 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 [12]. This network generation was static and did not progress through time. This network examined the company and its current state, and examined all changes that could occur. These changes were primarily tied to security properties such as physical break-ins on the property, firewalls being disabled, default system settings, and encryption expiration.

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

IV. Centralities and their Applications to Compliance Graphs

A. Introduction

The author of [13] 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 points 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 flow 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 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 [14], 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. From this, 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 [15], 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 the 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 [16] for the Google prototype for ranking web pages. The authors of [17] later introduced an Adapated PageRank 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 said to be 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 [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 [18] and [19]. 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. As a result, a transitive closure was identified 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.

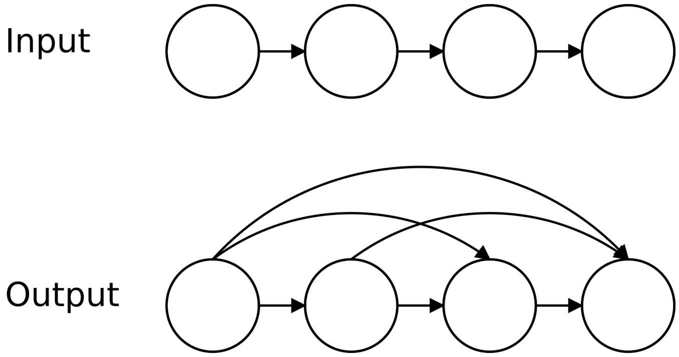

Fig. 1: Example of Transitive Closure

VI. Dominant Tree

A. Introduction and Application

Dominance, as initially introduced by the author of [20] in terms of flow, is defined as a node that is in every path to another node. For instance, 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 being 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 leads 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.

[1] Image origin can be located at: https://commons.wikimedia.org/wiki/File:Transitive-closure.svg, and this image has been licensed under the terms of the GNU Free Documentation License.

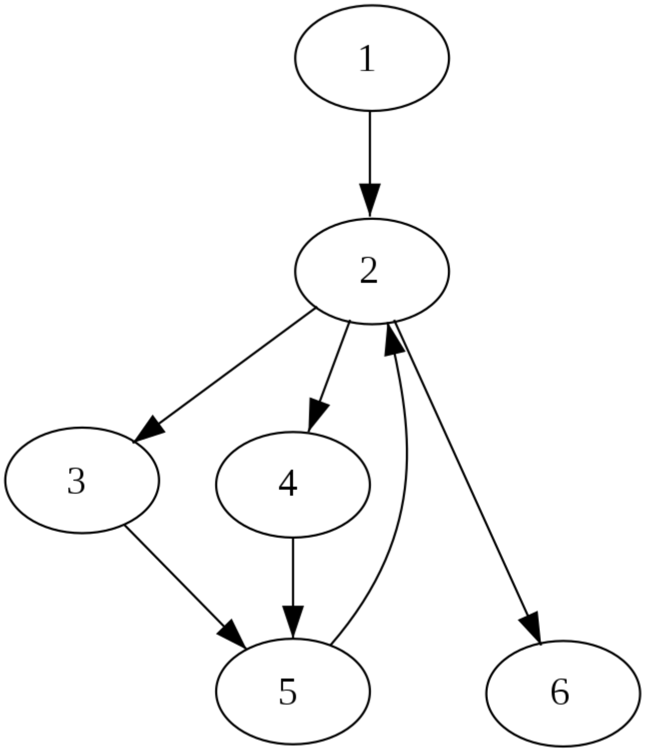

Fig. 2: Example Network for Illustrating Dominance [2]

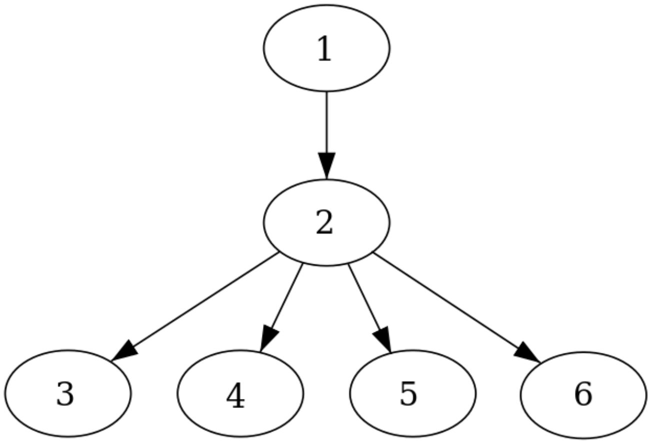

Fig. 3: Dominant Tree Derived from the Network Displayed in Figure 2 [3]

TABLE II: Top 15 Nodes with Degree Centrality

Base		**Transitive Closure**		**Dominant Tree**	
Node	**Value**	**Node**	**Value**	**Node**	**Value**
314	11	0	2490	1	1246
346	10	1	2489	3	934
362	10	3	2487	7	156
370	10	7	2479	42	115
374	10	15	2463	314	31
376	10	27	2447	0	1
377	10	42	2431	15	1
378	10	60	2367	27	1
379	10	87	2303	60	1
380	10	130	2239	87	1
381	10	187	2175	130	1
382	10	250	2111	187	1
398	9	314	2047	250	1
406	9	2	1244	2	0
410	9	4	1243	4	0

TABLE III: Top 15 Nodes with Katz Centrality

Base		**Transitive Closure**		**Dominant Tree**	
Node	**Value**	**Node**	**Value**	**Node**	**Value**
314	0.002459349	0	74.447935	1	0.0542337315
377	0.001870821	1	67.679941	3	0.0385235854
346	0.001870821	3	60.55317	7	0.0066730273
376	0.001870821	7	51.894146	0	0.0058248184
374	0.001870821	15	43.13118	42	0.0050225267
378	0.001870821	27	35.752083	314	0.0016459253
380	0.001870821	42	29.550411	27	0.0009036979
381	0.001870821	60	22.205831	250	0.0005660377
382	0.001870821	87	16.522142	15	0.000491815
262	0.001870821	130	12.155237	187	0.000458049
370	0.001870821	2	10.714534	130	0.0004472501
379	0.001870821	4	9.740485	87	0.0004461702
418	0.001469376	5	9.740485	60	0.0004460622
459	0.001469376	6	9.740485	2	0.0004014452
467	0.001469376	187	8.82693	4	0.0004014452

Results and a discussion of the results can be seen in Section VII.

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. For the HIPAA and PCI DSS networks, results can be seen in Appendices A and A, respectively.

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

Base		**Transitive Closure**		**Dominant Tree**	
Node	**Value**	**Node**	**Value**	**Node**	**Value**
314	231	0	2490	1	2336
346	175	1	1489	0	2181
362	175	3	2487	3	1091
370	175	7	2479	7	158
374	175	15	2463	15	117
376	175	27	2447	27	117
377	175	42	2431	42	117
378	175	60	2367	187	33
379	175	87	2303	250	32
380	175	130	2239	314	31
381	175	187	2175	60	3
382	175	250	2111	86	3
398	129	314	2047	130	3
406	129	2	1244	2	0
410	129	4	1243	4	0

TABLE V: Top 15 Nodes with PageRank Centrality

Base		**Transitive Closure**		**Dominant Tree**	
Node	**Value**	**Node**	**Value**	**Node**	**Value**
2490	0.0827	2490	0.1992	314	0.001655
1004	0.01506	2479	0.0158	250	0.001479
1467	0.00969	2480	0.0158	187	0.001272
2479	0.00948	2481	0.0158	130	0.001028
2480	0.00948	2482	0.0158	42	0.001025
2481	0.00948	2483	0.0158	87	0.00074
2482	0.00948	2484	0.014	27	0.00074
2483	0.00948	2485	0.014	1	0.00074
667	0.00919	2486	0.0139	378	0.00044
2484	0.0083	2487	0.0139	379	0.00044
2485	0.0083	2488	0.0139	380	0.00044
2486	0.0083	2489	0.0139	381	0.00044
2487	0.0083	2424	0.0029	382	0.00044
2488	0.0083	2425	0.0029	470	0.00044
2489	0.0083	2426	0.0029	471	0.00044

TABLE VI: Top 15 Nodes with Betweenness Centrality

Base		**Transitive Closure**		**Dominant Tree**	
Node	**Value**	**Node**	**Value**	**Node**	**Value**
42	9067.205	0	0	1	2489
27	8442.166	1	0	3	2486
60	8279.62	2	0	7	927
87	7580.359	3	0	42	906
15	7578.523	4	0	27	760
130	6868.21	5	0	15	612
7	6482.031	6	0	314	372
187	6111.862	7	0	250	352
50	5950.928	8	0	187	330
70	5822.054	9	0	130	306
104	5683.944	10	0	87	280
156	5474.525	11	0	60	252
1467	5299.985	12	0	0	0
250	5296.964	13	0	2	0
115	5196.398	14	0	4	0

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 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 Appendix A, 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 connectivity percent.

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 leaf node has the highest centrality value. Since the root node can reach all nodes, and the 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, nodes 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 nodes 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 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.

References

[1] C. Phillips and L. P. Swiler, "A graph-based system for network-vulnerability analysis," *Proceedings New Security Paradigms Workshop*, vol. Part F1292, pp. 71–79, 1998. doi: 10.1145/310889.310919.

[2] B. Schneier, "Modeling Security Threats," *Dr. Dobb's Journal*, 1999. vol. 24, no.12.

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

[4] 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.

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

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

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

[8] M. Li, *A System for Attack Graph Generation and Analysis*. PhD thesis, The University of Tulsa, 2021.

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

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

[11] "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.

[12] 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.

[13] 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.

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

[15] 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.

[16] 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.

[17] 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.

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

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

[20] 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.

APPENDIX

TABLE VII: Top 15 Nodes with Degree Centrality

Base		Transitive Closure		Dominant Tree	
Node	**Value**	**Node**	**Value**	**Node**	**Value**
1	4	0	2320	0	2203
4	4	1	2199	1	83
5	4	2	2109	2	1
6	4	6	1999	3	1
11	4	4	1980	4	1
12	4	5	1980	5	1
13	4	11	1980	8	1
15	4	15	1980	10	1
16	4	24	1980	11	1
18	4	33	1980	15	1
24	4	45	1980	20	1
25	4	61	1980	23	1
26	4	76	1980	24	1
28	4	101	1980	33	1
29	4	119	1980	40	1

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

Base		Transitive Closure		Dominant Tree	
Node	**Value**	**Node**	**Value**	**Node**	**Value**
1	29	0	2320	0	2292
4	29	1	2199	1	87
5	29	2	2109	2	3
6	29	6	1999	3	3
11	29	4	1979	4	3
12	29	5	1979	5	3
13	29	11	1979	8	3
15	29	15	1979	10	3
16	29	24	1979	11	3
18	29	33	1979	15	3
24	29	45	1979	20	3
25	29	61	1979	23	3
26	29	76	1979	24	3
28	29	101	1979	33	3
29	29	119	1979	40	3

TABLE VIII: Top 15 Nodes with Katz Centrality

Base		Transitive Closure		Dominant Tr	
Node	**Value**	**Node**	**Value**	**Node**	**Value**
250	0.01723395	0	0.0701156	0	0.0957149696
1313	0.01723395	1	0.06858208	1	0.004016468
2194	0.01723395	2	0.06240477	2	0.0004787209
396	0.01723395	11	0.06110052	3	0.0004787209
2300	0.01723395	24	0.06110052	4	0.0004787209
913	0.01723395	225	0.06110052	5	0.0004787209
983	0.01723395	101	0.06110052	8	0.0004787209
1048	0.01723395	45	0.06110052	10	0.0004787209
2268	0.01723395	76	0.06110052	11	0.0004787208
223	0.01723395	119	0.06110052	15	0.0004787209
385	0.01723395	15	0.06110052	20	0.0004787209
895	0.01723395	249	0.06110052	23	0.0004787209
1390	0.01723395	313	0.06110052	24	0.0004787208
1479	0.01723395	61	0.06110052	33	0.0004787208
1621	0.01723395	340	0.06110052	40	0.0004787208

TABLE X: Top 15 Nodes with PageRank Centrality

Base		Transitive Closure		Dominant Tree	
Node	**Value**	**Node**	**Value**	**Node**	**Value**
98	0.001622572	98	0.001149403	440	0.002213066
150	0.001517908	141	0.001149403	450	0.002213066
141	0.001473917	150	0.001149403	340	0.002118449
152	0.001458482	152	0.001149403	421	0.002118449
218	0.001430206	198	0.001149403	330	0.002117339
207	0.001394885	207	0.001149403	339	0.002117339
220	0.001378205	209	0.001149403	249	0.002006025
304	0.001370904	218	0.001149403	313	0.002006025
193	0.001363898	220	0.001149403	240	0.002004719
209	0.001340868	222	0.001149403	248	0.002004719
222	0.001338749	271	0.001149403	176	0.001873761
410	0.001334312	280	0.001149403	225	0.001873761
291	0.001321592	282	0.001149403	168	0.001872224
537	0.001312566	291	0.001149403	175	0.001872224
306	0.001304744	293	0.001149403	119	0.001718156

TABLE XI: Top 15 Nodes with Betweenness Centrality

Base		**Transitive Closure**		**Dominant Tree**	
Node	**Value**	**Node**	**Value**	**Node**	**Value**
57	30159.36	0	0	1	99
98	29440.04	1	0	45	25
150	28358.55	2	0	61	25
88	28108.58	3	0	70	25
95	28093.87	4	0	75	25
141	27812.05	5	0	24	24
152	27774.25	6	0	33	24
97	27495.11	7	0	40	24
218	27411.69	8	0	44	24
207	26972.75	9	0	76	24
220	26927	10	0	101	24
131	26605.99	11	0	112	24
304	26604.49	12	0	118	24
147	26591	13	0	11	21
138	26507.64	14	0	15	21

TABLE XII: Top 15 Nodes with Degree Centrality

Base		Transitive Closure		Dominant Tree	
Node	**Value**	**Node**	**Value**	**Node**	**Value**
1	4	0	60	0	43
4	4	1	47	1	8
7	4	4	36	2	1
11	4	11	36	3	1
14	4	22	36	4	1
22	4	2	31	7	1
25	4	3	31	8	1
38	4	7	31	10	1
0	3	8	24	11	1
2	3	10	24	19	1
3	3	14	24	21	1
5	3	19	24	5	0
6	3	21	24	6	0
8	3	25	24	9	0
10	3	33	24	12	0

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

Base		Transitive Closure		Dominant Tree	
Node	**Value**	**Node**	**Value**	**Node**	**Value**
1	23	0	60	0	57
4	22	1	47	1	11
11	22	4	35	2	3
22	22	11	35	3	3
0	21	22	35	4	2
7	19	2	31	8	2
14	18	3	31	10	2
25	18	7	31	7	1
38	18	5	23	11	1
2	15	6	23	19	1
3	15	8	23	21	1
5	14	10	23	5	0
6	14	14	23	6	0
8	14	19	23	9	0
10	14	21	23	12	0

TABLE XIII: Top 15 Nodes with Katz Centrality

Base		Transitive Closure		Dominant Tre	
Node	**Value**	**Node**	**Value**	**Node**	**Value**
0	21672.0619	0	38118.0529	0	0.08859508
1	14435.569	1	17874.4534	1	0.02985246
4	14435.569	4	17278.6824	2	0.01821311
11	14435.569	11	17278.6824	3	0.01821311
22	14435.569	22	17278.6824	4	0.01819672
3	4822.2923	2	6274.702	8	0.01819672
10	4822.2923	3	6274.702	10	0.01819672
21	4822.2923	8	6065.5717	7	0.01803279
35	4822.2923	19	6065.5717	11	0.01803279
2	4822.2923	33	6065.5717	19	0.01803279
8	4822.2923	10	6065.5717	21	0.01803279
19	4822.2923	21	6065.5717	5	0.01639344
33	4822.2923	35	6065.5717	6	0.01639344
38	801.1816	7	917.4427	9	0.01639344
25	801.1816	38	886.885	12	0.01639344

TABLE XV: Top 15 Nodes with PageRank Centrality

Base		Transitive Closure		Dominant Tree	
Node	**Value**	**Node**	**Value**	**Node**	**Value**
55	0.18443558	55	0.226570132	33	0.04233607
59	0.18084037	59	0.226570132	35	0.04233607
60	0.17618635	60	0.226570132	22	0.03508347
43	0.03052713	43	0.020661622	19	0.0342364
52	0.02771085	52	0.020661622	21	0.0342364
56	0.02561868	56	0.020661622	11	0.02570392
26	0.02106378	44	0.011922406	18	0.02570392
39	0.01874426	45	0.011922406	8	0.02470738
48	0.01726353	53	0.011922406	10	0.02470738
44	0.01305759	54	0.011922406	4	0.01466916
45	0.01305759	57	0.011922406	7	0.01466916
53	0.01156941	58	0.011922406	14	0.01466916
54	0.01156941	26	0.008406702	25	0.01466916
57	0.01073539	39	0.008406702	29	0.01466916
58	0.01073539	48	0.008406702	38	0.01466916

TABLE XVI: Top 15 Nodes with Betweenness Centrality

Base		**Transitive Closure**		**Dominant Tree**	
Node	**Value**	**Node**	**Value**	**Node**	**Value**
43	64.49143	0	0	1	11
27	61.97619	1	0	4	4
28	61.97619	2	0	8	4
26	60.86429	3	0	10	4
12	59.55619	4	0	2	3
13	59.55619	5	0	3	3
52	55.55238	6	0	11	3
14	55.4427	7	0	19	3
40	50.46952	8	0	21	3
41	50.46952	9	0	7	2
39	50.04095	10	0	0	0
23	45.18071	11	0	5	0
24	45.18071	12	0	6	0
4	42.77794	13	0	9	0
56	42.31905	14	0	12	0