Epidemiology Modeling for Compliance Graph Analysis

Noah L. Schrick
05 May 2023

Introduction

Compliance graphs are graphical representations of systems or a set of systems, with a focus on their regulation or compliance mandate status. Nodes in the graph represent the systems, with all their relevant information encoded within the node itself. Edges represent a change in a system, directed toward a new node that includes the change in information. As the graphs are generated, attributes can be applied. These attributes include labels for nodes or edges, or colors for violation severity.

Various approaches have been presented for analysis on attack graphs, which are similar to compliance graphs but with a focus on security. Common approaches include Bayesian Networks (such as that shown by the authors of [1] and the authors of [2]), Markov Models (shown by the authors of [3]), Game Theory (shown by the authors of [4]), and Uncertainty Analysis (shown by the authors of [5]). These approaches have shown success at highlighting various information of the original network. As an alternate approach, this work presents epidemiology modeling as an analysis technique. By using epidemiology modeling, the goal is to determine risk, predict future outcomes, and determine the success of correction schemes for a given environment.

Methodology

In order to apply epidemiology modeling to compliance graph, a few considerations were necessary. The compliance graphs are represented through a Graphviz dotfile [6], and must be imported as a graph object. This graph object must be iterated through, and must be used to derive the epidemiology model parameters and compartments. These parameters and compartments will then be used for an ODE solver. These processes are described in the following subsections.

Model

The selected model for compliance graph analysis was a SEIRDS (Susceptible – Exposed – Infected – Recovered - Deceased) model. Unlike a SIR model, the SEIRDS model includes an exposed group to represent nodes that *will* become infected, as compared to the susceptible group which *can* become infected. The exposed group has additional parameters, such as incubation time, which provide additional detail on when the group will transition to the infected group. In this model, there are also cases where the infected group can become deceased, rather than recover. After recovery, there is a waning immunity period where the population can become susceptible again. A similar model can be seen in the works presented by the authors of [7]. This model better fits the information present in a compliance graph, and their contextualization can be seen below in Tables 1 and 2. Figure 1 shows a block model representation of the SEIRDS model.

Table 1: Compartment Descriptors for Compliance Graphs. Each compartment is contextualized to compliance graphs, with their meanings and brief explanations of how nodes may fit in compartments.

Compartment	**Description**	**Contextualization to Compliance Graphs**
S	Susceptible	All other nodes.
E	Exposed	Nodes flagged as at risk of compliance violation by an IDS, timeframe trigger, or other.
I	Infected	Nodes that are in violation.
R	Recovered	Nodes that were infected and were capable of automatic correction (certificate renewal, scheduled maintenance, or other).
D	Deceased	Nodes removed from the network through quarantine, DMZ, legacy removal, or other.

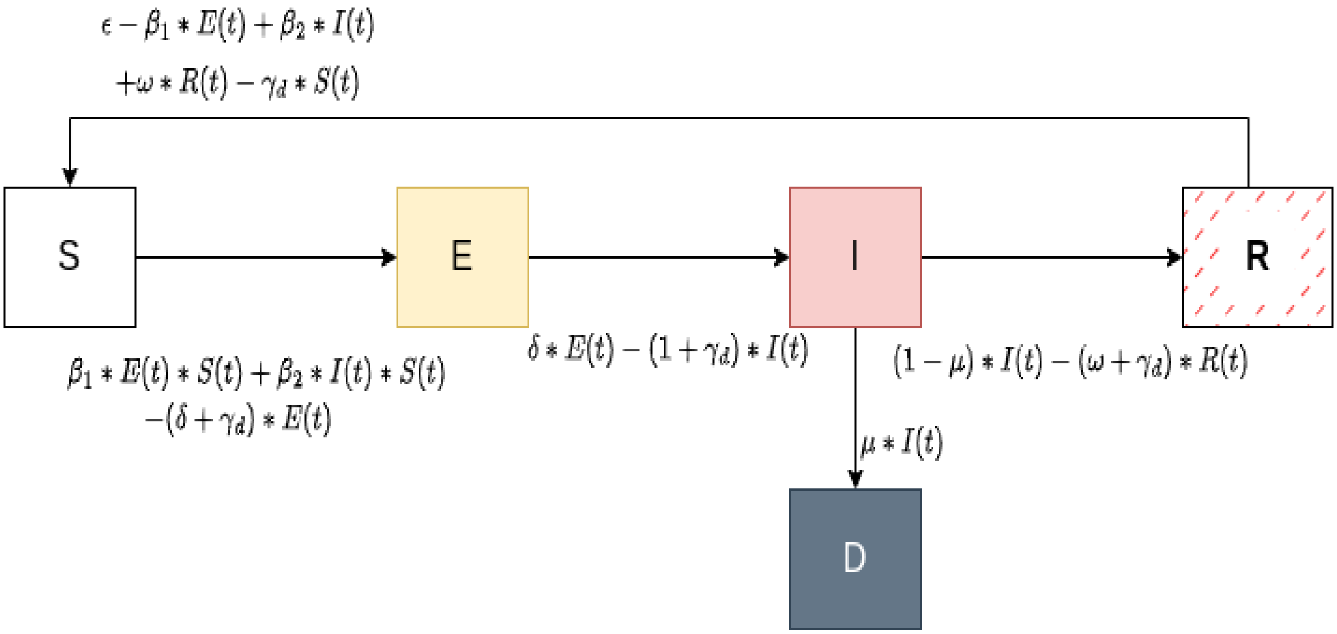
Table 2: Parameter Descriptors for Compliance Graphs.

Each parameter is contextualized to compliance graphs, with their meanings and brief explanations of how nodes relationships affect the parameters.

Parameter	Description	Contextualization to Compliance Graphs
β	Infection Rate	Probability of nodes falling out of compliance.
δ	Incubation Period	Once a node is at risk of falling out of compliance, how long it takes to actually violate a mandate or regulation.
γ_R	Recovery Rate	Probability of a system to correct its violation status.
γ_D	Death Rate	Probability of removal for a system in violation.
μ	Fatality Ratio	Natural rate at which any node may be removed from the network.
ϵ	Infected Import Rate	Systems that are already in violation; systems that do not fall out of compliance, but are already violating a mandate.
ω	Waning Immunity Rate	After a system recovers, how long for it to be available for violation.

Figure 1: SEIRDS Epidemiology Model.

Display of each compartment and the mathematical representation of the relationship between compartments.


$\epsilon - \beta_1 * E(t) + \beta_2 * I(t)$
$+\omega * R(t) - \gamma_d * S(t)$
S
E
I
R
D
$\beta_1 * E(t) * S(t) + \beta_2 * I(t) * S(t)$
$-(\delta + \gamma_d) * E(t)$
$\delta * E(t) - (1 + \gamma_d) * I(t)$
$(1 - \mu) * I(t) - (\omega + \gamma_d) * R(t)$
$\mu * I(t)$

Parameter and Compartments through Python

To obtain parameters and compartments for the SEIRDS model, the information present in the model needed to be loaded in and parsed. The network was stored in a Graphviz dotfile, and R's igraph library has limited functionality for interfacing with Graphviz. The network could be loaded, but no attributes would be preserved, included the directionality of the graph. Python has an equivalent library (NetworkX), so loading the library through Python was preferred. In NetworkX, many of the useful graph functions are in the DiGraph class. However, like igraph, importing the network through DiGraph loses attributes. The network could, however, be loaded into the NetworkX AGraph class. Though the network and all its attributes could be loaded with AGraph, AGraph does not have much functionality for working with the object. To account for this, the network was loaded through AGraph, converted to a DiGraph class, and both objects were used for the network operations.

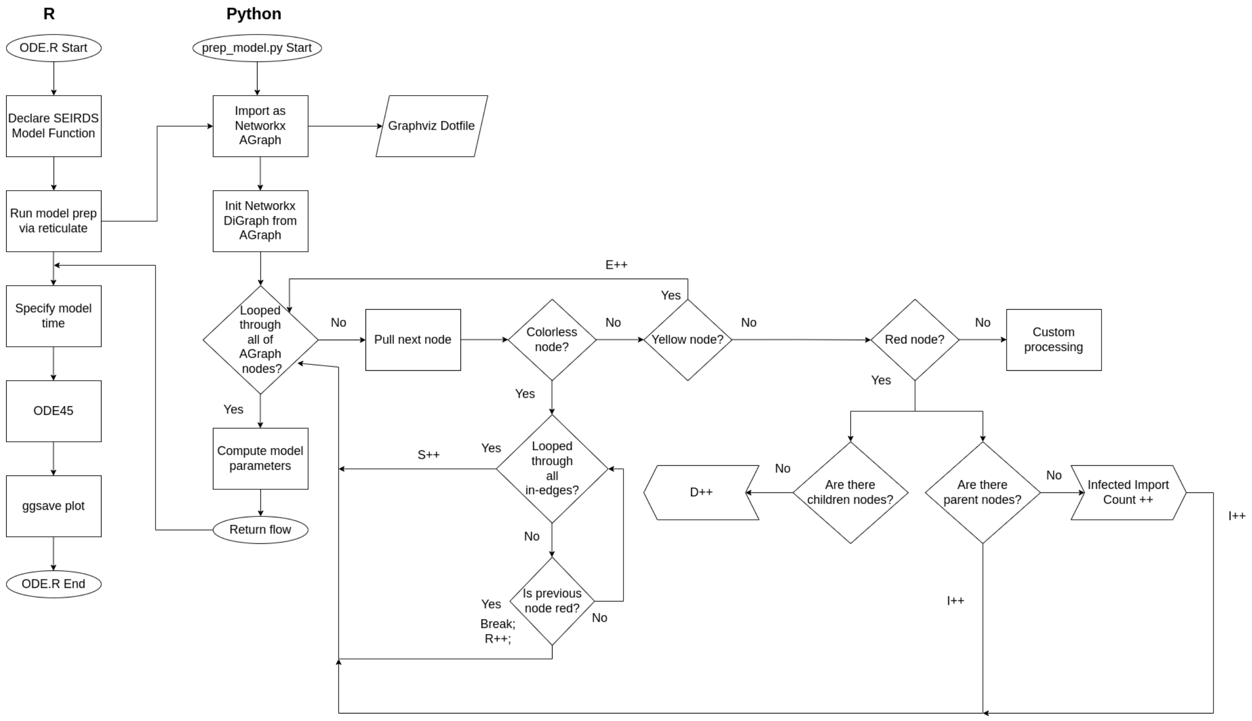
Once the network was loaded, its nodes were iterated through, with each node having its color checked. At each color (red for Infected, yellow for Exposed, white or colorless for Susceptible), the "in-edges" and/or "out-edges" were examined to determine if nodes were recovered, deceased, or if there was a transition from Susceptible to Exposed or Infected.

ODE Solver through R

The ODE solver used was the ODE45 library in R. Though a good portion of the work was performed in Python, the ODE solvers in Python did not have the same level as support or ease-of-use as the solvers present in R. To account for the data transfer between the Python script and the R ODE solver, the R reticulate library was used. Rather than using a system call to run the Python script and parse the resulting output in R, the reticulate library allows for environment sharing between Python and R, resulting in trivial data transferal. The program flow for this process is shown below in Figure 2.

Figure 2: Program Flow Diagram.

Display of the program flow, beginning with the R ODE solver code. Using reticulate, the Python script is ran to import the network file, and loop through node attributes and edges to derive compartments and parameters. These compartments and parameters are in a shared workspace with the ODE solver due to the reticulate library.



A function was created for the SEIRDS model. This function took inputs for the compartments, parameters, and a time step. The ODE45 library was called on the SEIRDS model function, where initial conditions were passed along with the specified time range. At each time step, the changes in populations were computed through the SEIRDS model function. After the solution was obtained, the result was melted based on time using the reshape2 library, and then plotted with ggplot2.

Incorporating Network Weights

Early shortcomings were identified after initial testing. In the initial tests, one difficulty was that topographical relationships were not accounted for in the parameter derivation. At each node, a static value of ‘1’ was added to the relative compartments or parameter counters. This approach does not account for nodes where edge weights may exist that describe the probability of a state transition.

To explore parameter alteration and potential addition of probability parameters, support for weighted networks was incorporated. This was performed by using trivial weighting. For each node, an out-edge weight was assigned based on the following:

$$Edge\,Weight = \frac{1}{number\ of\ out-edges} \tag{1}$$

At each node, rather than adding directly to compartments, temporary counter variables were leveraged to count edge weights appropriately. For instance, if a parent node had two children, with one exposed and one susceptible, the infection rate for this subset would be 0.5, rather than 1.0. This was also performed for the recovery rates.

Results and Analysis

Unweighted

A network with 395 total states was imported and examined. This network was predominantly Infected with only a limited number of Exposed nodes. This network depicted an environment where there was only a small window for a warning scheme, and repeated segments of Infected nodes. Nodes had a small recovery chance, and were likely to remain infected for long periods of time. Using this network, the compartments were obtained using the Python script. The values for the compartments are shown in Table 3. Likewise, the derived parameters are shown in Table 4. No natural increase was used for the network, so epsilon was set to 0. After nodes had recovered, they were capable of becoming susceptible at the next state transition, setting the waning immunity value to 1. The warning scheme used in the environment was equivalent to a 1 month warning scheme, so the incubation rate was also set to 1.

Table 3: Initial Values for the SEIRDS Model Compartments.
Obtained from the original network. Notably, the Infected population makes up the largest portion, with very low values for the Deceased and Exposed populations.

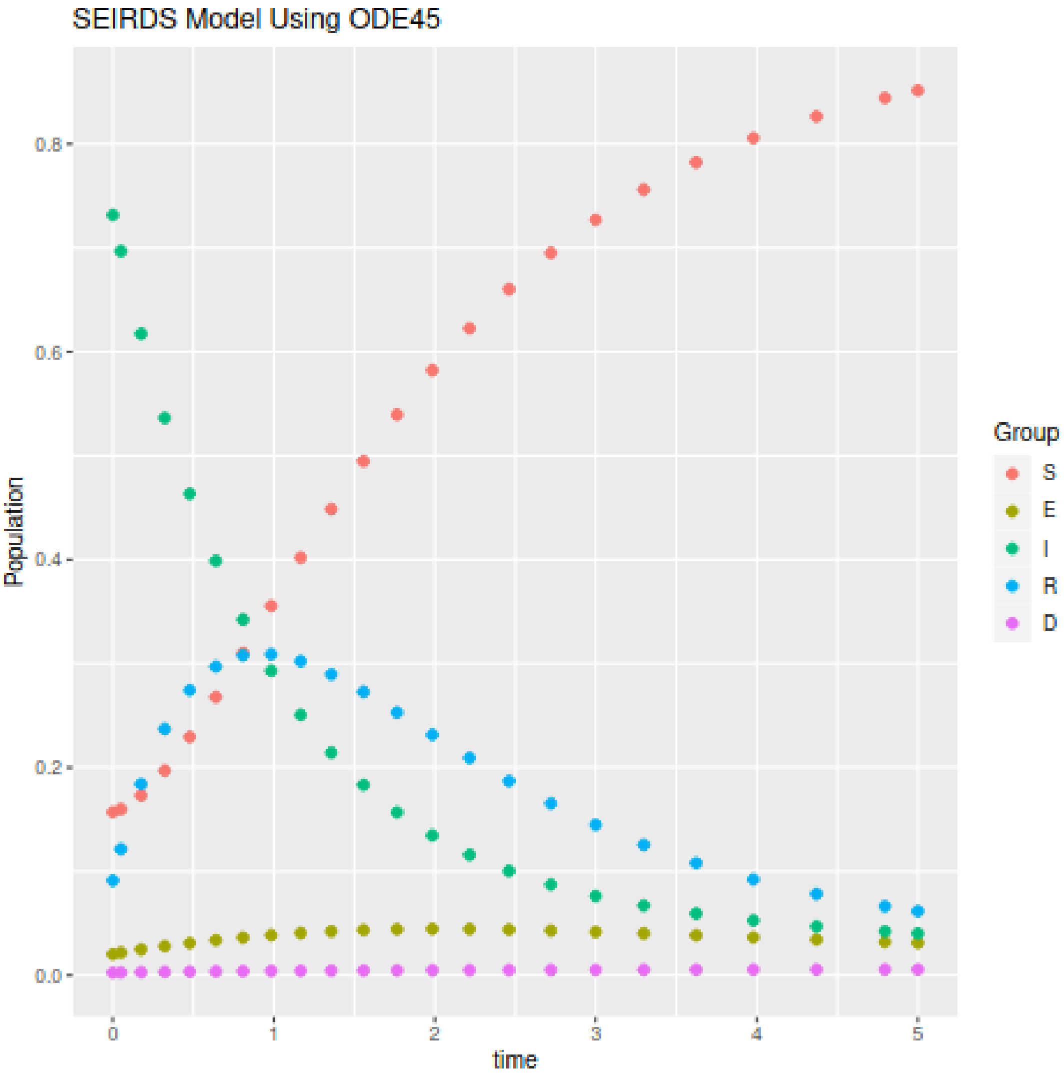
Compartment	Initial Value
S	62
E	8
I	289
R	36
D	1

Table 4: Initial Values for the Unweighted SEIRDS Model Parameters.
Obtained from the original network. Notably, incubation period and waning immunity were set to 1. Recovered rate, death rate, and fatality rate are low. No increase in population. Relatively high infection rate.

Parameter	Derived Value
β	0.4316
δ	1.0000
γ_R	0.0911
γ_D	0.0025
μ	0.0035
ϵ	0.0000
ω	1.0000

After all parameters and compartments were obtained, the ODE45 solver was ran with its solution melted and plotted. Figure 3 displays the compartment populations as they change over 5 time steps. The infected group starts as the largest portion of the population, and decreases by roughly 1/3 over 1 time step. During this time step, the recovered and susceptible groups rise to correspond with the decrease in the infected population. Over all 5 time steps, there is minimal increase in the exposed group or deceased group. The fatality rate was nearly negligible, and since the network only contained 1 deceased node to begin with, no notable changes were expected.

Figure 3: SEIRDS Epidemiology Model Results for an Unweighted Network. Display of compartment population changes over times 0 to 5. Infected population starts as the highest portion of the population, and decreases as nodes recover. Increase in the Susceptible population as the Infected population becomes Recovered, and then returns to a Susceptible state. Small rise in the Exposed population, with a negligible increase in the Deceased population.



<u>Weighted</u>

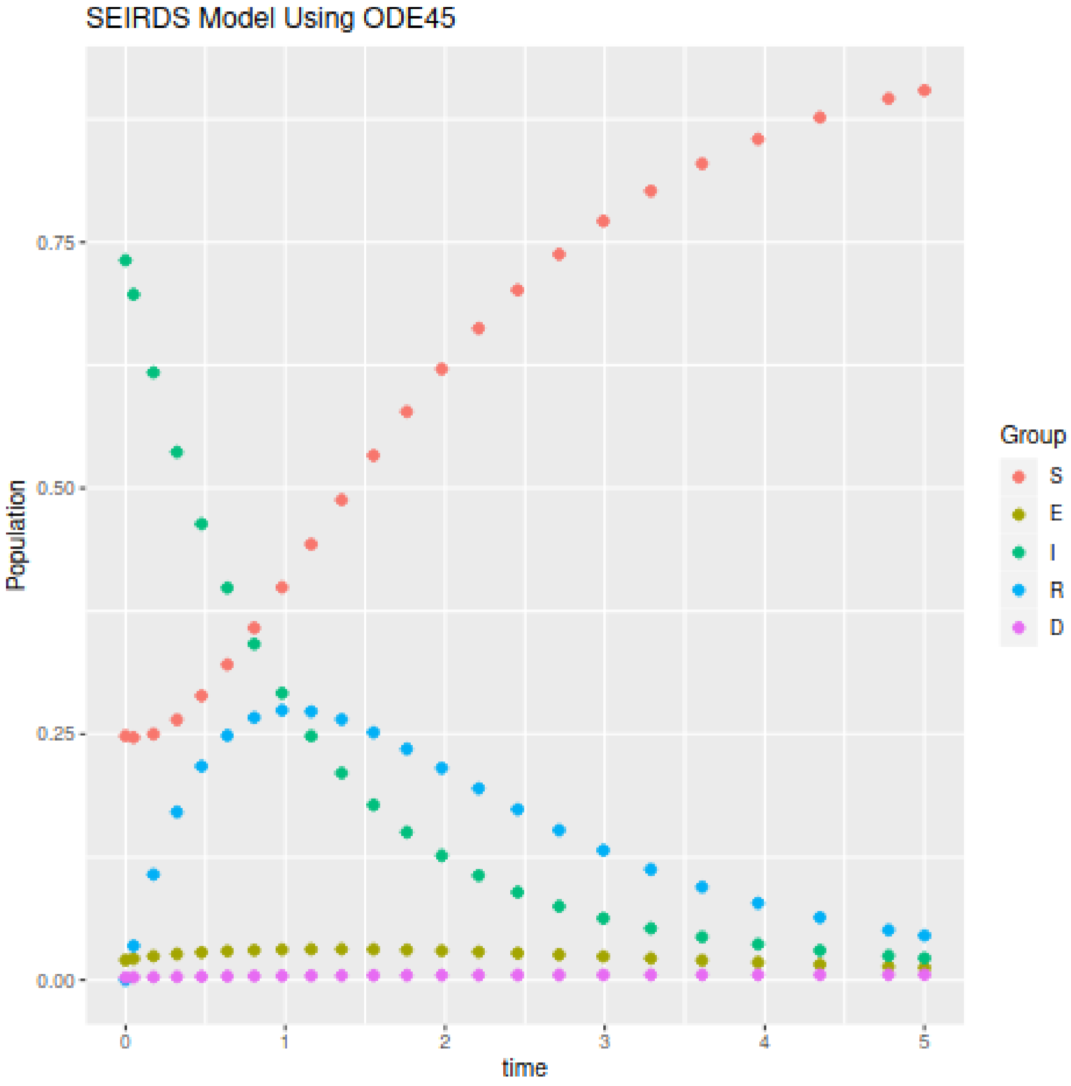
To investigate potential solutions to the difficulties described in Section 2.4, edge weights were added to the same network described in Section 3.1. As a result, there compartments are identical to those displayed in Table 3. Table 5 displays the new parameters after adding trivial weights to the network. It is notable that the infection rate and the recovery rate are roughly half to their counterparts in Table 4.

Table 5: Initial Values for the Weighted SEIRDS Model Parameters.
Obtained from the original network. Notably, incubation period and waning immunity were set to 1. Recovered rate, death rate, and fatality rate are low. No increase in population.
Infection rate and fatality rate are roughly half when compared to the unweighted network.

Parameter	**Derived Value**
β	0.2624
δ	1.0000
γ_R	0.0459
γ_D	0.0025
μ	0.0035
ϵ	0.0000
ω	1.0000

After the new parameters were obtained, the model was solved using ODE45, with its solution shown in Figure 4. Since the initial compartments were identical to those seen in Figure 3, and since all but two parameters were identical, the Figure appears very similar. In this test case, the amplitudes for all groups are slightly dampened, with time delays shown in the group curves. Though the exposed group in Figure 3 did not oscillate or rise by a significant amount, it changed by even less in Figure 4. Since the infection rate was cut in half, the exposed group did not experience as much population growth compared to Figure 3.

Figure 4: SEIRDS Epidemiology Model Results for a Weighted Network.
Display of compartment population changes over times 0 to 5. Infected population starts as the highest portion of the population, and decreases as nodes recover. Increase in the Susceptible population as the Infected population becomes Recovered, and then returns to a Susceptible state. Small rise in the Exposed population, with a negligible increase in the Deceased population.
Dampened amplitudes and delays when compared to the Unweighted Network.



Conclusions and Future Works

This work investigated the use of epidemiology modeling for compliance graph analysis. This work made use of one network, and used a SEIRDS model for analysis. The network was treated as both unweighted and weighted with trivial edge weights. In both cases, the epidemiology group populations were modeled over 5 time steps, noting the degree in which they changed over time. In both sets of results, room for potential future work was identified.

In the analyzed network, there were two groupings of infected nodes. The first grouping was mostly upstream, where the infected nodes were successfully able to recover due to a scheduled maintenance. The second grouping of infected nodes were mostly all downstream, and were unable to recover. It is noted that in the SEIRDS model, there is no account for the difference in upstream and downstream nodes. The recovery rate was obtained mostly through the upstream nodes, and the parameter is able to fit the first grouping relatively well. However, this recovery rate does not apply to the second grouping of infected nodes that never recover. As a result, future work is available for multi-strain epidemiology modeling. SEIRDS modeling can still be used as a basis, with separate subsets of exposed and infected groups. The first grouping of infected nodes could be fit into an E_1 and I_1 subset, with the downstream grouping of infected nodes fitting into an E_2 and I_2 subset. Separate transitional relationships between the susceptible group and the exposed and infected subsets could be modeled, each with their own set of parameters.

Additionally, new probability parameters could be used in the SEIRDS model. Rather than using an unweighted model or using trivial weights, probabilities that a node may fall out of compliance could be obtained independently, and can be represented as edge weights in the original network.

There are also difficulties when deriving parameters and compartments for the SEIRDS model from the same network. As an alternative, a network with time-steps could be used. In this type of network, subgraphs could be obtained for all nodes and edges belonging to a certain time step. In this subgraph, parameters could be derived. This process could be repeated until the entire network has been processed, with a method to combine the parameters. The compartments would be based on the entire network, but each subgraph would have its own set of computed parameters. Based on additional user-provided data, the subgraphs could be prioritized to add weighting to their computed parameters.

References

[1] M. Frigault, L. Wang, A. Singhal, and S. Jajodia, “Measuring network security using dynamic Bayesian network,” in Proceedings of the ACM Workshop on Quality of Protection, pp. 23–30, Alexandria, VA, USA, October 2008.

[2] J. Wu, L. Yin, and Y. Guo, “Cyber attacks prediction model based on Bayesian network,” in *Proceedings of the IEEE International Conference on Parallel and Distributed Systems*, pp. 730-731, Singapore, December 2012.

[3] S. Abraham and S. Nair, “Cyber security analytics: a stochastic model for security quantification using absorbing Markov chains,” *Journal of Communications*, vol. 9, no. 12, pp. 899–907, 2014.

[4] K. Durkota, V. Lisý, B. Bošanský, and C. Kiekintveld, “Approximate solutions for attack graph games with imperfect information,” in *Proceedings of the International Conference on Decision and Game Theory for Security*, pp. 228–249, London, UK, November 2015.

[5] H. H. Nguyen, K. Palani, and D. M. Nicol, “An approach to incorporating uncertainty in network security analysis,” in *Proceedings of the Hot Topics in Science of Security: Symposium and Bootcamp*, pp. 74–84, Hanover, MA, USA, April 2017.

[6] Graphviz, “Graphviz Dot”, https://graphviz.org/docs/layouts/dot/, Oct. 2022. Version 6153a896.

[7] M. B. Alaya, W. B. Aribi, S. B. Miled, “Mathematical analysis of a delayed SEIRDS epidemics models: deterministic and stochastic approach”, in *arXiv q-bio.PE,* 2208.07690. Aug. 2022.