The Effects of Modularity on Cascading Failures in Complex Engineering Systems

by
Elizabeth M. Parker

A THESIS

submitted to

Oregon State University

Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Industrial Engineering
(Honors Scholar)

Presented June 5, 2020
Commencement June 2020

AN ABSTRACT OF THE THESIS OF

Elizabeth M. Parker for the degree of Honors Baccalaureate of Science in Industrial Engineering presented on May 5, 2020. Title: The Effects of Modularity on Cascading Failures in Complex Engineering Systems.

Abstract approved:_____

Irem Y. Tumer

To design engineering systems that have improved reliability, it is important to understand what kind of system faults they will be susceptible to. Mitigation strategies are important to ensuring the performance of these engineering systems. Understanding how the modularity of complex engineering systems affects the risk of devastating failures such as cascading failures can help enable engineers to implement strategies in the design phase to increase reliability. The extent to which decreased system modularity propagates the spread of a cascading failure is unknown. This study analyzes how modularity in complex engineering systems affects resistance to the spread of cascading failures. In this research, synthetic networks are used to represent component models at differing degrees of modularity. These synthetic networks are then infected through epidemic spreading models that model cascading failures. The loss of functionality is determined by the percent of diseased nodes in the system, and the influence of the initial node is measured by eigenvector centrality. Increased modularity is associated with the improved ability of a system to inhibit the propagation of cascading failures over time through failure isolation within a module, measured by percent infected in the system, in comparison to less modular systems ($p < 0.001$). This finding indicates that the structural design of complex engineering systems could be crucial to increasing reliability in design with reference to cascading failures.

Key Words: Complex engineering systems, modularity, cascading failures

Corresponding e-mail address: parkerel@oregonstate.edu

The Effects of Modularity on Cascading Failures in Complex Engineering Systems

by
Elizabeth M. Parker

A THESIS

submitted to

Oregon State University

Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Industrial Engineering
(Honors Scholar)

Presented June 5, 2020
Commencement June 2020

Honors Baccalaureate of Science in Industrial Engineering project of Elizabeth M. Parker presented on June 5, 2020.

APPROVED:

_____

Irem Y. Tumer, Mentor, representing College of Engineering

_____

Andy Dong, Committee Member, representing Mechanical Engineering

_____

Hannah S. Walsh, Committee Member, representing Mechanical Engineering

_____

Toni Doolen, Dean, Oregon State University Honors College

I understand that my project will become part of the permanent collection of Oregon State University, Honors College. My signature below authorizes release of my project to any reader upon request.

_____

Elizabeth M. Parker, Author

ACKNOWLEDGMENTS

I express my gratitude and appreciation to my thesis committee. My advisor, Dr. Irem Tumer, has given continual support and encouragement. This opportunity was one of the most formative parts of my undergraduate education and would not have been possible without her willingness to take on an honors student amidst an impossible schedule. I would like to thank Dr. Andy Dong. I am extremely grateful for his input and enthusiasm. He has always encouraged critical thinking by asking the hard questions. I would also like to thank Hannah Walsh. Her patience, kindness, and encouragement through every step has been instrumental in this process, and without her this thesis would not have been possible.

1 INTRODUCTION

The 21st century is an incredible time for technological developments. Engineers are designing increasingly complex systems to address increasingly complex problems. Self-driving cars are on the road, and seats for commercial space flight are on the market. With these progressively complex systems, failures are likely and even expected. Key considerations in designing a product are complications to anticipate and redundancies to preemptively implement. As failures occur, they can expand beyond the point of origin to other parts of the system. In 2003, the East Coast power grid experienced such a problem [1]. The redundant lines that carry power are under high stress during the summer months. As the lines heat up, they sag. A failure in a line diverts the load to parallel lines, causing those lines to further sag. Such a failure in Ohio led to a massive overload in the system, causing further complications, and resulting in over 50 million people without power. Research into architectures that are resistant to cascading failures is crucial to preventing such accidents. Understanding how systems react to failures is essential to designing systems that are resistant to failures.

Modularity in engineering models specifies a division of tasks and is generally designed into the system for redundancy or functional purposes [2]. Modular design methods tend to emphasize the maximization of modularity rather than considering possible setbacks from other system attributes [3]. As modularity in these systems varies, so does reaction to a component failure. More specifically, it can be hypothesized that highly connected systems run a higher risk of falling victim to cascading failures, as in the power line failure described previously. A cascading failure is a type of failure that increases over time as one component triggers the failure of another, causing a positive feedback loop [4]. Highly connected systems provide more pathways for a failure to travel through, showing that more connections between modules may lead to a faster spread of a failure [5]. In contrast, if a system is highly modular with few connections between modules, a failure has a greater chance of being contained within the affected module. This ability is weakened if the infection begins at a highly connected point in the system. As in an infection spread, creating fewer connections between communities of people can lessen the spread of the disease and contain the infection to only those affected communities. If someone who is involved in many communities becomes infected, the social network is at a higher risk of infection. Allowing more interaction between communities increases the likelihood that the infection will travel. Like social systems, engineered systems can be complex and modular, and thus it is important to understand how the varying degrees of modularity affect the severity of cascading failures within the systems. Analyzing which levels of modularity are more susceptible to cascading failures enables engineers to determine which components need to be higher in survivability [6].

This study assesses the impacts of modularity on a complex engineering system's ability to contain the spread of a cascading failure. If systems have high levels of connectivity within modules but few

connections between modules, failures may be more easily contained within the affected module. Complex engineering systems with varying modularity in physical architecture, such as aircraft, weather satellites, or air traffic control systems, are required to perform with exceedingly high rates of dependability [7] [8]. This makes the problem of understanding how these systems react to failure an important problem. This study addresses this by generating synthetic networks with varying modularity, infecting them with an epidemic as a way to model the failure of a component and the potential spread of the failure to connected components, measuring the spread of that epidemic through the percent infected, and comparing the spread at different infection starting points. Understanding how modularity affects the spread of cascading failures provides vital information to the design and monitoring of complex engineering systems. If a system is less modular by design, implementing preventative methods to monitor the system for potential cascading failures can prevent system failure. This study investigates how increased network modularity will increase resistance to cascading failures.

2 BACKGROUND

Complex systems vary from technological, to biological, to social in nature and are often difficult to analyze, with many interrelated parts [9]. Complex engineering systems are a subset of complex systems. The terminology of 'complex engineering systems' differentiates from other complex systems to highlight that they are artificial, exist to perform specific functions, and are comprised of interdependent engineered systems [10]. Like complex systems, complex engineered systems have architectures and behaviors that cannot be fully understood and modeled due to their tendency to self-organize [11]. Self-organization is the spontaneous emergence of order in a system in the absence of external interference [12]. A complex system is neither completely random nor completely ordered [13]. This property makes complex networks a viable approach for analysis of complex engineered systems. Networks are made up of the interacting components of a complex system and are a useful tool to simplify systems into abstract structures by showing only basic connections [5]. In their most basic form, networks are collections of nodes connected by lines. These can then be labeled with additional information to represent something as simple as a family tree or as complex as the network structure of the Internet. Engineering system components can be modeled as the nodes, and relationships can be displayed as the lines connecting each node. Complex networks are used in research to understand a variety of complex systems [14]. Modeling product co-considerations as complex networks allows for the analysis of the relationship between customers and products in the vehicle industry [15]. Requirement change propagation in the design process can be predicted through complex network centrality metrics [16]. Visualizing these systems as complex networks creates a simpler method for analysis [17].

Furthering the use of using complex networks in the modeling of complex systems is the introduction of modules [17]. Networks represent each component in the system as a node. Clusters of

nodes make up each module. Modules are frequently designed to contain components that are more connected to each other than other components in the system [18]. The connections between each node are referred to as edges [19]. This technique assumes that all nodes are equal [10]. In reality, all components are not equal in importance. For the purpose of this study, the nodes that connect modules together are referred to as bridging nodes. More or fewer bridging nodes are created as interconnectedness fluctuates. Research suggests that when those particular nodes are subject to failure through a cascading failure, they have a moderate to strong effect on the failure tolerance of a system [17]. Despite this impact on failure spread, bridging nodes are modeled as commensurate with the other system components [20]. Nodes can be commensurate if they share functional or categorical similarities, or if their commensurate status supports the ultimate goal of the model [10]. For example, the ocean freight industry frequents certain ports. The Port of Los Angeles is significantly larger than the Port of Miami; however, in a network diagram, both would be modeled as equal nodes. In an argument against using complex networks in complex system modeling, some researchers highlight these heterogeneous parts as a reason this method is unfit for research [21]. Despite these limitations, network analysis is useful due to the inherent benefits of failure analysis, non-trivial topological features, and emergent interaction detection [10].

Failure analysis through complex networks can provide valuable insight. Haley et al. utilizes uni-partite and bi-partite networks for failure analysis [10]. A network with nodes of a single type is known as a uni-partite network, and a network with nodes of two types as a bi-partite network. Uni-partite behavioral networks can be used specifically in the analysis of commensurate node relationships to look at failure properties. A bi-partite network can be used to map the relationships in the same network between functional and design parameters to look at system behavior and performance. Research on cascading failures as a form of failure analysis is rapidly increasing. Crucitti et al. utilize complex networks to look at cascading failures and show that a single node failure can be catastrophic in highly heterogenous networks with large load distribution differentials [22]. Wang and Rong study how different targeted attacks on the United States power grid vary the impact of cascading failures [23]. Talukdar et al. investigate survivability in the aftermath of a power grid cascading failure, focusing on mission continuation during an inevitable blackout rather than prevention [24]. Mitigation strategies for overloaded edges in networks through routing can reduce the need to shut down nodes to stop a cascading failure [25]. Mehrpouyan et al. establish that networks with a higher node degree will propagate failures faster [26]. Modularity in networks is shown to have a relationship with cascading failures through intermodular links [27]. While prior studies have established network analysis as a means of understanding system failure, few studies have investigated the direct relationship between a system's modularity and its ability to decrease the rate of propagation of faults across a network. This study treats a cascading fault as an error in a node that then transmits to all other connected components, such as a contaminant in a part that flows to other parts and causes problems. Within

a highly modular system, the limited connections to outside modules create fewer paths for a fault to travel. This occurrence leads to the hypothesis that increased system modularity limits the spread of a cascading failure due to the isolation of the failure within a module, thus making the system more resistant to cascading failures.

3 METHOD

To ease the complexity of analyzing modularity, synthetic networks perform well in place of actual component models. This study utilizes synthetic network generation to create networks with varying degrees of modularity. These networks are then infected with an epidemic spreading model, and evaluated for Q-modularity, percent infected, and edges in the system. This process is depicted in Figure 3. A degree distribution plot is a common approach used to describe networks. Figure 1 shows the degree distribution plot of for the behavioral network of a simple drivetrain system [17]. Figure 2 shows the degree distribution plot for a synthetic network generated through the methodology used on the networks used in this study. The similarities in the two plots show that synthetic networks have a similar structure to real engineered systems, implying that the failure tolerances will be similar. Furthermore, synthetic networks allow for the performance of controlled experiments. Rather than finding models with differing levels of modularity, software is able to create network models with adjustable inter and intra module connectedness. Interconnectedness refers to the number of connections between modules, while intra-connectedness refers to the number of connections within modules. To increase modularity, intra-connectedness must be high, and interconnectedness must be low. By varying only one parameter in network generation and holding the other factors constant, the effect of that parameter's variation can be tested on the outcome.

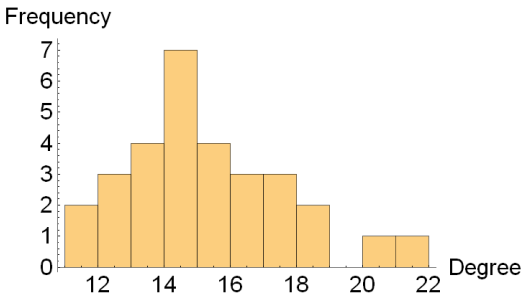

Figure 2: Degree distribution plot for synthetic network with $p_{c,m}$=0.8, $p_{c,i}$=0.35

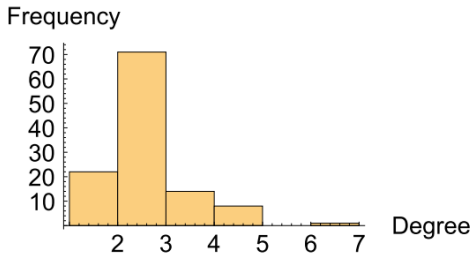

Figure 2: Degree distribution plot for simple drivetrain system

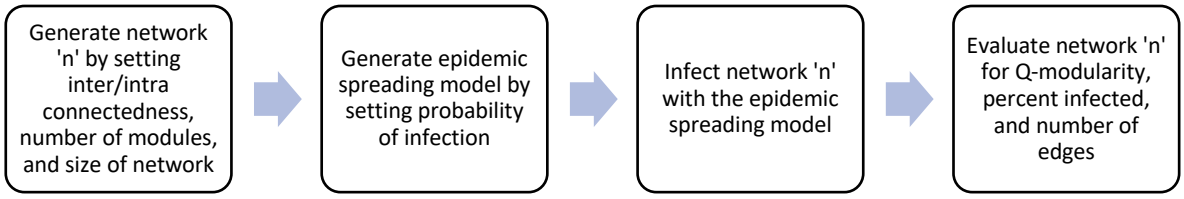

Figure 3: Algorithm for data collection

The generation of synthetic networks for this research was done within Mathematica. The synthetic networks generated consist of four inputs: $p_{c,m}$, $p_{c,i}$, number of modules, and network size. The $p_{c,m}$ refers to the interconnectedness, or the probability of nodes within a module being connected. The $p_{c,i}$ denotes intra-connectedness, or the probability of nodes between modules being connected. A network is considered modular when $p_{c,m} > p_{c,i}$. If $p_{c,m} \simeq p_{c,i}$, the network is considered integrated. The method for generating synthetic networks is based off of methodologies from Walsh [28], Sarkar et al [29] and Kasthurirathna et al [30]. Networks are generated by manipulating connections between nodes and connections between modules. The network size is set to contain 30 nodes, and each node must reside within a module. This means that the network size must be divisible by the number of modules with no remainder. The number of modules is held constant at 3, to ensure modules are large enough to contain more than just bridging nodes. Constraining the number of modules and nodes ensures that the differences between networks is solely due to connections produced by chance. Walsh et al. show that real engineered systems with 10, 19, and 375 components, random node removal does not significantly impact the network's Q-modularity [31]. This makes 30 nodes a reasonable value to analyze in this study. In the network, the adjacency matrix can be used to show node connections. If nodes i and j share a connection, A_{ij} is equal to 1. If there is not a connection between nodes i and j, A_{ij} is equal to 0. This matrix is utilized in the calculation of the network modularity as well as the eigenvector centrality.

3.1 Q-Modularity

Network modularity is evaluated using Q-modularity. Q-modularity is a quantitative criterion of the extent to which nodes in a module are connected [5]. The value of the Q-modularity quantifies the strength of modules themselves rather than the number of modules in a network [31]. Each module in the network represents a cluster of nodes that interact on a higher level of the system. Eq. 1 gives the calculation method for Q-modularity. The connections between nodes are represented by an adjacency matrix A_{ij}, m is total number of edges in the system, k_i is the degree of vertex i, and $\delta(c_i, c_j)$ is the Kronecker delta.

$$Q = \frac{1}{2m}\sum_{ij}\left(A_{ij} - \frac{k_i k_j}{2m}\right)\delta(c_i, c_j) \quad (1)$$

The value of Q falls between -1 and 1, but never exactly 1. If Q is positive, it shows to what degree there are more edges between nodes than would be expected by chance. If Q is negative, it quantifies the degree to which there are fewer edges between nodes than would be expected by chance. Q-modularity is ideal for network analysis and research because it can be used on both component and behavior models [31]. Figure 2 shows a highly modular network generated with a high interconnectedness ($p_{c,m}$) and low intra-connectedness ($p_{c,i}$). Figure 4 shows a network with equal $p_{c,m}$ and $p_{c,i}$. Figure 5 depicts a network with a very low $p_{c,m}$ and high $p_{c,i}$, which is shown in the increase in connections between modules from Figure 4 to Figure 5. The networks themselves do not dramatically vary visually; however, their Q-modularity values show that they are structurally very different.

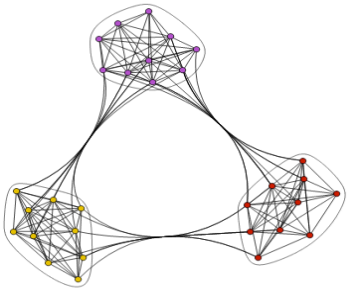

Figure 4: Highly modular network ($p_{c,m}$=0.9, $p_{c,i}$=0.1, Q=0.4406). Many edges within modules, but few edges between modules.

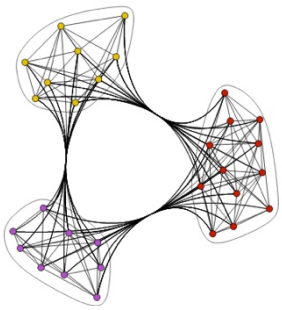

Figure 5: Low modular network ($p_{c,m}$=0.5, $p_{c,i}$=0.5, Q=0.0882). Equal probability of connection between nodes and modules.

3.2 Edge Count

Connections between nodes are referred to as edges. The number of edges in a network can contribute as a confounding variable in calculating the percentage of infected nodes. For example, social distancing is aimed to reduce social interaction and thus prevent the spread of infection. Each social interaction can be conceptualized as an edge, and each individual as a node. By limiting the edges in the network, infection can be minimized. Figure 6 shows a positive correlation between number of edges in the network and the final percentage of infected nodes. This relationship creates a need to control how many edges are in each network.

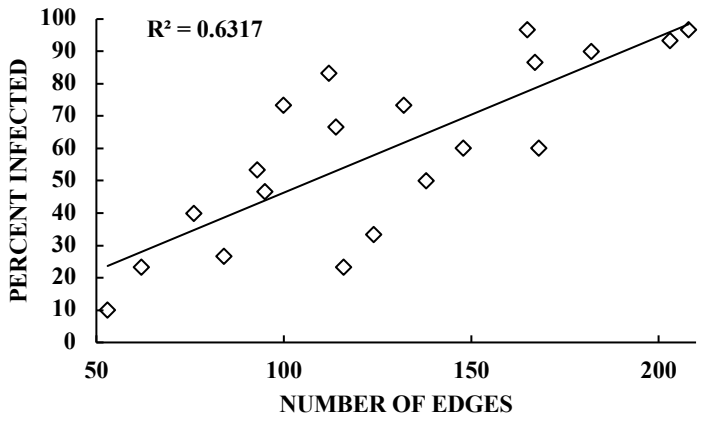

Figure 6: Plot of number of edges against percent infected using data from Table 2 of Appendix A

Edges can be held constant throughout network generation. Functionally, this means that edges do not fluctuate with modularity, they instead move around the system between different nodes. Eq. 2 can be utilized to calculate the number of edges in a system. Eqs. 2 and 3 are derived from Walsh [28] and based on the methodology from Sarkar et al [29] and Kasthurirathna et al [30].

$$E = \frac{N(\frac{N}{M}-1)}{2} * \left[p_{c,i} * (M-1) + p_{c,m}\right] \quad (2)$$

$$p_{c,i} = \frac{\frac{2E}{N(\frac{N}{M}-1)} - p_{c,m}}{M-1} \quad (3)$$

E is calculated in terms of N, M, $p_{c,i}$, and $p_{c,m}$. N denotes the network size as the number of nodes, while M is the number of nodes in the network. Once Eq. 2 has been used to determine the number of edges in a network for a specific $p_{c,m}$ and $p_{c,i}$ value, Eq. 3 can be used to calculate the $p_{c,i}$ value for the desired number of edges with an adjusted $p_{c,m}$.

3.3 Epidemic Spreading Function

Epidemic spreading models can be used in research of cascading failures. Guan et al. show how simulated cascading failures are accurate when compared to real-world cascading failure examples [32]. D. Valdez et al utilize the susceptible-infected-recovered (SIR) model of infection to study the role of bridging nodes in the spread of infection [33]. Mehrpouyan et al implement an epidemic spreading model to evaluate resiliency in complex engineering design [26]. The cascading failure model in this study utilizes the susceptible-infected (SI) model of disease spread. This model acknowledges two types of nodes: susceptible (S) or infected (I). This model simulates a system in which infected components experience an unrecoverable failure. Using an SI model represents an engineering system experiencing a failure that

incapacitates each node it interacts with. Population is denoted by N, so that N is the sum of S and I [34]. N refers to network size in the utilized model and is set at 30. The probability of infection per time step, or discrete state, is commonly denoted by β [5]. The probability of infection represents the likelihood of transmission between two nodes, one which is 'infected' and one which is 'healthy.' Thus, the number of new infections is βSI. This process makes a key assumption of no recovery of infected nodes. Once a node is infected, it cannot recover. In each infection probability time step β, the infected nodes retain their ability to infect healthy nodes, creating a cascading failure. This modeling technique is based off of the analysis of human interaction during an epidemic, with interaction modeled from social behavior [35]. An epidemic spreading models views a complex engineering system as a population existing in different discrete states [36]. The SI model can be represented by the following differential equation [37]:

$$\frac{dS}{dt} = -\beta SI = -\beta S(N - S) \tag{4}$$

While a two-state classification model overlooks many finer biological details in an infection spread, it captures the fundamental features of a disease dynamic. The SI model is a useful simplification to look at network level effects rather than what is happening internally within each node [5]. This study measures the number of infected nodes, rather than how recovering nodes can influence disease spread. The SI model is ideal for this due to its unchanging classification of infected nodes, no matter the passage of time steps. The use of percent infected as the metric for analyzing the spread of the cascading failure is useful to interpret the loss of functionality in the system. Infecting through three timesteps allows a look at the initial phase of an infection timeline where the majority of the system population is still susceptible, rather than letting the infection run its course and seeing the aftermath. Using an alternate method, such as time until 100% infection, would allow an analysis of which system survived the longest, but the interest of this study lies in how well modules contain the cascading failure. This is better described by percent infected rather than how long the system can survive the epidemic before total failure.

3.4 Eigenvector Centrality

Eigenvector centrality is a measurement of the influence a node has on a network [5]. Bonacich was the first to suggest that the eigenvector of the largest eigenvalue of an adjacency matrix could serve as a useful network centrality measure [38]. This method of degree centrality weights connections based on the centralities of the neighboring nodes. It is a weighted sum of not only direct connections, but the indirect connections as well, allowing it to take into account the entire network pattern [39]. The centrality of a node is proportional to the sum of the neighboring nodes, meaning that this property allows eigenvector centrality to be large because the node is highly connected or has important neighbors. Consider a social network. By this measure, one person can be important because they know a lot of people (who are not necessarily

incredibly important), or because they know a few very important people, or both. This can be represented mathematically as [5]:

$$x_i = k_1^{-1} \sum_j A_{ij} x_j \tag{5}$$

where x_i is the centrality of vertex i, and $k_1$is the largest eigenvalue of all eigenvalues k_i in adjacency matrix A_{ij}. This equation can also be written in matrix form [38] to say that centrality **x** satisfies

$$\mathbf{Ax} = \mathrm{k}_1\mathbf{x} \tag{6}$$

Calculating at the eigenvector centrality of each node shows the relative structural importance of each node within the network. Infecting the network starting at varying nodes and seeing how the eigenvector centrality correlates with the final percent infected offers the opportunity to investigate the importance of the starting node to the propagation of the epidemic.

3.5 Data Collection

The data collection was completed in six different phases. The initial collection of data consisted of creating 20 networks at varying levels of inter and intra connectedness, chosen at random. The probability of infection was initially set to 0.3. The initial data shown in Table 2 of Appendix A included the analysis of networks with $p_{c,i} > p_{c,m}$, which are not considered modular. After an adjustment in $p_{c,m}$ and $p_{c,i}$ values, the number of edges in each network was measured to determine if it was acting as a confounding variable. Table 2 of Appendix A shows that as modularity decreased, so did the number of edges in the system. This was due to the modularity in the system being adjusted without consideration of edges. Equations 2 and 3 were then implemented to hold the edges constant, and while Q modularity showed improvement in variety, the percent infected in each system at the end of the simulation was consistently high (see Table 3 of Appendix A). The next phase of data collection involved lowering the probability of infection to 0.1 and ensuring that $p_{c,m}$ remained greater than $p_{c,i}$. This data, shown in Table 4 of Appendix A, displayed the stochastic nature of the epidemic spreading model. The inherent stochasticity that exists within an infection algorithm can produce varying results each time it infects the same network. In response, a loop was incorporated into the code to infect the model 1000 times and use the average of the fluctuating results. The 1000 trials were plotted in Figure 9, and the large spread of infection results led to the measurement of eigenvector centrality. This was an unforeseen confounding variable, in response eigenvector centrality was plotted against percent infected to see if the infection start node of the epidemic played a role in the epidemic spread.

4 RESULTS

The data collected after the final phase of data collection, shown in Table 5 of Appendix A, shows the relationship between modularity and percent infected to be negative. Figure 7 plots the two and visually

displays a general negative trend is observable between infection and modularity. Spearman's rank correlation details the relationship between a monotonic relationship, rather than the strength of a linear relationship such as Pearson's correlation [40]. Spearman's rank correlation is useful to this study because it does not assume linearity in the data and is most useful in studies with a sample size below 28. It shows whether or not the relationship between the two variables occurred by chance. The Spearman's rank correlation shown in Table 1 is statistically significant at a 95% confidence level ($p < 0.01$). This statistically significant correlation means that there is only a 5% chance that the strength of ρ occurred by chance. A one-tailed t-test is appropriate for this study because the hypothesis is determining a directional relationship between Q-modularity and percent infected, rather than showing just a relationship.

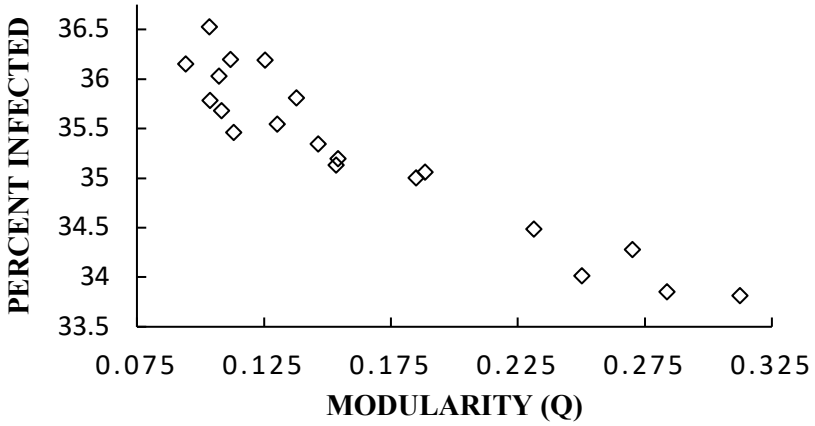

Figure 7: Plot of relationship between Q modularity and percent infected in a network

Table 1: Spearman's rank correlation: t-test results

Spearman's Rank Correlation	
Correlation value ρ	-0.9188
One-tailed T-stat	-9.8754
P-Value, $\alpha = 0.05$	5.4139×10^{-9}

When comparing different networks over five time steps, Figure 8 displays how the network with a higher Q-modularity is infected slower than the less modular network. As time increases, the graph shows that the gap in the infection rates also increases. Figure 9 shows the large variation in the infection spread. Each network had a minimum of 3.33% infected within the 1000 trials, an outcome that would occur if the infection never spread past the original node. The 25th through 75th percentile spread is large; however, this is not surprising due to the stochastic nature of the epidemic spreading model. The initial point of infection is not targeted; it occurs randomly with each simulation. The probability of infection is

just that, a probability. When the infection starts at a random node and the likelihood of the start node infecting neighboring nodes is a small probability, there are countless ways for the infection to spread through the system. This inherent randomness causes much variation in the study of cascading failures through epidemic spreading models.

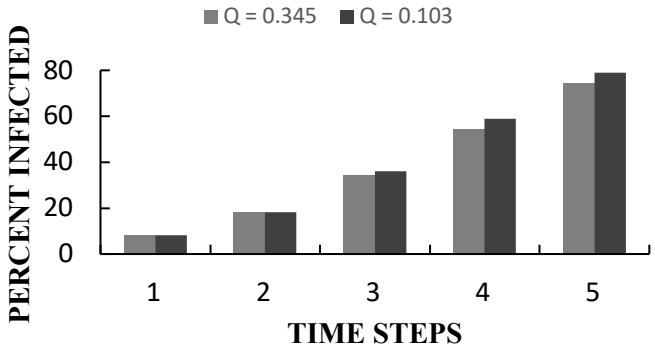

Figure 8: Comparison of two synthetic network infection rates over five time steps

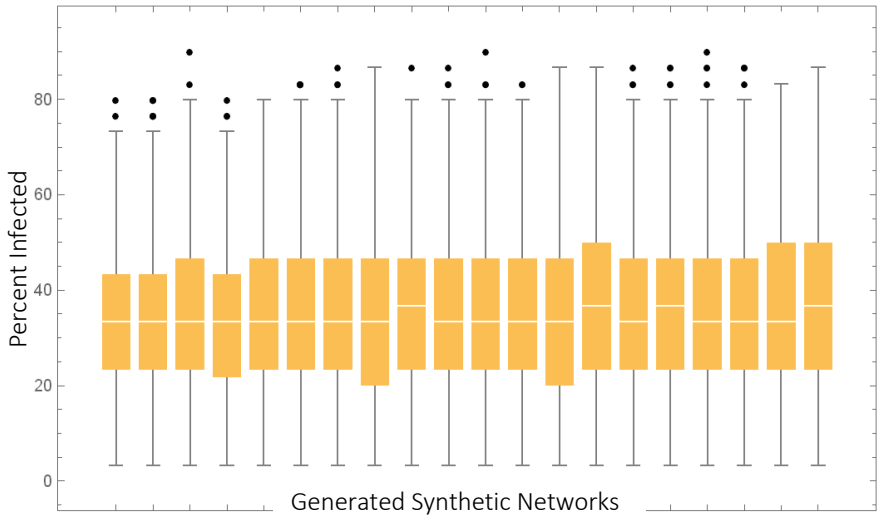

Figure 9: Boxplots of 1000 trials of epidemic spreading model on 20 generated synthetic networks

Controlling the start node of the infection shows a strong linear correlation between eigenvector centrality and percent infected. Fixing a node as the initial infection point and infecting it from that node allows for the comparison of infection rates to the eigenvector centrality of the start node. Figures 10, 11, and 12 show how the strength of the correlation increases with trials of infection. These figures illustrate the influence of the path of propagation in addition to the start node. As shown in Figure 9, the variation in final percent infected varies greatly in each trial. From 10 to 1000 infection trials, the Pearson's

Coefficient value increased from 0.202 to 0.919. The path through which the infection travels has a large impact on the cascading failures within the modular networks.

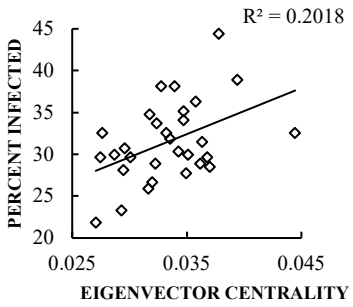

Figure 10: 10 epidemic trials

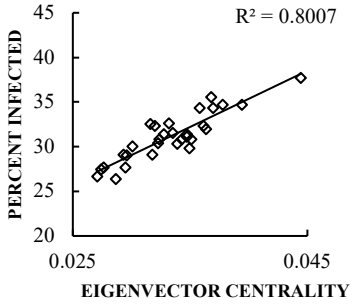

Figure 11: 100 epidemic trials

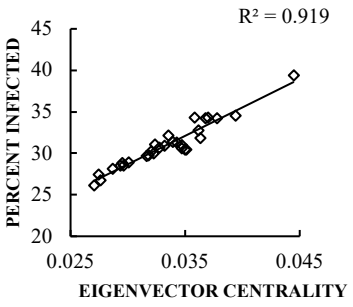

Figure 12: 1000 epidemic trials

Figure 13 depicts a synthetic network high in modularity with the infected nodes highlighted. Visualizing the infection spread in a network by highlighting infected nodes shows a higher concentration of infection in the module of origin. As the infection spreads and infects the entire module, chances of the failure cascading beyond the module to which it is contained increases. This figure highlights that the infection spread to a bridging node, and thus was able to continue its path through the network rather than being contained in one module.

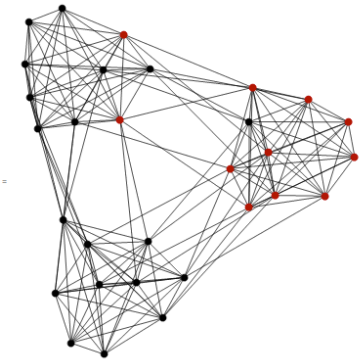

Figure 13: Visualization of infection in network with Q=0.5490

5 DISCUSSION

It has been shown that network modularity and a system's ability to resist cascading failures have a positive relationship. Networks with higher modularity are more resistant to a cascading failure. If a complex engineering system falls victim to a failure, the more modularized the system the more likely it is for the failure to be contained to the affected module. These findings agree with other studies on modularity as a means to resist an epidemic spread. Tightly connected clusters are shown to inhibit the Susceptible-Infected-Recovered phenomena [41]. Viruses can be annihilated faster by dividing networks into clusters; however, this comes at the cost of communication and reachability [42]. A fault will remain reasonably self-contained in one module if few edges between modules exist [43]. Mehrpouyan et al. show that networks with a higher degree will propagate failures faster [26]. Consider the power grid failure of 2003. There were redundancies in place to support the system, but the complex level of interconnected systems is what ultimately allowed the spread of the failure to over 50 million people in under eight minutes [1].

High modularity can serve as both a benefit and deficit in designing complex engineering systems. The more modular a system is, the more a failure can be contained; however, the lack of connections between modules can cause communication issues in the system. This also leaves key nodes susceptible to failures that could rapidly affect the entire system. System robustness is an important design decision. There is a tradeoff between modularity and robustness [31]. High modularity can penalize system performance [44].

The path through which an infection spreads has been shown to have a large impact on total infection within a network. The increasing Pearson's Coefficient shows that with just a few infection models, there is still much unpredictability. Only when a large number of trials at the same start node are performed can the average ending infection rate be seen. When an infection starts at a node with a higher eigenvector centrality, the infection on average infects a larger percentage of the network. For engineers, this means that systems must be designed to handle unpredictable cascading failure paths. Protecting the nodes that are more structurally important in the system could prevent detrimental failure propagation. Enabling engineers to make early design changes in modularity levels greatly reduces the potential cost of later adjustments. Furthermore, identifying a node with a higher eigenvector centrality provides the opportunity to implement protection measures in advance. This also allows for the premeditated analysis of likely cascading failure paths based on component relationships. Understanding early the implications of modularity and eigenvector centrality in design allows the integration of mitigation strategies into the system. This could be in the form of redundancies, sensors on bridging nodes, or a health management system. The benefits of modularizing a system are not universal; rather, they are extremely system-dependent [31]. If the goal of the system is increase containment of component failures, modularity is a helpful design factor. If the bridging nodes within the system are a high concern for failure, modularity is not a desirable feature.

6 CONCLUSION AND FUTURE WORK

The relationship between cascading failures in complex engineering systems and modularity has been demonstrated to be positive. In addition, the initial location of the failure has been shown to be crucial in predicting the impact of the failure. This connection serves as the first step towards understanding appropriate levels of modularity and protections strategies in complex system design. This research study approached the concept of modularity and cascading failures at a very broad level. Intuitively, conceptualizing a module as a very connected system with little connection to the outside world simplifies the understanding of why a component failure is easier to contain. If one of the few components connected to another module fails or if the module is highly connected to the other modules, the chance of spreading the failure increases. This study only analyzed systems with 30 nodes. Complex engineering systems, such as jet engines, can have hundreds of components. As part of future work to understand the ramifications of

modularity on a system's ability to resist failures, performing a specific case study would provide a clearer understanding of the problem in application. This would also allow for the observation of the specific path of the cascading failures through real components with functional connections, enabling an investigation of containment levels within modules to track the spread of infection. The explicit statement of modularity affecting cascading network failure assists in the continuation of research on robustness in system design. Failures in a system are inevitable; however, being able to contain and respond to the failures through the system architecture improve safety, reliability, and quality.

WORKS CITED

[1] "Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations," US Department of Energy, Washington DC, 2004.

[2] S. Valverde, "Breakdown of Modularity in Complex Networks," *Frontiers in Physiology,* vol. 8, no. 497, 2017.

[3] G. Paparistodimou, A. Duffy, R. I. Whitfield, P. Knight and M. Robb, "A network science-based assessment methodology for robust modular system architectures during early conceptual design," *Journal for Engineering Design,* vol. 31, no. 4, pp. 179-218, 2019.

[4] R. M. D'Souza, "Curtailing Cascading Failures," *Network Science,* vol. 358, no. 6365, pp. 860-861, 17 November 2017.

[5] M. Newman, Networks: An Introduction, New York: Oxford University Press, 2010.

[6] P. Hines, K. Balasubramaniam and E. Sanchez, "Cascading Failures in Power Grids," *IEEE Potentials,* vol. 28, no. 5, pp. 24-30, 2009.

[7] N. F. Soria, M. K. Colby, I. Y. Tumer, C. Hoyle and K. Tumer, "Design of Complex Engineering Systems Using Multiagent Coordination," in *42nd Design Automation Conference*, Charlotte, 2016.

[8] C. Bloebaum and A. McGowan, "Design of complex engineered systems," *Journal of Mechanical Design,* vol. 132, no. 12, pp. 120301-120301, 2010.

[9] P. Liu, Q. Zhang, X. Yang and L. Yang, "Passivity and optimal control of descriptor biological complex systems," *IEEE Transactions on Automatic Control,* vol. 53, no. Special Issue, pp. 122-125, 2008.

[10] B. M. Haley, *Evaluating Complex Engineering Systems using Complex Network Representations,* Corvallis, OR, 2014.

[11] K. Lewis, "Making sense of elegant complexity in design," *Journal of Mechanical Design,* vol. 134, no. 12, pp. 120801-120801, 2012.

[12] L. K. Comfort, "Self-Organization in Complex Systems," *Journal of Public Administration Research and Theory: J-PART,* vol. 4, no. 3, pp. 393-410, 1994.

[13] M. Newman, "Analysis of weighted networks," *Physical Review E,* vol. 70, no. 52, pp. 056131-1-056131-9, 2004.

[14] M. Mitchell, "Complex Systems: Network thinking," *Artificial Intelligence,* vol. 170, no. 18, pp. 1194-1212, 2006.

[15] M. Wang, W. Chen, Y. Huang, N. Contractor and Y. Fu, "A Multidimensional Network Approach for Modeling Customer-Proudct Relations in Engineering Design," in *ASME 2015 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Boston, 2015.

[16] P. Htet Hein, B. Morkos and C. Sen, "Utilizing Node Interference Method and Complex Network Centrality Metrics to Explore Requirement Change Propagation," in *Proceedings of the ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 1:37th Computers and Information in Engineering Conference*, Cleveland, 2017.

[17] H. Walsh, A. Dong and I. Tumer, "The role of bridging nodes in behavioral network models of complex engineered systems," *Design Science,* vol. 4, no. 8, 2018.

[18] P. Newcomb, B. Bras and D. Rosen, "Implications of modularity on product design for the life cycle," *Journal of Mechanical Design,* vol. 120, no. 3, pp. 483-490, 1998.

[19] M. E. J. Newman, "Modularity and Community Structure in Networks," *PNAS,* vol. 103, no. 23, pp. 8577-8582, 6 June 2006.

[20] B. M. Haley, A. Dong and I. Y. Tumer, "Creating Faultable Network Models of Complex Engineered Systems," in *40th Design Automation Conference*, Buffalo, 2014.

[21] P. Hines, E. Cotilla-Sanchez and S. Blumsack, "Do topological models provide good information about electricity infrastructure vulnerability?," *Chaos: An Interdisciplinary Journal of Nonlinear Science,* vol. 20, no. 3, 2010.

[22] P. Crucitti, V. Latora and M. Marchiori, "Model for cascading failures in complex networks," *Phsical Review E,* vol. 69, no. 4 pt 2, p. 045104, 2004.

[23] J.-W. Wang and L.-L. Rong, "Robustness of the western United States power grid under edge attack strategies due to cascading failures," *Safety Science,* vol. 49, no. 6, pp. 807-812, 2011.

[24] S. N. Talukdar, J. Apt, M. Ilic, L. B. Lave and M. G. Morgan, "Cascading Failures: Survival versus Prevention," *The Electricity Journal,* vol. 16, no. 9, pp. 25-31, 2003.

[25] H. Anh and A. Namatame, "Mitigation of Cascading Failures with Link Weight Control," *International Journal of Advanced Computer Science and Applications,* vol. 5, no. 7, 2014.

[26] H. Mehrpouyan, B. Haley, A. Dong, I. Y. Tumer and C. Hoyle, "Resilient Design of Complex Engineered Systems Against Cascading Failure," in *Proceedings of the ASME 2013 International Mechanical Engineering Congress and Exposition. Volume 12: Systems and Design.*, San Diego, 2013.

[27] M. Babaei, H. Ghassemieh and M. Jalili, "Cascading Failure Tolerance of Modular Small-World Networks," *IEEE Transactions on Circuits and Systems II: Express Briefs,* vol. 58, no. 8, pp. 527-531, 2011.

[28] H. S. Walsh, *Designing for Robustness: The Role of Bridging Nodes,* Corvallis, OR, 2018.

[29] S. Sarkar and A. Dong, "Characterizing Modularity, Hierarchy and Module Interfacing in Complex Design Systems," in *Proceedings of the ASME 2011 International Design Engineering Technical Conferences and Computer and Information in Engineering Conference.*, Washington DC, 2012.

[30] D. Kasthurirathna, A. Dong, M. Piraveenan and I. Y. Tumer, "The Failure Tolerance of Mechatronic Software Systems to Random and Targeted Attacks," in *Proceedings of the ASME Design Engineering Technical Conference*, Portland, 2013.

[31] H. S. Walsh, A. Dong and I. Y. Tumer, "An Analysis of Modularity as a Design Rule Using Network Theory," *Journal of Mechanical Design,* vol. 141, no. 3, 2019.

[32] X. Guan and C. Chen, "General methodology for inferring failure-spreading dynamics in networks," *Proceedings of the National ACademy of Sciences,* vol. 115, no. 35, pp. E8125-E8134, 2018.

[33] L. Valdez, H. H. A. Rêgo, H. E. Stanley, S. Havlin and L. A. Braunstein, "The role of bridge nodes between layers on epidemic spreading," *New Journal of Physics,* vol. 20, no. 12, 2018.

[34] L. Sattenspiel, "Modeling the Spread of Infectious Disease in Human Populations," *American Journal of Physical Anthropology,* vol. 33, no. S11, pp. 245-276, 1990.

[35] R. Anderson and R. Mary, Infectious Diseases of Humans, Oxford: Oxford University Press, 1991.

[36] H. Mehrpouyan, B. Haley, A. Dong and I. Y. Tumer, "Resiliency Analysis for Complex Engineered System Design," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* vol. 29, no. 1, pp. 93-108, 2015.

[37] N. Bailey, The Mathematical Theory of Infectious Diseases and Its Applications, New York: Hafner Press, 1975.

[38] P. Bonacich, "Power and Centrality: A famliy of measures," *American Journal of Sociology,* vol. 92, no. 5, pp. 1170-1182, 1987.

[39] P. Bonacich, "Some unique properties of eigenvector centrality," *Social Networks,* vol. 29, no. 4, pp. 555-564, 2007.

[40] J. Kenney and E. Keeping, Mathematics of Statistics, Princeton, NJ: Van Nostrand, 1951.

[41] M. Nadini, K. Sun, E. Ubaldi, M. Sarnini, A. Rizzo and N. Perra, "Epidemic spreading in modular time-varying networks," *Scientific Reports,* vol. 8, no. 2352, 2018.

[42] J. Omíc, J. Martín-Hernández and P. Van Mieghem, "Network protection against worms and cascading failures using modularity partitioning," in *2010 22nd International Teletraffic Congress (ITC 22)*, Amsterdam, 2010.

[43] H. Kitano, "Biological Robustness," *Nature Reviews Genetics,* vol. 5, no. 11, pp. 826-837, 2004.

[44] K. Hölttä, E. Suh and O. de Weck, "Trade-off between modularity and performance for engineered systems and products," in *Proceedings of 15th International Conference on Engineering Design ICED'05*, Melbourne, Australia, 2005.

APPENDICES

Appendix A: Data collected from Mathematica

Table 2: Initial Data Collection

PCM	PCI	PI	Q modularity	% Infected
1	0.25	0.3	0.315701	96.6667
1	0.2	0.3	0.331639	93.3333
1	0.15	0.3	0.40839	90
1	0.1	0.3	0.46539	60
1	0.05	0.3	0.578707	60
0.75	0.25	0.3	0.308962	96.6667
0.75	0.2	0.3	0.273692	86.6667
0.75	0.15	0.3	0.323146	50
0.75	0.1	0.3	0.463092	33.3333
0.75	0.05	0.3	0.563057	23.3333
0.5	0.25	0.3	0.165175	73.333
0.5	0.2	0.3	0.240467	33.333
0.5	0.15	0.3	0.280277	60
0.5	0.1	0.3	0.376286	60
0.5	0.05	0.3	0.527634	30
0.25	0.25	0.3	0.231124	70
0.25	0.2	0.3	0.234211	43.3333
0.25	0.15	0.3	0.292533	36.6667
0.25	0.1	0.3	0.368609	20
0.25	0.05	0.3	0.445176	26.6667

Table 3: Data collection with edges counted

PCM	PCI	PI	Q modularity	% Infected	Edges
1	0.25	0.3	0.315701	96.6667	208
1	0.2	0.3	0.331639	93.3333	203
1	0.15	0.3	0.40839	90	182
1	0.1	0.3	0.46539	60	168
1	0.05	0.3	0.578707	60	148
0.75	0.25	0.3	0.308962	96.6667	165
0.75	0.2	0.3	0.273692	86.6667	167
0.75	0.15	0.3	0.323146	50	138
0.75	0.1	0.3	0.463092	33.3333	124
0.75	0.05	0.3	0.563057	23.3333	116
0.5	0.25	0.3	0.216827	73.3333	132
0.5	0.2	0.3	0.245097	83.3333	112
0.5	0.15	0.3	0.320829	66.6667	114
0.5	0.1	0.3	0.308199	46.6667	95
0.5	0.05	0.3	0.45231	26.6667	84
0.25	0.25	0.3	0.2723	73.3333	100
0.25	0.2	0.3	0.254538	53.3333	93
0.25	0.15	0.3	0.302978	40	76
0.25	0.1	0.3	0.361212	23.3333	62
0.25	0.05	0.3	0.475792	10	53

Table 4: Data collection with edges held constant

PCM	PCI	PI	Q modularity	% Infected	Edges
1	0.25	0.3	0.305398	100	207
0.95	0.275	0.3	0.203419	93.3333	220
0.9	0.3	0.3	0.242095	100	203
0.85	0.325	0.3	0.191669	80	204
0.8	0.35	0.3	0.196823	66.6667	212
0.75	0.375	0.3	0.151444	90	212
0.7	0.4	0.3	0.10937	86.6667	219
0.65	0.425	0.3	0.118934	100	190
0.6	0.45	0.3	0.10751	100	216
0.55	0.475	0.3	0.110691	100	218
0.5	0.5	0.3	0.144046	100	208
0.45	0.525	0.3	0.100847	96.6667	229
0.4	0.55	0.3	0.113503	96.6667	213
0.35	0.575	0.3	0.117955	100	213
0.3	0.6	0.3	0.105604	93.3333	201
0.25	0.625	0.3	0.085739	96.6667	239
0.2	0.65	0.3	0.0867495	100	241
0.15	0.675	0.3	0.0948495	96.6667	227
0.1	0.7	0.3	0.0857579	93.3333	221
0.05	0.725	0.3	0.0851138	100	236

Table 5: Data collection with $p_{c,m}$>$p_{c,i}$, PI=0.1

PCM	PCI	PI	Q modularity	% Infected	Edges
1	0.25	0.1	0.318806	20	207
0.975	0.2625	0.1	0.293274	26.6667	209
0.95	0.275	0.1	0.289528	13.3333	207
0.925	0.2875	0.1	0.245564	16.6667	209
0.9	0.3	0.1	0.27052	16.6667	207
0.875	0.3125	0.1	0.22565	36.6667	200
0.85	0.325	0.1	0.198999	36.6667	216
0.825	0.3375	0.1	0.161769	40	210
0.8	0.35	0.1	0.165136	40	210
0.775	0.3625	0.1	0.19007	33.3333	212
0.75	0.375	0.1	0.143422	33.3333	216
0.725	0.3875	0.1	0.108444	36.6667	217
0.7	0.4	0.1	0.123319	50	221
0.675	0.4125	0.1	0.132101	53.3333	204
0.65	0.425	0.1	0.111279	50	225
0.625	0.4375	0.1	0.11357	26.6667	199
0.6	0.45	0.1	0.112127	60	230
0.575	0.4625	0.1	0.119619	56.6667	213
0.55	0.475	0.1	0.0957278	50	230
0.525	0.4875	0.1	0.11269	60	218

Table 6: Data collection with % Infected looped 1000 times

PCM	PCI	PI	Q modularity	% Infected	Edges
1	0.25	0.1	0.312527	33.8133	209
0.975	0.2625	0.1	0.283705	33.85	209
0.95	0.275	0.1	0.270216	34.28	207
0.925	0.2875	0.1	0.250143	34.01	209
0.9	0.3	0.1	0.231406	34.4833	216
0.875	0.3125	0.1	0.188458	35.06	221
0.85	0.325	0.1	0.185005	35.0033	213
0.825	0.3375	0.1	0.153563	35.1267	213
0.8	0.35	0.1	0.154169	35.1967	207
0.775	0.3625	0.1	0.146383	35.34	210
0.75	0.375	0.1	0.137872	35.8067	235
0.725	0.3875	0.1	0.13034	35.5433	210
0.7	0.4	0.1	0.111756	36.1933	211
0.675	0.4125	0.1	0.125389	36.1867	212
0.65	0.425	0.1	0.113094	35.4567	217
0.625	0.4375	0.1	0.0943271	36.1467	228
0.6	0.45	0.1	0.10828	35.6767	221
0.575	0.4625	0.1	0.103662	36.5267	231
0.55	0.475	0.1	0.103895	35.7833	220
0.525	0.4875	0.1	0.107355	36.027	220

Table 7: Eigenvector centrality and percent infected from targeted node attacks for varying infection trials

		Trials for Percent Infected Avg		
Node	**EigCent**	**10**	**100**	**1000**
1	0.0327373	38.1481	31.3805	30.7241
2	0.0274257	29.6296	27.4411	27.4875
3	0.0276047	32.5926	27.6768	26.7434
4	0.0295861	30.7407	29.0572	28.5085
5	0.034706	35.1852	31.3805	31.0244
6	0.0294764	28.1481	27.6768	28.8155
7	0.0320046	26.6667	32.3569	30.2366
8	0.0444492	32.5926	37.7441	39.3994
9	0.0369763	28.5185	34.3434	34.3143
10	0.0322513	28.8889	30.3704	30.01
11	0.031631	25.9259	32.5589	29.6864
12	0.0293184	23.3333	29.0909	28.5552
13	0.0286718	30	26.3636	28.1315
14	0.0317813	34.8148	29.0909	29.7731
15	0.0323734	33.7037	30.8418	31.041
16	0.0270567	21.8519	26.7003	26.1495
17	0.0335207	31.8519	31.5825	32.1622
18	0.0367671	29.6296	35.5892	34.2509
19	0.0357872	36.2963	34.3771	34.3477
20	0.0300808	29.6296	30.0337	28.9256
21	0.0363287	31.4815	31.9529	31.8719
22	0.03496	27.7778	29.798	30.4538
23	0.0377724	44.4444	34.6801	34.2242
24	0.0342604	30.3704	30.8418	31.3113
25	0.0393813	38.88889	34.6801	34.5813
26	0.0361533	28.8889	32.3232	32.7494
27	0.0347172	34.0741	31.1111	30.6273
28	0.0332081	32.5926	32.6263	30.9142
29	0.0339183	38.1481	30.303	31.3714
30	0.0350945	30	30.8081	30.4705