NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

DISSERTATION

SPECTRAL GRAPH-BASED CYBER DETECTION AND CLASSIFICATION SYSTEM WITH PHANTOM COMPONENTS

by

Jamie L. Safar

December 2020

Dissertation Supervisors: John C. McEachen
Murali Tummala

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 2020	Dissertation

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
SPECTRAL GRAPH-BASED CYBER DETECTION AND CLASSIFICATION SYSTEM WITH PHANTOM COMPONENTS	REPJH
6. AUTHOR(S) Jamie L. Safar	

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Naval Postgraduate School Monterey, CA 93943-5000	

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING / MONITORING AGENCY REPORT NUMBER
Laboratory for Telecommunications Science, Ft. George G. Meade, MD 20755	

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
Approved for public release. Distribution is unlimited.	A

13. ABSTRACT (maximum 200 words)

With cyber attacks on the rise, cyber defenders require new, innovative solutions to provide network protection. We propose a spectral graph-based cyber detection and classification (SGCDC) system using phantom components, the strong node concept, and the dual-degree matrix to detect, classify, and respond to worm and distributed denial-of-service (DDoS) attacks. The system is analyzed using absorbing Markov chains and a novel Levy-impulse model that characterizes network SYN traffic to determine the theoretical false-alarm rates of the system. The detection mechanism is analyzed in the face of network noise and congestion using Weyl's theorem, the Davis-Kahan theorem, and a novel application of the n-dimensional Euclidean metric. The SGCDC system is validated using real-world and synthetic datasets, including the WannaCry and Blaster worms and a SYN flood attack. The system accurately detected and classified the attacks in all but one case studied. The known attacking nodes were identified in less than 0.27 sec for the DDoS attack, and the worm-infected nodes were identified in less than one second after the second infected node began the target search and discovery process for the WannaCry and Blaster worm attacks. The system also produced a false-alarm rate of less than 0.005 under a scenario. These results improve upon other non-spectral graph systems that have detection rates of less than 0.97 and false alarm rates as high as 0.095 for worm and DDoS attacks.

14. SUBJECT TERMS	15. NUMBER OF PAGES
spectral graph detection, spectral graph-based cyber detection and classification, SGCDC, phantom component, strong node concept, dual-degree matrix, worm attack, distributed denial-of-service, DDoS, absorbing Markov chains, Levy-impulse model, SYN flood attack	179
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

SPECTRAL GRAPH-BASED CYBER DETECTION AND CLASSIFICATION SYSTEM WITH PHANTOM COMPONENTS

Jamie L. Safar
Lieutenant Commander, United States Navy
BA, University of California, San Diego, 2006
MS, Electrical Engineering, Naval Postgraduate School, 2014

Submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 2020

Approved by:	John C. McEachen Department of Electrical and Computer Engineering Dissertation Supervisor	Murali Tummala Department of Electrical and Computer Engineering Dissertation Supervisor
	Chad A. Bollmann Department of Electrical and Computer Engineering Department	Ralucca Gera Department of Graduate Education
	John D. Roth Department of Electrical and Computer Engineering	Murali Tummala Department of Electrical and Computer Engineering Dissertation Chair
Approved by:	Douglas J. Fouts Chair, Department of Electrical and Computer Engineering	
	Orrin D. Moses Vice Provost of Academic Affairs	

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

With cyber attacks on the rise, cyber defenders require new, innovative solutions to provide network protection. We propose a spectral graph-based cyber detection and classification (SGCDC) system using phantom components, the strong node concept, and the dual-degree matrix to detect, classify, and respond to worm and distributed denial-of-service (DDoS) attacks. The system is analyzed using absorbing Markov chains and a novel Levy-impulse model that characterizes network SYN traffic to determine the theoretical false-alarm rates of the system. The detection mechanism is analyzed in the face of network noise and congestion using Weyl's theorem, the Davis-Kahan theorem, and a novel application of the n-dimensional Euclidean metric. The SGCDC system is validated using real-world and synthetic datasets, including the WannaCry and Blaster worms and a SYN flood attack. The system accurately detected and classified the attacks in all but one case studied. The known attacking nodes were identified in less than 0.27 sec for the DDoS attack, and the worm-infected nodes were identified in less than one second after the second infected node began the target search and discovery process for the WannaCry and Blaster worm attacks. The system also produced a false-alarm rate of less than 0.005 under a scenario. These results improve upon other non-spectral graph systems that have detection rates of less than 0.97 and false alarm rates as high as 0.095 for worm and DDoS attacks.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

DDoS	distributed denial-of-service
DNS	domain name server
DoS	denial-of-service
GMT	Greenwich Mean Time
IP	internet protocol
IPv4	internet protocol version 4
MAWI	Measurement and Analysis on the WIDE Internet
MLE	maximum likelihood estimation
NPS	Naval Postgraduate School
PDF	probability distribution function
SGCDC	spectral graph-based cyber detection and classification
SI	susceptible-infectious
SIR	susceptible-infectious-recovered
SIS	susceptible-infectious-susceptible
SYN	synchronize
TCP	transmission control protocol
TDG	traffic dispersion graph
UDP	user datagram protocol

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgments

First and foremost, I give all honor and credit to God Almighty, who gave me the strength, discipline, and intellect to complete this research. Second, I am truly grateful for the love and support from my amazing husband, Branimir Safar. I would not have been able to complete this process without you! Finally, I would like to thank all of my dissertation committee members. You provided me with inspiration and encouragement and continually fostered my intellectual growth through this entire process. To all of you, I am truly grateful for and humbled by your belief in my abilities!

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

Despite continuous efforts to improve cybersecurity [1], cyberspace remains a vulnerable domain to a variety of attacks. In 2017, over 200,00 hosts across 150 countries experienced the devastating effects of the WannaCry ransomware cryptoworm [2]. In 2018, 4,800 websites were compromised each month by formjacking attacks, ransomware attacks increased by 12 percent, supply chain attacks increased by 78 percent, over 70 million cyber records were stolen from simple storage service buckets, phishing attacks continued to increase, and an increase was observed in extortion cases through cryptocurrency [3], [4]. In 2019, there were 7.2 billion malware attacks [5], successful phishing attacks increased by 25 percent [6], and cryptocurrency attacks increased by 29 percent in the first quarter alone [7]. Yet, in the face of a growing attack environment, the number of internally detected compromises continues to decrease, primarily because cyber attackers have an advantage over cyber defenders [4], [8]. Cyber attackers maintain the upper hand by controlling not only the target of the attack, but also the timing, tempo, and methodologies employed. Cyber defenders are left with the impossible task of predicting all potential attack vectors in an effort to proactively defend their network [9].

Seeking innovative solutions to shift the status quo toward or in favor of the cyber defender, we turned to biomimicry, the formal study of nature to develop technological or engineering solutions [10]. We examined biological defense mechanisms for relevant approaches and strategies because several similarities exist between the cyber and biological attack-defense domains [11], [12]. Since we live in a current environment where cyber attackers cannot be eradicated, the biological immune system stood out due to its principle of restoring balance in the attack-defense domain through immunity instead of eradication of a threat [11].

The vertebrate biological immune system is responsible for quickly and efficiently detecting, identifying, and eliminating various harmful pathogens in an animal including viruses, bacteria, and parasites. It is composed of an innate and an adaptive immune system. The innate immune system is responsible for detecting the harmful pathogens and signaling their presence to the adaptive immune system. The adaptive immune system is responsible for choosing and employing the correct immune response [13], as shown in Figure 1.1.

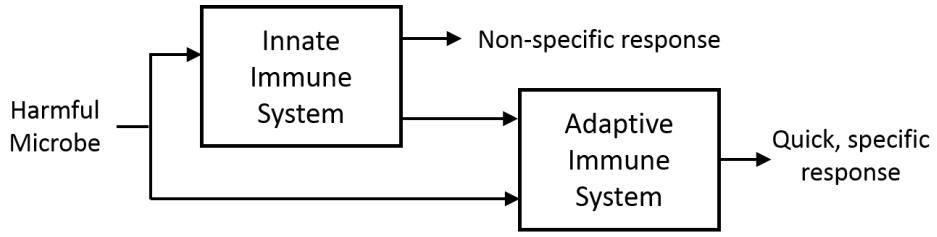


Figure 1.1. Subsystems of the biological immune system. Adapted from [11] and [13].

Working together, the subsystems perform four critical functions to enable immunity within an animal. First, the immune system recognizes the intruder through antibodies and immune cell receptors distributed throughout the body. Then, either an antibody binds to the pathogen to neutralize the antigen or a T cell kills the host cell. In either case, the pathogen is unable to further replicate. Next, the system learns to create and produce antibodies and T cell receptors that can quickly recognize that specific pathogen in the future in order to generate a quicker response. Finally, the system stimulates the reproduction of the immune cells and receptors that best match known pathogens in the memory of the system. As a result, the system contains the following critical properties: recognition of intruders, elimination or neutralization of intruders, ability to learn, ability to distinguish self from non-self, and system memory [14]. With the biological immune system in mind, we seek a path to implement the critical properties of the adaptive immune system in a cyber defensive system using spectral graph theory-based techniques.

1.1 Objective

The objective of this work is the development of a spectral graph-based cyber detection and classification (SGCDC) system capable of implementing the critical functions of the adaptive immune system against malicious worm and distributed denial-of-service (DDoS) attacks using spectral graph theory tools like phantom components, the strong node concept, and the dual-degree matrix. As depicted in Figure 1.2, the proposed system uses historical attack patterns coupled with real-time network traffic to detect malicious attack behaviors and trigger quick, specific countermeasures, just as the biological immune system uses information from the innate immune system coupled with the existence of a harmful microbe to trigger a quick, specific response. Accordingly, the proposed system must implement

components that identify, classify, and provide responsive actions against worm and DDoS attacks in real time using spectral graph theory tools and analysis.

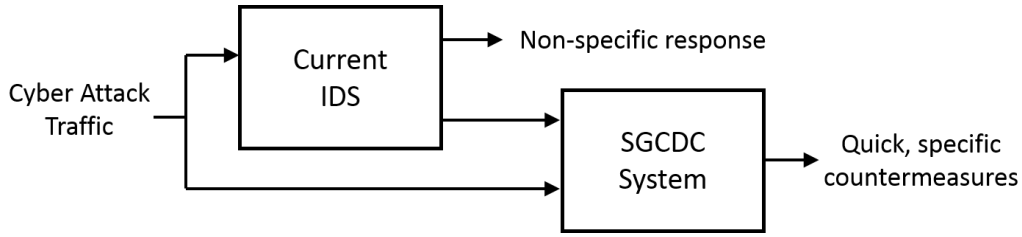


Figure 1.2. Biological immune system-inspired cyber defense system.

The objective of our approach stems from a comprehensive review of the literature, which indicated that four components are critical in developing a cyber system that mimics the biological immune system: behavior sensing, immunological memory, adaptive pattern recognition, and response [15]. The five processes of our design approach, depicted in Figure 1.3, align with these four critical components and provide the initial framework for developing the overall solution presented in Chapter 3. Specifically, the design approach makes use of a traffic dispersion graph (TDG) to perform network behavior sensing, phantom components to store behavioral characteristics of historical attacks, the strong node concept to recognize and detect the attack behavior, a dual-degree matrix to classify the detected attack behavior, and network countermeasures to perform response. The system proposed in this design approach is possible because each type (or category) of cyber attack has distinct core characteristics that distinguish it from other types of attacks [16]. Identifying those core characteristics and setting them as the attack behavior memory (i.e., phantom component), allows the use of spectral graph tools to detect the attack and trigger a quick and effective network response.

As part of the development process, the system must also be analyzed and evaluated to determine performance and limitations. Since the system relies on real-time network traffic for detection, the detection capabilities of the system must be analyzed under both ideal and non-ideal network conditions. For ideal network conditions, an absorbing Markov chain and a model of the identified network traffic feature associated with worm and DDoS attack behaviors will be developed to describe the proposed system for the purpose of analyzing false alarm rates. For non-ideal network conditions, the system detection capability will be analyzed using both traditional and novel eigenvalue and eigenvector

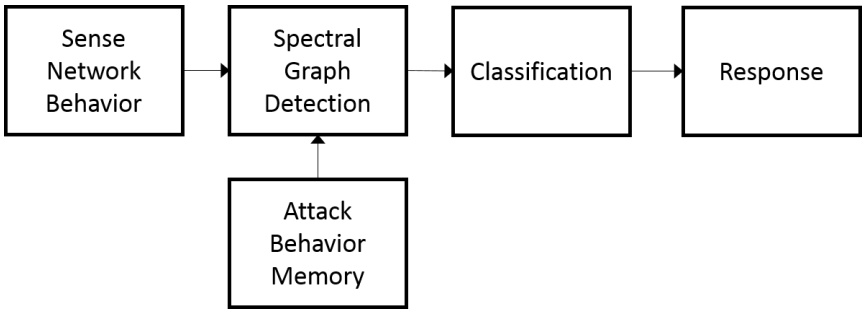


Figure 1.3. Design approach for SGCDC system.

perturbation techniques. Finally, the system detection capability to detect, classify, and respond to worm and DDoS attacks needs to be validated to determine detection and classification rates, detection times, false alarm rates, and response performance. This will be conducted using both real world and modeled datasets that contain normal and attack traffic.

With respect to design and implementation, we apply two constraints to the proposed SGCDC system. First, the SGCDC system is restrained to the detection and classification of scan-based transmission control protocol (TCP) worm attacks and synchronize (SYN) flood DDoS attacks. While this restraint is implemented for the research in this dissertation, the proposed SGCDC system has the potential to detect other forms of worm and DDoS attacks, along with other forms of cyber attacks, with minor modifications. Second, each node being analyzed by the SGCDC system is restrained to a single attack type. In other words, a node can be either normal, under worm attack, or under DDoS attack. A node cannot be under worm and DDoS attack simultaneously.

1.2 Related Work

There is no precedent for a biological immune system-inspired cyber defense system that employs spectral graph-based anomaly detection; however, there are six categories in literature that were influential to both the development of our solution approach and analysis of our proposed system: biological immune system-inspired cyber systems, graph theory-based anomaly detection, spectral graph theory-based anomaly detection, network anomaly detection metrics, traffic modeling, and perturbation analysis of eigenvectors. This section reviews the significant work from each of these categories that contributed to our work.

1.2.1 Biological Immune System-Inspired Cyber Systems

While a significant amount of research has been conducted on biological immune system-inspired cyber defense ([17]–[19], among others), we leaned heavily on material from [14] and [20] to understand the mechanics of and draw inspiration from a biological immune system-inspired cyber defense system. The work by Kephart in [14] produced one of the first biological immune system-inspired cyber defense systems. His host-based cyber defense approach uses system antibodies to remember previously encountered viruses and worms in an effort to quickly recognize and respond to future encounters of the same attack. The system antibodies are developed by using decoy programs to capture samples of signatures from the previously encountered virus or worm.

The network-based cyber defense approach by Hofmeyr and Forrest in [20] uses similar signature-based detection in their biological immune system-inspired cyber defense system. Their approach applies an artificial immune system to LISYS, a network intrusion detection system, in an attempt to mimic the processes observed in the biological immune system. Specifically, they utilize graph theory to model a local set of detectors, each of which is associated with a binary string for detection and can only interact with its directly connected neighbors. The system utilizes the detectors to perform binary string matching via an r-contiguous rule to identify non-self binary strings.

While both works produce good results, their systems do not perform optimally in the face of zero-day attacks. In [14], the cyber defense system develops system antibodies only if an anomaly is detected. As a result, the system will not develop antibodies until after the zero-day attack has occurred. In [20], the cyber defense system requires a human operator to identify the first occurrence of non-self for zero-day attacks of any form. As a result, we intend to employ modified portions of both systems in our solution approach. Specifically, we seek to modify the concept behind the antibody development process in [14] and the r-contiguous bit matching rule in [20] to transform historical attack patterns into a graphical attack model that can be utilized to form phantom component for behavioral detection instead of signature detection. As a result, we seek to increase the likelihood of the proposed SGCDC system to detect zero-day attacks through the implementation of these phantom components since the detection is not based on a previously encountered binary string but rather the intrinsic attack properties and behaviors that are critical across the different variations of the attack type. We also intend to adapt the concept of utilizing

human operators in [20] to develop the phantom component from historical attack records.

1.2.2 Graph Theory-Based Anomaly Detection

Graph theory has not been widely applied to solving anomaly detection problems. Le et al. [21] and Ellis et al. [22] utilized graph theory-based methods to study anomaly detection. The work in [21] uses a TDG to model network traffic and graph theory metrics to perform anomaly detection. The system then utilizes a different mechanism, the VF2 graph matching algorithm, to compare the graphical representation of the anomalous flow to known anomalous graphical representations for the purpose of classifying the anomaly. While the method is successful, it assumes that the system has all known attack patterns without providing any explanation as to how zero-day attack patterns are accounted for. Additionally, the system incurs increased computational overhead by performing two distinct mathematical processes for detection and classification. As a result, we only intend to adopt the use of TDGs as a method of graphically representing network traffic. We also seek to improve upon this work by using graph theory tools to decrease the computational overhead required to perform both detection and classification.

The work in [22] uses a specific graphical behavior signature of computer worm propagation to detect worm activity. Specifically, the worm detection algorithm employed by their system looks for worm causal-tree behavior via various graph metric thresholds (e.g., branching factor and depth). While their proposed solution is effective and efficient in identifying the presence of a worm attack, it fails to inherently identify the specific nodes infected by the attack without additional analysis and/or computation. As a result, our solution approach intends to adapt and expand upon their graphical behavior signature for worms by applying the concept to other types of attacks to develop phantom components that graphically describe the attack behavior. Our proposed system also seeks to improve upon their work by using a spectral graph detection process that inherently identifies not only the infected nodes but also the potential victims being targeted by the infected node.

1.2.3 Spectral Graph Theory-Based Anomaly Detection

Johnson [23] utilizes spectral graph-based tools to detect congestion in the physical layer of a software-defined network. The detection process uses a phantom node to produce a threshold in the eigenspectrum to identify nodes that are both under-utilized and over

congested for the purpose of load balancing. While the research provides non-specific immediate responses to network traffic anomalies, it cannot differentiate between normal network congestion and congestion brought about by attack. Additionally, it cannot detect attacks that do not produce network congestion. However, the work does identify through simulation a potential relationship between node connectivity and the eigenvector magnitude values, which is utilized in the detection mechanism. As a result, we intend to expand upon this spectral graph detection technique in our proposed system by defining and providing a mathematical foundation for the observed relationship. We also intend to expand upon the phantom node concept to develop phantom components for the purpose of detecting anomalies within the network logical flows at the transport layer.

1.2.4 Network Anomaly Detection Metrics

A significant amount of research exists on network anomaly detection techniques for worm ([22], [24]–[30], among others) and DDoS attacks ([31]–[34], among others); however, the vast majority of detection techniques proposed in literature are specific to one type of network attack. Additionally, a significant number of works fail to provide key empirical metrics that describe the capability and performance of their proposed detection system or technique. Table 1.1 lists the detection metrics that we found in literature for methods designed to detect both worm and denial-of-service (DoS) attacks. We seek to improve upon the detection and false alarm rates, and provide detection times, with our proposed SGCDC system.

Table 1.1. Empirical metrics in literature for methods designed to detect both worm and DoS attacks.

Source	Year	Target	Method	Detection Rate	Detection Time	False Alarm Rate
[35]	2004	DoS, probe, worms, etc.	Simplified Mahalanobiological immune system distance	> 0.526	–	< 0.01
[26]	2005	DoS, probe, worms, alpha flows, etc.	Multiway-subspace methods using feature entropy	0.8 (worm) 1.0 (DoS)	–	–
[29]	2016	DDoS, worms, botnets	Collaborative intrusion prevention architecture with neural net	0.92 – 0.9667 (worm) 0.8 – 0.9017 (DDoS)	–	0 – 0.095 (worm) 0 – 0.025 (DDoS)

1.2.5 Traffic Modeling

A significant amount of research [36]–[38] has been conducted to demonstrate that traffic features with high variance and asymmetry cannot be accurately modeled using traditional distributions, like Poisson and Gaussian. In [38]–[41], the stable distribution has been proposed to overcome the limitations observed in the traditional distributions for modeling network traffic features. In [42]–[46], mixture modeling has been proposed to improve model accuracy and outcomes. In fact, the mixture modeling truncation approach in [43]–[45] has been proposed in the financial world to capture the heavy tails of value-at-risk and asset return models. While both the stable distribution and the truncation approaches proposed in literature account for heavy-tailed traffic features, they do not account for traffic with extremely large zero-counts in combination with heavy-tails. As a result, we intend to develop a novel mixture model for traffic that has both a high zero-count and a heavy tail, for the purpose of analyzing our proposed system under ideal network conditions.

1.2.6 Eigenvector Perturbations

A significant amount of research has been conducted on perturbation analysis of the eigenvectors of a Hermitian matrix. Davis and Kahan [47] is prominent and foundational within this field of research. They developed a fundamental theorem describing the rotation of eigenvectors resulting from a perturbation in the Hermitian matrix. Since their work, several works [48]–[53], among others, have proposed variations or improvements to Davis and Kahan's work. Still, none of the works in literature analyze the distance between the perturbed and unperturbed vectors of the eigenvector matrix, specifically with respect to the nodal basis. As a result, we intend to expand upon the research in literature by applying the n-dimensional Euclidean distance formula to the n-dimensional vector of a single node within the nodal basis to analyze the distance between the perturbed and unperturbed vectors for the purpose of analyzing our proposed system under non-ideal network conditions.

1.3 Organization

The remainder of the dissertation is organized as follows. Background on topics and concepts that are necessary to describe and perform analysis of our proposed system are contained in Chapter 2. Our solution approach with detailed descriptions of each component are provided in Chapter 3. Analysis of the proposed system under ideal and non-ideal network conditions

is contained in Chapters 4 and 5. Chapter 6 provides validation results from a simulated implementation of the proposed SGCDC system. Chapter 7 concludes the dissertation with a summary of the work and contributions in this dissertation, along with recommendations for future research. There are two appendices. Appendix A contains the derivations of state transitions discussed in Chapter 4. Appendix B includes additional results from the false alarm rate analysis of the proposed system in Chapters 4 and 5.

The contents of this dissertation include material adapted from work to be published by the author. Sections from Chapter 3 contain some revised material from "Spectral Graph-based Cyber Worm Detection Using Phantom Components and Strong Node Concept" by Jamie Safar, Murali Tummala, and John McEachen, to be published in the 54th Hawaii International Conference on System Sciences. Sections from Chapter 4 contain some revised material from "A Novel Lévy-Impulse Mixture Based Connection Model for Computer Network Traffic" by Jamie Safar, Chad Bollmann, Murali Tummala, and John McEachen, to be published in the 14th International Conference on Signal Processing and Communication Systems.

CHAPTER 2:
Background

In this chapter, we present background material necessary to describe our proposed system. First, we introduce graph theory for modeling networks and network behaviors. Next, we explore spectral graph theory as a method of network analysis and behavior detection. Then, we examine the characteristics of worm and DDoS attacks to develop an understanding of the attack behavior the system must detect. Next, we utilize the attack characteristics to discuss network feature modeling as a means of developing a baseline of normal network behavior. Then, we discuss pertinent dual-basis perturbation theory to support the analysis of the system. Finally, we explore network countermeasure modeling as a means of evaluating system responses.

2.1 Graph Theory

The objective of this body of work is to develop a cyber defense system that utilizes spectral graph theory tools for detection and classification of worm and DDoS attacks. Thus, background on graph theory is necessary to explain the spectral graph techniques employed by the proposed SGCDC system. We begin this section with a discussion of graph theory terminology utilized in this work to model networks. We conclude this section with a review of the TDGs, a graph theory approach to model network traffic flows.

2.1.1 Terminology

The fundamentals of graph theory described in [54]–[56] are restated here in the notation adopted throughout this dissertation. A graph G is described as

$$G = (N, L, W), \tag{2.1}$$

where N is the set of nodes, L is the set of links, and W is the set of link weights [54], [55]. For the purpose of this dissertation, graph is used interchangeably with network, node is used interchangeably with both host and computer hardware device, and link is used interchangeably with communications path.

Any network can be represented either directionally or connectionally. An undirected graph G contains undirected links, and a directed graph G_{dir} contains directed links. For G_{dir}, links in each direction have independent link weights [54], [55]. For both G and G_{dir}, there are two types of graphs: complete and simple. The L of a complete graph K_n contains the complete set of links that connect every pair of nodes in N. If L does not contain the entire set of links, the graph is considered simple [54], [56]. Additionally, both G and G_{dir} can be described as connected or disconnected. A network is considered connected if a communications path exists between all pairs of nodes. Conversely, a network is considered disconnected if a communication path does not exist between all pairs of nodes. In the case of a disconnected network, the network is composed of a set of components (or clusters) where a communication path does not exist between the components [55].

To model G or G_{dir}, the adjacency matrix A or directed adjacency matrix A_{dir}, respectively, is formed from the information in Equation 2.1. The $n \times n$ adjacency matrices, A and A_{dir}, are both defined as

$$A(i,j) = A_{dir}(i,j) = \begin{cases} w_{i,j} & \text{if } a_{ij,} \in L, \\ 0 & \text{otherwise,} \end{cases} \tag{2.2}$$

where $w_{i,j}$ is the weight of $l_{i,j}$, $l_{i,j}$ is the link connecting node i to node j, and $a_{i,j}$ is the adjacency element associated with $l_{i,j}$. Since A is an undirected matrix, $w_{i,j} = w_{j,i}$, thus resulting in a symmetric matrix [54], [55].

For G_{dir}, node degrees are described by a set of degree matrices: the in-degree matrix D_{in} and the out-degree matrix D_{out}. The $n \times n$ matrix D_{in} describes the total number of links directionally pointed to the node and is defined as [54], [55]

$$D_{in}(i,j) = \begin{cases} \sum_{k=1}^{n} w_{k,j} & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{2.3}$$

The $n \times n$ matrix D_{out} describes the total number of links directionally pointed from the node and is defined as [54], [55]

$$D_{out}(i,j) = \begin{cases} \sum_{k=1}^{n} w_{i,k} & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{2.4}$$

For G, the $n \times n$ degree matrix D characterizes the total number of links that exist for each node. This means $D = D_{in}(i, j) = D_{out}(i, j)$ [54], [55].

The $n \times n$ Laplacian matrix Q of G is defined as [55]

$$Q = D - A, \tag{2.5}$$

or [23], [55]

$$Q(i, j) = \begin{cases} -w_{i,j} & \text{if } a_{i,j} \in L, \\ \sum_{k=1}^{n} w_{i,k} & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{2.6}$$

This matrix contains all necessary information to visually reconstruct the network data associated with G into a graphical format [55].

2.1.2 Traffic Dispersion Graphs

A traffic dispersion graph (TDG) is a special case of G_{dir} where network logical flows are utilized to create A_{dir}, D_{in}, and D_{out} for the purpose of modeling network traffic. TDGs are utilized in literature as a method to analyze network feature characteristics [57] and perform anomaly detection [21]. The graphical representation of a TDG provides a visual model of the end-to-end communications within the network. For a TDG, G_{dir} is described by the set of nodes containing the sources and destinations of logical end-to-end communications (or flows), the set of links between the source a destination logical flows, and the set of link weights [21]. The $w_{i,j} = 1$ if a logical flow exists from node i to node j, and $w_{i,j} = 0$ if a logical flow does not exist from node i to node j. The standard TCP-tuple is used to identify the existence of a flow and the source and destination information associated with the flow [21]. The resulting G_{dir} is composed of a set of components that represent the communication clusters.

2.2 Spectral Graph Theory

Concepts and techniques from spectral graph theory, a sub-field of graph theory, are employed by the spectral graph detection mechanism of the proposed SGCDC system for the purpose of detecting worm and DDoS attacks. Thus, background on spectral graph theory

is critical to explaining the detection process of the proposed system. We begin this section with a discussion of the dual basis as a method to represent the spectral characteristics of a graph. We then describe the information contained within each basis and explain how the dual basis can be utilized to extract different graph characteristics. We conclude with an examination of phantom nodes as a method to develop reference points within the dual basis for the purpose of analyzing network behavior.

2.2.1 Dual-Basis Representation

The fundamentals of spectral graph theory and the dual-basis representation described in [55], [58], [59] are restated here to provide a baseline of terminology and notation adopted throughout this thesis. The dual-basis representation is dependent on the eigenvalue and eigenvector of Q, which are defined as the solution to

$$Qv_j = \lambda_j v_j, \tag{2.7}$$

where λ_j is the j^{th} eigenvalue in the eigenspectrum, and v_j is the eigenvector corresponding to λ_j. The elements of v_j are represented as $v_{i,j}$, where i is the row of the eigenvector and j is the index of the eigenspectrum, and [55]

$$\sum_{i=1}^{n} v_{i,j}^2 = 1. \tag{2.8}$$

The order of the eigenvalues within the eigenspectrum is

$$0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n \tag{2.9}$$

where $n = N_{tot}$, and N_{tot} is the total number of nodes in the network [55].

The set of eigenvalue-eigenvector solutions are described in matrix form as

$$Q = V\Lambda V^T \tag{2.10}$$

where Λ is the $n \times n$ diagonal matrix of eigenvalues, V is the $n \times n$ orthonormal matrix of

n concatenated v_j [55], [59]. Given that

$$VV^T = I_n, \tag{2.11}$$

where I_n is the $n \times n$ identity matrix [59], the right eigenvector representation is defined as

$$QV = V\Lambda, \tag{2.12}$$

and the left eigenvector representation is defined as [23]

$$V^T Q = \Lambda V^T. \tag{2.13}$$

Utilizing the transformations defined in equations 2.12 and 2.13, the eigencentrality basis is defined as $V\Lambda$ and the nodal basis is defined as ΛV^T. The eigencentrality basis describes network centrality by measuring the degree of impact each node has on each of the eigenvalues across the eigenspectrum. The nodal basis describes nodal influence by measuring the degree of impact a single node has on each of the eigenvalues across the eigenspectrum [23], [60]. Combining together to form the dual-basis representation, these two bases allow for analysis of network robustness, centrality, and connectivity [58], [60].

2.2.2 Phantom Nodes

A phantom node is a concept employed in [23] to create a nodal reference point within the eigenspectrum of the dual-basis to perform connectivity analysis on other nodes in the network. A phantom node is defined in [23] as a single node that does not physically exist in the network but exists mathematically within A. The links connected to the phantom node have a static $w_{i,j}$. The number of links and their $w_{i,j}$ cause the phantom node to have the primary nodal influence $v_{i,j}^{pri}$ in a specific index of the eigenspectrum, where $v_{i,j}^{pri}$ is defined as the largest $|v_{i,j}|$ within v_j. The location of $v_{i,j}^{pri}$ of a node in the eigenspectrum is then utilized to determine the connectivity of a node relative to the phantom node [23]. From the simulation results in [23], the author concludes that nodes that have $v_{i,j}^{pri}$ in eigenspectrum indices larger than the eigenspectrum index where the phantom node has $v_{i,j}^{pri}$ are more connected than the phantom node. Conversely, nodes that have $v_{i,j}^{pri}$ in eigenspectrum indices smaller than the eigenspectrum index where the phantom node has $v_{i,j}^{pri}$ are less connected than the phantom node.

2.3 Worm and DDoS Attack Behaviors

The spectral graph detection mechanism of the proposed SGCDC system is designed to detect worm and DDoS attacks. Thus, background on worm and DDoS attack behaviors is necessary to explain the filtering and detection processes within our proposed model. We begin this section with a discussion of worm attack behavior before moving to a discussion of DDoS attack behavior.

2.3.1 Worm Attack Behavior

Due to its self-propagating nature, a worm requires the ability to locate vulnerable hosts in order to spread. The process of searching for vulnerable hosts is called the target search process, and the process of identifying these vulnerable hosts is called target discovery. As part of the target search and discovery processes, worms employ two main techniques to locate and identify vulnerable hosts: topology-based techniques and scan-based techniques [61]. Our SGCDC system is designed to detect scan-based TCP worms. As a result, the term "worm attack" implies a scan-based TCP worm attack for the remainder of this dissertation.

TCP worms employ the TCP three-way connection handshake [62] in conjunction with scan-based techniques [61] during the target search and discovery processes to identify potential victims to infect. In other words, TCP SYN packets are utilized to probe a specific set of IP addresses via random or localized scanning to identify vulnerable hosts. Random scanning randomly selects an IP address from the address space to probe while localized scanning has a higher probability of selecting an IP addresses closer to its own address than an address further away [61].

Random scanning utilizes three main scanning strategies: uniform, hit-list, and routable. Uniform scanning utilizes the entire internet protocol version 4 (IPv4) address space to identify and target vulnerable hosts without preference. This means that each internet protocol (IP) address has an equal probability of being selected from the entire address space, and it requires the use of a perfect random number generator to randomly select target IP addresses [61]. Hit-list scanning selects an IP address to target from a pre-developed list of vulnerable targets. The worm scans all IP addresses on the hit-list and infects the vulnerable targets before moving forward with the basic random scanning technique [61], [63]. This strategy, introduced in [63], results in an initial acceleration of worm propagation during

the early stages of infection since the worm scans are focused on known vulnerable hosts, and time is not wasted scanning invulnerable hosts [61], [63]. Routable scanning selects routable IP addresses to scan and target instead of the entire IPv4 address space [61].

Localized scanning also utilizes three main scanning strategies: local preference, sequential, and selective. Local preference scanning focuses on the concept that there is not a uniform distribution of vulnerable hosts across the internet [61]. Therefore, the target of the scan is more likely to have an IP addresses close to the infected host IP address than further away [61], [64]. In other words, the worm is more likely to target a host within the same /16 or /8 subnetwork as the source IP address than some other random IP address [64]. Sequential scanning selects consecutive IP addresses starting from an initial address determined by the worm. A specific type of sequential scanning is preferential sequential scanning where the worm chooses a starting IP address closer to its own IP addresses with a higher probability than addresses further away Selective scanning focuses on destroying a certain range of IP addresses. As a result, the scanning space is reduced from the entire IPv4 address space to a certain IP address area [61].

2.3.2 DDoS Attack Behavior

A DoS attack is a malicious attempt to severely degrade the availability of a network resource (host, router, server, network, etc.) [65], [66]. The majority of DoS attacks are distributed where the attacker, known as the bot master, acquires a large number of hosts, called bots, that form a network called a botnet [66], [67]. As a result, the bots, unaware of the compromise, simultaneously attack a target [66] through exploitation of vulnerabilities or flooding of packets [65].

DDoS attacks fall into one of three categories: volume-based, protocol-based, or application layer-based. The goal of a volume-based attack is to reduce the available bandwidth of the victim. Protocol-based attacks, on the other hand, attempt to expend the resources of the victim. Finally, application layer-based attacks attempt to crash a web server by using requests that appear legitimate [67]. In this dissertation, the SGCDC system is designed to detect a particular protocol-based attack, the SYN flood attack. As a result, the term “DDoS attack” implies a SYN flood attack for the remainder of this dissertation.

A SYN flood attack exploits a vulnerability that exists in the TCP three-way connection

handshake. To flood, the attacker sends numerous SYN packets to the victim but does not respond with an ACK to the SYN-ACK response. As a result, the TCP connections remain half-opened while the victim waits for an acknowledgement that the connection is opened, thus resulting in a degradation of victim availability [67].

2.4 Modeling Network Traffic

To evaluate the proposed theoretical performance of the SGCDC system under ideal network conditions, a mathematical model of network SYN traffic is required. Network traffic features like SYN packets have been known to exhibit behavioral characteristics like asymmetry and heavy tails that are not accurately modeled with traditional distributions, such as Poisson and Gaussian [36]–[38]. Both the stable distribution and mixture modeling have shown promise in providing a more accurate model for these types of network traffic features [39]–[43], [46]. We utilize both techniques in our research to develop a distribution that accurately fits the behavior of network SYN traffic. Thus, background on mixture modeling and the stable distribution is critical to the methodology we employ to determine the system theoretical false alarm rates. As a result, we define a mixture model and examine the characteristics of the stable distribution in the following subsections.

2.4.1 Mixture Modeling

Mixture models contain custom mixtures of different distributions intended to either leverage or mitigate properties of the individual components in order to provide a better distribution fit to the data. More formally, the M component distributions $f_1, f_2, ..., f_M$ form a mixture distribution f if

$$f(x) = \sum_{m=1}^{M} w_m f_m(x) \tag{2.14}$$

with $w_m > 0$ and $\sum w_m = 1$, where w_m is the mixture weight. Equation 2.14 describes a complete stochastic model that can be utilized to generate new data [68].

2.4.2 Stable Laws and the Lévy Distribution

The stable distribution stems from the stable laws, which were developed by Lévy in the 1920s while conducting research on the sum of two independent random variables [69].

The laws define random variable X as stable ($X \sim S_\alpha(\beta, \gamma, \mu)$) if

$$aX_1 + bX_2 = cX + d \tag{2.15}$$

is true for all $(a, b) \in \mathbb{R}^+ \times \mathbb{R}^+$, $c \in \mathbb{R}^+$ and $d \in \mathbb{R}$ [70]. This law is utilized to form a family of stable distributions, also known as: α-stable, Pareto stable, and Lévy stable [69]. Extensive theory and mathematical proofs regarding these distributions is outside of the scope of this dissertation and can be found in [42], [71]–[73], among other works.

The α-stable distribution is described by four parameters, α, β, γ, and μ, each of which control a specific aspect of the distribution [39]. The properties and ranges of these four parameters are listed in Table 2.1. Except for three special cases described in Table 2.2, the α-stable distribution does not have a closed-form solution [39], [71]. Instead, the α-stable RV $X \sim S(\alpha, \beta, \gamma, \mu)$ is described by its characteristic function [39]

$$E[\, e^{i\theta X}] \, = e^{-\gamma^\alpha |\theta|^\alpha [\, 1 - i\beta \tan(\frac{\pi\alpha}{2}) \mathrm{sign}(\theta)] + i\mu\theta} \tag{2.16}$$

for $\alpha \neq 1$, and

$$E[\, e^{i\theta X}] \, = e^{-\gamma|\theta| [\, 1 - i\beta \frac{2}{\pi} (\ln|\theta|) \mathrm{sign}(\theta)] + i\mu\theta} \tag{2.17}$$

for $\alpha = 1$. This characteristic function is defined as the Fourier transform of the probability distribution function (PDF) [39], [42], [71].

Table 2.1. α-Stable distribution parameters. Source: [39].

Parameter	Property	Range
α	Tail Size	(0,2]
β	Asymmetry	[-1,1]
γ	Spread	[0,∞]
μ	Location	$\mathbb{R}$

Table 2.2. Parameters of stable distribution special cases. Source: [42].

Distribution	α	β
Cauchy	1	0
Gaussian	2	-
Levy	0.5	1

The Lévy distribution, a special case of the α-stable distribution, sets $\alpha = 0.5$ and $\beta = 1$, resulting in a closed-form PDF of X, defined as

$$f(x;\mu,\gamma) = \sqrt{\frac{\gamma}{2\pi}} \frac{e^{-\frac{\gamma}{2(x-\mu)}}}{(x-\mu)^{3/2}} \tag{2.18}$$

for $x > \mu$ [39], [42]. The closed-form PDF solution of the Lévy distribution facilitates applying traditional statistical techniques for fit estimation and to perform distribution analysis.

2.5 Dual-Basis Perturbation Theorems

To evaluate the proposed theoretical performance of the SGCDC system under non-ideal network conditions, the spectral graph detection mechanism must be analyzed for potential vulnerabilities that exist within the dual-basis. In the case of network analysis, the dual-basis is prone to perturbations as a result of noise or congestion among the network communications [48]–[50], [74]. Therefore, background on eigenvalue and eigenvector perturbations is critical to analyzing the effects of noise and congestion on the ability of the system to accurately detect attacks. While a significant amount of research has been conducted on perturbations within the dual-basis ([48]–[50], [52], [53], among others), this section focuses on the two main theorems that are utilized in this work to analyze the detection mechanism of the proposed system: Weyl's theorem and the Davis-Kahan theorem.

2.5.1 Weyl's Theorem

Weyl's theorem [48], often referred to as the eigenvalue stability inequality, provides an upper bound on the change of λ_j given any perturbation in Q as a result of perturbation(s)

in A. Let the change of λ_j be defined as

$$\Delta\lambda_j = \mid \lambda_j(\hat{Q}) - \lambda_j(Q) \mid, \tag{2.19}$$

where

$$\hat{Q} = Q + Q_{error}, \tag{2.20}$$

$\hat{Q}$ is the perturbed Laplacian matrix, and Q_{error} is the Laplacian error matrix resulting from the $n \times n$ matrix that contains the errors applied to A [75]. Additionally, let

$$\parallel Q_{error} \parallel_{op} = \lambda_{max}, \tag{2.21}$$

where

$$\lambda_{max} = max(\mid \lambda_1(Q_{error}) \mid, \mid \lambda_n(Q_{error}) \mid) \tag{2.22}$$

and $||\bullet||$ is the spectral norm [48], [75]. We have:

Theorem 1 *(Weyl's theorem [48]). For any* $j \in [n]$, $\Delta\lambda_j \leq \parallel Q_{error} \parallel_{op}$.

Weyl's theorem is derived by combining the Courant-Fischer min-max theorem and the Wielandt minimax linear relationship. The Courant-Fischer min-max theorem and proof and the Wielandt minimax formula and linear relationship are found in [75] along with the derivation of the eigenvalue stability inequality.

2.5.2 The Davis-Kahan Theorem and the Eigencentrality Basis

The Davis-Kahan theorem [47] provides an upper bound on the angle of rotation θ_j between v_j and $\hat{v}_j$, where $\hat{v}_j$ is the j^{th} eigenvector of $\hat{Q}$. Let $\hat{\lambda_j}$ be the j^{th} eigenvalue of $\hat{Q}$. Additionally, assume that λ_j and $\hat{\lambda_j}$ have a multiplicity of one. We then have:

Theorem 2 *(Davis-Kahan theorem [48]). Define the min change in eigenvalues as* $\kappa_j = min\{\mid \hat{\lambda_k} - \lambda_j \mid : j \neq t\}$. *Then* $\sin\theta_j \leq \frac{\|Q_{error}\|}{\kappa_j}$.

The proof and additional cases and/or variations of the Davis-Kahan theorem are found in [48]–[50], among other works.

2.6 Modeling Network Countermeasures

In addition to evaluating the detection and classification performance of the SGCDC system, the countermeasure response mechanism is also modeled for analysis. Two types of models have been proposed in literature to model network countermeasures: epidemic models [76] and a passivity-based model [77]. Since epidemic modeling is more widely used in literature, it is the methodology employed in this work. Thus, background on epidemic modeling is critical to the approach employed in this dissertation to evaluate the proposed response of the system. In this section, we begin with a brief overview of epidemic models. Then we explore the susceptible-infectious-recovered (SIR) model utilized in this work to evaluate the SGCDC system.

2.6.1 Epidemic Models

Epidemic models have become the standard in literature for malware analysis because they provide researchers with the ability to analyze the attack spread behavior and the effects of a specific countermeasure while balancing performance and cost [76]. The inclusion of different countermeasures has resulted in numerous compartmental models to simulate various defensive strategies, including the SI, SIS, SIR, SEIR, SEIQRS, SEIQV, [76], and SEIUR [78] models, among others [76].

While one model has not been used more often in literature than others for defensive strategy analysis, the various proposed models result from modifications of the fundamental susceptible-infectious (SI), susceptible-infectious-susceptible (SIS), and SIR [79], [80] epidemic models. For defensive countermeasures employed against worm and DDoS attacks, the SIR model is the best-fit since we assume that recovery from an attack is possible, and re-infection from the same attack is not possible.

2.6.2 SIR Model

The SIR model, otherwise known as the Kermark-McKendrick model or the Classical General Epidemic model [79], is depicted in Figure 2.1. It contains three compartments: susceptible S, infectious I, and recovered R [80]. With respect to computer networks, S typically contains all nodes in the network at initialization. A node transitions from S to I if it becomes infected/attacked, and from I to R if it recovers from the infection/attack. A node

that is in R is considered immune to infection and will remain in that state forever [80]. In other words, the model describes a closed population in which the infection will eventually die out [81].

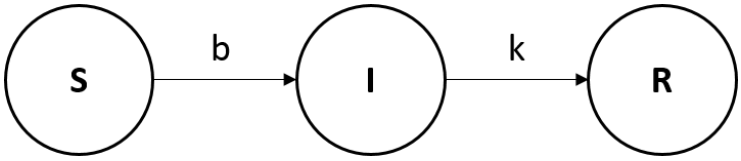


Figure 2.1. Susceptible-Infectious-Recovered (SIR) model with infection contact rate b and recovery rate k. Adapted from [80].

The analytical behavior of the SIR model is described by a set of deterministic, differential equations given by

$$\begin{aligned} \frac{dS_{tot}}{dt} &= -bS_{tot}I_{tot} \\ \frac{dI_{tot}}{dt} &= bS_{tot}I_{tot} - kI_{tot} \\ \frac{dR_{tot}}{dt} &= kI_{tot} \end{aligned} \tag{2.23}$$

where b is the infection contact rate, k is the recovery rate, S_{tot} is the number of susceptible, I_{tot} is the number of infectious, and R_{tot} is the number of recovered [81].

Utilizing Equation 2.23, the basic reproduction number of the system R_0 is derived. Given an infection within a susceptible population, R_0 describes the number of secondary infections that can be expected from the spread of the infection [82]. It describes the stability of both the infection-free and epidemic equilibrium, providing a quantitative measure of the infection spread [83]. If $R_0 \leq 1$, the system will head toward in infection-free equilibrium over time. On the other hand, if $R_0 \geq 1$, the system will head toward an epidemic equilibrium over time [76]. For the SIR model [81],

$$R_0 = \frac{b}{k}. \tag{2.24}$$

As a result, the stability of the system is directly dependent on b and k.

In summary, this chapter summarized the background technical material necessary for

describing the fundamental concepts of our proposed model, and for understanding the methods employed to analyze the performance of our proposed system. With this baseline knowledge established, the next chapter describes the inner workings of our proposed system.

CHAPTER 3:
Spectral Graph-Based Cyber Defense System

In this chapter, we propose a novel cyber detection and classification system for worm and DDoS attacks utilizing spectral graph theory tools. The proposed detection and classification methods take advantage of the extensive theory and mathematical techniques associated with graph theory and spectral graph theory, which are discussed in Chapter 2. While we limit our proposed methods to worm and DDoS attacks, the methodologies presented in this chapter could be utilized against other various cyber attacks with minor modifications. Furthermore, we narrow our proposed system to the detection of TCP-based attacks, but the system could be employed against UDP-based attacks with minor modification to the filters.

Referring to Figure 3.1, our proposed SGCDC system contains five functional components. First, the relevant traffic flows are filtered and processed into a usable format as part of the network behavior sensing component. Graph spectral analysis techniques are then applied to the data to detect attack behaviors in the spectral graph detection component. Then, the detected attack behaviors are classified as either worm or DDoS in the classification component. In the case of worm attacks, the detected attack behaviors are stored in system memory as part of a feedback loop process. Finally, the attack parameter outputs of the detection and classification mechanisms are utilized to determine appropriate network response actions in the response component.

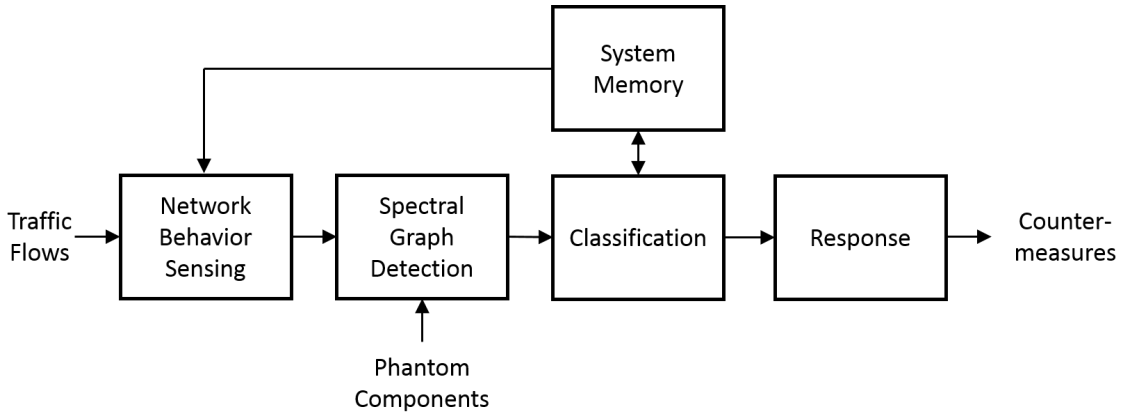


Figure 3.1. Proposed SGCDC system component model.

The remainder of this chapter is organized as follows. First, we provide a detailed description of the five components of the proposed SGCDC system. Then, we define the external system parameters that are required to be initialized by the network administrator.

This chapter includes material adapted from work to be published by the author. Specifically, sections 3.1, 3.2, and 3.6 contain some revised material from "Spectral Graph-based Cyber Worm Detection Using Phantom Components and Strong Node Concept" by Jamie Safar, Murali Tummala, and John McEachen, to be published in the 54th Hawaii International Conference on System Sciences.

3.1 Network Behavior Sensing

The network behavior sensing component shown in Figure 3.2 utilizes the network traffic flows to perform the following functions: data windowing, behavior filtering, and TDG formation. The outputs of this component are the undirected, first-level offspring adjacency matrix A_1, the undirected, second-level offspring adjacency matrix A_2, and the dual-degree matrix, which are utilized by the spectral graph detection and classification components. To fully describe the network behavior sensing component, we discuss the three functional processes of the component in detail in the order that they are performed.

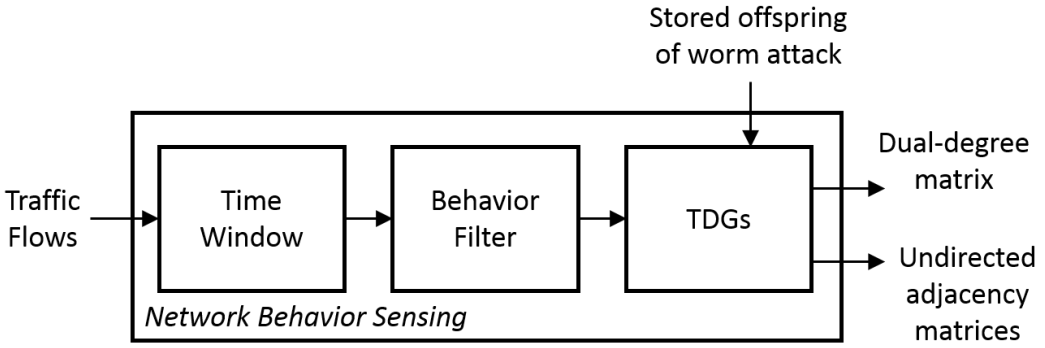


Figure 3.2. Functional processes of the network behavior sensing component.

3.1.1 Time Window

The network behavior sensing component makes use of time windows to decrease the amount of data being processed by the system at any given time. The system employs an overlapping time window, as shown in Figure 3.3, where each window begins halfway through the previous window. The length of the time window L_w is externally determined by the network administrator, which is discussed in Section 3.6.

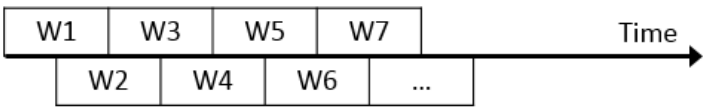


Figure 3.3. Overlapping time window representation

The overlapping time windows are necessary to overcome the missed detection problems that arise from a single shifted-window. When an attack exists in time across two single shifted-windows, there is a possibility that the attack will not be detected if the traffic in each window does not meet threshold requirements. Figure 3.4 shows the probability of window detection of a modeled Blaster worm utilizing the SGCDC system with a one second single shifted-window, a worm phantom component threshold $T^{worm}_{phantom}$ = 13, and a memory log time t_{log} = 40 seconds (see Section 3.6 for discussion of $T^{worm}_{phantom}$ and t_{log}). For the single shifted-window, the number of windows that failed to detect worm activity increase as the size of the infection increases. For the same simulation, the overlapping time window had a window detection probability of one for all end infection sizes. Unlike the

single shifted-window, the overlapping time window allows the system to account for the occurrence of an attack across multiple windows.

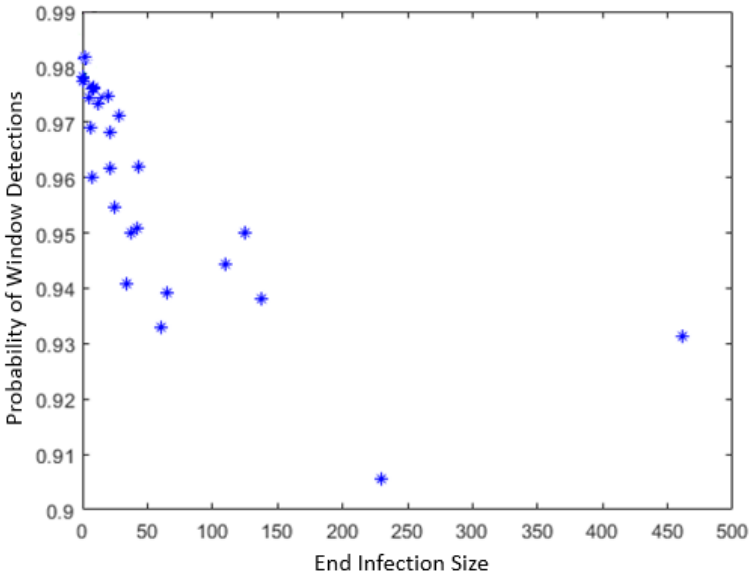


Figure 3.4. Probability of window detections for modeled Blaster worm using a one-second single shifted-window.

3.1.2 Behavior Filter

The network behavior sensing component utilizes a behavior filter to further decrease the computational overhead for the system and to eliminate traffic that does not match the known behavioral characteristics of a worm or DDoS attack. In other words, not all traffic flows within a window are utilized to form the TDG (discussed in Section 3.1.3). Depending on the known graphical and behavioral characteristics of the attack to be detected, the network logical flows are filtered for behavior prior to forming A_{dir}. Additionally, the spectral graph detection component only requires information on the presence or absence of a flow between any two nodes. As a result, the elements of A_{dir} are defined as

$$a_{i,j} = \begin{cases} 0 & \text{if no flow exists,} \\ 1 & \text{if at least one flow exists .} \end{cases} \tag{3.1}$$

In other words, A_{dir} is normalized so that each matrix element either equals zero, representing the absence of a flow between the two nodes in either direction or one, representing the presence of a flow between the two nodes in either direction. This further reduces the system computational overhead by only utilizing unique traffic flows.

Since our research specifically focuses on TCP worm attacks and SYN-flood DDoS attacks, the behavior filter eliminates traffic flows that do not match the known SYN behaviors of these attacks. For worm attacks, the sending of SYN packets in the TCP three-way handshake results in a causal-tree [22], [84] graphical representation of the attack behavior regardless of the specific scanning technique employed. The graphical representation of the worm attack behavior is shown in Figure 3.5 where $P^w_{i,j}$ is j^{th} worm parent node on the i^{th} attack level (representing an infected host) and $O^w_{i,j}$ is the j^{th} worm offspring node on the i^{th} attack level (representing a host that is scanned by the infected node). It is important to note that only two levels of infected parent nodes are shown in Figure 3.5, but worm attacks typically result in many levels of infected parents due to the causal-tree behavior [22], [84].

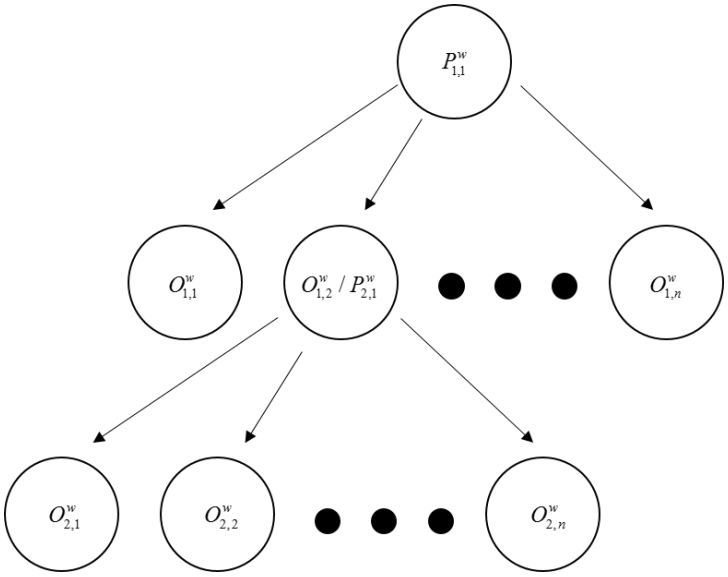


Figure 3.5. Graphical representation of worm attack behavior from SYN scan traffic.

For DDoS attacks, the sending of SYN packets results in a graphical representation that appears as an inverted, single-level worm attack. The graphical representation of the DDoS attack behavior is shown in Figure 3.6 where P^d is the DDoS parent node (representing the

attacked host) and O_i^d are the DDoS offspring nodes (representing hosts that are attacking the parent node).

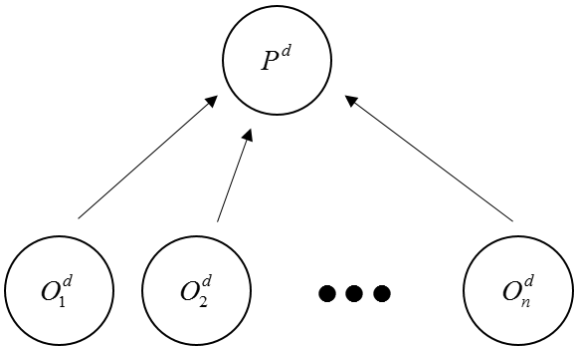


Figure 3.6. Graphical representation of DDoS attack behavior from SYN attack traffic.

Due to the feedback loop from the data storage sub-system (see Section 3.4), two filtering functions are performed on the windowed traffic flows. First, the traffic is filtered for first-level offspring. To conform with worm and DDoS attack behaviors, only traffic flows that contain a packet with flags SYN=1 and ACK=0 (i.e., the first packet of the TCP three-way handshake) and a unique destination address are utilized to form the directed, first-level offspring adjacency matrix A_{dir}^1. Second, the traffic is filtered for second-level offspring. To conform with the worm attack behavior, only traffic flows from A_{dir}^1 that have a source IP address that was previously flagged by the spectral graph detection component as offspring of detected worm attack behavior are utilized to form the directed, second-level offspring adjacency matrix A_{dir}^2.

3.1.3 Traffic Dispersion Graphs

The directed adjacency matrices, A_{dir}^1 and A_{dir}^2, are the mathematical representations of the filtered TDG of the network (see Section 2.1.2 for more information on TDGs). The network behavior sensing component uses the TDGs formed by A_{dir}^1 and A_{dir}^2 to create A_1, A_2, and the dual-degree matrix. For the purposes of eigendecomposition, A_{dir}^1 and A_{dir}^2 are utilized to form A_1 and A_2, respectively, via the process shown in Aglorithm 1. The dual-degree matrix is formed by combining D_{in} and D_{out}, which is discussed in detail in Section 3.3.1.

Algorithm 1: Algorithm to form A_i from A^i_{dir}

```
for i from 1 to n do
    for j from 1 to i do
        if A[i,j]==1 OR A[j,i]==1 then
            A[i,j]=A[j,i]=1;
        else
            A[i,j]=A[j,i]=0;
        end
    end
end
```

3.2 Spectral Graph Detection Component

The spectral graph detection component is the central anomaly detection mechanism of the proposed SGCDC system. This component performs eigenanalysis to identify anomalous behaviors through the employment of the strong node concept and phantom components. More specifically, as shown in Figure 3.7, the undirected adjacency matrices, A_1 and A_2, produced by the network behavior sensing component are utilized by the spectral graph detection component to perform the following functions: phantom component attachment, eigendecomposition, and anomaly detection using the strong node concept. The outputs of this component are the detected parent and offspring nodes exhibiting attack behavior.

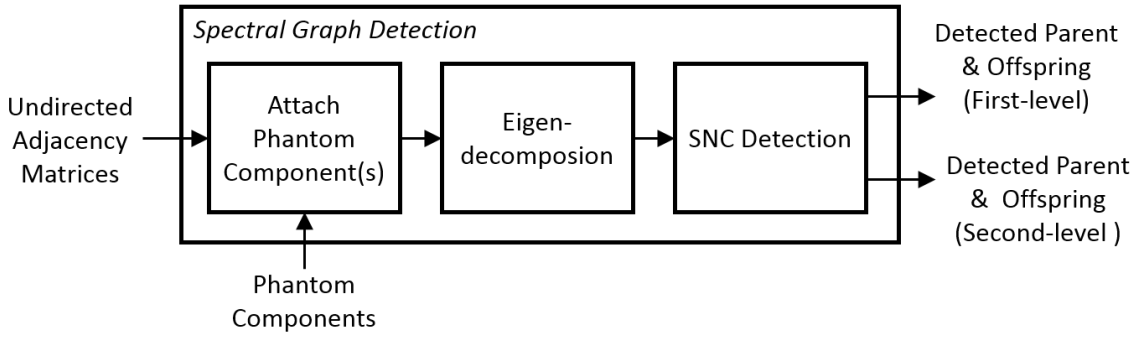


Figure 3.7. Functional processes of the spectral graph detection component.

Before we can delve into the strong node concept detection process utilized by the spectral

graph detection component, we must first define phantom components and provide a description of the phantom component associated with worm and DDoS attack detection. We must also define the strong node concept, which is fundamental to the detection mechanism. Once this foundation is established, we discuss how phantom components are combined with the strong node concept to produce thresholds in the eigenspectrum of the dual-basis, which are utilized by the spectral graph detection component to detect worm and DDoS attacks.

3.2.1 Phantom Components

Phantom components are crucial elements for the detection mechanism of the spectral graph detection component. Expanding upon the phantom node concept in [23] (described in Chapter 2), a phantom component is defined as a group of connected nodes that do not exist in the directed adjacency matrix of the TDG but exist mathematically within the undirected adjacency matrix for the purpose of spectral analysis. In broader terms, a phantom component provides a graphical depiction of a malicious behavior that is utilized for detection. The number of nodes that make up the phantom component and the connection of those nodes describe both the desired detection threshold and the attack behavior for detection, respectively.

While the worm and DDoS attack behaviors are graphically directional (see figures 3.5 and 3.6), the phantom component exists within the symmetric, undirected adjacency matrix. As a result, the undirected graphical representation of a worm and DDoS attack is identical when only considering first-level offspring. As a result, the same phantom component construct, shown in Figure 3.8, is employed by the spectral graph detection component for both worm and DDoS attack detection. For the purposes of the proposed system, the links between the nodes within the phantom component have a static link weight equal to one. The difference between the phantom component utilized for worm detection and the phantom component utilized for DDoS detection is the number of offspring nodes connected to the parent node. The number of offspring nodes within the phantom component, defined as the phantom threshold $T_{phantom}$, is externally determined by the network administrator (see Section 3.6).

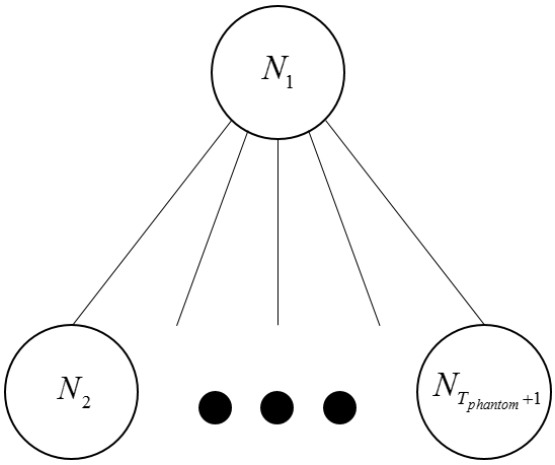


Figure 3.8. Graphical representation of the phantom component utilized by the SGCDC system for worm and DDoS attack detection.

It is important to note that the phantom component for worm attack detection only covers the first-level offspring from the graphical attack behavior of the worm (see Figure 3.5). In order to attain additional levels of offspring using this phantom component, memory must be introduced into the system with a feedback loop for detection of additional levels of offspring (see Section 3.4).

3.2.2 Strong Node Concept

The strong node concept provides a framework for the phantom components to be utilized to detect worm and DDoS attack behavior within network traffic flows. Using the spectral graph theory techniques discussed in Chapter 2, the strong node concept is derived from analysis of the dual-basis with specific emphasis given to the relationship between an eigenvector element $v_{i,j}$ and its corresponding eigenvalue λ_j.

Given Equation 2.5, Equations 2.7 can be rewritten as

$$0 = d_i v_{i,j} - (\sum_{k=1}^{n} a_{i,k} v_{k,j} \mid k \neq i) - \lambda_j v_{i,j} \tag{3.2}$$

or

$$v_{i,j} = \frac{1}{d_i - \lambda_j}(\sum_{k=1}^{n} a_{i,k} v_{k,j} \mid k \neq i), \tag{3.3}$$

where d_i is the degree associated with the i^{th} node [85]. Keeping all other variables constant, d_i and λ_j provide information on the behavior of $v_{i,j}$. The magnitude of the scaling factor $\frac{1}{d_i - \lambda_j}$ in Equation 3.3 is larger in lower eigenspectrum indices for smaller degree nodes than larger degree nodes. Conversely, the magnitude of $\frac{1}{d_i - \lambda_j}$ from Equation 3.3 is larger in higher eigenspectrum indices for larger degree nodes than smaller degree nodes. An example of this is shown in Figure 3.9 for a network containing six nodes with $d_1 \leq d_2 \leq d_3 \leq d_4 \leq d_5 \leq d_6$. In this example, d_1 has the largest scaling factor in λ_2, and d_6 has the largest scaling factor in λ_6.

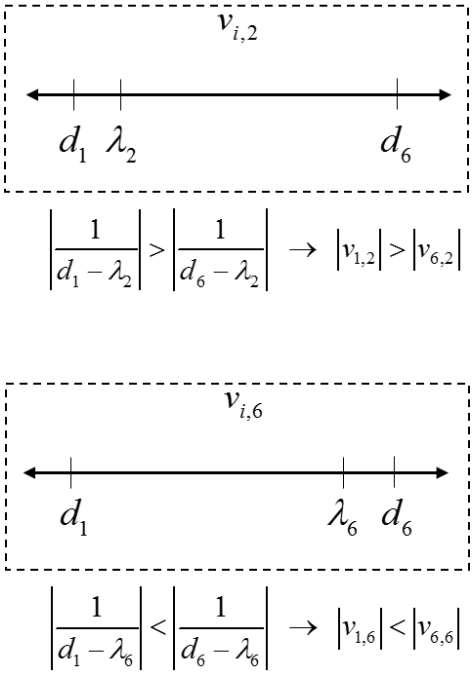


Figure 3.9. Visual number line depiction of effects of the magnitude of the scaling factor $\frac{1}{d_i - \lambda_j}$ from Equation 3.3 on the magnitude of $v_{i,j}$ for a network of six nodes where $d_1 \leq d_2 \leq d_3 \leq d_4 \leq d_5 \leq d_6$. Source: [86].

As depicted in Figure 4.10, we expand this line of thought to develop a mathematical foundation for the strong node concept, which directly connects node strength (or connectivity) to the eigenvector magnitude value across the eigenspectrum. The summation

$(\sum_{k=1}^{n} a_{ik} v_{kj} \mid k \neq i)$ from Equation 3.3 is directly influenced by the degree (or connectivity) of the connected neighbors of the node. As a result, larger degree nodes have more terms in the summation than smaller degree nodes. Each term in the summation is directly scaled by the eigenvector value of the neighboring node, which in turn is being scaled by the scaling factor $\frac{1}{d_i - \lambda_j}$ of that node. Using deductive reasoning, $\frac{1}{d_i - \lambda_j}$ appears to be the dominating parameter in Equation 3.3. Consequently, the effects depicted in Figure 4.10 for a six-node network should hold true across all networks and graphs.

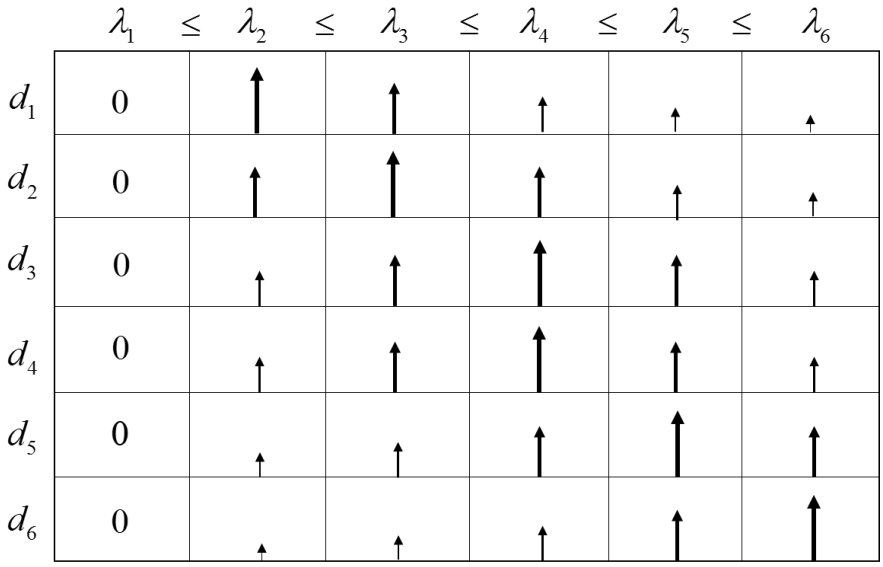


Figure 3.10. Depiction of the strong node concept using the magnitude of $v_{i,j}$ for a six node network with $d_1 \leq d_2 \leq d_3 \leq d_4 \leq d_5 \leq d_6$. Source: [86].

To summarize, the strong node concept relates node connectivity directly to its eigenvector magnitude value (or nodal influence) across the eigenspectrum. Nodes that are strongly connected will have larger nodal influence in higher eigenspectrum indices while nodes that are weakly connected will have larger nodal influence in lower eigenspectrum indices [23].

3.2.3 Detection Process

The detection process utilizes the strong node concept and phantom components to develop thresholds within the eigenspectrum for the purpose of detecting malicious behavior/patterns within the network traffic flows. While most applications of spectral graph theory focus on

analysis of static networks [87], [88], the strong node concept detection process compares eigenspectrum changes over time for the purpose of analyzing real-world dynamic networks.

The detection process begins with the attachment of both the worm and DDoS phantom components to A_1 to produce A^1_{phan}, and the worm phantom component to A_2 to produce A^2_{phan}. The second-level matrices, A_2 and A^2_{phan}, are employed as part of the system feedback loop (see Section 3.4) to attain second-level offspring of a worm attack. Eigendecomposition is then performed on the newly formed $(n + m) \times (n + m)$ adjacency matrices, A^1_{phan} and A^2_{phan}, to produce the eigenvalue and eigenvector matrices where $m = (T^{worm}_{phantom} + 1) + d_{comp}(T^{DDoS}_{phantom} + 1)$, $T^{worm}_{phantom}$ is the number of offspring nodes in the worm phantom component, $T^{DDoS}_{phantom}$ is the number of offspring nodes in the DDoS phantom component, and the DDoS component scale factor

$$d_{comp} = \begin{cases} 1 & \text{if } A^1_{phan}, \\ 0 & \text{if } A^2_{phan}. \end{cases} \tag{3.4}$$

Given the strong node concept, one of the nodes within the phantom component will have $v^{pri}_{i,j}$ in at least one of the $n \times 1$ vectors of V (i.e., in one eigenspectrum index). The phantom eigenvalue λ_{phan} is the eigenvalue associated with the highest eigenspectrum index that a phantom component node has $v^{pri}_{i,j}$. The λ_{phan} is utilized to develop an eigenspectrum threshold T_{eigen} in the dual-basis. Since an eigenvalue can have a multiplicity greater than one, T_{eigen} is established in the lowest eigenspectrum index that has an eigenvalue equal to λ_{phan}. The T_{eigen} is then utilized to detect threshold crossings. A threshold crossing occurs when a node is detected as having $v^{pri}_{i,j}$ in an eigenspectrum index greater than or equal to the T_{eigen} established by the phantom component. Due to the strong node concept, nodes that have a connectivity greater than or equal to the most connected node in the phantom component will have a $v^{pri}_{i,j}$ in an eigenspectrum index greater than or equal to T_{eigen}.

The strong node concept detection process is demonstrated using the filtered TDG shown in Figure 3.11. In this TDG, the black components represent normal traffic flows, the green component represents the phantom component utilized for detection, and the red

components represent worm attack traffic. The phatom component has $T^{worm}_{phantom} = 2$. Nodes 5, 9, 11, and 15 are infected by a worm and are exhibiting worm attack behavior. The undirected, phantom adjacency matrix of this TDG,

$$A^1_{phan} = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0
\end{bmatrix}, \quad (3.5)$$

is utilized to form Q, which is decomposed into the eigenvalues and eigenvectors depicted in Figure 3.12. The largest eigenspectrum index that the phantom component has $v^{pri}_{i,j}$ is 17 (indicated by the green box), which has an associated $\lambda_{phan} = 3$. The smallest eigenspectrum index that has an eigenvalue equal to λ_{phan} is 16. As a result, $T_{eigen} = 16$ (indicated by the dashed green line). All nodes that have $v^{pri}_{i,j}$ in an eigenspectrum index greater than or equal to T_{eigen} are detected as anomalous, or in this case infected by a worm (indicated by the red boxes). The nodes detected as exhibiting worm attack behavior (nodes 5, 9, 11, and 15, indicated by red boxes) are in fact the worm infected nodes from the TDG in Figure 3.11.

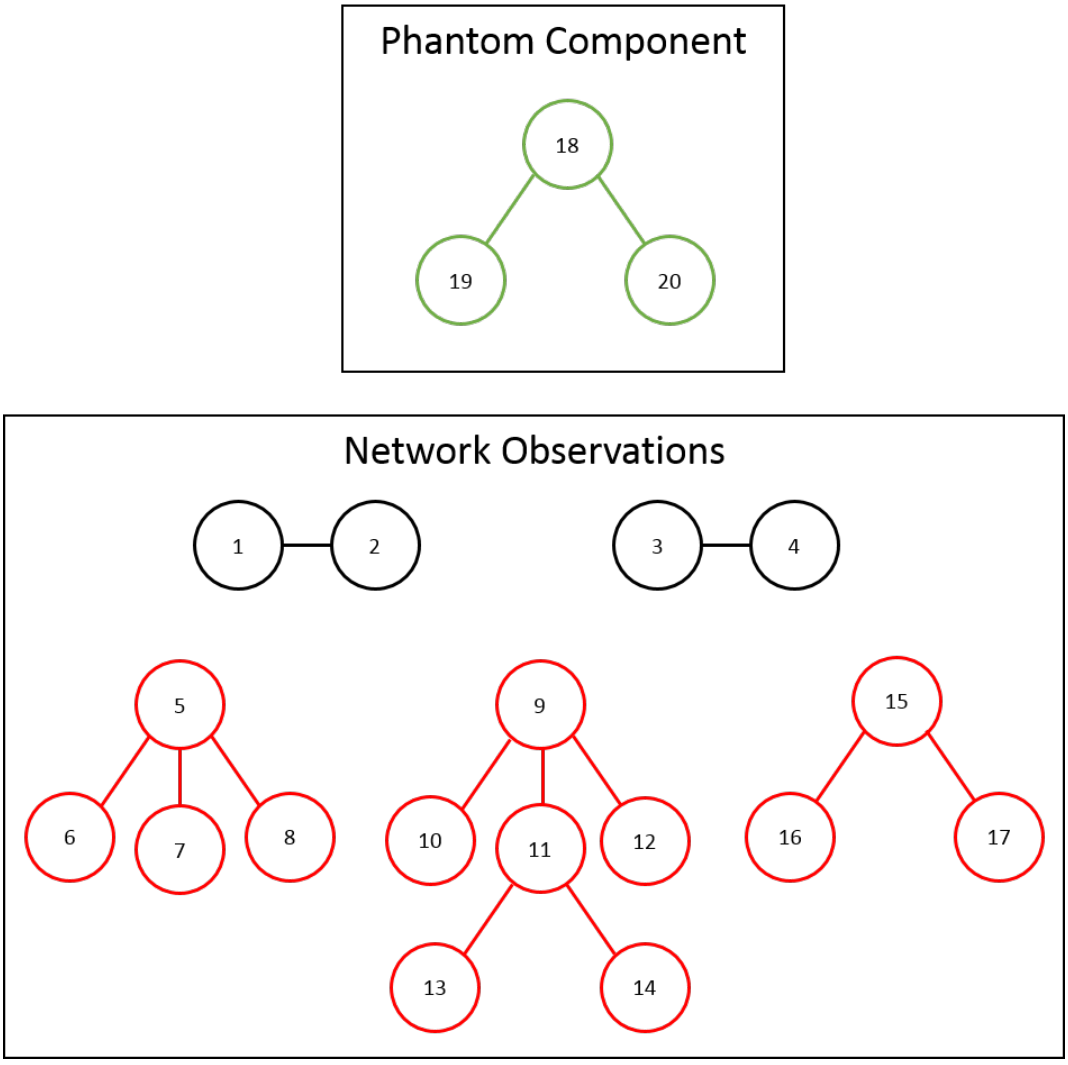


Figure 3.11. Example TDG containing a phantom component (green), normal traffic flows (black), and worm attack traffic (red).

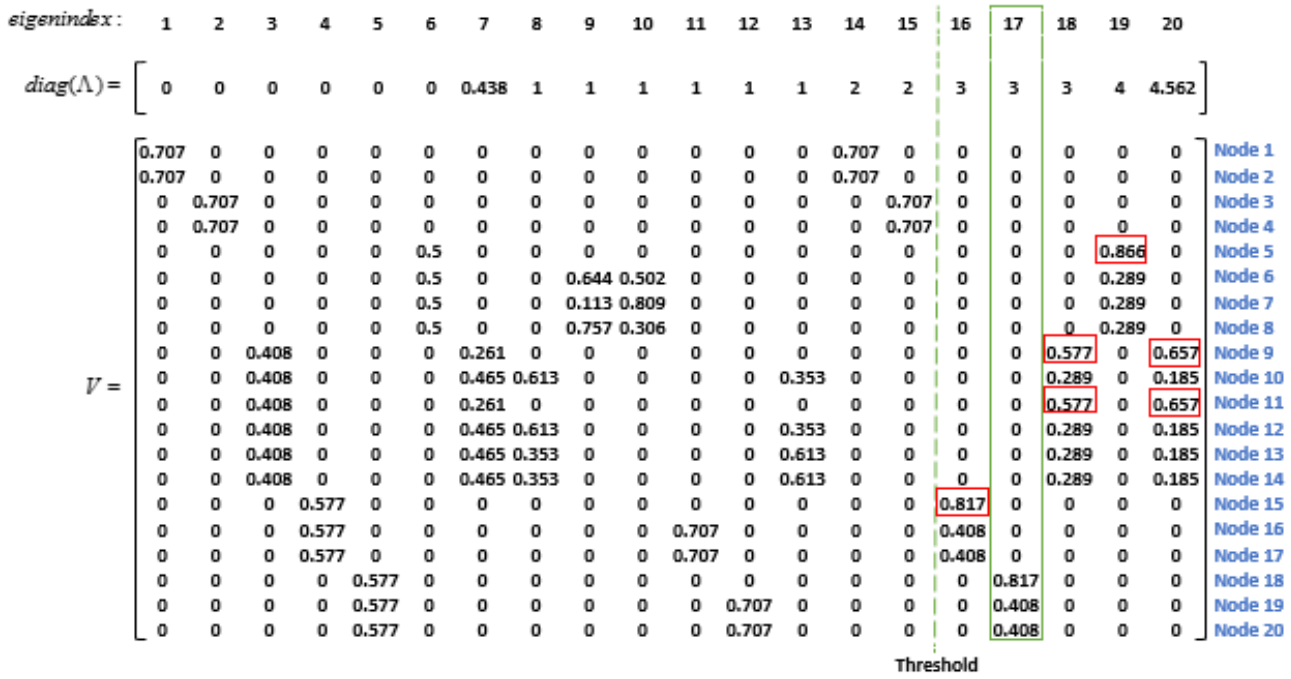


Figure 3.12. Example of the strong node concept detection process using a phantom component (green box) to create a threshold in the eigenspectrum index (green dashed line) of the dual basis to identify worm infected nodes (red boxes) associated with the TDG depicted in Figure 3.11.

A node detected as anomalous by the strong node concept detection process is considered the parent node of the attack behavior. For each node detected as anomalous, the other nodes that have $|v_{i,j}| > 0$ in the same eigenspectrum index are considered its offspring. For example, in the strong node concept detection process shown in Figure 3.12, the offspring of node 5 are nodes 6, 7, and 8. Comparing these results to Figure 3.11, the strong node concept detection process accurately identifies the worm infected nodes and potential victims of the worm infected nodes.

When a threshold crossing is detected by the spectral graph detection, the node (or nodes) responsible are classified as anomalous. Parent and offspring nodes that are detected as anomalous in A^1_{phan} and A^2_{phan} are sent to the classification component to determine if they are part of a DDoS attack or worm attack.

3.3 Classification

The classification component shown in Figure 3.13 is responsible for classifying detected anomalous network behaviors and providing that classification and additional known attack parameters to the response component. The dual-degree matrix, the detected anomalous node(s) and offspring from A^1_{phan}, the detected anomalous parent and offspring node(s) from A^2_{phan}, and the anomalous parent and offspring node(s) stored in the system memory component are inputs for the classification component. The classification component utilizes these inputs to perform the following functions: anomaly classification and attack parameter determination. The outputs of this component include the victim, potential attackers, and estimated attack rate of a DDoS attack, and the first-level and second-level infected nodes, the first-level and second-level potential victim nodes, and estimated scan rate of a worm attack. In the remainder of this section, we first discuss the anomaly classification process before describing the outputted attack parameters.

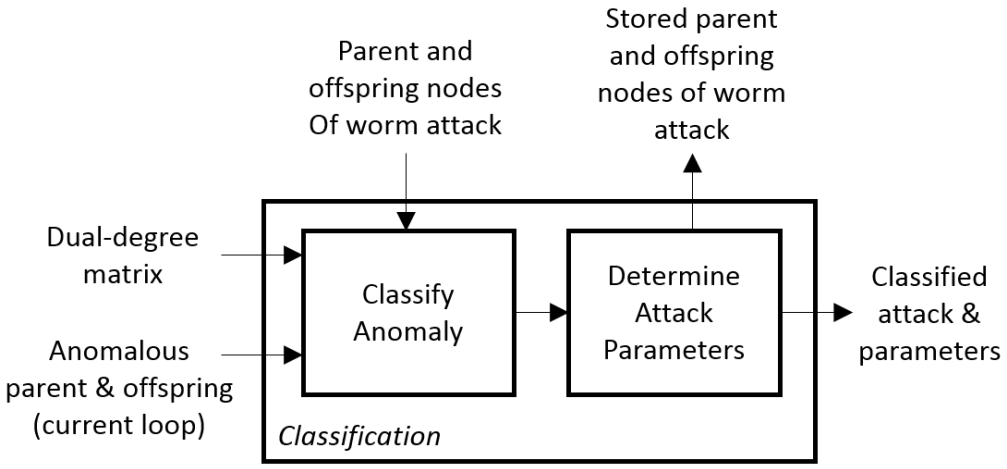


Figure 3.13. Functional processes of the classification component.

3.3.1 Anomaly Classification

Since the same phantom component structure is utilized to identify both worm and DDoS attacks, an additional mechanism must be implemented by the classification component to distinguish/classify the detected anomalous nodes from the spectral graph detection component. The worm and DDoS phantom components are identical in graphical structure but potentially differ in the value of $T_{phantom}$. While it is possible that $T^{worm}_{phantom} = T^{DDoS}_{phantom}$, where $T^{worm}_{phantom}$ is the worm phantom component threshold and $T^{DDoS}_{phantom}$ is the DDoS phantom component threshold, every application of $T_{phantom}$ in this work has $T^{worm}_{phantom} < T^{DDoS}_{phantom}$ due to the historical attack characteristics of worm and DDoS attacks. If $T^{worm}_{phantom} < T^{DDoS}_{phantom}$, then $T^{worm}_{eigen} < T^{DDoS}_{eigen}$ due to the strong node concept where T^{worm}_{eigen} is the eigenspectrum index threshold established by the worm phantom component, and T^{DDoS}_{eigen} is the eigenspectrum index threshold established by the DDoS phantom component. As a result, nodes detected as crossing T^{DDoS}_{eigen} could be either a worm attack or a DDoS attack. Additionally, nodes that cross T^{worm}_{eigen} could be either a worm attack or normal traffic flows that exhibit DDoS-like behavior. As a result, we developed the dual-degree matrix as a method to distinguish and classify a worm attack from a DDoS attack.

The dual-degree matrix produced by the network behavior sensing component contains link directional information, which is used to differentiate between worm and DDoS attacks. The dual-degree matrix is a $N_{filt} \times 2$ matrix in which the first column vector contains the diagonal elements of D_{in}, the second column vector contains the diagonal elements of D_{out}, and N_{filt} is the number of filtered nodes that compose the TDG. If anomalous behavior is detected by an eigenspectrum threshold crossing in the spectral graph detection component, the classification component compares the in-degree to the out-degree of the detected anomalous (i.e., parent) node. If the in-degree is greater than the out-degree, then the node and its offspring are classified as a DDoS attack. If the out-degree is greater than the in-degree, then the node and its offspring are classified as a worm attack. Since a node is restrained to a single attack type for the purpose of this work, additional research needs to be conducted to determine the risk of a mild worm or DDoS attack being masked by a simultaneous DDoS or worm attack, respectively.

All detected anomalous nodes from A^1_{phan} that are classified as worm attack represent the first-level graphical attack behavior of a worm. While a single iteration of the spectral graph detection process provides the necessary information associated with the DDoS attack behavior, it does not satisfy the graphical worm attack behavior. As a result, the anomalous node and its offspring are sent to the system memory component to be utilized as a filter in subsequent iterations of the strong node concept detection process to attain an additional level of attack behavior (see Section 3.4). It is important to note that the SGCDC system is designed to detect worm attacks only after the second node is infected. This means that the first node infected by the worm will not be identified as infected until a second node is infected and begins exhibiting the worm attack behavior.

Only nodes from A^1_{phan} that cross T^{DDoS}_{eigen} and are classified as DDoS attack from the dual-degree matrix are sent to the next functional process in the classification component to identify the DDoS attack parameters. Similarly, only nodes from A^2_{phan} that cross T^{worm}_{eigen} and are classified as worm attack from the dual-degree matrix are sent to the next functional process in the classification component to identify the worm attack parameters.

3.3.2 Attack Parameters

The classification component utilizes the classified anomalies to provide the response component with critical attack parameter information for network response. In the case of a detected and classified DDoS attack, the classification component uses the nodes detected by the spectral graph detection component to identify the victim (i.e., the detected parent node) and the potential attackers (i.e., the detected offspring nodes). Additionally, the component estimates the DDoS attack rate (i.e., the number of unique communication links per window time unit associated with the detected attacked node), defined as

$$R_{DDoS}^{est} = \frac{N_{offspring}}{L_w}, \tag{3.6}$$

where $N_{offspring}$ is the total number of offspring nodes associated with the detected parent node.

In the case of a detected and classified worm attack, the classification component uses the nodes stored in memory and the nodes detected by the spectral graph detection component to identify the infected nodes (i.e., detected parent nodes) and the potential victims (i.e., detected offspring nodes) of the worm attack. Additionally, the component estimates the worm spread rate (i.e., the number of unique communication links per window time unit associated with the detected infected node), defined as

$$R_{worm}^{est} = \frac{N_{offspring}^{P1} + N_{offspring}^{P2}}{2 \times L_w}, \tag{3.7}$$

where $N_{offspring}^{P1}$ is the total number of offspring nodes associated with the first-level parent node, and $N_{offspring}^{P2}$ is the total number of offspring nodes associated with the second-level parent node.

3.4 System Memory

The system memory component is responsible for storing information on classified first-level worm attack behavior, which is utilized by both the network behavior sensing component and the classification component. The system memory component stores information on both the parent and offspring nodes of a classified first-level worm attack for a set period of

time t_{log}. The length of t_{log} is externally determined by the network administrator, which is discussed in Section 3.6.

In order to accurately differentiate and identify worm attack behavior from potentially normal traffic flows, two levels of offspring must be exhibited and identified. To attain the second-level offspring, the first-level offspring must be utilized as a filter in subsequent iterations of the SGCDC process. The implementation of the feedback loop in the SGCDC system as shown in Figure 3.14 allows the system memory component to share its stored information with the network behavior sensing component. For the length of t_{log}, information on detected anomalous offspring nodes are sent to the network sensing component to be utilized by the behavior filter to find TDGs containing second-level offspring (i.e., for A^2_{dir}). This process allows the single-level phantom component (discussed in Section 3.2.1) to be utilized to detect additional levels of offspring.

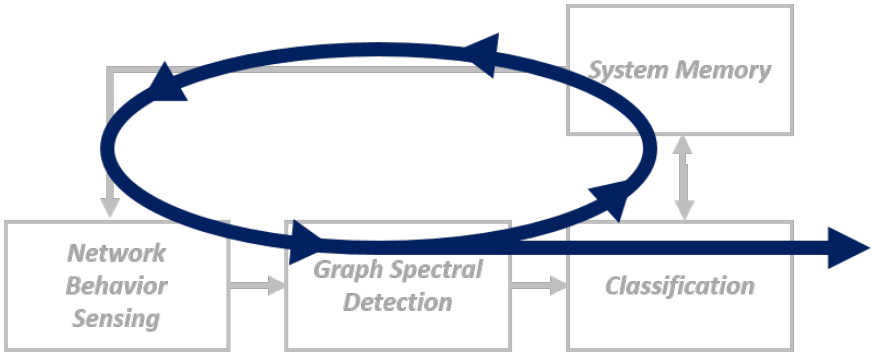


Figure 3.14. Feedback loop within the SGCDC system.

For implementation of the proposed SGCDC system, a single feedback cycle is utilized to detect second-level offspring; however, the feedback loop could be implemented for additional iterations to detect additional levels of offspring. While this may potentially decrease the number of false alarms in the system, additional iterations increase the computational costs and length of time for detection.

The system memory component is also responsible for providing information to the classification component to correlate the first-level worm attack behavior to the second-level worm attack behavior and to determine additional attack parameters. For the length of t_{log}, information on a detected anomalous node and its offspring are sent to the classification for this purpose.

3.5 Response

The response component is responsible for selecting the appropriate network actions based on the attack parameters outputted by the classification component. These attack parameters provide the SGCDC system with critical information to defend the network against the detected worm and/or DDoS attack(s). The response component could be implemented in a variety of ways depending on the needs or desires of the network administrator. For example, the response component could represent a user interface where the attack parameters are sent to a human monitoring the network to make decisions and take action to defend the network. The attack parameters also provide the opportunity for the response component to be a module with pre-programmed network responses to attain quicker response times. Regardless of the implementation, the response component takes action to defend the network against the detected and classified attacks.

3.6 External Input Parameters

There are four external, input parameters critical to the SGCDC system that must be determined by the network administrator as part of the system initialization: L_w, $T_{phantom}^{worm}$, $T_{phantom}^{DDoS}$, t_{log}. The L_w controls the number of traffic flows being processed by the system and should be as small as possible to decrease the computational complexity and to provide quicker identification and classification of attacks. The $T_{phantom}^{worm}$ represents the number of offspring in the worm phantom component, and $T_{phantom}^{DDoS}$ represents the number of offspring in the DDoS phantom component. Both $T_{phantom}^{worm}$ and $T_{phantom}^{DDoS}$ should be as large as possible to keep benign network traffic below T_{eigen}, yet small enough that worm and DDoS attack traffic will still be detected. To ensure $T_{phantom}^{worm}$ is small enough to detect worm attack traffic, the relationship given by

$$T_{phantom}^{worm} < R_{scan} \times L_w \tag{3.8}$$

must hold true where R_{scan} is the worm attack scan rate defined as the number of SYN packets sent per unit time to unique hosts. Similarly, to ensure $T_{phantom}^{DDoS}$ is small enough to

detect DDoS attack traffic,

$$T_{phantom}^{DDoS} < R_{attack} \times L_w \tag{3.9}$$

must hold true where R_{attack} is the DDoS attack rate defined as the number of unique nodes sending SYN packets per unit time. Finally, t_{log} is a parameter that is solely used for worm attack detection and classification. The t_{log} must be chosen such that

$$t_{log} > t_T - t_{SYN} \tag{3.10}$$

where t_T is the time when the second-level parent node sends a number of SYN packets equal to $T_{phantom}^{worm}$ in the defined L_w, and t_{SYN} is the time when the second-level parent node first received the SYN-only packet of the TCP three-way handshake from the first-level parent node. In other words, t_{log} must be greater than the sum of the infection time t_{infect} and the time it takes to send a number SYN packets equal to $T_{phantom}^{worm}$. The initialization of these four external parameters by the network administrator directly impacts the detection accuracy and false alarm rate of the SGCDC system (see Chapter 4).

All simulations in this work use one instantiation of each of these external, input parameters; however, when implementing the system in hardware and software, multiple instantiations of each parameter can be employed to provide additional detection services. Two or more L_w, $T_{phantom}^{worm}$, $T_{phantom}^{DDoS}$ and/or t_{log} values may be implemented to provide detection against both slow-spreading and fast-spreading attacks. For such cases, a confidence level mechanism may need to be introduced into the system to differentiate between attacks detected by the different parameters.

In summary, this chapter introduced our solution approach for worm and DDoS detection using spectral graph theory tools. We proposed a system that requires external input and performs detection using the strong node concept and phantom components. The next chapter explores the theoretical false alarms for the proposed system under ideal network conditions for a given set of external, input parameters.

CHAPTER 4:
Theoretical False Alarm Rates

The spectral graph detection approach employed in the proposed SGCDC system is susceptible to false alarms. In this chapter, we examine the proposed system states and state transition probabilities to develop theoretical upper bounds on the system false alarm rates under ideal network conditions. We begin by describing state models for both the proposed system and the individual nodes within the system. From the two models, equations are developed to define the system and node false alarm rates. The equations require a PDF of network SYN traffic, so we develop a novel mixture modeling methodology and propose a novel Lévy-impulse model to describe network SYN traffic. Utilizing the Lévy-impulse model and false alarm equations, the theoretical upper bounds are then defined for the system false alarm rates. Finally, nodal state visitation is explored to provide an addition layer of theoretical analysis on the length of time between false alarms.

This chapter includes material adapted from work to be published by the author. Specifically, Section 4.3 contains some revised material from "A Novel Lévy-Impulse Mixture Based Connection Model for Computer Network Traffic" by Jamie Safar, Murali Tummala, and John McEachen, to be published in the 14th International Conference on Signal Processing and Communication Systems.

4.1 System State Model Specifications

To determine the proposed SGCDC system theoretical false alarm rates, a system state model is developed to describe the transitional probabilities that result in the detection of an attack. The proposed SGCDC system is modeled using the two-state absorbing Markov chain shown in Figure 4.1, where the first state s_1 represents normal, and the second state s_2 represents attacked. The state transition probabilities are defined as $p_{i,j}$ where i is the current state and j is the new state [89].

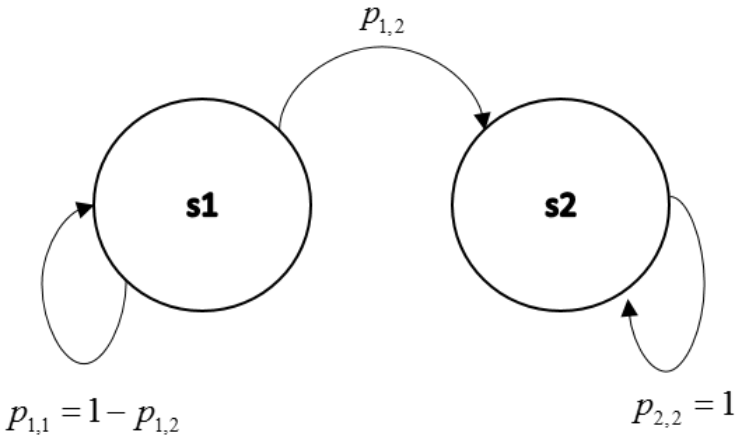


Figure 4.1. System state model for the proposed SGCDC system.

The system initially resides in s_1 and will remain in s_1 as long as an attack is not detected. The system will transition from s_1 to s_2 if either a worm attack or DDoS attack is detected from the filtered TDGs. Once an attack is detected, the system remains in s_2 until network intervention stops the attack and restores the system to its normal state [90]. The attack response and recovery efforts are variable and typically occur on a different time scale [91] compared to the proposed system detection and classification process. As a result, s_2 is modeled as an absorbing state.

To determine $p_{1,2}$ of the system state model, the probability of worm detection $P(D_{worm})$ and the probability of DDoS detection $P(D_{DDoS})$ must be defined. For the purposes of this dissertation, we assume that the probability that any two nodes communicate is equally likely. As a result,

$$P(D_{worm}) = c_{det}^{worm} \left(\int_{T_{phantom}^{worm}}^{\infty} f_{SYN}^{L_w}(x)dx \right)^2 \tag{4.1}$$

where c_{det}^{worm} is the worm detection scaling factor defined as

$$c_{det}^{worm} = \left(\frac{2 \times S_{tx}^{n_i} \times t_{log}}{L_w} \right), \tag{4.2}$$

D_{worm} is the event describing the detection of a worm attack, $S_{tx}^{n_i}$ is the number of SYN

packets transmitted by node i in the given L_w, and $f_{SYN}^{L_w}(x)$ is the PDF of SYN traffic for the given L_w. The worm detection scaling factor is a result of the three conditions that must be met for worm traffic to be detected. In Appendix A.1, these three conditions are defined and the worm scaling factor is derived as part of the derivation of $P(D_{worm})$.

For DDoS attacks,

$$P(D_{DDoS}) = \int_{T_{phantom}^{DDoS}}^{\infty} f_{SYN}^{L_w}(x)dx \tag{4.3}$$

where D_{DDoS} is the event describing the detection of a DDoS attack. See Appendix A.1 for a complete derivation of $P(D_{worm})$ and $P(D_{DDoS})$. From equations A.6 and 4.3, $p_{1,2}$ for the system state model is

$$p_{1,2}^{sys} = P(D_{worm}) + P(D_{DDoS}). \tag{4.4}$$

In the SGCDC system, a system false alarm D_{false}^{sys} occurs if the system transition from s_1 to s_2 when no worm or DDoS attack exists in the network. As a result, the probability of D_{false}^{sys} is

$$P(D_{false}^{sys} \mid \text{no worm or DDoS attack in network}) = p_{1,2}^{sys}. \tag{4.5}$$

This means that the system false alarm rate is directly dependent on the set of external, input parameters (L_w, $T_{phantom}^{worm}$, $T_{phantom}^{DDoS}$, and t_{log}) and the PDF of network SYN traffic.

4.2 Nodal State Model Specifications

While the system state model provides an overarching depiction of the detection process, it does not provide any insight on the intermediary steps resulting from the implementation of the feedback loop to detect second-level offspring of worm attacks. As a result, a nodal state model is developed to describe node transition probabilities that result in the detection of an attack. This model is then utilized to develop an expression that describes the probability of a node being identified as attacked when no attack exists.

Each node monitored by the SGCDC system is independently modeled using the three-state absorbing Markov chain shown in Figure 4.2 where s_1 represents normal, s_2 represents abnormal, and the third state s_3 represents attacked. While each node has its own independent model, dependency exists between the models with respect to state transitions.

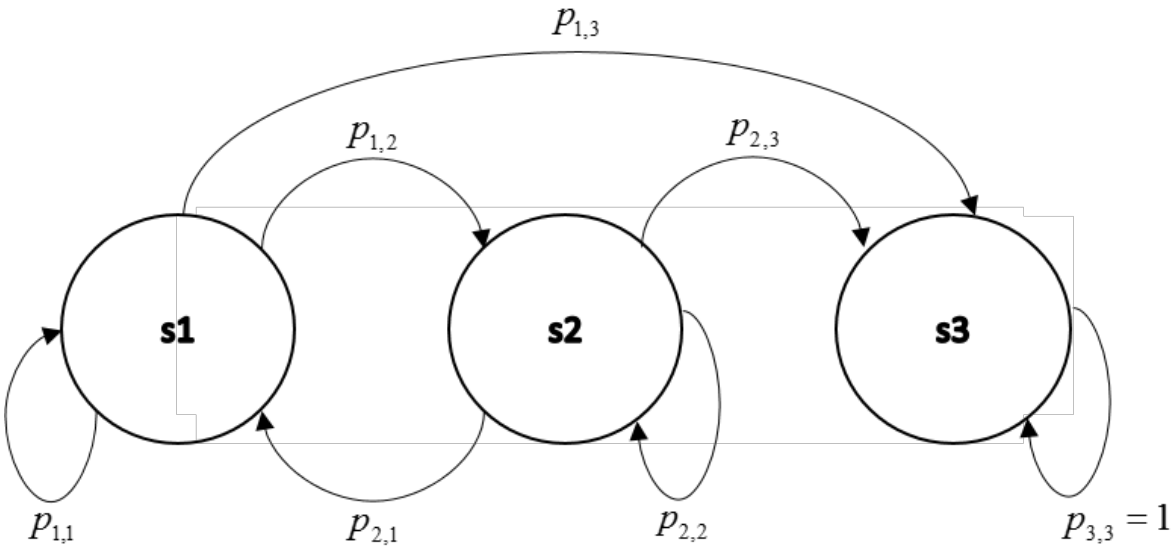


Figure 4.2. Nodal state model for the proposed SGCDC system.

The node initially resides in s_1 and will remain in s_1 unless one of three conditions is met: 1) a DDoS attack is detected, 2) the node sends a number of SYN packets greater than or equal to $T^{worm}_{phantom}$, and one of its offspring subsequently sends a number of SYN greater than or equal to $T^{worm}_{phantom}$ while it is being stored in system memory, or 3) the node receives a SYN packet from a node that sent a number of SYN packets greater than or equal to $T^{worm}_{phantom}$. The combination of condition one and two results in a transition from s_1 to s_3. For condition one, the node is identified as the victim of a DDoS attack, which has a probability equal to $P(D_{DDoS})$ as defined in Equation 4.3. For condition two, the node is identified as a first-level parent of a worm attack, which has a probability equal to $P(D_{worm})$ as defined in Equation A.6. Combining the probabilities, the probability of a node transitioning from s_1 to s_3 is given by

$$p^{node}_{1,3} = P(D_{DDoS}) + P(D_{worm}). \tag{4.6}$$

For condition three, the node becomes a first-level offspring of a potential worm attack and transitions from s_1 to s_2. Assuming a uniform distribution for the probability that any two nodes communicate, the probability of a node transitioning from s_1 to s_2 is obtained as

$$p^{node}_{1,2} = S^{n_i}_{tx} \int_{T^{worm}_{phantom}}^{\infty} f^{L_w}_{SYN}(x)dx. \tag{4.7}$$

If a node in s_1 does not transition to s_2 or s_3, then it remains in s_1. Thus, the probability of a node remaining in s_1 is given by

$$p_{1,1} = 1 - p_{1,2} - p_{1,3}. \tag{4.8}$$

From s_2, a node will transition to s_3 if it sends a number of SYN packets greater than or equal to $T_{phantom}^{worm}$. This means the probability of a node transitioning from s_2 to s_3 is given by

$$p_{2,3}^{node} = \int_{T_{phantom}^{worm}}^{\infty} f_{SYN}^{L_w}(x)dx. \tag{4.9}$$

A node in s_2 will transition back to s_1 if two conditions are met: 1) all other nodes that send a number of SYN packets greater than or equal to $T_{phantom}^{worm}$ do not send this node a SYN while it is being stored in system memory as an offspring node, and 2) if the node does not send a number of SYN packets greater than or equal to $T_{phantom}^{worm}$ while it is being stored in system memory. Assuming a uniform value of $S_{tx}^{n_i}$ for all nodes found above T_{eigen}^{worm}, the probability of a node transitioning from s_2 to s_1 can be shown to be

$$\begin{aligned} p_{2,1}^{node} = & \left(\int_0^{T_{phantom}^{worm}} f_{SYN}^{L_w}(x)dx \right)^{(2\times t_{log})/Lw} \\ & \times \left(\prod_{k=1}^{S_{tx}^{n_i}} \frac{N_{sending} - k}{N_{sending}} \right)^{\left(2\times(N_{tot}-1)\times t_{log}\times \int_{T_{phantom}^{worm}}^{\infty} f_{SYN}^{L_w}(x)dx\right)/L_w} . \end{aligned} \tag{4.10}$$

If a node in s_2 does not transition to s_1 or s_3, then it remains in s_2. Thus, the probability of a node remaining in s_2 is given by

$$p_{2,2}^{node} = 1 - p_{2,3}^{node} - p_{2,1}^{node}. \tag{4.11}$$

Once an attack is detected, the node remains in s_3 until network intervention stops the attack and restores the system to its normal state [90]. As a result, s_3 is modeled as an absorbing state. The derivations of $p_{1,3}^{node}$, $p_{1,2}^{node}$, and $p_{2,1}^{node}$ are provided in Appendix A.2.

In the SGCDC system, a nodal false alarm D_{node}^{false} occurs if any node transitions to s_3 when

no worm or DDoS attack exists on the node. As a result, the probability of of a nodal false alarm can be expressed as

$$P(D^{node}_{false} \mid \text{no worm or DDoS attack on the node}) = p_{1,3} + (p_{1,2} \times p_{2,3}). \tag{4.12}$$

This means a node false alarm rate is also directly dependent on the set of external, input parameters (L_w, $T^{worm}_{phantom}$, $T^{DDoS}_{phantom}$, and t_{log}) and the PDF of network SYN traffic.

4.3 SYN Traffic Distribution Model

In order to further analyze the theoretical false alarm rates of the proposed SGCDC system, the PDF of network SYN traffic must be determined. We begin this section by describing the characteristics of SYN traffic. Next, we show that traditional distributions do not accurately describe the SYN traffic behavior. Since none of the previously proposed models that we reviewed in literature fit the SYN traffic characteristics, we develop an innovative mixture model methodology to develop the novel Lévy-impulse model for network SYN traffic. We conclude this section with an analysis of the accuracy of the proposed Lévy-impulse model.

4.3.1 SYN Traffic Characteristics

To develop an understanding of the behavior and characteristics of network SYN traffic, we utilize a dataset containing one hour of real-world traffic flows collected in 2013 from the Naval Postgraduate School (NPS) network. For the remainder of this dissertation, we refer to the SYN traffic flows within this collected dataset as the NPS2013 dataset. The NPS2013 dataset contains 1341 nodes, and the overall traffic has an average SYN rate of approximately 16.17 packets per second. The number of SYN packets sent by each individual node to unique destination addresses for varying L_w is shown in Figure 4.3. The data distributions across the different L_w all exhibit a high zero-probability and a very long tail. While this work focuses on SYN packets sent to unique destination addresses to support the worm and DDoS attack behavior modeled by the phantom component, it is important to note that similar results are observed when we do not constrain the data to unique destination addresses.

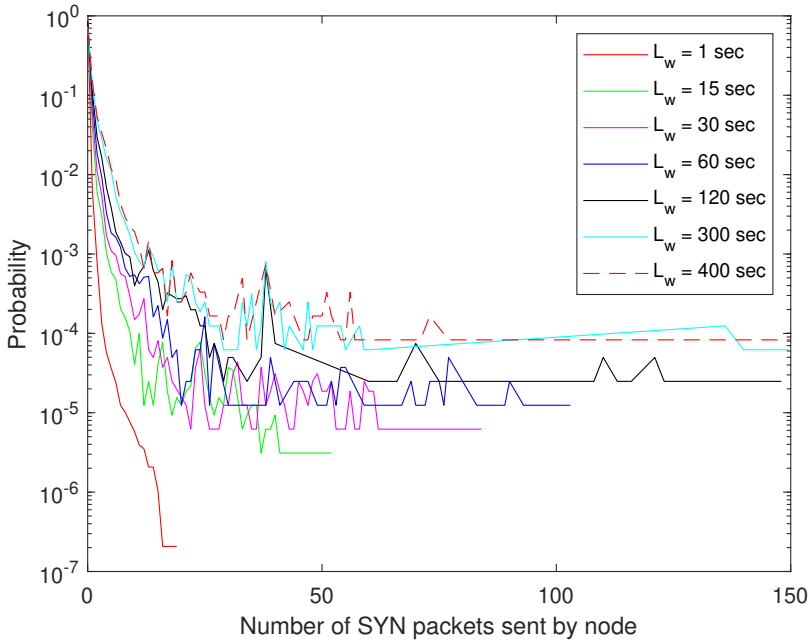


Figure 4.3. Probability of a node in the NPS2013 dataset sending a specified number of SYN packets to unique destination addresses for various L_w. Source: [92].

4.3.2 Distribution Fit of Traditional Models

As discussed in Chapters 1 and 2, many real-world network traffic features cannot be accurately modeled using traditional distribution models, such as exponential, gamma, normal, and Rayleigh [36]–[38]. Utilizing the SYN packets transmitted to unique destination addresses in the NPS2013 dataset, we evaluate and compare the fit of these traditional distribution models using the “fitmethis" program [93] for MATLAB. This program utilizes MATLAB built-in maximum likelihood estimation (MLE) function [94] to find the best distribution fit for the data vector under consideration. The output includes both the distribution parameters and log-likelihood values for the continuous and discrete distributions available in MATLAB MLE function.

The log-likelihood results of the exponential, gamma, normal and Rayleigh distribution fits for the transmitted SYN traffic in NPS2013 dataset are listed in Table 4.1. The exponential was suggested as the "best-fit" for all of the windows.

Table 4.1. Log-likelihoods of MLE fits of various commonly used network modeling distributions fitted to the NPS2013 SYN dataset. Source: [92].

L_w (sec)	Exponential	Gamma	Normal	Rayleigh
1	18010000	-217600	3601000	-2682000
5	2271000	-118800	-168300	-399700
15	449200	-80020	-315900	-109000
30	125900	-62520	-253700	-60220
45	49480	-52060	-196600	-46600
60	17970	-45830	-165200	-39400
75	3344	-40930	-139600	-35670
90	-5020	-37240	-124300	-32680
120	-13340	-32360	-102200	-29280
180	-16540	-25910	-74210	-22450
240	-16690	-22220	-60530	-18900
300	-15960	-19420	-51090	-17140

Yet, when the suggested exponential distribution is compared to the NPS2013 dataset, the exponential distribution fails to accurately fit both the high zero-probability and heavy tail of the SYN traffic. For example, a comparison of the MLE-suggested exponential distribution and the NPS2013 SYN data for L_w = 45 seconds is shown in Figure 4.4. Similar to the results observed for all L_w, the suggested exponential distribution does not accurately reflect the NPS SYN traffic for L_w = 45 seconds; the exponential distribution fails to accurately reflect the high zero-probability of SYN packets being sent.

From the results, traditional distributions fail to accurately model network SYN traffic behavior. As a result, we turn to mixture modeling methodologies in an attempt to develop a more accurate model.

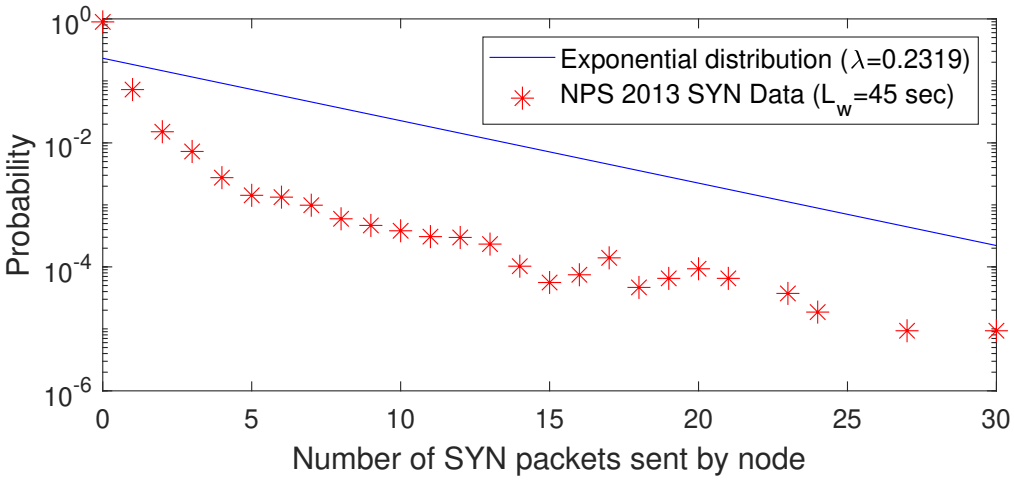


Figure 4.4. Comparison of the exponential distribution with the MLE-suggested $\lambda = 0.2319$ (blue line) and SYN packet counts from the NPS2013 dataset (red asterisk) for $L_w = 45$ seconds. Source: [92].

4.3.3 Mixture Modeling and the Lévy-Impulse Model

None of the previously proposed models in mixture modeling literature (see Section 1.2.5), accurately describe the SYN traffic characteristics observed in the NPS2013 dataset. Additionally, the traditional truncation methodologies proposed in mixture modeling literature (see Chapters 1 and 2) did not work well for the SYN traffic characteristics since the zero-count density is consistently greater than the non-zero transmission counts by a few orders of magnitude. As a result, we propose an innovative truncation methodology where an impulse is combined with a heavy-tailed distribution to develop a novel mixture model. In accordance with this methodology, the windowed NPS2013 SYN data is split into two groups to account for both the high zero-probability and the heavy tail. An impulse is applied to the high-zero probability data, and the heavy-tailed data is analyzed for "best-fit." The results are then combined and normalized to form the mixture model.

The first group contains the data associated with SYN transmission counts equal to zero (i.e., the data associated with the high zero-probability). An impulse located at zero captures the high zero-probability of SYN traffic that appears to skew a single-distribution model. The zero-count probabilities for the first group for various L_w are shown in Figure 4.5 as red asterisks. We then apply MATLAB polynomial curve fitting tools to this data using the

"polyfit" function [94] to determine the polynomial coefficients of the least-squares best-fit to the data [95]. As a result, the probability of a node sending zero SYN packets for a given window length is given by

$$P(N_{SYN}^{L_w} = 0) = -0.02261 L_w^{0.4528} + 1.021, \tag{4.13}$$

as shown by the blue line in Figure 4.5. Equation 4.13 provides the scaling factor of the impulse for the proposed Lévy-impulse model.

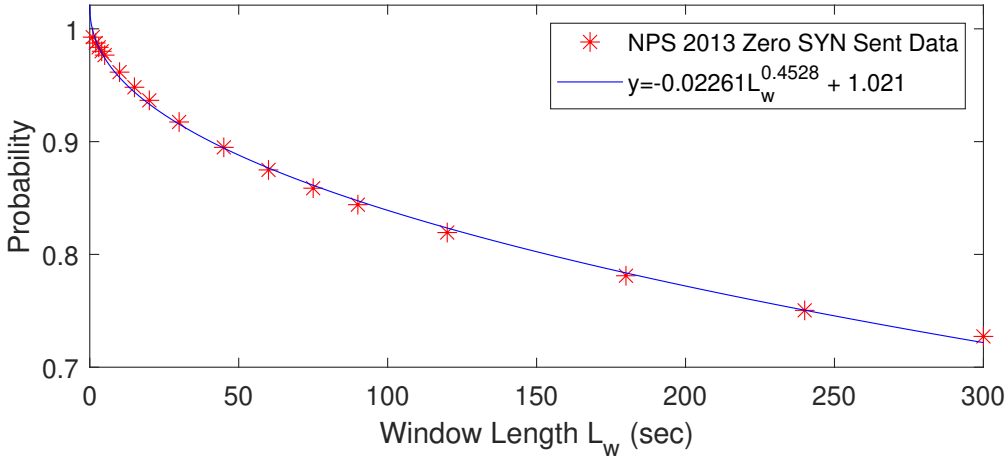


Figure 4.5. Curve-fit (blue line) for the probability of a node to send zero SYN packets within a defined window using NPS 2013 SYN data (red asterisk). Source: [92].

The second group contains the remaining SYN data (i.e., the data associated with the long-tail). Utilizing the "fitmethis" program, the log-likelihood results of various distributions applied to the second group of data is listed in Table 4.2 for various L_w. From the log-likelihood results, the stable distribution is suggested as the "best-fit" for all L_w tested.

Table 4.2. Log likelihoods of various commonly used network modeling distributions fitted to the second group of NPS2013 SYN data (data excluding zero-count transmissions). Source: [92].

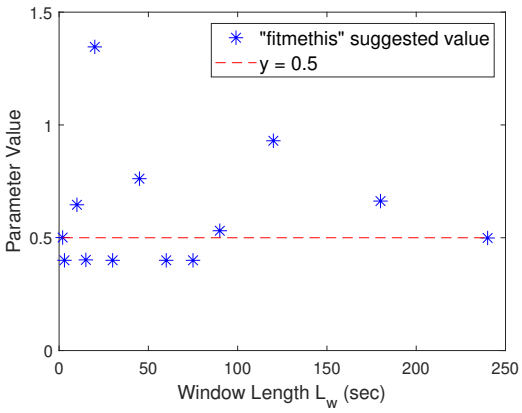
L_w (sec)	Stable	Gamma	Weibull	Exponential	Gaussian
1	71870	-26660	-35380	-43750	-47750
5	0	-26980	-30240	-31700	-45450
15	20300	-24480	-25880	-26080	-41100
30	6389	-22230	-22750	-22750	-38130
45	-4813	-19940	-20170	-20200	-34170
60	-4833	-18540	-18600	-18670	-31690
75	-6258	-17250	-17230	-17330	-29350
90	-4554	-16290	-16190	-16320	-27520
120	-8482	-14840	-14650	-14840	-25050
180	-6876	-12540	-12310	-12540	-21140
240	-5772	-11120	-10870	-11130	-18790
300	0	-10040	-9794	-10050	-16830

As discussed in Chapter 2, the stable distribution contains four parameters: α, β, γ, and μ [39]. The MLE-suggested values of these stable distribution parameters for various L_w are shown in Figure 4.6 where the blue asterisks are the suggested parameter values and the red dashed lines are our suggested fits. For α and β, no discernable trend exists in relation to L_w; however, the majority of the L_w have an α parameter value close to 0.5 and a β parameter value close to 1.0. These parameter results align with Lévy distribution, a special case of the stable distribution (see Chapter 2). Applying MATLAB polynomial curve fitting tools [94] to the γ and μ datasets in Figure 4.6, the parameter value of γ for a given L_w is given by

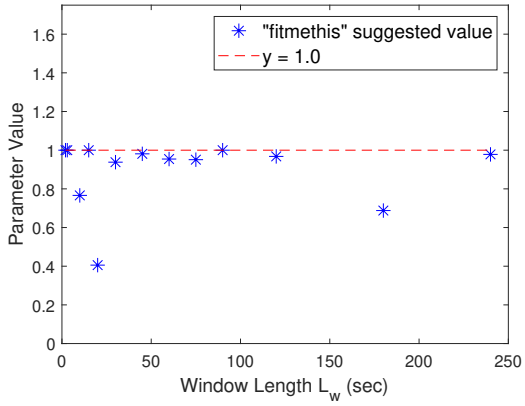
$$\gamma_{L_w} = 0.001244 L_w + 0.09255, \tag{4.14}$$

and the parameter value of μ for a given L_w is obtained as

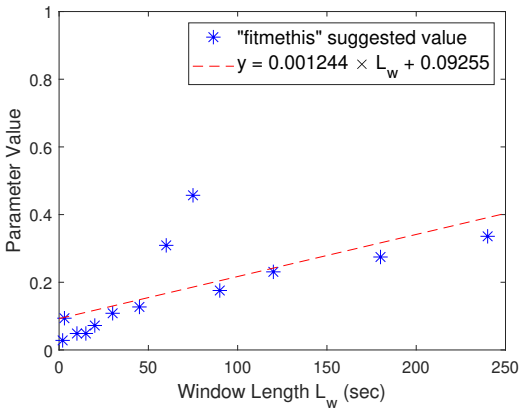
$$\mu_{L_w} = 0.0007453 L_w + 1.047. \tag{4.15}$$



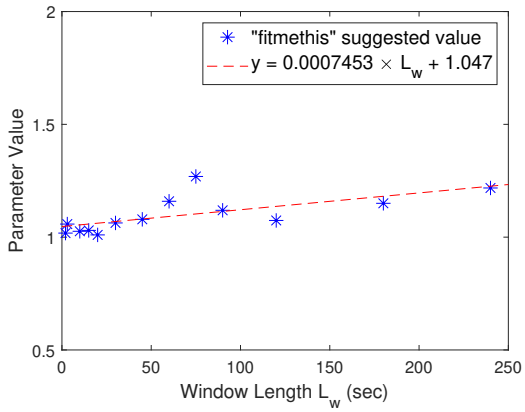
(a) α parameter values



(b) β parameter values



(c) γ parameter values



(d) μ parameter values

Figure 4.6. Parameters of the stable distribution suggested by the "fitmethis" program for the second group of NPS2013 SYN data (data excluding zero-count transmissions). Source: [92].

Continuing with our proposed methodology, the Lévy-impulse model is formed by first concatenating the result from the two groups of data and then normalizing the distribution to ensure the law of total probability is not violated. As a result, the PDF of the proposed Lévy-impulse model describing network SYN traffic is defined as

$$f_{SYN}^{L_w}(x) = \xi_{L_w}\delta[0] + (1-\xi_{L_w})\sqrt{\frac{\gamma_{L_w}}{2\pi}}\frac{e^{-\left(\frac{\gamma_{L_w}}{2(x-\mu_{L_w})}\right)}}{(x-\mu_{L_w})^{3/2}}, \tag{4.16}$$

where

$$\xi_{L_w} = P(N_{SYN}^{L_w} = 0). \tag{4.17}$$

This model accurately accounts for the dominant features (i.e., high zero-probability and heavy tail) exhibited by network SYN traffic.

4.3.4 Accuracy of the LI Model

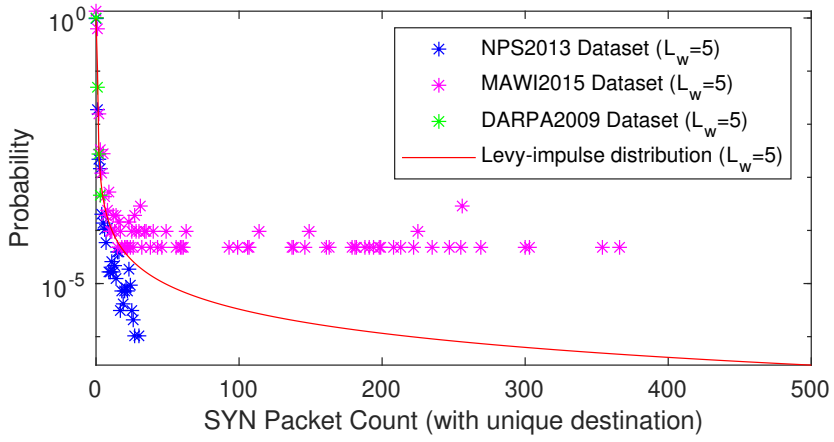
To demonstrate the accuracy of the proposed Lévy-impulse model in characterizing network SYN traffic, we analyze the SYN packets sent by nodes and the SYN packets received by nodes for two different real-world network datasets and one synthesized dataset: the NPS2013 dataset, the Measurement and Analysis on the WIDE Internet (MAWI) 20151114 trace [96] (referred to as the MAWI2015 dataset in the remainder of this dissertation), and the 2009 DARPA Scalable Network Monitoring Program Traffic 2009-11-05 trace [97] (referred to as the DARPA2009 dataset in the remainder of this dissertation).

The MAWI2015 dataset consists of a 15-minute .pcap trace collected between 14:00 and 14:15 Greenwich Mean Time (GMT) from the trans-Pacific backbone link between Japan and the United States, which operates at a rate of approximately one Gbps [96], [98]. Due to the extremely large amount of data within each trace and the size of the network, we only analyze the SYN traffic between zero and five seconds (containing 20,820 nodes) for the results in this section.

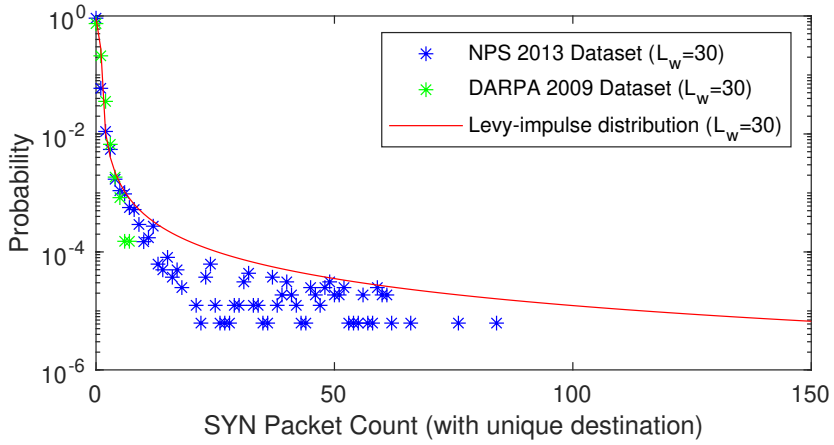
The DARPA2009 dataset contains synthesized network traffic between a /16 subnet (172.28.0.0/16) and the Internet. At some point within the trace (we estimate sometime between 90 and 100 seconds from visual analysis of the .pcap data), a SYN flood DDoS attack occurs on IP address 172.28.4.7 from approximately 100 different IPs [99]. As a result, we utilize the traffic between zero and 90 seconds (containing 8,807 nodes) in this section for SYN traffic analysis.

To evaluate the accuracy of the Lévy-impulse model against transmitted SYN traffic, the datasets are windowed for various L_w and analyzed for the number of SYN packets sent to unique destination addresses by each individual node. The results for $L_w = 5$ and $L_w = 30$ seconds are shown in Figure 4.7, and the results for $L_w = 90$ seconds are shown in Figure 4.8. Since only five seconds of the MAWI2015 dataset are analyzed, the MAWI2015

results are only shown in Figure 4.7(a). For all L_w, the Lévy-impulse model accurately characterizes the transmitted SYN data for the NPS2013 and DARPA2009 datasets. For the MAWI2015 dataset, the Lévy-impulse model accurately models the body of the data but slightly underestimates the tail by approximately 10^{-4}.



(a) SYN transmission for L_w = 5 seconds



(b) SYN transmission for L_w = 30 seconds

Figure 4.7. Accuracy of LI model for SYN traffic sent from nodes in the NPS2013, MAWI2015, and DARPA2009 datasets for L_w = 5 and L_w = 30 seconds.

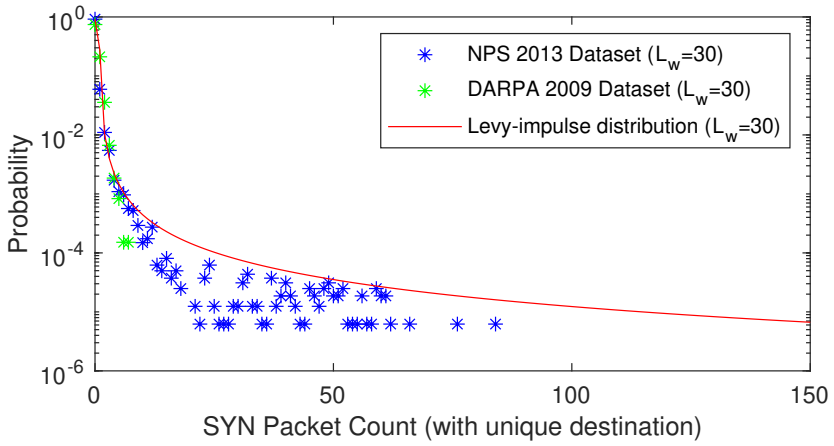
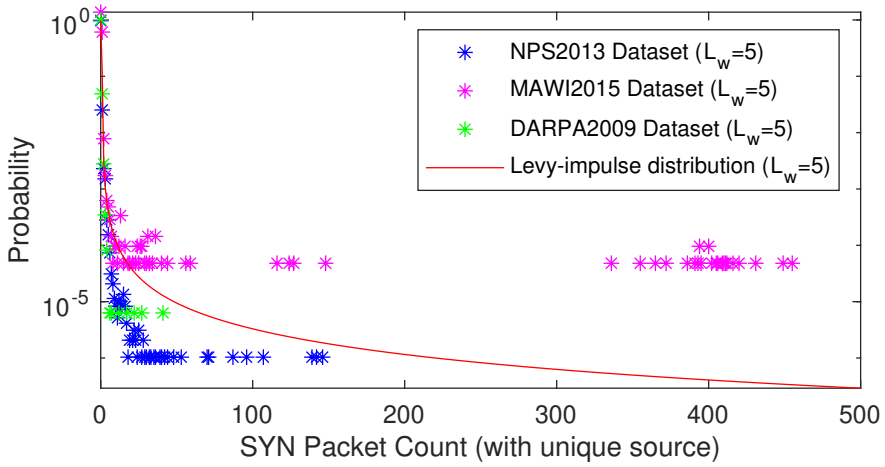


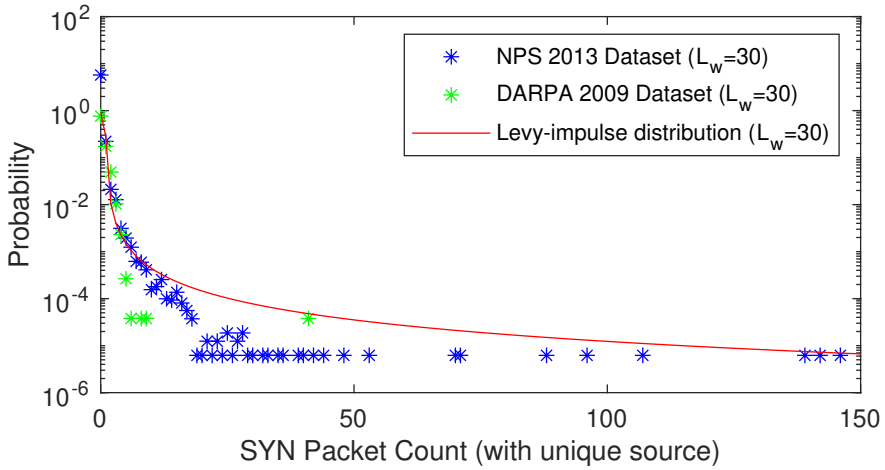
Figure 4.8. Accuracy of LI model for SYN traffic sent from nodes in the NPS2013, MAWI2015, and DARPA2009 datasets for L_w = 90 seconds.

To evaluate the accuracy of the Lévy-impulse model against received SYN traffic, the datasets are windowed for various L_w and analyzed for the number of SYN packets received by each individual node from unique sending addresses in each L_w. The results for L_w = 5 and L_w = 30 seconds are shown in Figure 4.9 and the results for L_w = 90 seconds are shown in Figure 4.10. Again, since only five seconds of the MAWI2015 dataset are analyzed, the MAWI2015 results are only shown in Figure 4.9(a). Similar to the results of the transmitted SYN traffic, the Lévy-impulse model accurately characterizes the received SYN data for the NPS2013 and DARPA2009 datasets. It also accurately models the body of the data but slightly underestimates the tail of the MAWI2015 dataset by approximately 10^{-4}.

The results shown in figures 4.7-4.10 are representative of the results observed across all evaluated L_w values. From these results, the Lévy-impulse model appears to accurately model both the body and the tail of SYN traffic for smaller networks (i.e., approximately 10,000 nodes or less), but very slightly underestimates the tail for larger networks. This suggests that larger networks have heavier tails. Since the underestimation appears to be on the order of approximately 10^{-4}, the error is considered negligible for the purposes of this work.



(a) Received SYNs for L_w = 5 seconds



(b) Received SYNs for L_w = 30 seconds

Figure 4.9. Accuracy of LI model for SYN traffic received from nodes in the NPS2013, MAWI2015, and DARPA2009 datasets for L_w = 5 and L_w = 30 seconds.

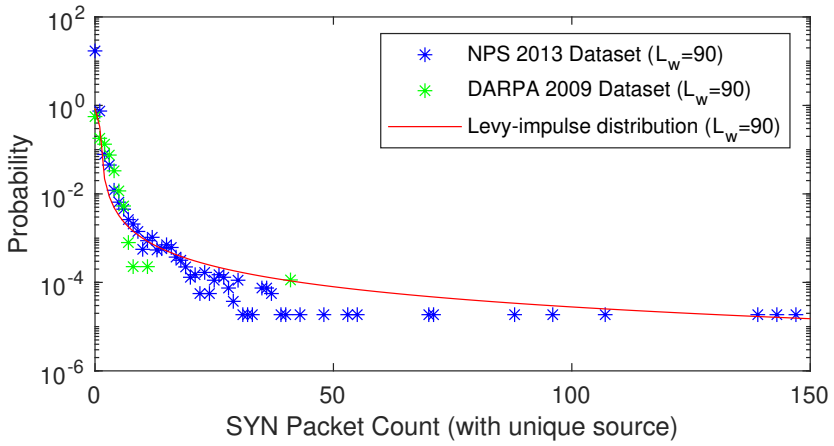


Figure 4.10. Accuracy of LI model for SYN traffic received by nodes in the NPS2013, MAWI2015, and DARPA2009 datasets for L_w = 90 seconds.

4.4 False Alarm Rate Upper Bounds

In this section, the Lévy-impulse model from Section 4.3.4 is combined with the system and nodal Markov models from Sections 4.1 and 4.2 to analyze the theoretical false alarm rate of the SGCDC system under ideal network conditions. This analysis is then utilized to describe both the upper bound and most-likely false alarm rates for both the system and the nodes being monitored by the system.

4.4.1 Theoretical Rates

The Lévy-impulse model in Equation 4.16 is combined with equations 4.5 and 4.12 to develop equations describing the theoretical false alarm rate of the SGCDC system. Substituting Equation 4.16 into Equation 4.5 yields

$$P(D^{sys}_{false}) = c^{sys}_1 \left(\int_{T^{worm}_{phantom}}^{\infty} \frac{e^{-\left(\frac{\gamma_{L_w}}{2(x-\mu_{L_w})}\right)}}{(x-\mu_{L_w})^{3/2}} dx \right)^2 + c^{sys}_2 \int_{T^{DDoS}_{phantom}}^{\infty} \frac{e^{-\left(\frac{\gamma_{L_w}}{2(x-\mu_{L_w})}\right)}}{(x-\mu_{L_w})^{3/2}} dx \qquad (4.18)$$

where

$$c^{sys}_1 = \frac{2(\gamma_{L_w})(S^{n_i}_{tx})(t_{log})(1-\xi_{L_w})^2}{2\pi L_w} \qquad (4.19)$$

and

$$c_2^{sys} = (1 - \xi_{L_w})\sqrt{\frac{\gamma_{L_w}}{2\pi}}, \tag{4.20}$$

given no worm or DDoS attack exists. Similarly, Equation 4.12 yields

$$P(D_{false}^{node}) = c_1^{node}\left(\int_{T_{phantom}^{worm}}^{\infty} \frac{e^{-\left(\frac{\gamma_{L_w}}{2(x-\mu_{L_w})}\right)}}{(x-\mu_{L_w})^{3/2}}dx\right)^2 + c_2^{node}\int_{T_{phantom}^{DDoS}}^{\infty} \frac{e^{-\left(,\frac{\gamma_{L_w}}{2(x-\mu_{L_w})}\right)}}{(x-\mu_{L_w})^{3/2}}dx, \tag{4.21}$$

where

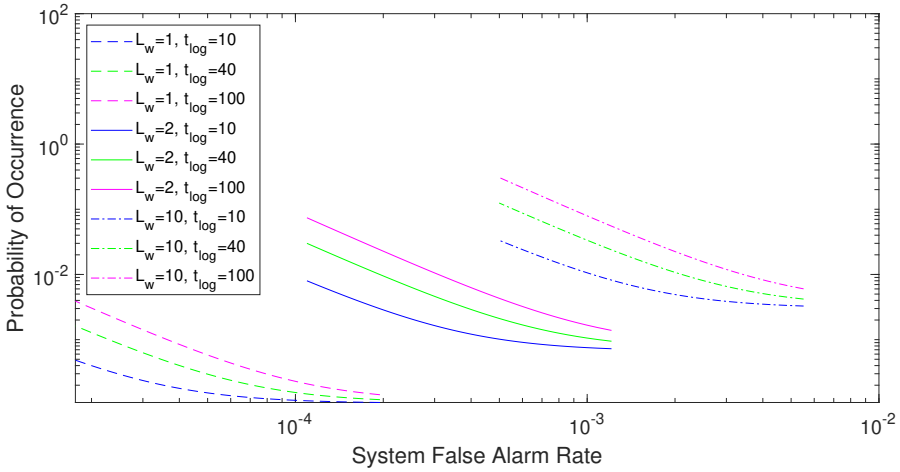
$$c_1^{node} = \left(\frac{S_{tx}^{n_i}(\gamma_{L_w})(1-\xi_{L_w})^2}{2\pi}\right) \times \left(1 + \frac{2(t_{log})}{L_w}\right), \tag{4.22}$$

and

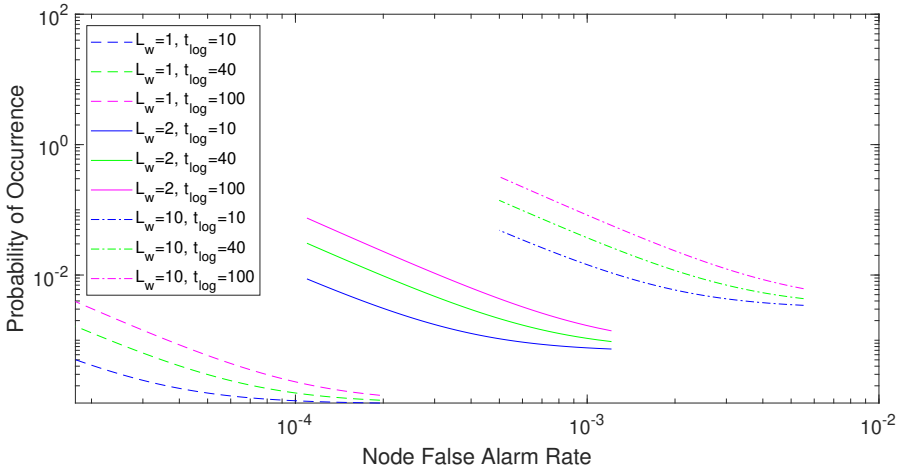
$$c_2^{node} = c_2^{sys}, \tag{4.23}$$

given no worm or DDoS attack exists.

Both $P(D_{false}^{sys})$ and $P(D_{false}^{node})$ are dependent on the set of external, input parameters (i.e., $L_w, T_{phantom}^{worm}, T_{phantom}^{DDoS}$, and t_{log}) and $S_{tx}^{n_i}$. Utilizing various sets of external, input parameters that align with the evaluated characteristics of historical worm and DDoS attacks and support the required external, input parameter relationships described by equations 3.8, 3.9, and 3.10, the theoretical false alarm rates are analyzed for determining the probability of occurrence. The probability of occurrence is determined by equations 4.18 and 4.21 for various $x = S_{tx}^{n_i}$. Given $T_{phantom}^{worm} = 5$, $T_{phantom}^{DDoS} = 15$, the results of $P(D_{false}^{sys})$ and $P(D_{false}^{node})$ are shown in Figure 4.11 for various combinations of L_w and t_{log}. Additional results from other sets of external, input parameters can be found in Appendix B.1. For both $P(D_{false}^{sys})$ and $P(D_{false}^{node})$, it is far more likely to have a false alarm rate near or equal to zero than to have a large false alarm rate. Additionally, the probability of a larger false alarm rate occurring increases as L_w and t_{log} increase. This is due to the increased amount of time within the system to allow for a node to send and/or receive a number of SYN packets equal to or greater than $T_{phantom}^{worm}$ and/or $T_{phantom}^{DDoS}$, respectively.



(a) $P(D^{sys}_{false})$ results for $T^{worm}_{phantom}$ = 5 and $T^{DDoS}_{phantom}$ = 15

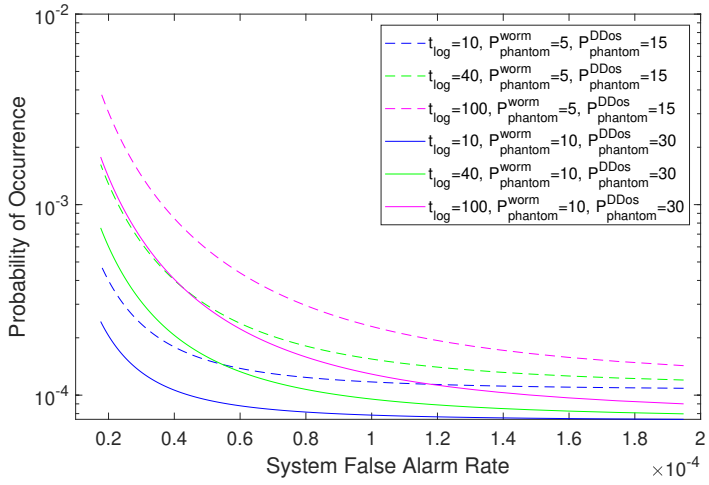


(b) $P(D^{node}_{false})$ results for $T^{worm}_{phantom}$ = 5 and $T^{DDoS}_{phantom}$ = 15

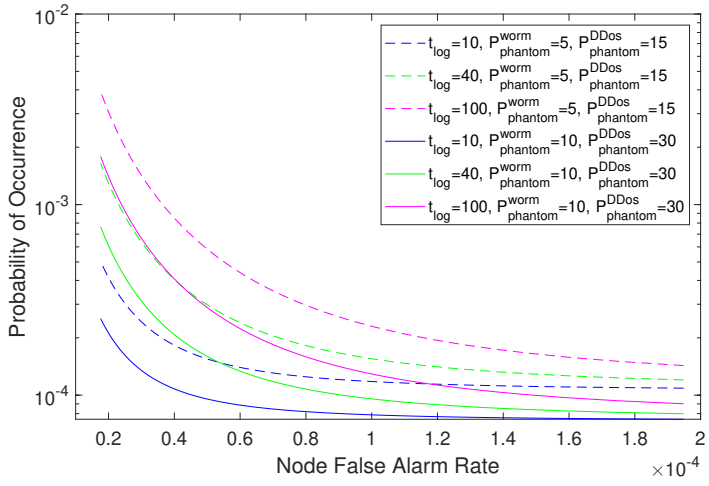
Figure 4.11. Probability of given system and nodal false alarm rates occurring for various sets of external, input parameters, given $T^{worm}_{phantom}$ = 5, $T^{DDoS}_{phantom}$ = 15, and the units of L_w and t_{log} are seconds.

Given L_w = 1 second, the results of $P(D^{sys}_{false})$ and $P(D^{node}_{false})$ are shown in Figure 4.12 for various combinations of t_{log}, $T^{worm}_{phantom}$ and $T^{DDoS}_{phantom}$. Additional results from other sets of external, input parameters can be found in Appendix B.1. Again, the results demonstrate that it is far more likely to have a false alarm rate equal to, or near, zero than to have a large

false alarm rate. Additionally, for a given L_w and t_{log}, the probability of a larger false alarm rate occurring decreases as $T^{worm}_{phantom}$ and $T^{DDoS}_{phantom}$ increase. This is due to the increased number of SYN packets equal to or greater than $T^{worm}_{phantom}$ and/or $T^{DDoS}_{phantom}$ that a node must send and/or receive, respectively, to trigger a false alarm.



(a) $P(D^{sys}_{false})$ results for L_w = 1 second



(b) $P(D^{node}_{false})$ results for L_w = 1 second

Figure 4.12. Probability of given system and nodal false alarm rates occurring for various sets of external, input parameters, given L_w = 1 seconds and the unit of t_{log} is seconds.

4.4.2 Upper Bound

To develop a false alarm rate upper bound from equations 4.18 and 4.21, we assume, for the purpose of analytical simplicity, that $S_{tx}^{n_i}$ is a uniform value for all nodes that are detected crossing T_{eigen}^{worm}. Given this assumption, the false alarm rate upper bound for any set of external, input parameters is determined by setting $S_{tx}^{n_i} = N_{tot} - 1$, which represents a node sending the maximum number of SYN packets to unique destination addresses within the network. Thus, the false alarm rate upper bound for the system is given by

$$D_{UpperFalse}^{sys} = c_{1,upper}^{sys} \left(\int_{T_{phantom}^{worm}}^{\infty} \frac{e^{-\left(\frac{\gamma_{L_w}}{2(x-\mu_{L_w})}\right)}}{(x-\mu_{L_w})^{3/2}} dx \right)^2 + c_2^{sys} \int_{T_{phantom}^{DDoS}}^{\infty} \frac{e^{-\left(\frac{\gamma_{L_w}}{2(x-\mu_{L_w})}\right)}}{(x-\mu_{L_w})^{3/2}} dx \quad (4.24)$$

where

$$c_{1,upper}^{sys} = \frac{2(\gamma_{L_w})(N_{tot}-1)(t_{log})(1-\xi_{L_w})^2}{2\pi L_w}, \quad (4.25)$$

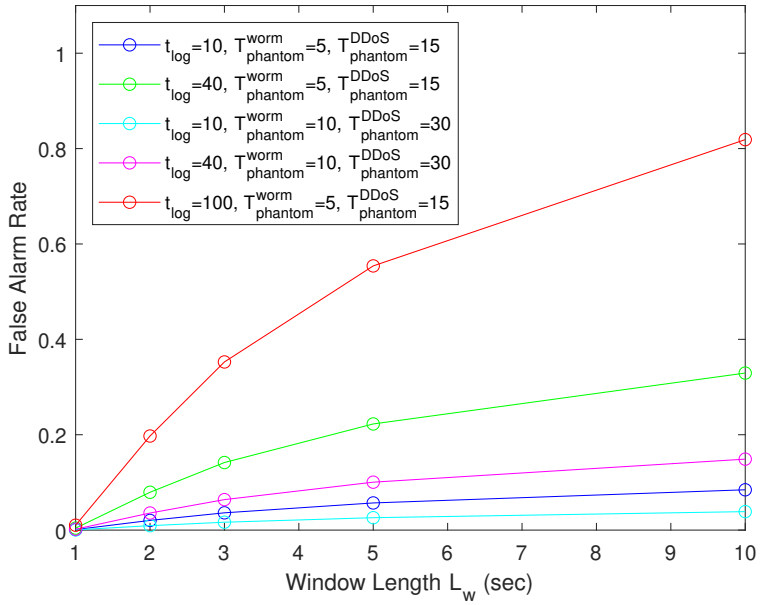
and the false alarm rate upper bound of a node being monitored by the system is given by

$$D_{UpperFalse}^{node} = c_{1,upper}^{node} \left(\int_{T_{phantom}^{worm}}^{\infty} \frac{e^{-\left(\frac{\gamma_{L_w}}{2(x-\mu_{L_w})}\right)}}{(x-\mu_{L_w})^{3/2}} dx \right)^2 + c_2^{node} \int_{T_{phantom}^{DDoS}}^{\infty} \frac{e^{-\left(\frac{\gamma_{L_w}}{2(x-\mu_{L_w})}\right)}}{(x-\mu_{L_w})^{3/2}} dx, \quad (4.26)$$

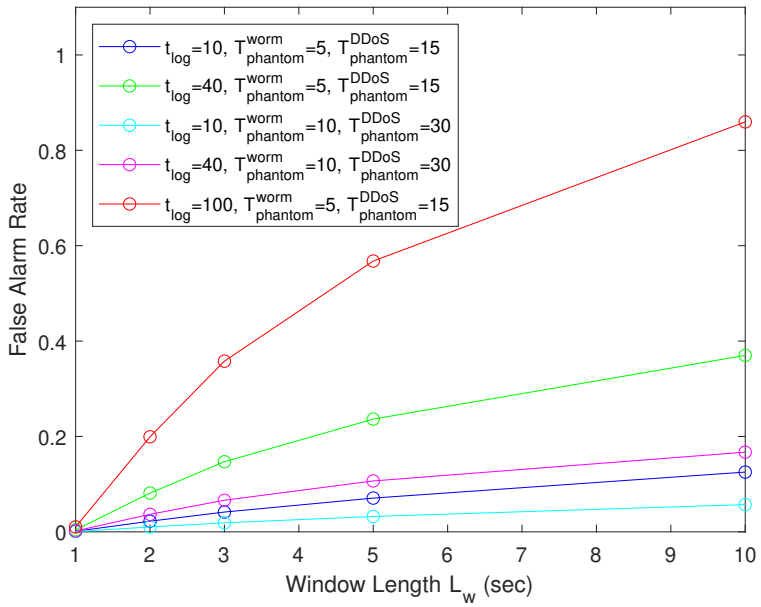
where

$$c_{1,upper}^{node} = \left(\frac{(N_{tot}-1)(\gamma_{L_w})(1-\xi_{L_w})^2}{2\pi} \right) \times \left(1 + \frac{2(t_{log})}{L_w} \right). \quad (4.27)$$

Equations 4.24 through 4.27 are utilized to evaluate the theoretical false alarm rate upper bounds for various sets of external, input parameters. For evaluation purposes, the number of nodes in the network is set to 1,341, which is the total number of nodes in the NPS2013 dataset. The $D_{UpperFalse}^{sys}$ and $D_{UpperFalse}^{node}$ results are shown in Figure 4.13 for various combinations of t_{log}, $T_{phantom}^{worm}$ and $T_{phantom}^{DDoS}$ that align with evaluated characteristics of numerous historical worm and DDoS attacks and support the required external, input parameter relationships described by equations 3.8, 3.9, and 3.10.



(a) Theoretical $D^{sys}_{UpperFalse}$ results



(b) Theoretical $D^{node}_{UpperFalse}$ results

Figure 4.13. Theoretical upper bound false alarm rates for NPS2013 dataset for various combinations of t_{log}, $T^{worm}_{phantom}$ and $T^{DDoS}_{phantom}$.

The results indicate that both $D^{sys}_{UpperFalse}$ and $D^{node}_{UpperFalse}$ increase as L_w and/or t_{log} increase or as $T^{worm}_{phantom}$ and $T^{worm}_{phantom}$ decrease. Similar to the results for $P(D^{sys}_{false})$ and $P(D^{node}_{false})$, if L_w or t_{log} increase, the amount of time within the system to allow for a node to send and/or receive number of SYN packets equal to or greater than $T^{worm}_{phantom}$ and/or $T^{DDoS}_{phantom}$, respectively, also increases, making a false alarm more probable. If $T^{worm}_{phantom}$ and $T^{worm}_{phantom}$ decrease, the number of SYN packets equal to or greater than $T^{worm}_{phantom}$ and/or $T^{DDoS}_{phantom}$ that a node must send and/or receive, respectively, to trigger a false alarm also decreases, making a false alarm more probable.

4.4.3 Most-Likely Rates

To determine the most-likely false alarm rate from equations 4.18 and 4.21, we assume for the purpose of analytical simplicity that $S^{n_i}_{tx}$ is a uniform value for all nodes that are detected crossing T^{worm}_{eigen}. Given this assumption, the most-likely false alarm rate for any set of external, input parameters is determined by setting $S^{n_i}_{tx} = T^{worm}_{phantom}$, which represents a node sending the minimum number of SYN packets to unique destination addresses while still being detected as crossing T^{worm}_{eigen}. Thus, the most-likely false alarm rate for the system can be shown to be

$$D^{sys}_{LikelyFalse} = c^{sys}_{1,likely}\left(\int_{T^{worm}_{phantom}}^{\infty} \frac{e^{-\left(\frac{\gamma_{L_w}}{2(x-\mu_{L_w})}\right)}}{(x-\mu_{L_w})^{3/2}}dx\right)^2 + c^{sys}_2 \int_{T^{DDoS}_{phantom}}^{\infty} \frac{e^{-\left(\frac{\gamma_{L_w}}{2(x-\mu_{L_w})}\right)}}{(x-\mu_{L_w})^{3/2}}dx \quad (4.28)$$

where

$$c^{sys}_{1,likely} = \frac{2(\gamma_{L_w})(T^{worm}_{phantom})(t_{log})(1-\xi_{L_w})^2}{2\pi L_w}, \quad (4.29)$$

and the most-likely false alarm rate a node being monitored by the system is given by

$$D^{node}_{LikelyFalse} = c^{node}_{1,likely}\left(\int_{T^{worm}_{phantom}}^{\infty} \frac{e^{-\left(\frac{\gamma_{L_w}}{2(x-\mu_{L_w})}\right)}}{(x-\mu_{L_w})^{3/2}}dx\right)^2 + c^{node}_2 \int_{T^{DDoS}_{phantom}}^{\infty} \frac{e^{-\left(\frac{\gamma_{L_w}}{2(x-\mu_{L_w})}\right)}}{(x-\mu_{L_w})^{3/2}}dx, \quad (4.30)$$

where

$$c^{node}_{1,likely} = \left(\frac{T^{worm}_{phantom}(\gamma_{L_w})(1-\xi_{L_w})^2}{2\pi} \right) \times \left(1 + \frac{2(t_{log})}{L_w} \right). \tag{4.31}$$

Equations 4.28 through 4.31 are utilized to evaluate the most-likely false alarm rates for various sets of external, input parameters. For evaluation purposes, the number of nodes in the network is set to 1,341, which is the total number of nodes in the NPS2013 dataset. The $D^{sys}_{LikelyFalse}$ results are shown in Figure 4.14 and the $D^{node}_{LikelyFalse}$ results are shown in Figure 4.15 for various combinations of external input parameters that align with evaluated characteristics of numerous historical worm and DDoS attacks, and support the required external, input parameter relationships described by equations 3.8, 3.9, and 3.10.

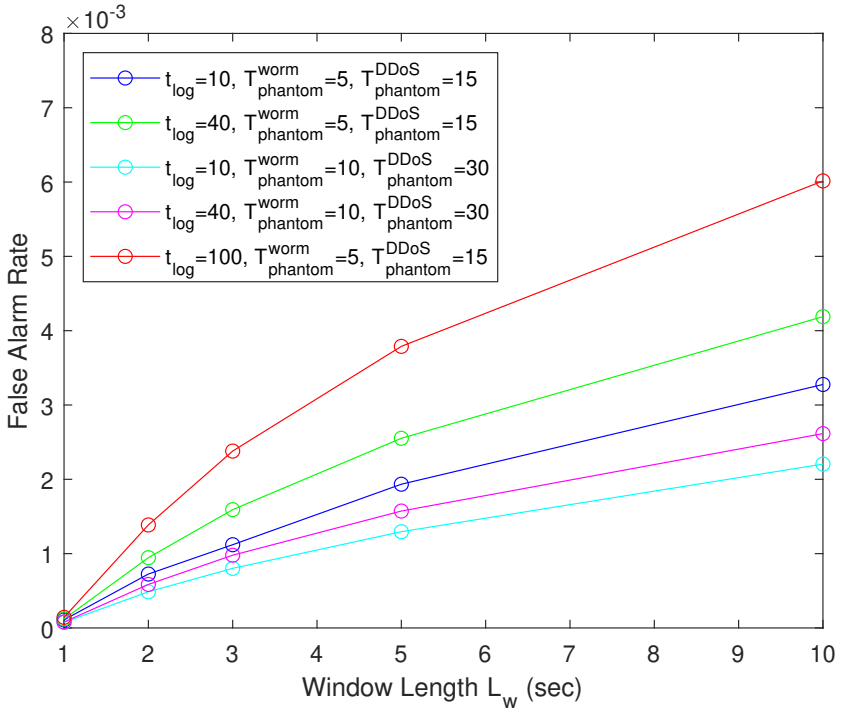


Figure 4.14. System theoretical most-likely false alarm rates for NPS2013 dataset for various combinations of t_{log}, $T^{worm}_{phantom}$ and $T^{DDoS}_{phantom}$.

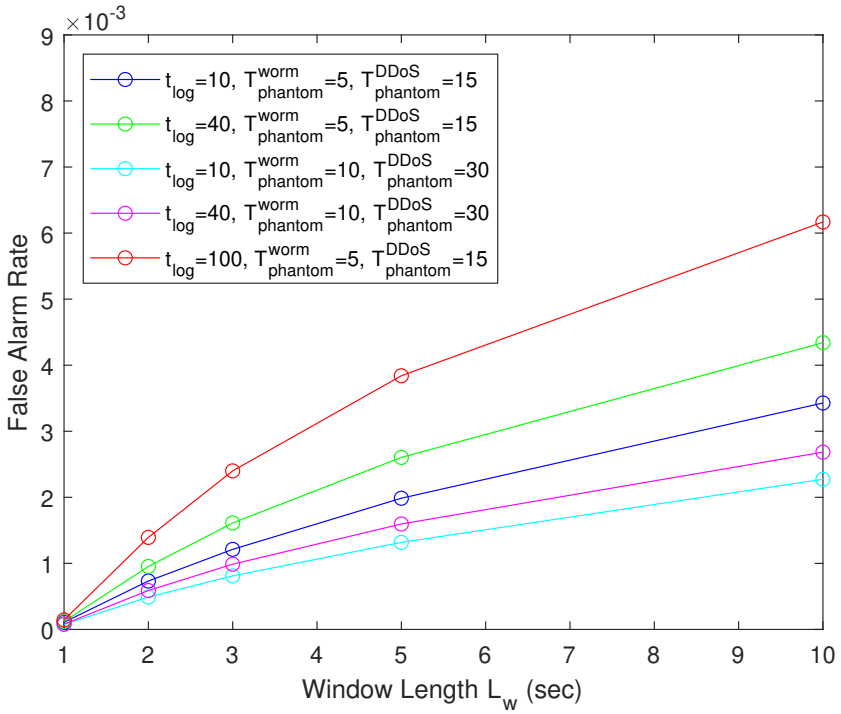


Figure 4.15. Nodal theoretical most-likely false alarm rates for NPS2013 dataset for various combinations of t_{log}, $T^{worm}_{phantom}$ and $T^{DDoS}_{phantom}$.

The results indicate that both $D^{sys}_{LikelyFalse}$ and $D^{node}_{LikelyFalse}$ increase as L_w and/or t_{log} increase because the amount of time within the system to allow for a node to send and/or receive number of SYN packets equal to or greater than $T^{worm}_{phantom}$ and/or $T^{DDoS}_{phantom}$, respectively, also increases. Additionally, $D^{sys}_{LikelyFalse}$ and $D^{node}_{LikelyFalse}$ both increase as as $T^{worm}_{phantom}$ and $T^{worm}_{phantom}$ decrease because the number of SYN packets equal to or greater than $T^{worm}_{phantom}$ and/or $T^{DDoS}_{phantom}$ that a node must send and/or receive, respectively, to trigger a false alarm also decreases.

4.5 State Visitation Analysis

To provide an additional layer of analysis to the theoretical false alarm rates, the state diagrams representing the proposed SGCDC system are analyzed for the expected state visitation count to determine the theoretical length of time between false alarms. Utilizing

the system and nodal state diagrams in Sections 4.1 and 4.2, respectively, state visitation analysis provides a theoretical count of the number of times each of the non-absorbing (i.e., transient) states are visited prior to absorption. Since a single visitation occurs during each L_w, the length of time between false alarms can be determined from the total number of transient state visitations prior to absorption.

In accordance with [89], the state visitation count is determined by the fundamental matrix N_{fund} of an absorbing Markov chain, defined as

$$N_{fund} = (I_{n_s-1} - Q_i), \tag{4.32}$$

where Q_i is the $(n_s - 1) \times (n_s - 1)$ upper left partition of the probability transition matrix P_{trans} [89], P_{trans} is the $n_s \times n_s$ matrix containing all $p_{i,j}$ of the absorbing Markov chain [100], I_{n_s} is the identity matrix of size $(n_s - 1) \times (n_s - 1)$, and n_s is the number of states in the absorbing Markov chain [89].

For the SGCDC system state model (see Figure 4.1), the system probability transition matrix is given by

$$P_{trans}^{sys} = \begin{bmatrix} 1 - p_{1,2}^{sys} & p_{1,2}^{sys} \\ 0 & 1 \end{bmatrix}, \tag{4.33}$$

where $p_{1,2}^{sys}$ is defined in Equation 4.4. The 1×1 upper left partition of P_{trans}^{sys} is given by

$$Q_i^{sys} = 1 - p_{1,2}^{sys}, \tag{4.34}$$

and the system fundamental matrix is given by

$$N_{fund}^{sys} = (p_{1,2}^{sys})^{-1}. \tag{4.35}$$

As a result, the system theoretical length of time between false alarms can be shown to be

$$t_{FA}^{sys} = \frac{L_w}{p_{1,2}^{sys}}. \tag{4.36}$$

For the SGCDC nodal state model (see Figure 4.2), the nodal probability transition matrix

is given by

$$P_{trans}^{node} \begin{bmatrix} 1-(p_{1,2}^{node}+p_{1,3}^{node}) & p_{1,2}^{node} & p_{1,3}^{node} \\ p_{2,1}^{node} & p_{2,2}^{node} & p_{2,3}^{node} \\ 0 & 0 & 1 \end{bmatrix}, \tag{4.37}$$

where $p_{1,2}^{node}$ is defined in Equation 4.7, $p_{1,3}^{node}$ is defined in Equations 4.6, $p_{2,1}^{node}$ is defined in Equations 4.10, $p_{2,2}^{node}$ is defined in Equation 4.11, and $p_{2,3}^{node}$ is defined in Equation 4.9. The 2×2 upper left partition of P_{trans}^{node} is defined as

$$Q_i^{node} = \begin{bmatrix} 1-(p_{1,2}^{node}+p_{1,3}^{node}) & p_{1,2}^{node} \\ p_{2,1}^{node} & p_{2,2}^{node} \end{bmatrix}, \tag{4.38}$$

and the nodal fundamental matrix is given by

$$N_{fund}^{node} = (I_2 - Q_i^{node})^{-1}. \tag{4.39}$$

As a result, the nodal theoretical length of time between false alarms can be shown to be

$$t_{FA}^{node} = L_w \times \left(\sum_{i=1}^{n_s-1} \sum_{j=1}^{n_s-1} n_{i,j}^{fund} \right), \tag{4.40}$$

where $n_{i,j}^{fund}$ is the i^{th} element of the j^{th} column of N_{fund}^{node}.

For both t_{FA}^{sys} and t_{FA}^{node}, the length of time between false alarms is directly dependent on $S_{tx}^{n_i}$ and the set of external, input parameters of the system (i.e., L_w, $T_{phantom}^{worm}$, $T_{phantom}^{DDoS}$, and t_{log}). Assuming that $S_{tx}^{n_i}$ has a uniform value for all nodes that cross T_{eigen}^{worm}, the t_{FA}^{sys} and t_{FA}^{node} for various sets of external, input parameters is listed in Table 4.3 using the 1341 node network from the NPS2013 dataset. The largest t_{FA}^{sys} and t_{FA}^{node} occur when $L_w = 1$ second, $t_{log} = 10$ second, $T_{phantom}^{worm} = 10$, and $T_{phantom}^{DDoS} = 30$. These results indicate that the length of time between false alarms can be increased by decreasing t_{log} and L_w and/or by increasing $T_{phantom}^{worm}$ and $T_{phantom}^{DDoS}$.

Table 4.3. Example theoretical times between false alarms for SGCDC system using the 1341 node network from the NPS2013 dataset.

L_w (sec)	t_{log} (sec)	$T_{phantom}^{worm}$	$T_{phantom}^{DDoS}$	$S_{tx}^{n_i}$	t_{FA}^{sys} (hrs)	t_{FA}^{node} (hrs)
1	10	5	15	5	2.56	2.96
1	40	5	15	5	2.31	2.95
1	100	5	15	5	1.95	2.92
1	10	10	30	10	3.64	4.34
1	40	10	30	10	3.21	4.32
1	100	10	30	10	2.59	4.29
2	10	5	15	5	0.77	0.95
2	40	5	15	5	0.59	0.93
2	100	5	15	5	0.40	0.91
2	10	10	30	10	1.07	1.39
2	40	10	30	10	0.77	1.38
2	100	10	30	10	0.50	1.35

In summary, this chapter explored the theoretical system and nodal false alarms rates for the proposed SGCDC using absorbing Markov chains. Analysis of the false alarm rates required the development of the novel Lévy-impulse model to describe network SYN traffic. The absorbing Markov chains were also analyzed for state visitation count to determine the theoretical length of time between false alarms. While it is important to understand the theoretical system performance and limitations under ideal network conditions, it is just as important to understand how adverse network conditions affect system performance. The proposed SGCDC system is susceptible to network noise and congestion, which may impact both detection and false alarm rates. The effect of these adverse conditions on the spectral graph detection component of the proposed system is examined in the next chapter.

CHAPTER 5:
Detection among Noise and Congestion

In this chapter, we examine how perturbations in the adjacency matrix impact the dual-basis to develop an understanding of how network noise and congestion affect the spectral graph detection mechanism employed in the proposed SGCDC system. We begin by describing three traffic scenarios that would impact the ability of the proposed system to detect an attack. We find that only one case requires further analysis of the dual-basis with respect to perturbations in the adjacency matrix to determine the impact on the SGCDC system. In support of that case, we then explore the impact of adjacency matrix perturbation on the dual-basis using the eigenvector and eigenvalue perturbation theorems discussed in Chapter 2. Next, we propose a novel method of analyzing the perturbation of the eigenvectors of the nodal basis. We conclude this chapter with a discussion on how the perturbation findings of the dual-basis affect the proposed SGCDC system.

5.1 Network Perturbation Cases

In this section, we describe computer network scenarios in which network traffic is impacted by either congestion or the presence of noise resulting in a perturbation of the data utilized by the proposed SGCDC system to perform detection. From the proposed system operating under ideal network conditions as depicted in Figure 5.1, there are three network congestion/noise scenarios that will affect the SGCDC system: 1) all traffic flows are not received, 2) some of the traffic flows are received, and 3) errors exist within some of the traffic flows.

In context of the network in Figure 5.1, severe network congestion or a cyber attack directed at the SGCDC system can result in the system not receiving all of the traffic flows in the network as shown in Figure 5.2. In this scenario, all nodes will have $v_{i,j}^{pri}$ in an eigenspectrum index less than the eigenspectrum index associated with T_{eigen}^{worm} and T_{eigen}^{DDoS}. If no attack is present, then this scenario has no effect on the system; however, if a worm or DDoS attack exists within the network, the SGCDC system will be unable to detect the attack.

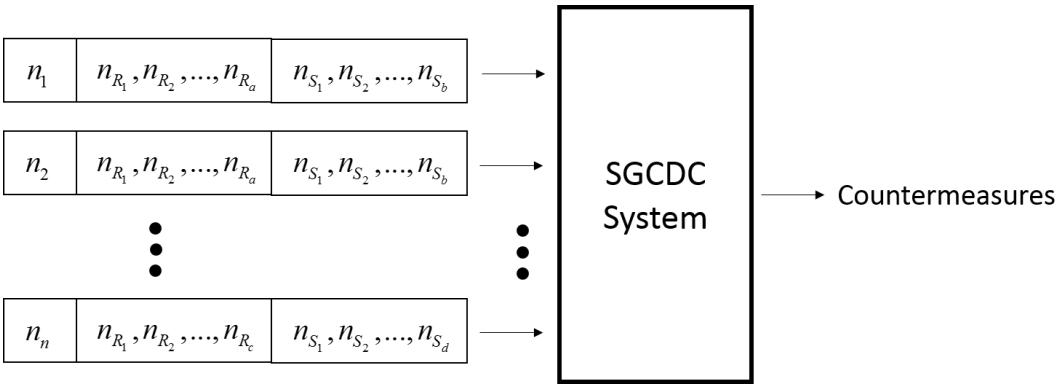


Figure 5.1. SGCDC system operating under ideal network conditions.

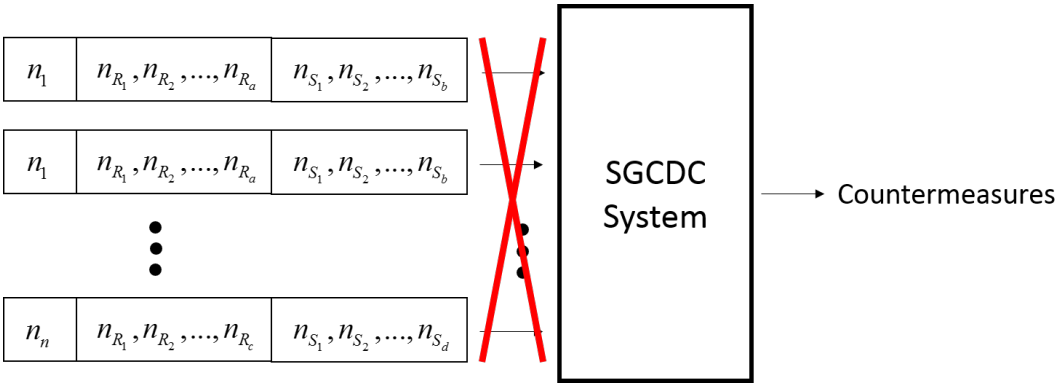


Figure 5.2. SGCDC system operating under severe network congestion or cyber attack resulting in the system not receiving all traffic flows.

In context of the network in Figure 5.1, network congestion or a cyber attack directed at the SGCDC system can result in the system receiving only a portion of the traffic flows as shown in Figure 5.3. If a worm attack exists within the network, then one of three cases result from the SGCDC system receiving only a portion of the traffic flows. First, all of the flows of the infected node and the flows associated with the nodes that the infected node sent a SYN to are not received. Consequently, the SGCDC system is unable to detect the infected node. Second, the of the infected node flows are received and/or all of the flows associated with the nodes that the infected node sent a SYN to are received. This case has no effect on the ability of the system to detect the infected node. Third, the flows of the infected node are not received and only a portion of the flows associated with the nodes that the infected node sent a SYN to are received. For this case, the ability of the SGCDC

system to detect the infected node is dependent on the number of node flows received and requires further analysis on the effects of adjacency matrix perturbation on the dual-basis.

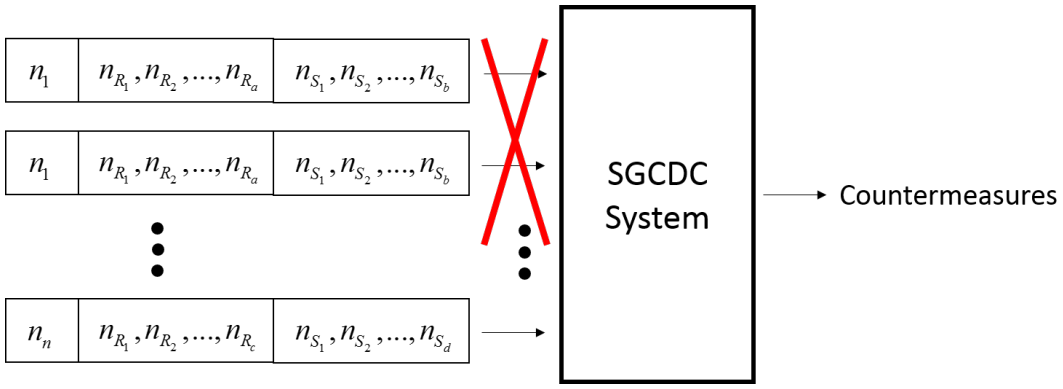


Figure 5.3. SGCDC system operating under network congestion or cyber attack resulting in the system not receiving a portion of the traffic flows.

If a DDoS attack exists within the network in Figure 5.3, then one of three cases result from the SGCDC system receiving only a portion of the traffic flows. First, the flows of the attacked node and all of the flows of the attacking nodes are not received. Consequently, the SGCDC system is unable to detect the attacked node. Second, the flows of the attacked node are received and/or all of the flows of the attacking nodes are received. This case has no affect on the ability of the system to detect the attacked node. Third, the flows of the attacked node are not received and only a portion of the flows of the attacking nodes are received. Similar to the worm attack, the ability of the SGCDC system to detect the attacked node in this case is dependent on the number of node flows received and requires further analysis on the effects of adjacency matrix perturbation on the dual-basis.

In context of the network in Figure 5.1, noise within the network results in bit errors (i.e., flipped bits) in either the payload or frame check sequence. While typical TCP protocols would disregard data with errors, we analyze both the typical TCP response of distrusting the data and the atypical response of trusting the data to provide comprehensive analysis of the system. We also recognize the possibility that a case exists where the TCP protocols might find no errors even though bit errors exist, which would produce synonymous results as trusting data with errors. Given this understanding, the SGCDC system receives all of the traffic flows but has to decide whether to trust the data contained in certain flows, as depicted in Figure 5.4. The system can either accept that errors exist and keep all flows for

analysis or not trust the flows that contain errors and disregard those flows.

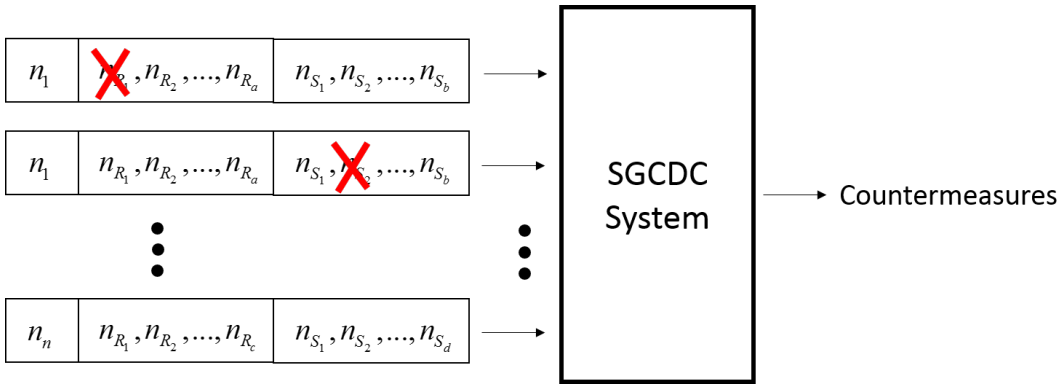


Figure 5.4. SGCDC system operating under network noise resulting in bit errors in the payload.

If the proposed SGCDC system is initialized to trust the data, then three cases exist with respect to errors. First, errors exist and are ignored in the section of the payload describing flows received. Second, errors exist and are ignored in the section of the payload describing flows sent. Third, errors exist and are ignored in the frame check sequence. Given these cases, there is only one situation given a worm attack and one situation given a DDoS attack that could affect the SGCDC system. Should a worm attack exist in the network, if one of the offspring nodes is altered in the payload and that offspring is the second infected node, then the system will not detect the worm attack. Should a DDoS attack exist in the network, if one or more of the attacking nodes is altered in the payload, then the system will detect the attack but the response mechanism will produce countermeasures against the wrong node(s).

If the proposed SGCDC system is initialized to distrust the data, then two cases exist with respect dropped data given a worm attack exists in the network depicted in Figure 5.4. First, the flows of the infected node and all of the flows of its offspring are not trusted and discarded. Consequently, the SGCDC system is unable to detect the infected node. Second, the flows of the infected node and some of the flows of its offspring are not trusted and dropped. For this case, the ability of the SGCDC system to detect the infected node is the same as the third worm and DDoS cases for the network in Figure 5.3 (system receiving only a portion of the traffic flows). In other words, attack detection is dependent on the number of node flows that are discarded. This requires further analysis on the effects of

adjacency matrix perturbation on the dual basis.

If the proposed SGCDC system is initialized to distrust the data, then two cases exist with respect dropped data given a DDoS attack exists in the network depicted in Figure 5.4. First, the flows of the attacked node and all of the flows of its attackers are not trusted and dropped. As a result, the SGCDC system fails to detect the attacked node. Second, the flows of the attacked node and some of the flows of its attackers are not trusted and discarded. Given this scenario, the ability of the SGCDC system to detect the infected node is the same as the third worm and DDoS cases for the network in Figure 5.3 (system receiving only a portion of the traffic flows); attack detection is dependent on the number of node flows that are disregarded and requires further analysis on the effects of adjacency matrix perturbation on the dual-basis.

In summary, the only case across all three network noise/congestion scenarios that requires further analysis is when a certain number of flows of an individual node are perturbed in the adjacency matrix. The effects of this specific matrix perturbation on the spectral graph detection mechanism are unclear. The remaining sections in this chapter are dedicated to analyzing this perturbation in order to determine the impact on the SGCDC system.

5.2 Eigenvalues of Perturbed Adjacency Matrix

In support of the network case that requires further analysis, we focus in this section on the $\Delta\lambda_j$ (defined in Equation 2.19) that results from the links associated with one node being perturbed in the adjacency matrix. We begin by developing a theoretical upper bound of $\Delta\lambda_j$ for K_n using Weyl's theorem. Next, we analyze the accuracy of the upper bound by simulating link removals for one node within K_n. We conclude this section by removing the K_n constraint and comparing the upper bound of $\Delta\lambda_j$ for K_n to the simulation results of $\Delta\lambda_j$ for simple graphs.

5.2.1 Theoretical Upper Bound for Complete Graphs

The theoretical upper bound of $\Delta\lambda_j$ for K_n is determined from the eigenvalue stability inequality defined in Theorem 1 in Chapter 2. In support of the network case requiring further analysis, we assume that the link perturbation in the adjacency matrix is confined to a single node. If the number of link perturbations $l_{perturb}$ is equal to $n-1$ (i.e., all links

for one node are disconnected), Q_{error} is a $n \times n$ star graph S_n, which has $\lambda = 0$ with a multiplicity of one, $\lambda = 1$ with a multiplicity of $n - 2$, and $\lambda = n$ with multiplicity of one [101]. As a result, we have

$$\lambda_{max}(S_n) = l_{perturb} + 1, \tag{5.1}$$

where λ_{max} is defined in Equation 2.22. If $1 < l_{perturb} < n - 1$ (i.e., some of the links for one node are disconnected) Q_{error} results in a $n \times n$ error star graph S_n^{error} where the number of links in the star is equal to $l_{perturb}$. The S_n^{error} has $\lambda = 0$ with a multiplicity of $n - 1$ and $\lambda = l_{perturb}$ with a multiplicity of one [101]. As a result, we have

$$\lambda_{max}(S_n^{error}) = l_{perturb}. \tag{5.2}$$

If $l_{perturb} = 1$ (i.e., only one link is disconnected) Q_{error} is a $n \times n$ error path graph P_n^{error} where only one path exists between two nodes. For a path graph P_n [101], we have

$$\lambda = 2 - cos\left(\frac{2\pi k}{d}\right), \tag{5.3}$$

resulting in $\lambda_{max} = l_{perturb}$. Given $\lambda_{max}(P_n^{error}) \leq \lambda_{max}(P_n)$, then

$$\lambda_{max}(P_n^{error}) \leq l_{perturb}. \tag{5.4}$$

Given equations 5.1, 5.2, and 5.4, the upper bound of $\Delta\lambda_j$ for K_n is defined as

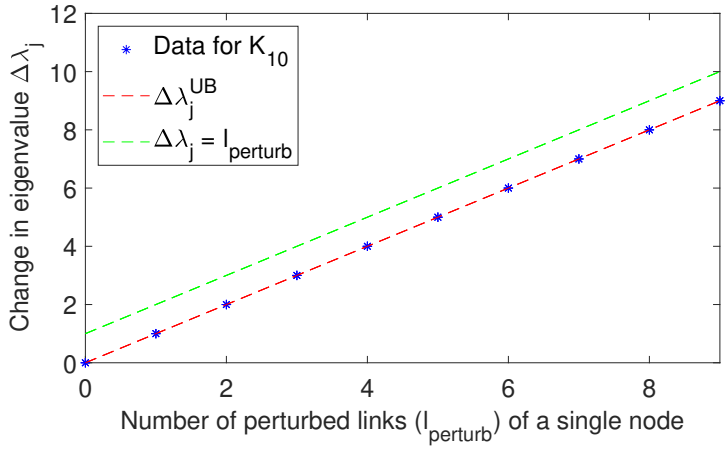
$$\Delta\lambda_j^{UB} = l_{perturb} + 1. \tag{5.5}$$

This implies a linear relationship between $\Delta\lambda_j$ and the number of $l_{perturb}$ in the adjacency matrix of K_n.

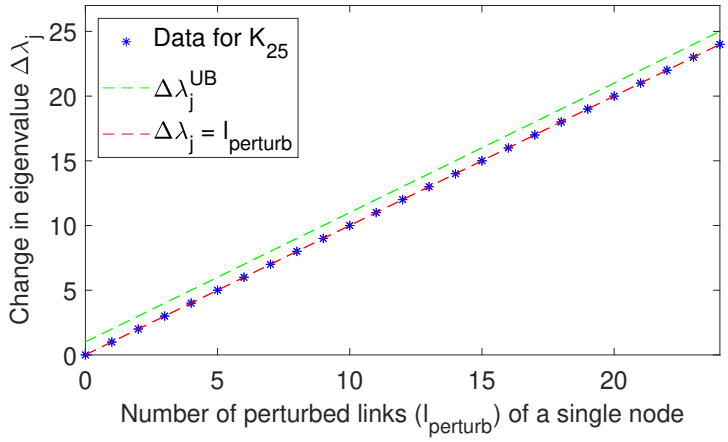
5.2.2 Simulation Results for K_n

The accuracy of Equation 5.5 is analyzed by simulating link removals from one node of K_n for various n between five and 500. The results of K_{10} and K_{25} are shown in Figure 5.5, where the blue asterisks are the data results from the simulation, the green dashed line is $\Delta\lambda_j^{UB}$, and the red dashed line is the equation $\Delta\lambda_j = l_{perturb}$. The simulation results for all

evaluated n including those shown in Figure 5.5 produce a $\Delta\lambda_j < \Delta\lambda_j^{UB}$.



(a) Link removals from one node in K_{10}



(b) Link removals from one node in K_{25}

Figure 5.5. Comparison of $\Delta\lambda_j^{UB}$ to the $\Delta\lambda_j$ results from K_{10} and K_{25}, where links are removed from a single node.

From these results, the theoretical $\Delta\lambda_j^{UB}$, defined in Equation 5.5, is validated as an upper bound; however, the results also suggest that a more accurate (i.e., tighter-fit) upper bound of $\Delta\lambda_j$ for K_n is given by

$$\Delta\lambda_j^{UB,sim} = l_{perturb}. \tag{5.6}$$

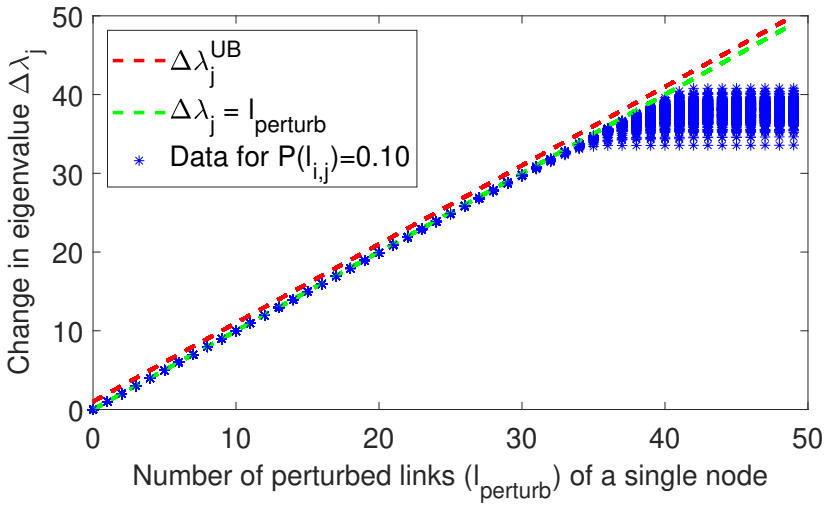
Additional research needs to be conducted to validate Equation 5.6, which is outside the scope of this dissertation.

5.2.3 Simulation Results for Simple Graphs

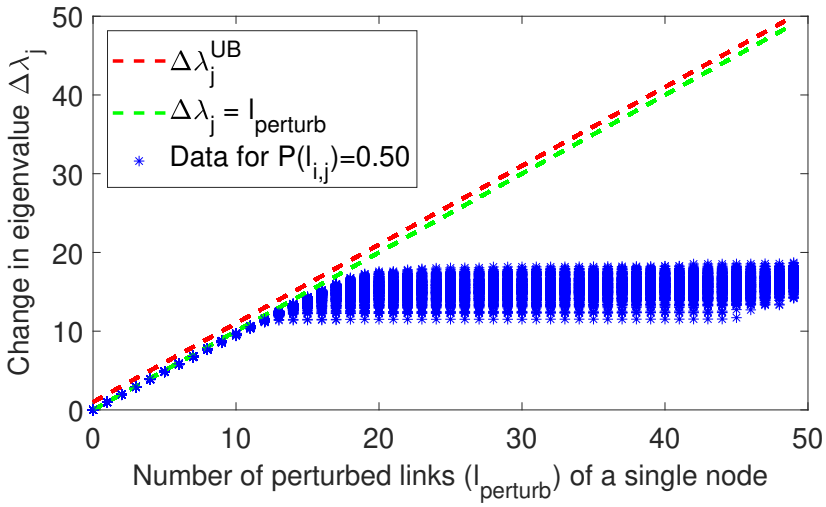
The accuracy of Equation 5.5 as an upper bound for $\Delta\lambda_j$ in simple graphs is analyzed by simulating link removals from one node in a variety of graphs with different levels of connectivity. For each simulation, 300 random simple graphs containing 50 nodes are formed using a static probability of link connection $P(l_{i,j})$ between each pair of nodes, one node in each graph is selected and edited to be completely connected (i.e., have links connecting it to every other node in the network), and $\Delta\lambda_j$ is evaluated as the links of the selected node are disconnected. Simulation results are obtained for $P(l_{i,j})$ between 0.01 and 0.99.

To demonstrate the effects of connectivity on $\Delta\lambda_j$, the results of $P(l_{i,j}) = 0.1$ and $P(l_{i,j}) = 0.5$ are shown in Figure 5.6, and the results of $P(l_{i,j}) = 0.75$ and $P(l_{i,j}) = 0.99$ are shown in Figure 5.7. In figures 5.6 and 5.7, the blue asterisks are the data results from the simulation, the green dashed line is $\Delta\lambda_j^{UB}$, and the red dashed line is $\Delta\lambda_j^{UB,sim}$. Additional results for other $P(l_{i,j})$ can be found in Appendix B.2.

From the simulation results, $\Delta\lambda_j$ is impacted differently depending on the connectivity of the simple graph. In general, graphs that are sparsely connected or densely connected have a $\Delta\lambda_j$ close to $\Delta\lambda_j^{UB}$ for all $l_{perturb}$. Graphs that are neither sparsely nor densely connected have a $\Delta\lambda_j$ close to $\Delta\lambda_j^{UB}$ only for small $l_{perturb}$ and a $\Delta\lambda_j \ll \Delta\lambda_j^{UB}$ for large $l_{perturb}$. This means $\Delta\lambda_j \approx \Delta\lambda_j^{UB}$ for very small perturbations in the adjacency matrix of all simple graphs, but $\Delta\lambda_j$ is dependent upon the connectivity of the graph for any other size perturbation. Regardless, the results validate $\Delta\lambda_j^{UB}$ for perturbations in the adjacency matrix of simple graphs. It is important to also note that the majority of the simple graph results validate $\Delta\lambda_j^{UB,sim}$, but the results for $P(l_{i,j}) = 0.99$ do not validate $\Delta\lambda_j^{UB,sim}$. Additional research needs to be conducted to determine the specific cases for which $\Delta\lambda_j^{UB,sim}$ is not valid, which is outside the scope of this dissertation.

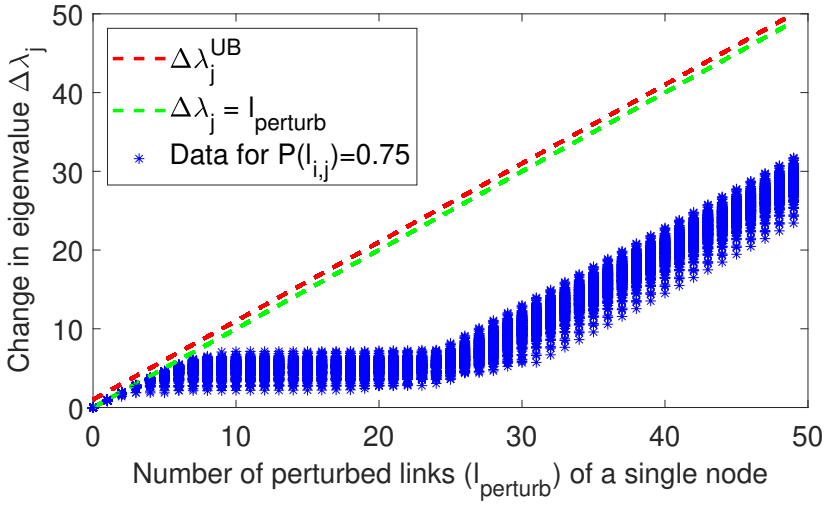


(a) $\Delta\lambda_j$ from graphs with $P(l_{i,j}) = 0.1$

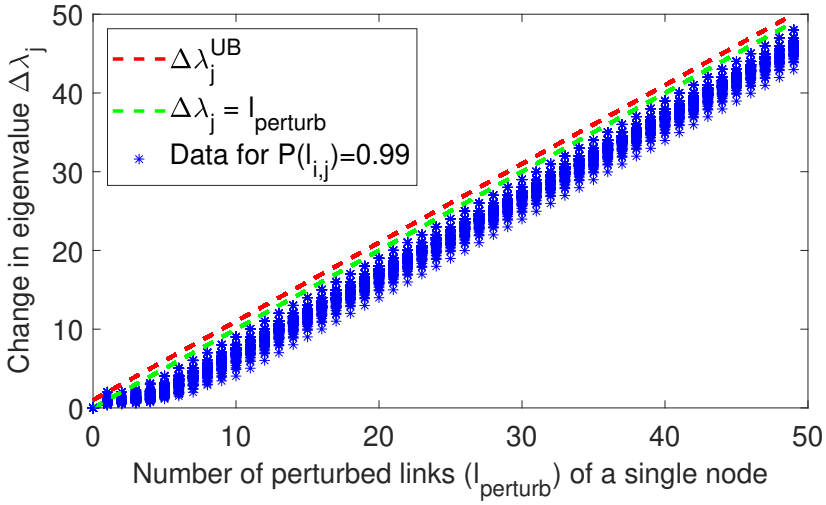


(b) $\Delta\lambda_j$ from graphs with $P(l_{i,j}) = 0.5$

Figure 5.6. Comparison of $\Delta\lambda_j^{UB}$ to the $\Delta\lambda_j$ results from a simulation of 300 random simple graphs containing 50 nodes and $P(l_{i,j}) = \{0.1, 0.5\}$ where links are removed from a single, fully-connected node.



(a) $\Delta\lambda_j$ from graphs with $P(l_{i,j}) = 0.75$



(b) $\Delta\lambda_j$ from graphs with $P(l_{i,j}) = 0.99$

Figure 5.7. Comparison of $\Delta\lambda_j^{UB}$ to the $\Delta\lambda_j$ results from a simulation of 300 random simple graphs containing 50 nodes and $P(l_{i,j}) = \{0.75, 0.99\}$ where links are removed from a single, fully-connected node.

5.3 Eigenvectors of Perturbed Adjacency Matrix

Continuing to support the network case requiring further analysis, we focus in this section on the change in v_j resulting from the links associated with one node being perturbed in the adjacency matrix. We begin by developing a theoretical upper bound on θ_j using the Davis-Kahan theorem. We then use known properties of v_j to develop a theoretical upper bound on the change in the magnitude of the eigenvector element $\Delta|v_{i,j}|$. Network simulations are utilized to analyze and validate the magnitude bound for graphs with various degrees of connectivity. We conclude with a description of a novel application of the n-dimensional Euclidean distance formula that provides an upper bound on the eigenvector distance D_{v_j}, where D_{v_j} is the Euclidean distance between v_j and $\hat{v_j}$. Network simulations are utilized to analyze and validate the distance bound for graphs with various degrees of connectivity.

5.3.1 Bounding the Eigenvector Rotation

The theoretical upper bound of θ_j is determined from the Davis-Kahan theorem (defined in Theorem 2 in Chapter 2), which is dependent on $||Q_{error}||$ and the perturbed and unperturbed eigenvalues. In support of the network case requiring further analysis, we assume that link perturbations in the adjacency matrix are confined to a single node. For mathematical simplicity, we confine our analysis of θ_j to K_n in this section.

For all cases of $l_{perturb}$ in K_n [101],

$$\| Q_{error} \| = l_{perturb} + 1. \tag{5.7}$$

It is difficult to further reduce or extrapolate information from κ_j for link removals in K_n. As a result, the upper bound of θ_j for K_n is given by

$$\theta_j^{UB} = \arcsin\left(\frac{l_{perturb} + 1}{\kappa_j}\right) \leq 180 \text{ degrees} \tag{5.8}$$

due to the properties of arcsin. It is important to note that the bound of $\theta_j^{UB} \leq 180$ degrees could also be easily determined from the orthonormal property of the eigenvectors of Q [55]. Regardless, this information tells us that link perturbations in the adjacency matrix can result in an eigenvector rotating into a different eigenspectrum index with a different λ_j. While inherently logical, this proves from a nodal basis perspective that the eigenspectrum index

for which a node has $v_{i,j}^{pri}$ may shift as a result of link perturbations.

5.3.2 Bounding the Change in Eigenvector Element

The theoretical upper bound of a change in $v_{i,j}$ is determined from known properties of eigenvectors discussed in Chapter 2. We utilize these properties to develop a theoretical upper bound and then utilize network simulations of link removals to validate the theoretical bound.

Given the sum of squares property in Equation 2.8, the range of $v_{i,j}$ is [-1 1]. As a result, the upper bound for a change in $v_{i,j}$ is given by

$$\Delta v_{i,j}^{UB} = 2, \tag{5.9}$$

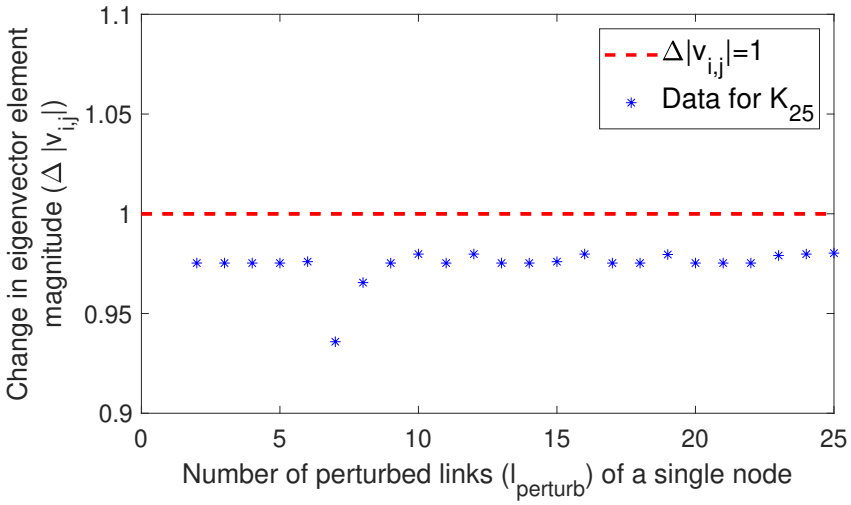
and the upper bound for a change in $|v_{i,j}|$ is given by

$$\Delta |v_{i,j}^{UB}| = 1. \tag{5.10}$$

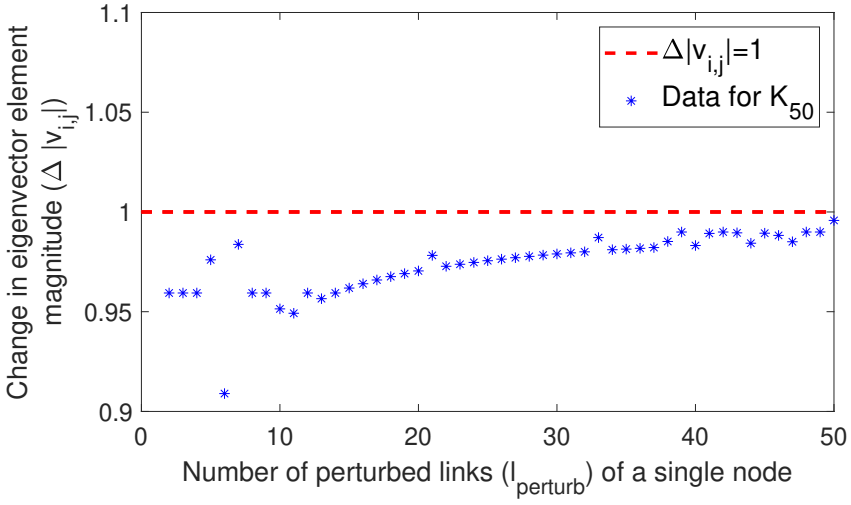
Similar to the eigenvector rotation bound, this eigenvector element bound also proves from a nodal basis perspective that the eigenspectrum index in which a node has $v_{i,j}^{max}$ may shift as a result of link perturbations.

Since $|v_{i,j}|$ is utilized in the proposed SGCDC system, the accuracy of $\Delta |v_{i,j}^{UB}|$ is analyzed through the simulation of single node link removals in K_n and simple graphs. For K_n, $\Delta |v_{i,j}|$ is analyzed for networks ranging from $n = 25$ to $n = 200$ where a single node experiences link perturbations (i.e., removals). The results for K_{25} and K_{50} are shown in Figure 5.8 and the results of K_{100} and K_{200} are shown in Figure 5.9. In both figures, the blue asterisks are the data results from the simulation and the red dashed line is $\Delta |v_{i,j}| = 1$.

For all network sizes evaluated, the simulations results are similar to those shown in figures 5.8 and 5.9. In other words, the $\Delta |v_{i,j}|$ resulting from link perturbations for a single node are less than or equal to one for all evaluated K_n. These results validate $\Delta |v_{i,j}^{UB}|$ in Equation 5.10 as an upper bound of the eigenvector element change for complete graphs.

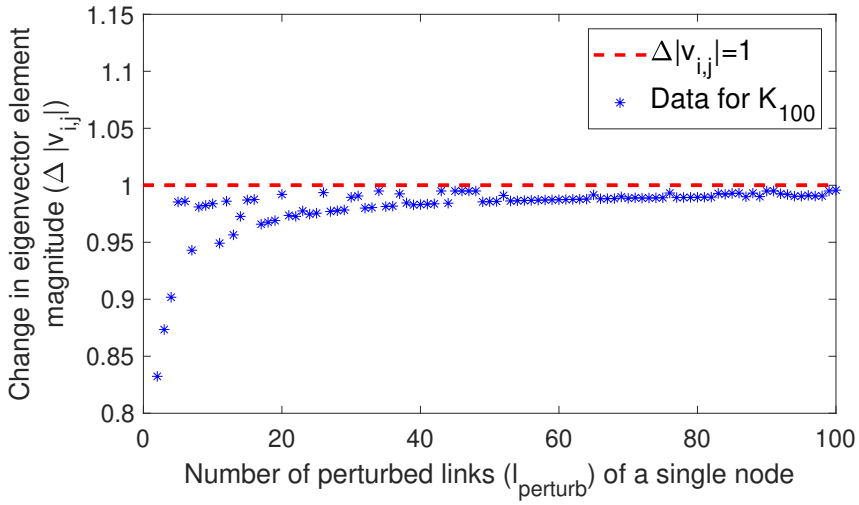


(a) Link removals from one node in K_{25}

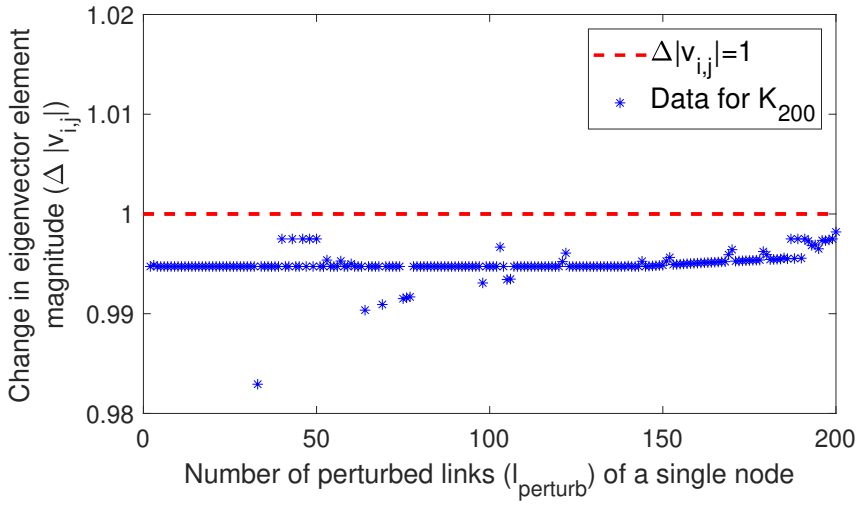


(b) Link removals from one node in K_{50}

Figure 5.8. Comparison of $\Delta|v_{i,j}^{UB}|$ to the $\Delta|v_{i,j}|$ simulation results from K_{25} and K_{50}, where links are removed from a single node.



(a) Link removals from one node in K_{100}



(b) Link removals from one node in K_{200}

Figure 5.9. Comparison of $\Delta|v_{i,j}^{UB}|$ to the $\Delta|v_{i,j}|$ simulation results from K_{100} and K_{200}, where links are removed from a single node.

To further validate the accuracy of $\Delta|v_{i,j}^{UB}|$ for other types of networks, single node link removals are analyzed for 300 random simple graphs containing 50 nodes and a static $P(l_{i,j})$. In the simulation, one node in each graph is selected and edited to be completely connected and $\Delta|v_{i,j}|$ is evaluated as the links of the selected node are disconnected. Simulations results are obtained for $P(l_{i,j})$ between 0.01 and 0.99.

To demonstrate the effects of connectivity on $\Delta|v_{i,j}|$, the results of $P(l_{i,j}) = 0.1$ are shown in Figure 5.10, the results of $P(l_{i,j}) = 0.5$ and $P(l_{i,j}) = 0.75$ are shown in Figure 5.11, and the results of $P(l_{i,j}) = 0.99$ are shown in Figure 5.12. In all three figures, the blue asterisks are the data results from the simulation and the red dashed line is $\Delta|v_{i,j}| = 1$. Additional results for other $P(l_{i,j})$ can be found in Appendix B.3.

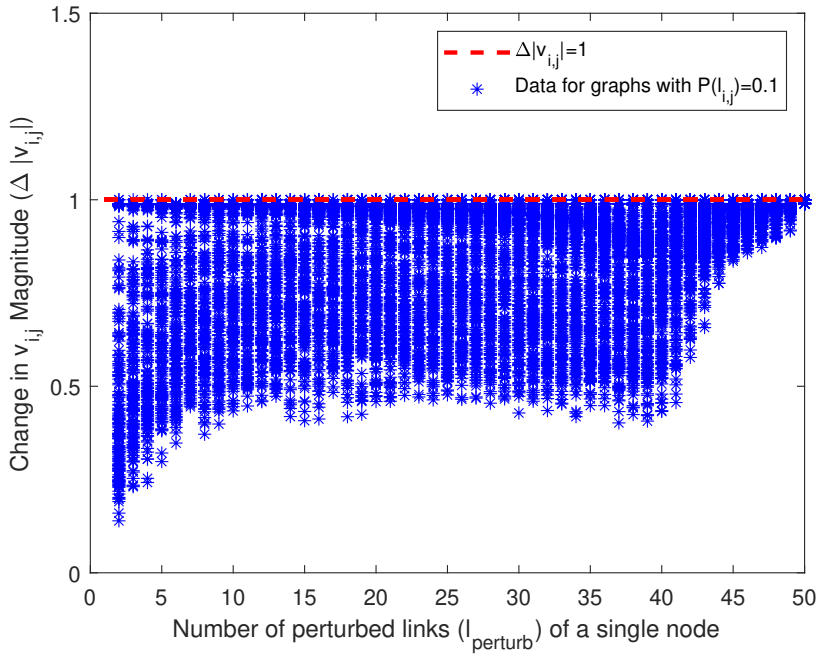
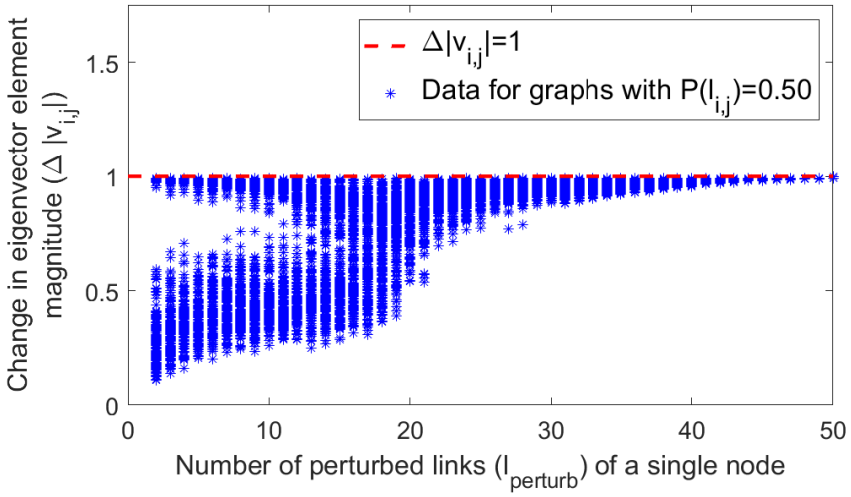
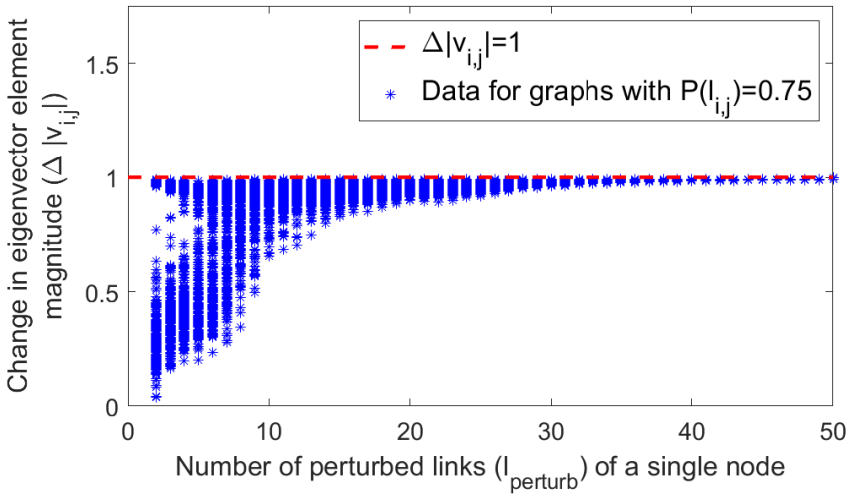


Figure 5.10. Comparison of $\Delta|v_{i,j}^{UB}|$ to the $\Delta|v_{i,j}|$ simulation results of 300 random simple graphs containing 50 nodes and $P(l_{i,j}) = 0.1$ with link removals from a single, fully-connected node.



(a) $\Delta|v_{i,j}|$ from graphs with $P(l_{i,j}) = 0.5$



(b) $\Delta|v_{i,j}|$ from graphs with $P(l_{i,j}) = 0.75$

Figure 5.11. Comparison of $\Delta|v_{i,j}^{UB}|$ to the $\Delta|v_{i,j}|$ simulation results of 300 random simple graphs containing 50 nodes and $P(l_{i,j}) = \{0.5, 0.75\}$ with link removals from a single, fully-connected node.

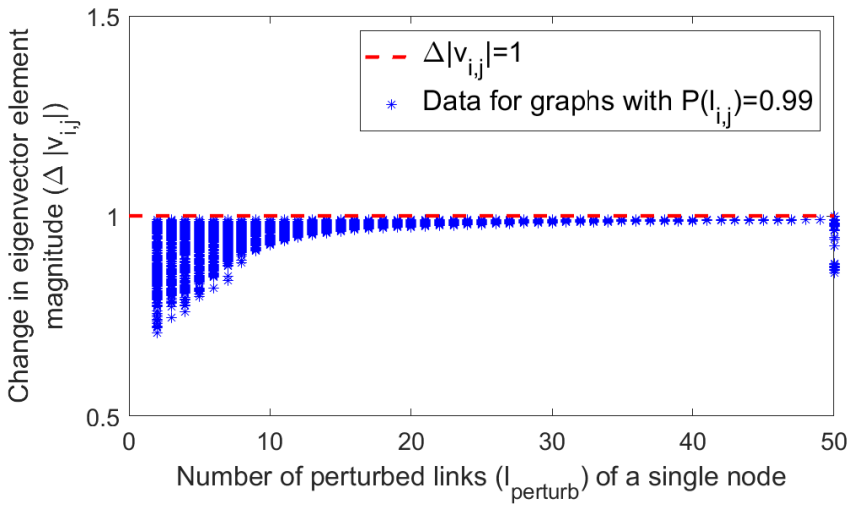


Figure 5.12. Comparison of $\Delta|v_{i,j}^{UB}|$ to the $\Delta|v_{i,j}|$ simulation results of 300 random simple graphs containing 50 nodes and $P(l_{i,j}) = 0.99$ with link removals from a single, fully-connected node.

From the simulation results, $\Delta|v_{i,j}|$ is impacted differently depending on the connectivity of the simple graph. In general, graphs that are more sparsely connected have a higher probability of a smaller $\Delta|v_{i,j}|$ for all $l_{perturb}$ compared to graphs that are more densely connected; however, graphs that are very sparsely connected or very densely connected appear to have a large $\Delta|v_{i,j}|$. Still, graphs with all levels of connectivity appear to have a higher probability of a smaller $\Delta|v_{i,j}|$ for a small $l_{perturb}$ than a large $l_{perturb}$. This implies that small perturbations in the adjacency matrix have a higher probability of producing $\Delta|v_{i,j}| \ll \Delta|v_{i,j}^{UB}|$. For larger $l_{perturb}$, $\Delta|v_{i,j}|$ is directly dependent on the graph connectivity. Regardless, the majority of sparse graphs have a higher probability across all $l_{perturb}$ to produce $\Delta|v_{i,j}| \ll \Delta|v_{i,j}^{UB}|$.

5.3.3 Bounding the N-dimensional Change in Distance

A novel application of the n-dimensional Euclidean distance formula (defined in [102]) provides information on eigenvector perturbation that is not provided by the Davis-Kahan theorem or the sum of squares property. This application is of particular interest when applied to the nodal basis because it provides information on the distance and angle of rotation of the nodal influences of a single node across the eigenspectrum index from a perturbation of

its links in the adjacency matrix. We begin by defining the $(n-1)$-dimensional coordinate system before determining the upper bound on the distance and angle of rotation. Then, we utilize simulations to determine that accuracy of the distance upper bound for graphs with various levels of connectivity.

Given the $n-1$ nodal influences associated with λ_2 to λ_n for a single node, a depiction of the $(n-1)$-dimensional coordinate system established by the perturbation of v_j^T is shown in Figure 5.13 where v_j^T is the eigenvector of the nodal basis associated with node j, $\hat{v}_j^T$ is the perturbed eigenvector of the nodal basis associated with node j, D_{v_j} is the Euclidean distance between v_j^T and $\hat{v}_j^T$, and θ_{v_j} is the angle of rotation between v_j^T and $\hat{v}_j^T$.

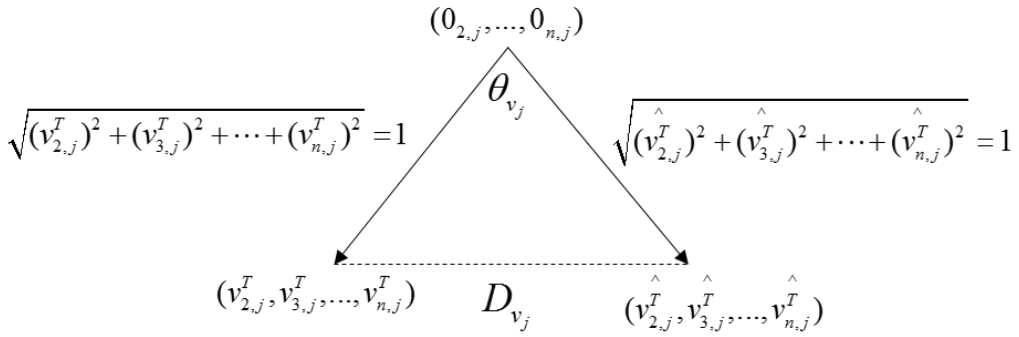


Figure 5.13. $(n-1)$-dimensional coordinate system for eigenvector perturbation in the nodal basis.

Given the n-dimensional Euclidean distance formula in [102] and the $(n-1)$-dimensional coordinate system, D_{v_j} is defined as

$$D_{v_j} = \sqrt{\sum_{i=2}^{n} (\hat{v_{i,j}}^T - v_{i,j}^T)^2}, \tag{5.11}$$

and θ_{v_j} is defined as

$$\theta_{v_j} = \arccos\left(\frac{(D_{v_j})^2 - 2}{2}\right). \tag{5.12}$$

Applying logical reasoning, the upper bound on D_{v_j} is given by

$$D_{v_j}^{UB} = 2 \tag{5.13}$$

and the upper bound on θ_{v_j} can be shown to be

$$\theta_{v_j}^{UB} = 180^{\circ}, \tag{5.14}$$

as depicted in Figure 5.14 due to the orthonormal properties of eigenvectors and the properties of arccos. Equation 5.14 does not provide any new information since 180 degrees is also the maximum upper bound of Equation 5.8 from the Davis-Kahan theorem. As a result, we only need to use simulation results to validate the accuracy of $D_{v_j}^{UB}$.

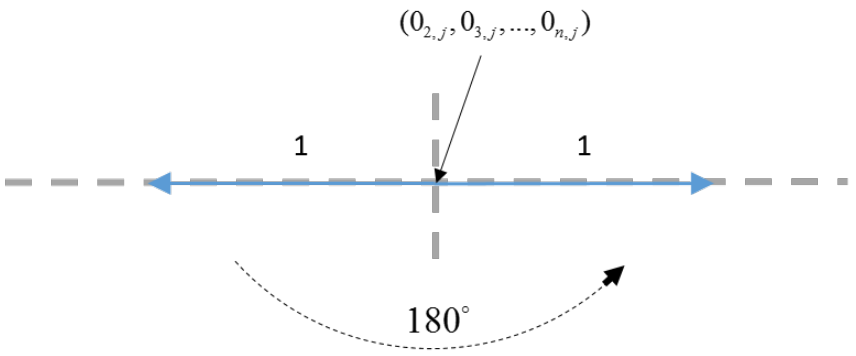
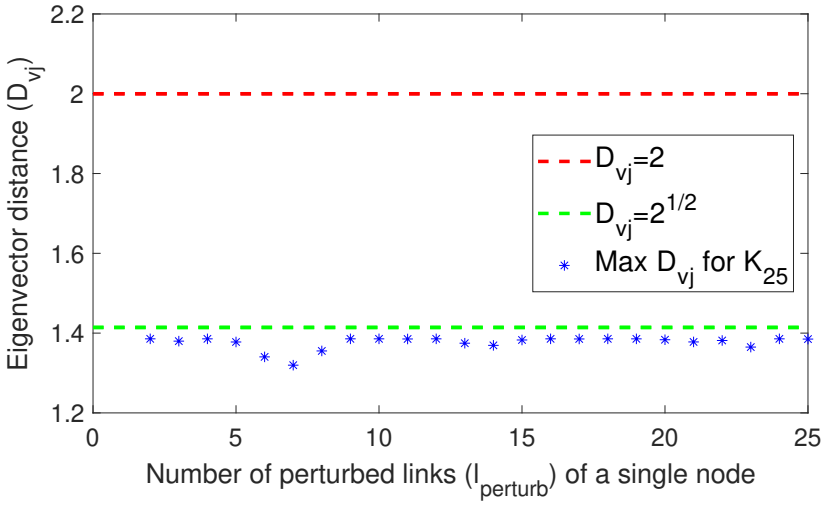
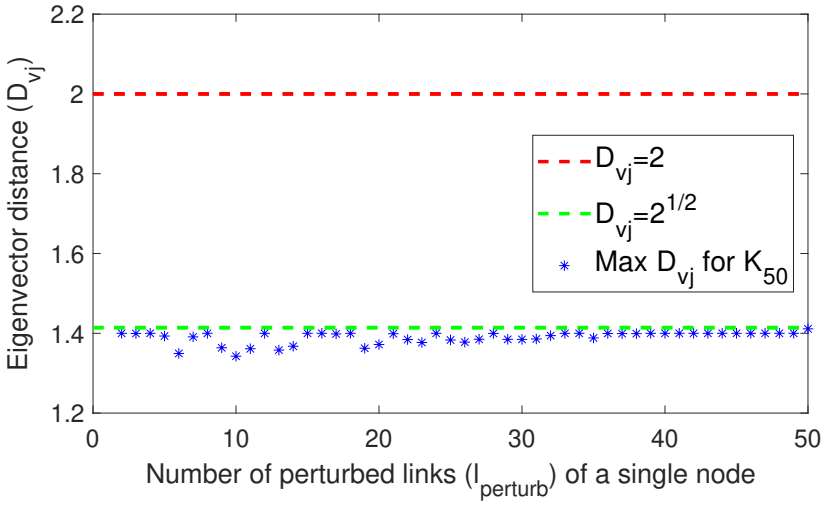


Figure 5.14. Maximum distance and angle of rotation for eigenvector perturbation in the nodal basis.

The accuracy of $D_{v_j}^{UB}$ is analyzed through the simulation of single node link removals for K_n and simple graphs. For K_n, the maximum D_{v_j} is analyzed for networks ranging from $n = 25$ to $n = 200$ where a single node experiences link removals. The results for the maximum D_{v_j} for K_{25} and K_{50} are shown in Figure 5.15, and the results for K_{100} and K_{200} are shown in Figure 5.16. For both figures, the blue asterisks are the maximum eigenvector distance change results from the simulation, the red dashed line is $D_{v_j} = 2$, and the green dashed line is $D_{v_j} = \sqrt{2}$.

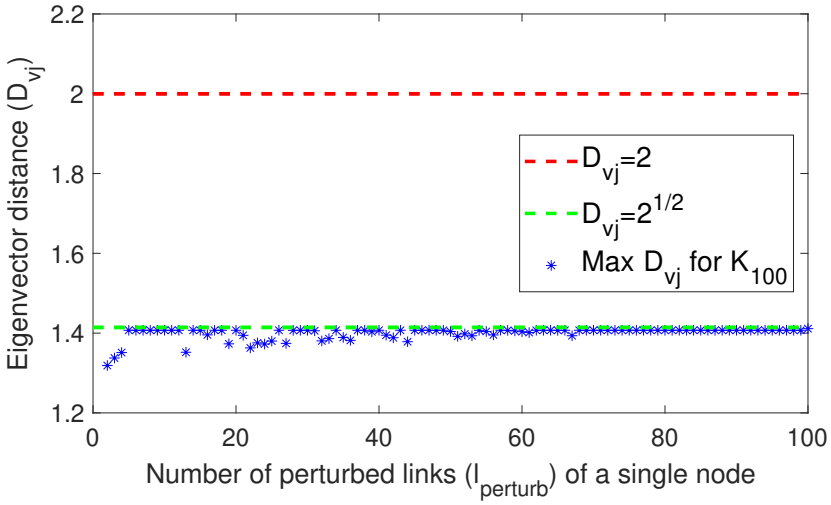


(a) Link removals from one node in K_{25}

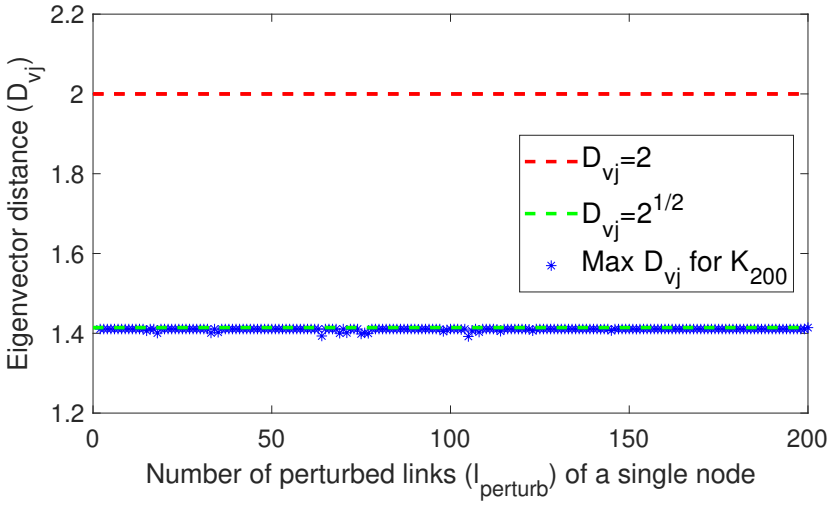


(b) Link removals from one node in K_{50}

Figure 5.15. Comparison of $D_{v_j}^{UB}$ (red dashed line) to the maximum D_{v_j} observed (blue asterisks) in K_{25} and K_{50}, where links are removed from a single node.



(a) Link removals from one node in K_{100}



(b) Link removals from one node in K_{200}

Figure 5.16. Comparison of $D_{v_j}^{UB}$ (red dashed line) to the maximum D_{v_j} observed (blue asterisks) in K_{100} and K_{200}, where links are removed from a single node.

All simulations of K_n exhibit similar results as those shown in Figure 5.11. Since $D_{v_j} < D_{v_j}^{UB}$ for all simulations, the results validate Equation 5.13. Additionally, the simulation results appear to show that a tighter upper bound may exist for D_{v_j}. For K_n, the simulation-suggested upper bound of eigenvalue distance is given by

$$D_{v_j}^{UB,sim} = \sqrt{2}. \tag{5.15}$$

For simple graphs, 300 random graphs containing 50 nodes are formed using a static $P(l_{i,j})$ between each pair of nodes, one node in each graph is selected, the selected node is edited to be completely connected, and the maximum D_{v_j} is evaluated as the links of the selected node are disconnected. Simulations results are obtained for $P(l_{i,j})$ between 0.01 and 0.99. To demonstrate the impact of link perturbations on D_{v_j}, the results of $P(l_{i,j}) = 0.1$ are shown in Figure 5.17, the results of $P(l_{i,j}) = 0.5$ and $P(l_{i,j}) = 0.75$ are shown in Figure 5.18, and the results of $P(l_{i,j}) = 0.99$ are shown in Figure 5.19. For all three figures, the blue asterisks are the maximum eigenvector distance change results from the simulation, and the red dashed line is $D_{v_j}^{UB}$, and the green dashed line is $D_{v_j}^{UB,sim}$. Additional results for other $P(l_{i,j})$ can be found in Appendix B.4.

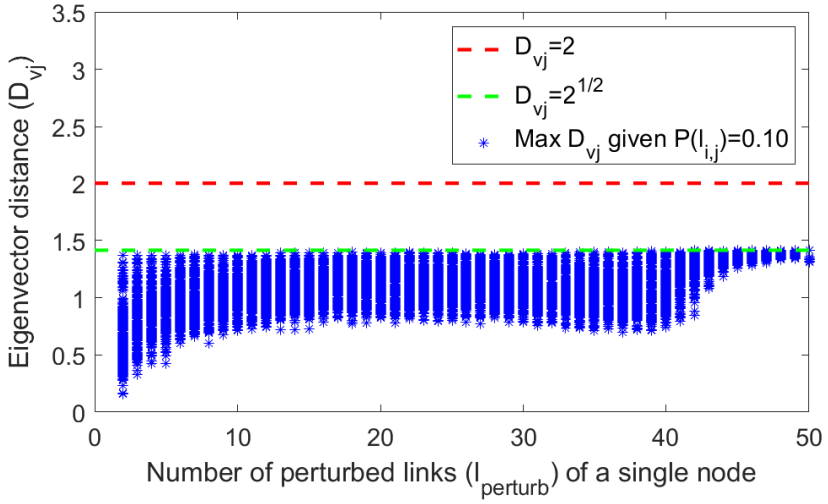
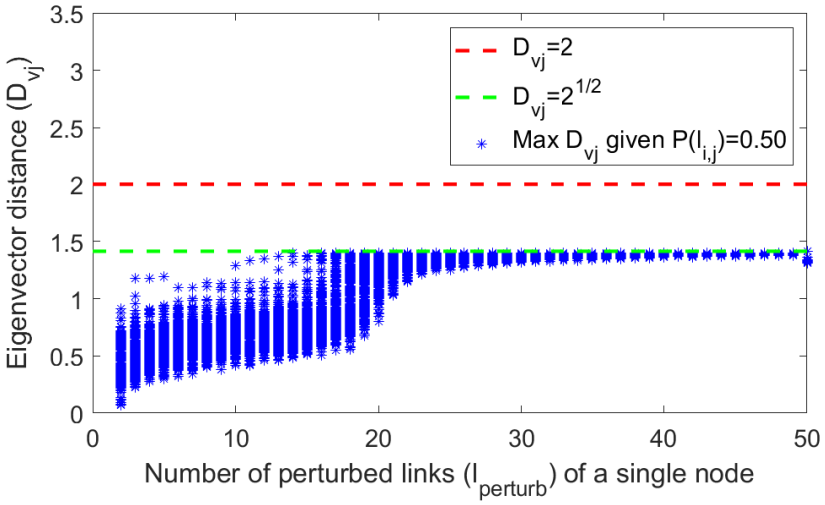
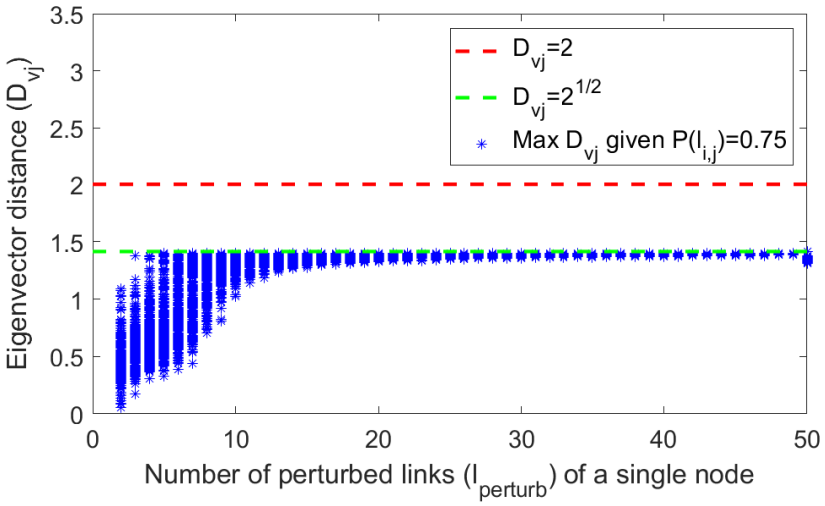


Figure 5.17. Comparison of $D_{v_j}^{UB}$ (red dashed line) to the maximum D_{v_j} observed (blue asterisks) in 300 random simple graphs containing 50 nodes and $P(l_{i,j}) = 0.1$ with link removals from a single, fully-connected node.



(a) Max D_{v_j} from graphs with $P(l_{i,j}) = 0.5$



(b) Max D_{v_j} from graphs with $P(l_{i,j}) = 0.75$

Figure 5.18. Comparison of $D_{v_j}^{UB}$ (red dashed line) to the maximum D_{v_j} observed (blue asterisks) in 300 random simple graphs containing 50 nodes and $P(l_{i,j})$ = $\{0.5, 0.75\}$ with link removals from a single, fully-connected node.

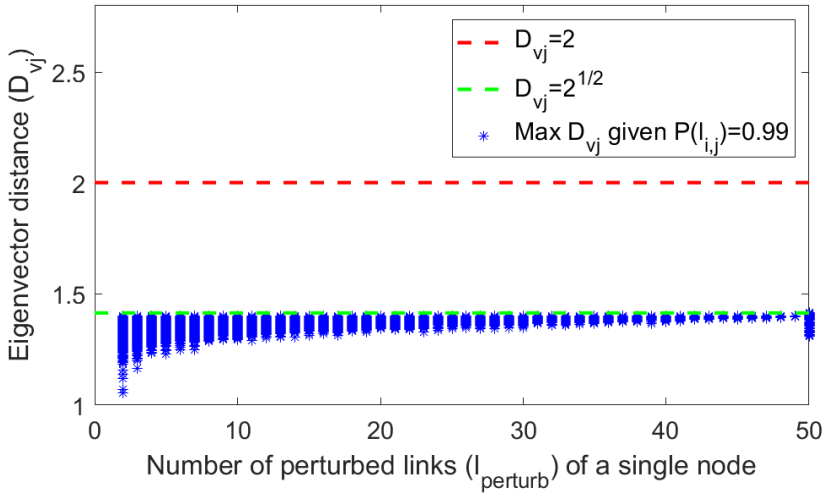


Figure 5.19. Comparison of $D_{v_j}^{UB}$ (red dashed line) to the maximum D_{v_j} observed (blue asterisks) in 300 random simple graphs containing 50 nodes and $P(l_{i,j}) = 0.99$ with link removals from a single, fully-connected node.

From the simulation results, the maximum D_{v_j} is impacted differently depending on the connectivity of the graph. In general, all graphs have a a maximum D_{v_j} close to $D_{v_j}^{UB}$ for very large $l_{perturb}$, but graphs that are very sparsely or densely connected have a maximum D_{v_j} close to $D_{v_j}^{UB}$ for all $l_{perturb}$. On the other hand, graphs that are neither sparsely nor densely connected have a higher probability of a small maximum D_{v_j} for a small $l_{perturb}$. This implies that small perturbations in the adjacency matrix have a higher probability of producing a maximum $D_{v_j} \ll D_{v_j}^{UB}$ if the graph connectivity is not extremely sparse or dense. Regardless, all simulation results validate $D_{v_j}^{UB,sim}$ as an upper bound for the maximum D_{v_j}.

5.4 Perturbation and Spectral Graph Detection

The spectral graph detection mechanism employed in the proposed SGCDC system analyzes the eigenvalues and eigenvectors of Q (formed by A) and, therefore, is susceptible to link perturbations as described in Section 5.1. Given the eigenvalue and eigenvector perturbation results in Sections 5.2 and 5.3 (summarized in Table 5.1), the spectral graph detection component is examined to determine the effects of link perturbations in A on attack detection. We begin with an analysis of the eigenvalues before addressing the eigenvectors.

Table 5.1. Summary of the effects of adjacency matrix perturbations on the dual basis.

$l_{perturb}$	Graph Connectivity	$\Delta\lambda_j$	$\Delta \mid v_{i,j} \mid$	D_{v_j}
Small	Very Sparse	Large value	Large value	Large value
	Medium Sparse	Large value	High prob. of small value	High prob. of small value
	Medium	High prob. of small value	High prob. of small value	High prob. of small value
	Medium Dense	High prob. of small value	High prob. of small value	High prob. of small value
	Very Dense	Large value	Large value	Large value
Medium	Very Sparse	Large value	Large value	Large value
	Medium Sparse	High prob. of small value	High prob. of small value	High prob. of small value
	Medium	High prob. of small value	High prob. of small value	High prob. of small value
	Medium Dense	High prob. of small value	Large value	Large value
	Very Dense	Large value	Large value	Large value
Large	Very Sparse	Large value	Large value	Large value
	Medium Sparse	High prob. of small value	Large value	Large value
	Medium	High prob. of small value	Large value	Large value
	Medium Dense	High prob. of small value	Large value	Large value
	Very Dense	Large value	Large value	Large value

5.4.1 Eigenvalue Perturbations and System Detection

Given the eigenvalue perturbation results described in Section 5.2 (summarized in Table 5.1) and the strong node concept described in Section 3.2.2, the detection capability of the spectral graph detection mechanism is directly impacted by changes in the eigenvalues resulting from link perturbations in A. Specifically, the degree of the perturbed node, defined as

$$\hat{d}_i = d_i - l_{perturb} \tag{5.16}$$

directly impacts λ_j^{pri} where λ_j^{pri} is the λ_j associated with the $v_{i,j}^{pri}$ of the node within the eigenspectrum index.

If $\hat{d}_i$ results in $\hat{\lambda}_j^{pri} \geq \lambda_{thresh}$ where $\hat{\lambda}_j^{pri}$ is the λ_j^{thresh} of the perturbed node and λ_{thresh} is the λ_j associated with T_{eigen}, then the node will be detected as crossing the eigenspectrum threshold. Given this case, the spectral graph detection mechanism accurately detects the attack if the node is under attack and falsely detects an attack if the node is not under attack. Conversely, if $\hat{d}_i$ results in $\hat{\lambda}_j^{pri} < \lambda_{thresh}$, then the node will not be detected as crossing the threshold. Given this case, the spectral graph detection mechanism fails to detect the attack if the node is under attack and correctly classifies the node as acting under normal conditions if the node is not under attack.

Combining this information with the simulation results summarized in Table 5.1, the impact of noise or congestion on the ability of the SGCDC system to accurately detect worm and DDoS attacks is highly dependent upon the level of connectivity of the components within the TDG. Theoretically, network noise and congestion will have less impact on the ability of the system to detect an attack in a TDG containing components with medium connectivity than a TDG containing components with sparse or dense connections.

5.4.2 Eigenvector Perturbations and System Detection

Given the eigenvector perturbation results described in Section 5.3 (summarized in Table 5.1) and the strong node concept described in Section 3.2.2, link perturbations in the adjacency matrix can impact the ability of the SGCDC system to detect attacks due to changes in the eigenvector matrix. Similar to the eigenvalues, $\hat{d}_i$ directly impacts the location of $v_{i,j}^{pri}$ of a node in the eigenspectrum index. If $\hat{d}_i$ results in a $\hat{v}_{i,j}^{pri}$ in a smaller eigenspectrum index than the eigenspectrum index of T_{eigen}, where $\hat{v}_{i,j}^{pri}$ is the primary nodal influence of the

perturbed node, then the node will not be detected as crossing the eigenspectrum threshold. As a result, the SGCDC system will not detect the node as a parent node of an attack. If the node is not under attack, then the detection mechanism correctly classifies the nodes as normal; however, if the node is under attack, then the detection mechanism fails to detect the attack.

On the other hand, if $\hat{d}_i$ results in a $\hat{v}_{i,j}^{pri}$ in a larger eigenspectrum index than the eigenspectrum index of T_{eigen}, then the node will be detected as crossing the eigenspectrum threshold. As a result, the SGCDC system will detect the node as a parent node of an attack. If the node is under attack, then the system accurately detects the attack; however, if the node is not under attack, then the system produces a false alarm.

Combining this analysis with the simulation results summarized in Table 5.1, the detection accuracy of the SGCDC system is impacted by network noise and congestion. The degree of impact is primarily dependent on the size of the perturbation and the rate of the attack. In general, small perturbations in the adjacency matrix from network noise or congestion should not impact the ability of the SGCDC system to detect medium- to fast-spreading worm attacks and overt DDoS attacks. Yet, as the worm scan rate and/or DDoS attack rate approach $T_{phantom}^{worm}$ and/or $T_{phantom}^{DDoS}$, respectively, small perturbations in the adjacency matrix from network noise or congestion will have a significant impact on the ability of the system to detect the attack.

In summary, this chapter analyzed the impact of network noise and congestion on the spectral graph detection mechanism employed by the proposed SGCDC system. Network noise and congestion results in errors in the adjacency matrix, which in turn affects the ability of the proposed SGCDC system to accurately detect attacks from the eigenvalues and eigenvectors of the TDG. As a result, network administrators must have an accurate understanding of noise and congestion within their network to accurately select the set of external, input parameters that will increase the probability of detection and decrease the probability of false alarms. The next chapter validates the detection capability and false alarm rates of the proposed system using multiple datasets.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 6:
Validation

In this chapter, we present results that support the analysis conducted in this dissertation and validate the performance of the proposed SGCDC system. We first demonstrate the accuracy of the proposed system in detecting worm and DDoS attacks. Then, we present the system false alarm rates using real-world and synthetic datasets. Finally, we demonstrate the potential response capability of the proposed system.

6.1 Scan-Based TCP Worm Attack

In this section, we use several different datasets of various computer worms with different propagation speeds to examine the detection accuracy of the SGCDC system via MATLAB implementation and simulation. We begin with an evaluation of the implemented system against two WannaCry worm datasets that exhibit a slower worm propagation speed. Then, we evaluate the implemented system against a very fast-propagating modeled hit-list worm. Finally, we evaluate the implemented system against a slow-to-medium propagating Blaster worm model.

6.1.1 WannaCry Worm

The detection capability of the proposed SGCDC system is evaluated against two WannaCry worm datasets that have slower estimated propagation speeds due to the number of infectable hosts in the networks. The first dataset combines Stratosphere Lab CTU-Malware-Capture-Botnet-284-1 [103] and CTU-Malware-Capture-Botnet-285-1 [104] WannaCry worm datasets. For the remainder of this dissertation, we refer to this collated set of data as the Stratosphere_WC dataset.

CTU-Malware-Capture-Botnet-284-1 provides the traffic captures of host 192.168.1.112, and CTU-Malware-Capture-Botnet-285-1 provides the traffic captures of host 192.168.1.135. There is an approximately 10-second timing difference between the two datasets, which is accounted for in the Stratosphere_WC dataset. Within the data, host 192.168.1.112 infects host 192.168.1.135 with the WannaCry worm at approximately 470 seconds in CTU-

Malware-Capture-Botnet-284-1 (approximately 480 seconds in CTU-Malware-Capture-Botnet-285-1). Host 192.168.1.135 then begins scanning for new victims at approximately 482.03 seconds in CTU-Malware-Capture-Botnet-285-1. The number of SYN scans by host 192.168.1.135 are initially small in quantity, but the quantity increases as time passes. As a result, we focus on the data between 450 seconds and 515 seconds in the compiled dataset during our evaluation. The final infection size is two nodes.

We developed the second WannaCry worm dataset from the NPS WannaCry virtual network depicted in Figure 6.1, which contains four laptops with the Windows 7 operating system, five virtual hosts with the Windows 7 operating system, and one virtual Apache web server.

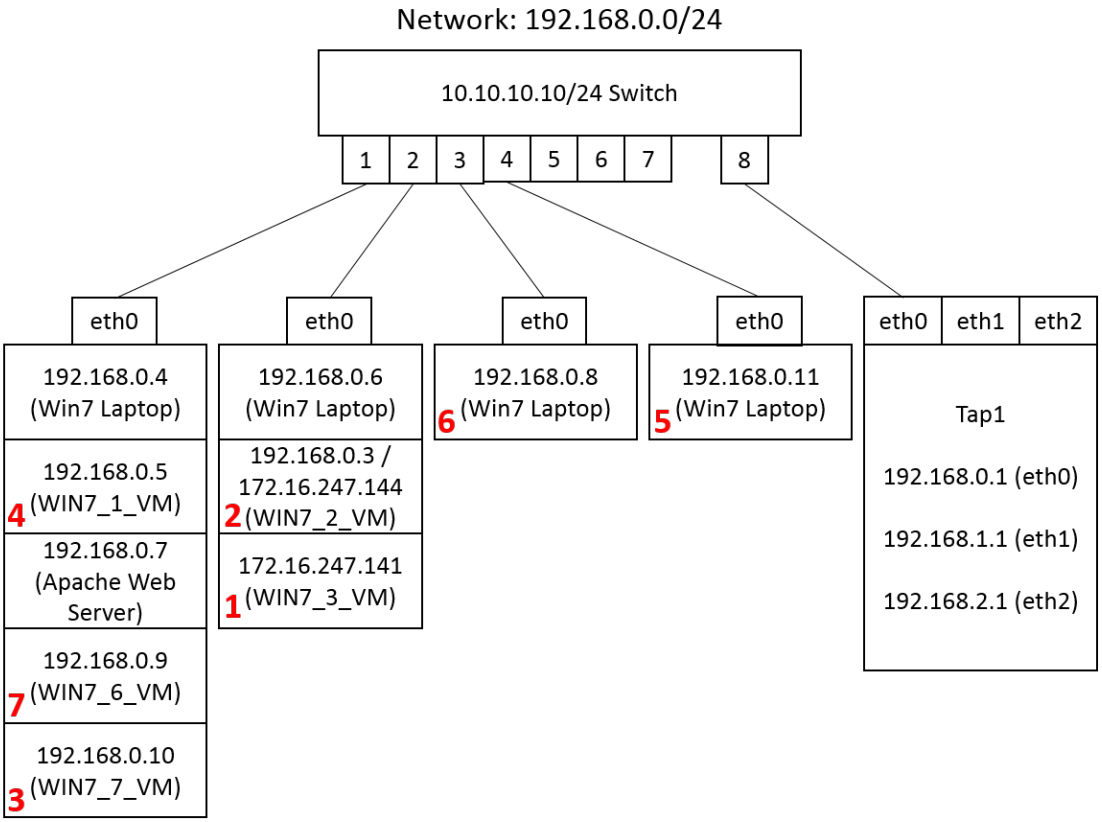


Figure 6.1. Diagram of WannaCry virtual network developed for the NPS_WC dataset collection with order of infection depicted by the red numbers.

To acquire data on the WannaCry worm propagation, host 172.16.247.141 is infected with

the WannaCry worm acquired from [105], and six additional hosts are rapidly infected in the order depicted in Figure 6.1. For the remainder of this dissertation, we refer to this collection of data as the NPS_WC dataset. Upon receipt of the worm payload as part of the infection process, each of the infected hosts attempts to reinfect the other infected hosts for a period of time; every instance of reinfection resets the WannaCry worm infection for the host. After the period of reinfections concludes, the infected host reaches out to a domain name server (DNS). Approximately 25 seconds after the DNS query, the infected host begins the traditional target search and discovery process for a scan-based TCP worm. As a result, we estimate the time of infection as the time that the infected host sends the standard query of "www.ifferfsodp9ifjaposdfjhgosurijfaewrwergwea.com" to the DNS server instead of the time that the host initially received the worm payload.

Table 6.1 contains the approximate infection times for each of the infected hosts relative to the beginning of the capture along with the time that each host sends its first SYN packet after sending the DNS query. It is important to note that while host 192.168.0.5 receives the worm payload in the NPS_WC dataset, it is unable to execute to the point of sending the DNS query. As a result, host 192.168.0.5 is not included as an infected host in Table 6.1.

Table 6.1. Infection times and post-infection SYN activity of hosts in the NPS_WC dataset.

Host IP	Approx. Infection Time	Time of First SYN after Infection
172.16.247.144 (dual homed as 192.168.0.3)	96 sec	120.98 sec
192.168.0.8	294 sec	318.469 sec
192.168.0.9	387 sec	411.956 sec
192.168.0.10	159 sec	184.327 sec
192.168.0.11	242 sec	267.384 sec

We evaluate the detection accuracy and timeliness of our proposed solution by setting t_{log} = 40 seconds to account for longer infection times due to system reboots [106]. The remaining external, input parameters are selected to account for both slow and fast worm propagation speeds, align with evaluated characteristics of numerous historical worm attacks, and support

the required external, input parameter relationships described by equations 3.8, 3.9, and 3.10.

The times for the SGCDC system to detect the infected node from both infection and from when the node sent its first SYN packet post-infection (i.e., when began exhibiting worm attack behavior) are provided in Table 6.2 for the Stratophere_WC dataset and Table 6.3 for the NPS_WC dataset.

Table 6.2. SGCDC system detection results against Stratosphere_WC dataset.

L_w (sec)	t_{log} (sec)	$T^{worm}_{phantom}$	Time to Detect		SYNs Sent Prior to Detection
			From Infection	From 1^{st} SYN	
1	40	5	8.31 sec	6.16 sec	15
2	40	5	6.31 sec	4.16 sec	9
3	40	5	4.31 sec	2.16 sec	5
4	40	5	4.31 sec	2.16 sec	5

Table 6.3. SGCDC system detection results against NPS_WC dataset.

IP Address	L_w (sec)	t_{log} (sec)	$T^{worm}_{phantom}$	Time to Detect	
				From Infection	From 1^{st} SYN
172.16.247.144	0.5	40	5	25.5 sec	0.52 sec
	1	40	5	25.5 sec	0.52 sec
	2	40	5	26 sec	1.02 sec
192.168.0.8	0.5	40	5	24.75	0.281
	1	40	5	25	0.531
	2	40	5	25	0.531
192.168.0.9	0.5	40	5	25.75	0.794
	1	40	5	25.5	0.544
	2	40	5	26	1.044
192.168.0.10	0.5	40	5	26	0.673
	1	40	5	26	0.673
	2	40	5	26	0.673
192.168.0.11	0.5	40	5	26	0.616
	1	40	5	26.5	1.116
	2	40	5	27	1.616

The SGCDC system had a worm detection and classification rate of one and a worm false alarm rate of zero for all sets of external, input parameters for both the Stratoshphere_WC dataset and the NPS_WC dataset. For the NPS_WC dataset, the SGCDC system detected all infected nodes in less than two seconds after the node began exhibiting worm attack behavior with the vast majority detected in less than one second. The SGCDC system detected the infected nodes more slowly for the Stratosphere_WC dataset due to the initially slow scan rate of the newly infected node. This indicates that the SGCDC system performs better with a longer L_w when a slower ramp up in worm scan behavior occurs post infection, and better with a shorter L_w when a quicker ramp up in worm scan behavior occurs post infection. In both cases, the SGCDC system accurately detected all of the infected nodes once they begin exhibiting worm scan behavior.

6.1.2 Hit-List Worm

The detection capability of the proposed SGCDC system is evaluated against a dataset produced by a modeled hit-list worm (discussed in Section 2.3.1) using MATLAB to determine the system performance against a worm with a faster propagation speed. Using various R_{scan} and t_{infect}, the modeled hit-list worm sequentially infects 512 specified nodes from the NPS2013 dataset. Once the last node on the hit-list is infected, all of the infected nodes cease scanning activity. The worm traffic is then spliced into the first four seconds of the NPS2013 dataset to produce the a dataset containing traffic from a hit-list worm. We refer to this dataset as the NPS_hit-list dataset for the remainder of this dissertation.

Within the NPS_hit-list dataset, the infected nodes send a different total number of SYN packets depending on when they are infected. As shown in Table 6.4, the vast majority of infected nodes send a minimal number of SYN packets after infection while a minority of the infected nodes send the majority of the SYN packets. In fact, 419 of the 512 nodes send fewer than five SYN packets.

Table 6.4. Number of the infected nodes that sent a specific number of SYN packets in the NPS_hit-list dataset.

# of the Infected Nodes	# of SYN Packets Sent
256	0
128	1
64	2
32	3
16	4
8	5
4	6
2	7
1	8
1	9

To comparatively evaluate the detection accuracy and timeliness of the proposed system against the faster-spreading worm, the t_{log} and $T^{worm}_{phantom}$ that is utilized for the Stratosphere_WC and NPS_WC datasets is employed for the NPS_hit-list datasets. The detection results of the implemented system against the NPS_hit-list datasets with a L_w = 0.5 seconds are provided in Table 6.5, L_w = 1 second are provided in Table 6.6, and L_w = 2 second are provided in Table 6.7.

The SGCDC system produced a worm detection and classification rate of one and a worm false alarm rate of zero for all simulated cases except L_w = 0.5 sec, R_{scan} = 10 packets per second, and t_{infect} = 0.1 second, which produced worm detection, classification, and false alarm rates of zero. While the SGCDC system was very accurate in detecting the existence of any attack, the ability of the system to identify the infected nodes was variable across the L_w simulated.

With respect to the accurate identification of infected nodes, the number of identified infected nodes increased as L_w increased. For L_w = 0.5, the SGCDC system produced the least number of identified infected nodes for all worm variations due the smaller amount of worm traffic that occurs within each window. In other words, as the length of time increased between SYN packets being sent for infection, the number of SYN packets sent during any window decreased resulting in missed detections.

Table 6.5. SGCDC detection results against NPS_hit-list datasets with L_w = 0.5 sec and various R_{scan} and t_{infect}.

R_{scan} (pps)	t_{infect} (sec)	Time Infection Start (sec)	Time Node 512 Infected (sec)	Spectral Graph Detection Window							
				1.75-2.25 sec		2-2.5 sec		2.25-2.75 sec		2.5-3 sec	
				# Infect	# Detect	# Infect	# Detect	# Infect	# Detect	# Infect	# Detect
10	0.05	1.7	2.9325	16	0	64	1	128	1	512	0
	0.1	1.7	3.3439	8	0	16	0	64	0	128	0
20	0.05	1.85	2.6863	16	0	128	4	512	19	512	1
	0.1	1.85	3.0964	8	0	32	0	64	0	256	1
50	0.05	1.82	2.4023	74	1	512	17	512	2	512	1
	0.1	1.82	2.8172	16	0	64	1	256	6	512	2

Table 6.6. SGCDC detection results against NPS_hit-list datasets with L_w = 1 sec and various R_{scan} and t_{infect}.

R_{scan} (pps)	t_{infect} (sec)	Time Infection Start (sec)	Time Node 512 Infected (sec)	Spectral Graph Detection Window							
				1.5-2.5 sec		2-3 sec		2.5-3.5 sec		3-4 sec	
				# Infect	# Detect	# Infect	# Detect	# Infect	# Detect	# Infect	# Detect
10	0.05	1.9	3.1409	16	0	256	7	512	26	512	0
	0.1	1.9	3.5453	8	0	64	1	256	11	512	4
20	0.05	1.85	2.6996	128	2	512	23	512	2	512	0
	0.1	1.85	3.0950	32	0	256	6	512	22	512	0
50	0.05	1.82	2.4221	512	4	512	32	512	1	512	0
	0.1	1.82	2.8168	64	1	512	15	512	3	512	0

Table 6.7. SGCDC detection results against NPS_hit-list datasets with L_w = 2 sec and various R_{scan} and t_{infect}.

R_{scan} (pps)	t_{infect} (sec)	Time Infection Start (sec)	Time Node 512 Infected (sec)	Spectral Graph Detection Window							
				0-2 sec		1-3 sec		2-4 sec		3-5 sec	
				# Infect	# Detect	# Infect	# Detect	# Infect	# Detect	# Infect	# Detect
10	0.05	1.9	3.145	1	0	255	4	512	30	512	1
	0.1	1.9	3.5407	1	0	64	1	512	20	512	5
20	0.05	1.85	2.7009	3	0	512	6	512	32	512	0
	0.1	1.85	3.0964	1	0	255	4	512	31	512	1
50	0.05	1.82	2.4332	7	0	512	4	512	32	512	0
	0.1	1.82	2.8207	3	0	512	6	512	32	512	0

It is important to note that while the SGCDC system did not perform as well against this faster-spreading worm, it still performed better than a traditional packet counting solution. With $T^{worm}_{phantom} = 5$, a traditional packet counting solution would require a node to send five or more SYN packets to be detected as anomalous. As annotated in Table 6.4, only 16 nodes send five or more SYN packets in the worm traffic. Yet, the SGCDC system with a $L_w = 2$ seconds identified more than 16 of the infected nodes for all worm variations, a $L_w = 1$ second identified more than 16 of the infected nodes for the majority of the worm variations, and a $L_w = 0.5$ seconds identified more than 16 of the infected nodes for a couple of the worm variations.

Comparing the hit-list results to the WannaCry worm results, the SGCDC system produced relatively similar worm detection and false alarm rates, but our system performed better in identifying the infected nodes for the slower propagating WannaCry worm. It is important to note that the hit-list worm model ceased worm traffic once the 512 nodes were infected, which is atypical of worm propagation. As discussed in Chapter 2, once the hit-list worm infects the designated nodes on the list, the worm typically resorts back to some other form of random scanning technique like that utilized by the WannaCry worm. As a result, the system would begin to produce results similar to those observed from the Stratosphere_WC and NPS_WC datasets following the completion of the hit-list.

6.1.3 Blaster Worm

The detection capability of the proposed SGCDC system is evaluated against a dataset produced by a modeled Blaster worm using MATLAB to determine the system performance against a worm with a slow-to-medium propagation speed due to a longer infection time. The modeled Blaster worm utilizes the primary characteristics of the real-world Blaster worm (described in [106]) with some minor modifications. The resulting worm traffic is then spliced into the NPS2013 dataset to produce the NPS_blaster datasets. It is important to note, since the NPS2013 dataset does not contain operating system information and the Blaster worm model requires operating system information for infection, that the hosts in the NPS2013 dataset are randomly assigned a Windows XP or Windows 2000 operating system.

In the Blaster worm model, an initial node is infected at a random time within the first

detection window of the SGCDC system. The infected node attempts to infect a new node by generating an IP address. The IP address is generated randomly over the entire IP address space with a 60 percent probability, and generated in the form of A.B.C.0 with a 40 percent probability, given A, B, and C are the first, second, and third octets, respectively, of the infected node IP address. Next, the worm selects the operating system offset it will use for the exploit; the worm selects the Windows XP exploit with an 80 percent probability and the Windows 2000 exploit with a 20 percent probability. Using the selected exploit offset, the worm then attempts to infect 20 sequential nodes by starting with the generated IP address and incrementing the fourth octet by one. The SYN scans occur at a rate of 100 scans per second [24], and the infection time takes 36.0058 seconds (approximate time to conduct the TCP handshake, to send a UDP packet and to reboot the system [106], [107]). Once the victims are infected, or if the targeted IP address is either non-exploitable or non-responsive, the worm waits 20 seconds and then generates a new IP address using the rules previously stated.

To comparatively evaluate the detection accuracy and timeliness of the proposed system against the Blaster worm, the t_{log} and $T^{worm}_{phantom}$ that is utilized for the Stratosphere_WC, NPS_WC dataset, and the NPS_hit-list datasets is employed for the NPS_blaster datasets. Instead of evaluating the system against all the L_w employed in the previous datasets, a $L_w = 2$ is fixed and the Blaster worm is simulated for 30 trials to determine the performance against various end infection sizes.

Across the 30 trials, the end infection size ranged from two nodes to 240 nodes. Regardless of the end infection size, the SGCDC system produced a worm detection and classification rate of one. Additionally, the system was able to rapidly detect the presence of the worm after the second node was infected and began exhibiting worm scan behavior as shown in Figure 6.2 and Figure 6.3. In fact, the second infected node was detected within less than one second after it sent its first SYN packet following infection for all trials.

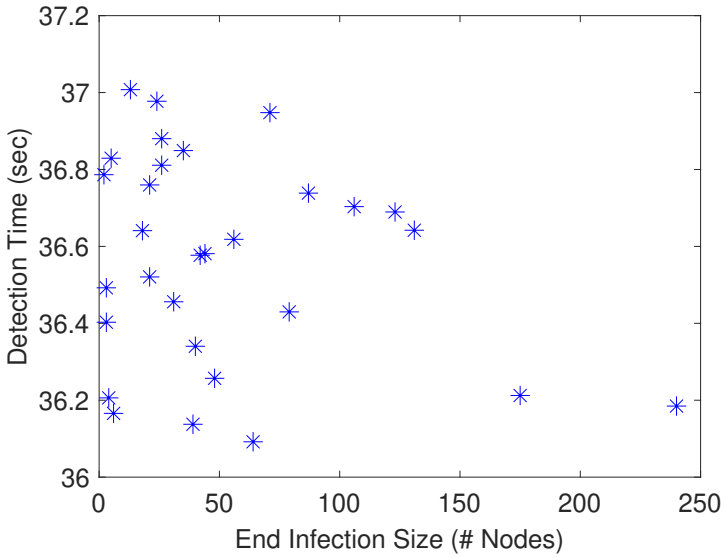


Figure 6.2. Amount of time for the SGCDC system to detect the second infected node in the NPS_blaster datasets from the time the second node is infected.

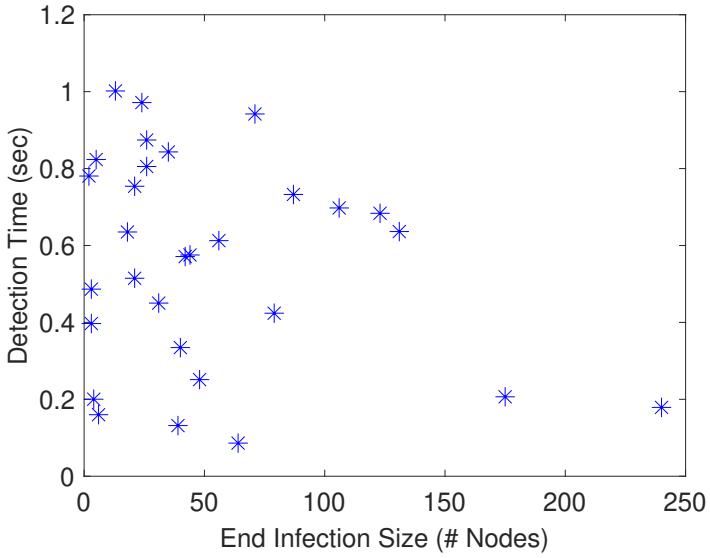


Figure 6.3. Amount of time for the SGCDC system to detect the second infected node in the NPS_blaster datasets from the time the second node sends the first SYN packet post-infection.

The SGCDC system also performed well in detecting all subsequently infected nodes in the simulation, resulting in an infected node detection and classification rate of one across all 30 trials. The amount of time for the SGCDC system to detect each of the subsequent infected nodes from their first post-infection SYN packet was less than 1.05 seconds across all trials as depicted in Figure 6.4. In fact, over 96 percent of the infected nodes were detected in less than one second after they began exhibiting worm infection behavior.

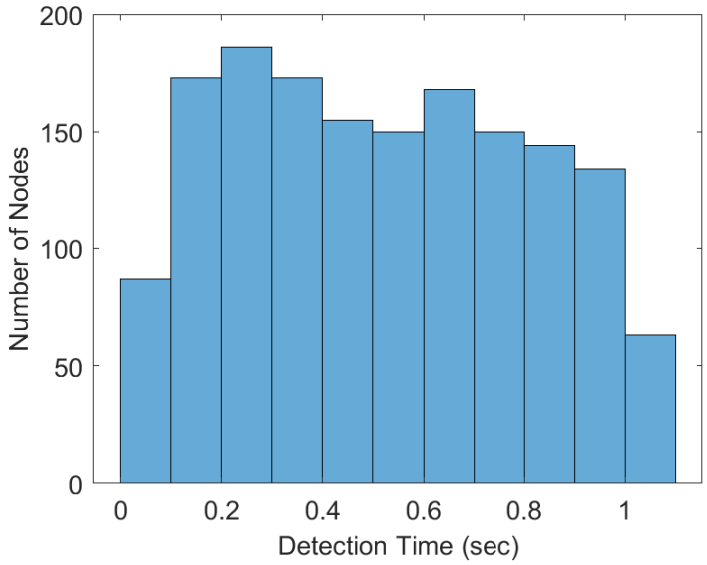


Figure 6.4. Histogram of the amount of time for the SGCDC system to detect each of the infected nodes from the time the node sends the first post-infection SYN packet, across 30 trials of the Blaster worm in the NPS_blaster datasets.

While the SGCDC system produced excellent worm and node detection and classification rates, it also produced some false alarms for node infections (i.e., identified a node as infected when not infected). Across the 30 trials, a total of 314 nodes were falsely identified by the SGCDC as infected during the worm attack. While the SGCDC system produced less than 25 nodal false alarms in the vast majority of the trials, four trials produced more than 25 false alarms with one trial producing 34 false alarms as depicted in Figure 6.5. This number of falsely identified nodes is an arguably acceptable cost given the SGCDC system correctly identifies all of the nodes that are infected by the worm. This could potentially halt any further spread of the infection. Nonetheless, further analysis is required to determine

the root cause of the nodal false alarms to develop a solution to decrease this cost.

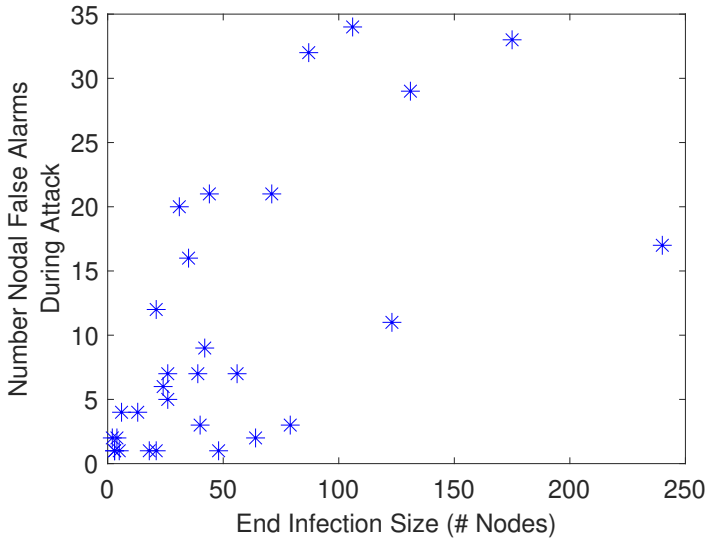


Figure 6.5. Number of nodes falsely detected as infected by the SGCDC system across 30 trials of the Blaster worm in the NPS_blaster datasets.

Upon further investigation of the source of the false alarms, the nodal false alarms are produced by under two different cases. First, a false alarm occurred when an infected node (i.e., parent node) attempted to exploit a host (i.e., offspring) in the network with the wrong operating system offset, and then that offspring node sent $T_{phantom}^{worm}$ or more SYN packets to unique destinations, which occurs with a 0.0012 probability according to Equation A.2, within one of the following windows while it was stored in the SGCDC system memory. Second, a false alarm occurred when T_{eigen} occupied one of the middle eigenspectrum indices. From this, we learn that while the strong node concept is strictly upheld by the principal (e.g., leading and trailing) eigenvalues and eigenvectors, there is some ambiguity in the very middle eigenspectrum indices. As a result, the SGCDC system might not be as accurate when the infection begins to grow in size. This requires additional investigation and is outside the scope of this dissertation.

Comparing the Blaster worm results to the WannaCry worm and hit-list worm results, the SGCDC system produced relatively similar worm detection and classification rates for the worm attacks, but the increased infection size of the Blaster worm revealed an underlying occurrence of increased nodal false alarms by the SGCDC system as the worm infection

spreads. If the identification of an infected node results in the removal of the node from the network until it is assessed, the network incurs an additional node unavailability cost up to approximately 2.5 percent for nodes that are falsely identified as infected. This is arguably an acceptable cost if it means the SGCDC system accurately identifies and removes all of the infected nodes from the network prior to further infection spread. In other words, the network could see a significantly larger number of unavailable nodes if the worm were to continue to rapidly spread and infect additional hosts. It is also important to note that this 2.5 percent unavailability cost assumes that no action is taken until the end of the simulation since no response mechanism was employed during these 30 trials; the unavailability cost theoretically drops closer to zero if the response mechanism removes the infected nodes from the network immediately upon detection and classification. Overall, the SGCDC system has reasonable performance results in detecting a variety of worm attacks.

6.2 SYN-flood DDoS Attack

In this section, we use the DARPA2009 dataset to examine the detection accuracy of the SGCDC system via MATLAB implementation and simulation against a SYN flood DDoS attack. As discussed in Section 4.3.4, the DARPA2009 dataset contains synthesized network traffic between a "/16" subnet (172.28.0.0/16) and the Internet. While a SYN flood DDoS attack exists within the traffic, the exact start time and all attacking nodes are not known. It is estimated that approximately 100 different hosts contribute to the attack, most of which contribute stealthily, but only hosts 19.202.221.71, 19.202.221.72, and 19.202.221.73 are specifically identified by the owners of the dataset as known attackers with significant contribution to the attack. Host 172.28.4.7 is identified as the victim of the attack [99]. While we estimate the DDoS attack to start sometime between 90 and 100 seconds in the dataset (from brief visual analysis of .pcap data), the known hosts do not begin contributing to the attack until around 114 seconds. The specific times of the first SYN for each of the three known attacking hosts are listed in Table 6.8.

Table 6.8. Start of SYN flood attack times for known attacking nodes in DARPA2009 dataset.

Host IP	Time of First SYN of Attack (sec)
19.202.221.71	114.748961
19.202.221.72	114.735132
19.202.221.73	114.74934

To evaluate the accuracy and timeliness of the proposed system against a DDoS attack, we set $T_{phantom}^{worm} = 15$ to account for Equation 4.3 and evaluated characteristics of numerous historical DDoS attacks. We then evaluate the SGCDC system using the same L_w utilized in the worm attack simulations. The detection times and detected attacked node by the SGCDC system are listed in Table 6.9 for the various L_w. For all L_w, the known attacking hosts are detected and correctly classified approximately 0.265 seconds after host 19.202.221.72 sent its first attacking SYN packet.

Table 6.9. Detection times and detected attacked nodes by the SGCDC system for the three known attacking hosts in the DARPA2009 dataset given $T_{phantom}^{DDoS} = 15$.

L_w (sec)	Attacker	Detected Victim	Detection Time (sec)
0.5	19.202.221.71	172.28.4.7	115
1	19.202.221.72	172.28.4.7	115
2	19.202.221.73	172.28.4.7	115

It is important to note that the SGCDC system detected and classified additional nodes in the DARPA2009 dataset, but unfortunately those nodes cannot be validated since only three of the approximately 100 attacking nodes have been specifically identified by the owners of the dataset as attackers. For $L_w = 2$ seconds, the SGCDC detected and classified 28 attacking nodes prior to the time in which host 19.202.221.72 sends its first attacking SYN. Additionally, 33 nodes other than the three known attacking nodes were detected during the window in which the three known attacking nodes were detected. For $L_w = 0.5$ and $L_w = 1$ seconds, the SGCDC detected and classified zero attacking nodes prior to the time in which

host 19.202.221.72 sends its first attacking SYN. The system also detected 16 nodes other than the three known attacking nodes during the window in which the three known attacking nodes were detected. From this, we conclude that the SGCDC system requires a longer L_w to detect stealthier attacks and a shorter L_w to detect more overt attacks. Still, a longer L_w increases the theoretical system false alarm rate (see Section 4.1), which might explain the additional nodes detected by the $L_w = 2$ seconds during the window in which the three known attackers are detected.

Overall, the SGCDC system performed well in detecting the DDoS attack in the DARPA2009 dataset. Similar to the detection performance of the proposed system against worm attacks, the system produced a detection and classification rate of one for all evaluated sets of external, input parameters.

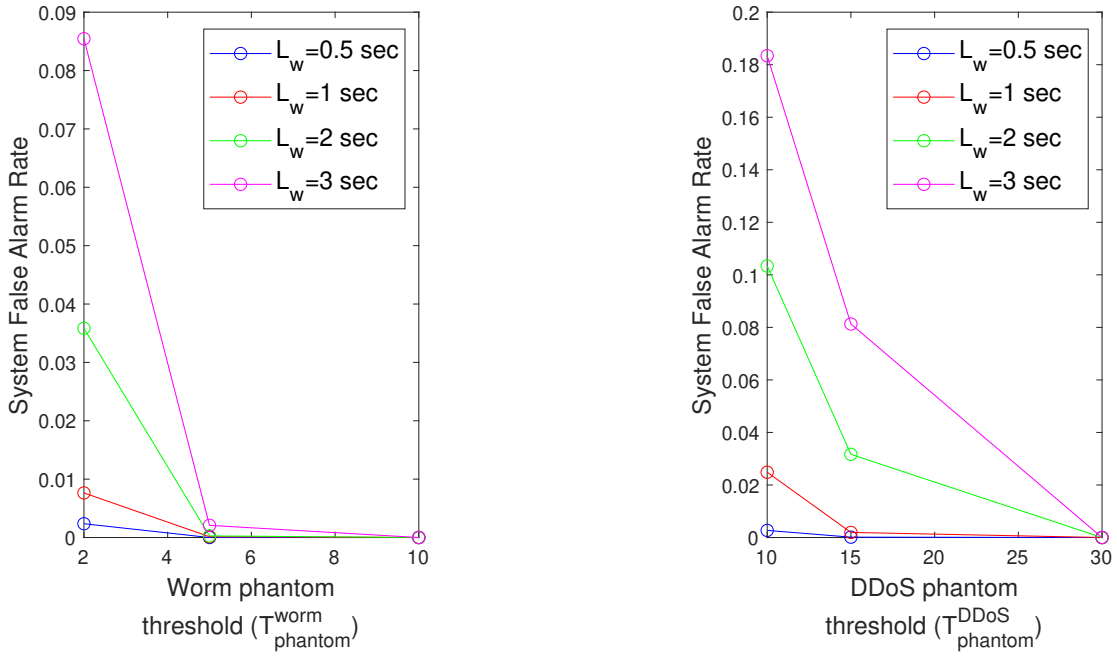
6.3 False Alarm Rates

In this section, we use portions of the DARPA2009, NPS2013, and MAWI2015 datasets that contain normal network traffic to examine the false alarm rates of the SGCDC system via MATLAB implementation and simulation. Due to the difference between the $T^{worm}_{phantom}$ and $T^{DDoS}_{phantom}$ values utilized in this dissertation, each dataset is evaluated for worm false alarms independent of DDoS false alarms. In accordance with Sections 6.1 and 6.2, we set $t_{log} = 40$ seconds to account for longer infection times due to system reboots [106]. We also select $T^{worm}_{phantom}$ and $T^{DDoS}_{phantom}$ values that align with evaluated characteristics of numerous historical attacks and support the required external, input parameter relationships described by equations 3.8, 3.9, and 3.10.

For the DARPA2009 dataset, we evaluated the worm false alarm rates of the SGCDC system for all combinations of $T^{worm}_{phantom} = \{2, 5, 10\}$ and $L_w = \{0.5, 1, 2, 3\}$ and DDoS false alarm rates for all combinations of $T^{DDoS}_{phantom} = \{10, 15, 30\}$ and $L_w = \{0.5, 1, 2, 3\}$. For all combinations, both the worm false alarm rate and DDoS false alarm rate was zero. These results were expected due to the relatively small amount of traffic that was synthetically

generated across the nodes.

For the NPS2013 dataset, we evaluate the worm false alarm rates of the SGCDC system for all combinations of $T^{worm}_{phantom}$ = {2, 5, 10} and L_w = {0.5, 1, 2, 3} and DDoS false alarm rates for all combinations of $T^{DDoS}_{phantom}$ = {10, 15, 30} and L_w = {0.5, 1, 2, 3}. The SGCDC system worm false alarm rates are shown in Figure 6.6a and DDoS false alarm rates are shown in Figure 6.6b. In general, the false alarm rate increased as L_w increased, and the false alarm rate decreased as $T^{worm}_{phantom}$ or $T^{DDoS}_{phantom}$ increased. As expected, the largest false alarm rate for both worm and DDoS occurred when the smallest threshold value was combined with the largest L_w. It is also important to note that a $L_w > 1$ second provided DDoS false alarm rates that are too high to be usable in a real-world system.



(a) Worm false alarm rates

(b) DDoS false alarm rates

Figure 6.6. Worm and DDoS false alarm rates of the SGCDC system in the NPS2013 dataset for various $T_{phantom}$ and a t_{log} = 40 seconds.

To compare these observed results to the theoretical system false alarm rate from Equation

4.18, the worm false alarm rate for $L_w = 1$ and $T^{worm}_{phantom} = 5$ was combined with the DDoS false alarm rate for $L_w = 1$ and $T^{DDoS}_{phantom} = 15$. The observed worm false alarm rate was approximately 0.00013, and the observed DDoS false alarm rate was approximately 0.00194, resulting in an observed system false alarm rate of approximately 0.0021. This rate falls approximately halfway between the theoretical most-likely system false alarm rate of 0.00012 and the theoretical system false alarm upper bound rate of 0.00417. To add further analysis, this set of external, input parameters also produced a worm detection and classification rate of one for the Stratosphere_WC dataset, NPS_WC dataset, NPS_hitlist dataset, and NPS_blaster dataset, and a DDoS detection and classification rate of one for the DARPA2009 dataset.

For the MAWI2015 dataset, we only utilize the first five seconds due to the network size (20,820 nodes) and volume of traffic across the nodes. The SGCDC system worm false alarm rates were evaluated for $L_w = \{0.5, 1, 2\}$ with a $T^{worm}_{phantom} = \{5, 10\}$. The SGCDC system produced a worm false alarm rate of zero across all L_w. The system DDoS false alarm rates are evaluated for all combinations of $T^{DDoS}_{phantom} = \{15, 30\}$ and $L_w = \{0.5, 1, 2\}$. The DDoS false alarm rates produced by the SGCDC system are provided in Table 6.10. While we expected the DDoS false alarm rates to be slightly skewed due to the small number of windows being analyzed and due to results shown in figures 4.7 and 4.8 (see Section 4.3.4), the observed false alarm rates were extremely high. These large false alarm rates indicate that modifications would need to be made to the SGCDC system in order to produce a solution that is viable for large real-world networks.

Table 6.10. False alarm rates of the SGCDC system for first 5 seconds of MAWI dataset.

$T^{DDoS}_{phantom}$	L_w (sec)	False Alarm Rate
15	0.5	0.564
	1	0.684
	2	1.0
30	0.5	0.128
	1	0.632
	2	1.0

One solution to counter the extremely large false alarm rates is to significantly decrease the

L_w or significantly increase the $T_{phantom}^{DDoS}$. This solution would be easy to implement and would theoretically decrease the false alarm rate; however, it might also negatively affect the detection rates.

Upon further investigation of the false alarms, there was a lack of uniformity in SYN traffic across the nodes in the network that produced the MAWI2015 dataset. Some nodes were more active and regularly sent a significantly greater number of SYN packets compared to other nodes in the network. Equation 4.17 assumes a uniform probability of nodes sending and/or receiving SYN packets, which is clearly not the case for this network. Given a network with non-uniform nodal SYN traffic, the better solution would be establishing multiple T_{eigen}^{DDoS} in the eigenspectrum index. While this would require additional baseline knowledge of nodal SYN traffic (e.g., baseline of SYN traffic for different node functions or identification of nodes that regularly send a larger amount of SYN traffic), it would theoretically decrease the false alarm rate while maintaining a high detection rate. Due to time constraints, we were unable to explore this solution; additional research needs to be conducted to determine the degree to which the false alarm rates could be decreased using this solution.

In summary, the SGCDC system produces low false alarm rates in smaller networks that have uniform SYN traffic across the nodes but produces extremely large DDoS false alarm rates in larger networks that contain disparate nodal SYN traffic.

6.4 System Response

In this section, we use epidemic modeling to evaluate the system countermeasure response given the detection rates determined in Section 6.1. For defensive countermeasures, there are two fundamentally different approaches to thwarting cyber attacks: reactive and proactive. Each defensive countermeasure provides some degree of mitigation to the overall spread of the attack; however, each countermeasure also has some associated cost (i.e., system downtime, network delays, resource consumption, etc.) [77]. Since the response component of the proposed SGCDC system is reactive, we analyze the system response by implementing a packet blocking countermeasure in the SGCDC system response component. This countermeasure, similar to the packet filtering response in [77], drops all incoming and outgoing packets of an attacked node. In essence, the packet blocking response removes the

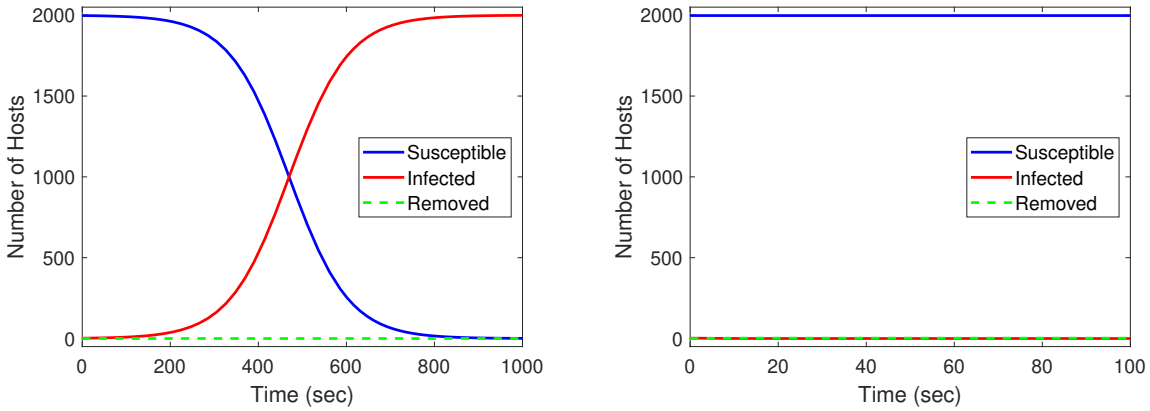
node from the network which in turn nullifies the DDoS attack or prevents further spread of the worm attack. As a result, the SIR model (see Chapter 2) is employed to evaluate this countermeasure response against slow- and fast-propagating worm attacks. To simulate the SIR system described by the set of deterministic equations in Equation 2.23, we utilize the Runge-Kutta fourth-order method in MATLAB to account for the iterative process.

6.4.1 Slow-Propagating Worm

The SIR system is evaluated with and without the deployment of the packet blocking response for a slow-spreading worm attack. For both simulations, $N_{tot} = 2000$, $S_{tot} = 1998$, $I_{tot} = 2$, and $R_{tot} = 0$ to account for the inability of the SGCDC system to detect the worm attack until the second node is infected. We also utilize attack parameters gleaned from the NPS_blaster dataset for both simulations.

Across the 30 one-hour trials in Section 6.1.3, the average propagation rate is estimated to be 0.0147 nodes per second, resulting in $b = 0.0147/N_{tot}$. The rate that the nodes are removed from the network k is dependent on the deployment of the packet blocking countermeasure. For the system that does not deploy the packet blocking countermeasure, $k = 0$. For the system that deploys the packet blocking countermeasure, we assume that the time between detection and response is negligible and set $k = 1$ to correspond with the SGCDC detection results from the Blaster worm simulations. The resulting dynamical behavior of the network for the slow-spreading worm, with and without countermeasure response, is shown in Figure 6.7.

For the simulation without countermeasure, the number of infected nodes slowly grows over time and heads toward an epidemic equilibrium. This makes sense given $R_0 \geq 1$. For the simulation with countermeasure employment, the SGCDC system quickly detects and identifies the infected nodes and removes their traffic from the network, resulting in the elimination of the attack. Overall, the detection and classification capabilities of the SGCDC system allowed for a network response that could potentially stop a slow-spreading worm attack before it gains traction.



(a) Dynamical behavior without countermeasure (b) Dynamical behavior with countermeasure

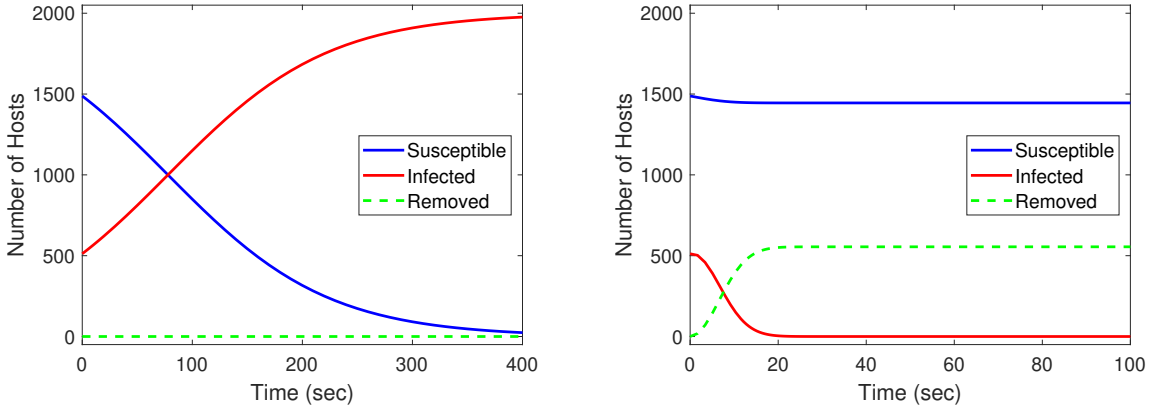
Figure 6.7. Dynamical behavior of a network experiencing a slow-propagating worm attack with $b = 0.0147/N_{tot}$ and: a) no deployment of countermeasures by SGCDC system resulting in $k = 0$, and b) deployment of packet blocking countermeasure by SGCDC system resulting in $k = 1$.

6.4.2 Fast-Propagating Worm

The SIR system is evaluated with and without the deployment of the packet blocking response for a fast-spreading worm attack. We assume that the worm has a hit-list of 512 nodes and then transitions to a random scanning technique similar to the WannaCry worm. As a result, $N_{tot} = 2000$, $S = 1488$, $I = 512$, and $R = 0$ for both simulations to account for the extremely fast infection of the 512 nodes on the hit-list. We also utilize attack parameters gleaned from the NPS_WC dataset for both simulations.

For both simulations, $b = 0.0137/N_{tot}$ since the average propagation rate is estimated to be approximately 0.0137 nodes per second from analysis of the detection times in Table 6.1. To simulate the SGCDC system without countermeasure deployment, $k = 0$. On the other hand, for the simulation of the SGCDC system with countermeasure deployment, we assume the time between detection and response is negligible and set $k = 0.519$ to correspond with the average SGCDC detection results from both the hit-list and WannaCry worm simulations. The resulting behavioral dynamics of the network for the fast-spreading worm with and without countermeasure response, is shown in Figure 6.8.

For a fast-spreading worm attack like the hit-list worm, the SGCDC system has a very



(a) Dynamical behavior without countermeasure (b) Dynamical behavior with countermeasure

Figure 6.8. Dynamical behavior of a network experiencing a fast-propagating worm attack with $b = 0.0137/N_{tot}$ and: a) no deployment of countermeasures by SGCDC system resulting in $k = 0$, and b) deployment of packet blocking countermeasure by SGCDC system resulting in $k = 0.519$.

low nodal detection rate during the hit-list period resulting in an initially large infected set of nodes. While the SGCDC system is capable of identifying the infected nodes (see Section 6.1), the number of infected nodes continued to rapidly grow and head toward on epidemic if the system does not deploy countermeasures against the attack. This makes sense given $R_0 \geq 1$. On the other hand, for the SGCDC system that deployed a countermeasure response, the initially infected nodes were quickly removed from the network after the worm transitioned from hit-list to traditional random scanning techniques, resulting in the elimination of the attack.

Similar to the response results for the slow-propagating worm, the detection and classification capabilities of the SGCDC system allowed for network responses that can potentially stop a worm attack even after it gains initially gains some traction. Still, the response component of the SGCDC system is directly tied to the detection and classification results. Our system is unable to accurately identify the infected nodes in the very beginning seconds of a fast-propagating worm if the infected nodes send a number of SYN packets less than the phantom component threshold within the specified L_w. As a result, the response mechanism would not be able to isolate those infected nodes from the network and the infection would continue to spread. As long as the SGCDC has an accurate infected node identification

rate less than or equal to b, the worm would continue to spread and infect. This means the response mechanism performs optimally when the SGCDC has an accurate infected node identification rate significantly larger than b.

This chapter has validated the performance of the proposed SGCDC system in accurately detecting worm and DDoS attacks. The system provides optimal defense against slow- to medium-spreading worms attacks and overt DDoS attacks but effectively detects and classifies the existence of all attack types. We also validated the system performance against normal network traffic and determined that the system has lower false alarm rates for smaller networks and for networks that have uniform SYN traffic across all nodes. We provided a potential solution in the form of multiple thresholds in the eigenspectrum index for larger networks and networks that do not contain uniform SYN traffic across all nodes. Finally, we validated the system response component given the observed detection rates of the system and found that the response mechanism is effective in countering an attack in the network. Overall, the system performs as designed and is adaptable for different size networks and different types of attacks.

CHAPTER 7:
Conclusion

In this dissertation, we formulated and validated a biological immune system-inspired SGCDC system using spectral graph theory techniques, such as phantom components, the strong node concept and the dual-degree matrix, to detect, classify and respond to TCP scan-based worm attacks and SYN flood DDoS attacks. Using real-time network flows as the input to the SGCDC system, we identified the five functional components necessary to detect, classify and respond to such attacks, and provided a detailed explanation of the processes that occur within each functional component.

We then conducted analysis of the proposed system under both ideal and non-ideal network conditions. With respect to ideal network conditions, we developed false alarm equations to predict and assess the performance of the proposed system for different sets of external, input parameters. We found that the proposed system is more likely to have a near-zero false alarm rate than a high false alarm rate for the sets of external, input parameters analyzed. With respect to non-ideal network conditions, we utilized perturbation theorems and a novel application of the n-dimensional Euclidean distance formula to analyze the effects of network congestion and noise on the spectral graph detection mechanism. Regarding the specific network scenario evaluated, we determined that the degree of impact depended on the amount of link perturbations relative to the rate of the attack.

Finally, we developed a simulation of the proposed SGCDC system and examined its performance against synthetic and real-world datasets containing both normal and attack traffic. The results validated the detection, classification, and response mechanisms. First, the attack datasets were evaluated for detection rate and detection time. The results suggested that the proposed system provides optimal defense against slow- to medium-spreading worm attacks and overt DDoS attacks. Regardless, the system detected the existence of an attack in all but one of the simulations. Next, the normal traffic datasets were evaluated for false alarm rates. The results suggested that the system experiences lower false alarm rates for networks that contain uniform SYN traffic across all nodes. We identified additional practices that could be implemented to optimize the proposed system performance in networks with

higher observed false alarm rates, effectively providing flexibility and adaptability of the SGCDC system. Lastly, the response mechanism was analyzed using a packet blocking countermeasure and the observed detection rates of the system against the worm attack datasets. Overall, the results from our simulations support the conclusion that the proposed SGCDC system quickly and effectively detects, classifies, and responds to worm and DDoS attacks.

7.1 Contributions

The results of this work have made several novel and notable contributions. Specifically, this work provides a novel method of spectral graph-based cyber attack detection and classification using phantom components, the strong node concept, and the dual-degree matrix. Extensive simulations, including real-world attack data, validate the performance and flexibility of the proposed system to detect, classify, and respond to a variety of cyber attacks under different network types. Additionally, this work develops a novel mixture model to describe network SYN traffic, which led to the derivation of a theoretical upper bound on system false alarm rates to facilitate system analysis. Finally, this work develops a novel method of analyzing eigenvector perturbations within the nodal basis of the dual basis, which provided a basis for theoretical analysis of the system in a network with noise and/or congestion.

7.1.1 Spectral Graph-Based Cyber Defense Model

The proposed SGCDC system is the first demonstration of a spectral-graph based cyber system designed to detect, classify, and respond to cyber attacks. Our simulations have vetted the proposed SGCDC system against multiple attack scenarios to determine the ability of the system to detect various worm and DDoS attacks. The proposed system accurately detected the existence of an attack in every simulation except one where the set of external, input parameters was not optimal for detection. The system also accurately classified the attack type in every simulation. Given a $L_w < 1$ seconds, $T^{worm}_{phantom} > 5$, and $T^{DDoS}_{phantom} > 15$, the system had a false alarm rate of approximately 0.002 for worm and DDoS attack detection. Overall, these combined detection and false alarm rates are better than those previously observed in literature [26], [29], [35] for methods to detect both worm and DoS attacks.

In addition to overall attack detection, the proposed system was 100 percent accurate in identifying all attacked nodes for the overt DDoS attack in the DARPA2009 dataset and the slow- to medium-spreading worm attacks in the Stratosphere_WC, NPS_WC, and NPS_blaster datasets. Furthermore, the attacked or infected nodes of the simulated DDoS and worm attacks (excluding the first node of the worm attack) were all identified in less than 1.6 seconds after it began exhibiting attack behavior in the NPS_WC, NPS_blaster, and DARPA2009 datasets. In fact, the vast majority of those nodes were identified in less than one second. With an absence of detection times in literature for methods to detect both worm and DoS attacks [26], [29], [35], these results provide a new empirical benchmark for detection.

Due to the variety of attacks simulated, the observed system performance, and the fundamental concept behind the phantom component, it is reasonable to conclude that this system is theoretically capable of detecting zero-day worm attacks that exhibit traditional scan-based propagation characteristics, which cannot be claimed in the biological immune system-inspired cyber defense systems [14], [20], and two out of the three graph theory-based, and spectral graph theory-based, cyber defense systems [21], [23], reviewed in literature. For the one graph theory-based cyber defense system capable of detecting zero-day worm attacks [22], that system is unable to intrinsically identify the nodes involved in the attack and is limited to detecting worm attacks only.

7.1.2 Spectral Graph Detection and Classification Techniques

The SGCDC system utilizes innovative techniques to perform spectral graph-based detection and classification of worm and DDoS attacks. The novel graphical attack behavior representation via phantom components and the newly-defined strong node concept contribute to the development of attack behavior thresholds within the eigenspectrum index of the dual-basis to detect worm and DDoS attacks. This detection mechanism provides an original method of not only identifying the existence of an attack, but also intrinsically identifying the specific attacked node(s), additional nodes involved in the attack, and estimated attack rates. Combined with the innovative dual-degree matrix for attack classification, the system produces the necessary information to quickly and effectively respond with network countermeasures.

7.1.3 Lévy-Impulse Mixture Model

To facilitate theoretical false alarm rate analysis of the proposed SGCDC system under ideal network conditions, the novel Lévy-impulse model was developed. Through simulation, we found that traditional single-distribution models did not accurately describe the SYN traffic characteristics observed in real-world traffic. We also found that none of the previously proposed mixture models in literature accurately characterized SYN traffic. Additionally, we found that traditional truncation methodologies in mixture modeling literature, which utilize multiple weighted distributions, did not produce the most accurate model for the high zero-count and heavy tail characteristics of network SYN traffic. Thus, we developed an unconventional truncation methodology by combining an impulse with a single weighted distribution. The resulting Lévy-impulse model was shown to accurately characterize network SYN traffic in multiple real-world and synthetic datasets. It also allowed us to derive false alarm rate upper bounds for the SGCDC system given a set of external, input parameters. As a result, network administrators are able to analyze sets of external, input parameters to produce a desired level of SGCDC system performance.

7.1.4 Nodal Basis Perturbation Analysis

To facilitate detection analysis of the propose SGCDC system under non-ideal network conditions, we proposed a novel application of the n-dimensional Euclidean distance formula. This $(n-1)$-dimensional space application provided additional information that traditional perturbation analysis techniques could not provide with respect to the effects of noise or congestion on the detection mechanism of the proposed SGCDC model. Specifically, we found that the spectral graph detection component is robust against noise in the adjacency matrix when the $R_{attack} \gg l_{perturb}$ or $R_{scan} \gg l_{perturb}$; however, the detection accuracy of the spectral graph detection mechanism can be decreased if $R_{attack} \approx l_{perturb}$ or $R_{scan} \approx l_{perturb}$. In other words, we found that network noise and congestion theoretically have no impact on the detection capability of the SGCDC for fast scanning worms and large, overt DDoS attacks.

7.2 Future Work

An expansion of the contributions presented in this dissertation, along with the removal of certain assumptions and constraints, offer opportunity for future work. Specifically, we

suggest additional research in four areas: implementation against other cyber attacks, node-dependent SYN traffic modeling, evaluation of multiple attack-specific thresholds in the eigenspectrum index, and determining the breakdown point for the strong node concept.

7.2.1 Detection of Other Cyber Attacks

In this work, the detection and classification of cyber attacks was limited to TCP scan-based worm attacks and SYN flood DDoS attacks. In a realistic application, the SGCDC system needs to be able to also detect user datagram protocol (UDP) worm attacks and other types of DDoS attacks. While the UDP worm implementation would take very little modification, other types of DDoS attacks would require the identification of the attack behavior to implement the necessary traffic filters.

Additionally, the implementation of the SGCDC system against other forms of cyber attacks would greatly complement the work in this dissertation. Similar to other types of DDoS attacks, this would require the identification of the attack behavior to develop the necessary phantom component and to implement the necessary traffic filters. This line of research could be further stretched to produce a modified SGCDC host-based system to other attacks, like viruses, that may not produce network traffic. In this case, researchers should consider other mechanisms to describe the phantom component attack behavior, such as assembly language or machine processes.

7.2.2 Node-Dependent SYN Traffic Modeling

In developing the theoretical upper bounds for the SGCDC system false alarm rate, we assumed a uniform SYN traffic behavior across all nodes in the network. As a result, we averaged the SYN traffic behaviors across all the nodes in the NPS2013 dataset to develop the Lévy-impulse model. Yet, during validation of the proposed system, we observed a higher than expected false alarm rate within real-world networks and traced the false alarms back to a lack of uniformity in nodal SYN traffic. Since certain nodal functions can result in more network SYN traffic, function-dependent SYN traffic models need to be developed for a variety of network node types in order to provide researchers with a more accurate assessment of the performance of the proposed system relative to their network.

For research into implementation of the proposed system against other cyber attacks, these

function-dependent traffic models can be further expanded upon for other network features that characterize the different attack behaviors. Additionally, it would be interesting to determine if the proposed Lévy-impulse model accurately describes other network traffic features and is a better fit for general network feature modeling than those previously proposed in literature.

7.2.3 Multiple Eigenspectrum Index Thresholds

During the SGCDC system validation, we observed higher than expected false alarm rates in larger networks and networks that do not have uniform SYN traffic across all nodes in the network. We proposed the implementation of multiple attack-specific eigenspectrum index thresholds to decrease the false alarm rate while maintaining a high detection rate. Due to time constraints, we were unable to investigate this proposed solution. To explore this concept, nodes within the network would need to be independently analyzed for SYN traffic behavior to determine baseline normal traffic levels. Then, the nodes would need to be grouped together based on their normal baseline levels, and a T_{eigen}^{worm} and T_{eigen}^{DDoS} would be set for each group of nodes. In addition, the SGCDC system would need to be modified to account for nodes associated to specific thresholds. The resulting research could potentially provide more flexibility in the implementation SGCDC system across various networks.

7.2.4 Strong Node Concept Breakdown Point

During the SGCDC system validation, we also observed an increase in false alarms when T_{eigen} occupied one of the middle eigenspectrum indices, and we concluded that ambiguity exists for the strong node concept within the very middle eigenspectrum indices. Additional research needs to be conducted to analyze the strong node concept in the middle eigenvectors in association to both the size of the eigenspectrum index and the number of nodes above T_{eigen}. This would provide more evidence toward the capability of the SGCDC to detect attacks of various sizes.

APPENDIX A:
State Model Transition Derivations

In this appendix, we provide the derivation of $P(D_{worm})$ associated with the system state model discussed in Section 4.1 and the derivations of $p_{1,3}$, $p_{1,2}$, and $p_{2,1}$ associated with the node state model discussed in Section 4.2.

A.1 Probability of Worm Detection

In this section, the probability of worm detection $P(D_{worm})$ using the proposed SGCDC system is derived. In order for worm traffic to be detected, three conditions must be met: 1) node A sends a number of SYN packets greater than or equal to $T^{worm}_{phantom}$, 2) node B receives one of those SYN packets sent by node A, and 3) node B sends a number of SYN packets greater than or equal to $T^{worm}_{phantom}$ while it is being stored in system memory as a offspring of a potential worm attack. As a result, we have

$$P(D_{worm}) = N_{sending} \times P(S^{n_A}_{tx} > T^{worm}_{phantom}) \times P(n^{SYN}_{A,B}) \times P(S^{n_B}_{tx} > T^{worm}_{phantom}) \times W_{logged} \quad \text{(A.1)}$$

where $N_{sending}$ is potential sending nodes in the network for a given communication, $P(S^{n_A}_{tx} > T^{worm}_{phantom})$ is the probability that node A is detected as exhibiting worm attack behavior, $P(n^{SYN}_{A,B})$ is the probability that node A sends one of the SYN packets to node B, $P(S^{n_B}_{tx} > T^{worm}_{phantom})$ is the probability that node B is detected as exhibiting worm attack behavior, and W_{logged} is the number of windows that an offspring is stored in memory.

Given N_{total} nodes in the network, the potential number of sending nodes in the network can be shown to be

$$N_{sending} = N_{total} - 1 \quad \text{(A.2)}$$

since one node must be the receiving node of a communication. The probability that a given node n_i is detected as exhibiting worm attack behavior is given by

$$P(S^{n_i}_{tx} > T^{worm}_{phantom}) = \int_{T^{worm}_{phantom}}^{\infty} f^{L_w}_{SYN}(x)dx. \quad \text{(A.3)}$$

Assuming a uniform distribution for the probability that any two nodes communicate, it can be shown that

$$P(n_{A,B}^{SYN}) = \frac{S_{tx}^{n_A}}{N_{total} - 1}. \tag{A.4}$$

Given L_w and t_{log}, we have

$$W_{logged} = \frac{2 \times t_{log}}{L_w}. \tag{A.5}$$

From equations A.2, A.3, A.4, and A.5, Equations A.1 can be reduced to

$$P(D_{worm}) = c_{det}^{worm} \left(\int_{T_{phantom}^{worm}}^{\infty} f_{SYN}^{L_w}(x)dx \right)^2 \tag{A.6}$$

where c_{det}^{worm} is the worm detection scaling factor defined as

$$c_{det}^{worm} = \left(\frac{2 \times S_{tx}^{n_i} \times t_{log}}{L_w} \right). \tag{A.7}$$

A.2 Probability of Nodal Transitions

In this section, the nodal state transition probabilities $p_{1,2}$, $p_{1,3}$, and $p_{2,1}$ are derived. For $p_{1,2}$, a given node n_i becomes a first-level offspring of a potential worm attack if one of the network sending nodes sends a number of SYN packets greater than or equal to $T_{phantom}^{worm}$ and n_i is the recipient of one of those SYN packets. Assuming a uniform distribution for the probability that any two nodes communicate, we have

$$p_{1,2} = N_{sending} \times P(S_{tx}^{n_i} > T_{phantom}^{worm}) \times P(n_{A,B}^{SYN}). \tag{A.8}$$

Substituting equations A.2, A.3, and A.4 into Equation A.8 and reducing the result produces

$$p_{1,2} = S_{tx}^{n_A} \int_{T_{phantom}^{worm}}^{\infty} f_{SYN}^{L_w}(x)dx. \tag{A.9}$$

For $p_{1,3}$, either a DDoS attack is detected, or the node is detected as a $P_{1,j}^{w}$ of a worm attack. Consequently, we have

$$p_{1,3} = P(D_{DDoS}) + P(n_i = P_{1,j}^{w}) \tag{A.10}$$

where $P(n_i = P^w_{1,j})$ is the probability that a node is detected as a first-level-parent of a worm attack. In order for n_i to be detected as $P^w_{1,j}$, n_i must send a number of SYN packets greater than or equal to $T^{worm}_{phantom}$, and one of the recipients of one of those SYN packets must send a number of SYN packets greater than or equal to $T^{worm}_{phantom}$ while it is being stored in system memory as a offspring of a potential worm attack. As a result, it can be shown that

$$P(n_i = P^w_{1,j}) = P(S^{n_A}_{tx} > T^{worm}_{phantom}) \times S^{n_A}_{tx} \times P(S^{n_B}_{tx} > T^{worm}_{phantom}) \times W_{logged}. \tag{A.11}$$

Substituting equations A.3 and A.5 into Equation A.11, and reducing the results yields

$$P(n_i = P^w_{1,j}) = \frac{2 \times S^{n_A}_{tx} \times t_{log}}{L_w} [\int_{T^{worm}_{phantom}}^{\infty} f^{L_w}_{SYN}(x)dx]^2. \tag{A.12}$$

Substituting Equation A.12 into Equation A.10 results in

$$p_{1,3} = P(D_{DDoS}) + \frac{2 \times S^{n_A}_{tx} \times t_{log}}{L_w} [\int_{T^{worm}_{phantom}}^{\infty} f^{L_w}_{SYN}(x)dx]^2. \tag{A.13}$$

For $p_{2,1}$, a node transitions back to the normal state if two conditions are met: 1) the node does not send a number of SYN packets greater than or equal to $T^{worm}_{phantom}$ while it is being stored in system memory as a offspring of a potential worm attack, and 2) the node does not receive another SYN packet from another node that sends a number of SYN packets greater than or equal to $T^{worm}_{phantom}$ while it is being stored in system memory as a offspring of a potential worm attack. Assuming uniform value of $S^{n_A}_{tx}$ for all nodes found above $T^{worm}_{eigenspace}$, we can write

$$p_{2,1} = [1 - P(S^{n_i}_{tx} > T^{worm}_{phantom})]^{W_{logged}} \times [\prod_{k=1}^{S^{n_A}_{tx}} \frac{N_{sending} - k}{N_{sending}}]^{O_{repeat}} \tag{A.14}$$

where O_{repeat} is the number of repeated occurrences, defined as

$$O_{repeat} = N_{sending} \times P(S^{n_i}_{tx} > T^{worm}_{phantom}) \times W_{logged}. \tag{A.15}$$

Substituting equations A.2, A.3, A.5, and A.15 into Equation A.14 and reducing the results

yields

$$p_{2,1} = [\int_0^{T_{phantom}^{worm}} f_{SYN}^{L_w}(x)dx]^{(2\times t_{log})/Lw} \times [\prod_{k=1}^{S_{tx}^{n_A}} \frac{N_{sending}-k}{N_{sending}}]^{[(2\times(N_{total}-1)\times t_{log}\times\int_{T_{phantom}^{worm}}^{\infty} f_{SYN}^{L_w}(x)dx)/L_w]}. \quad \text{(A.16)}$$

APPENDIX B:
Additional Results

In this appendix, we provide additional results for simulations conducted in Chapters 4 and 5 in support of analysis of the SGCDC system.

B.1 Theoretical False Alarm Rate

In this section, additional results for $P(D^{sys}_{false})$ and $P(D^{node}_{false})$ are provided for various sets of external, input parameters. The chosen parameter values align with equations 3.8, 3.9, and 3.10, and the evaluated characteristics of historical worm and DDoS attacks. The results for $P(D^{sys}_{false})$ are shown in figures B.1 and B.2. The results for $P(D^{node}_{false})$ are shown in figures B.3 and B.4. See Section 4.4 for additional information.

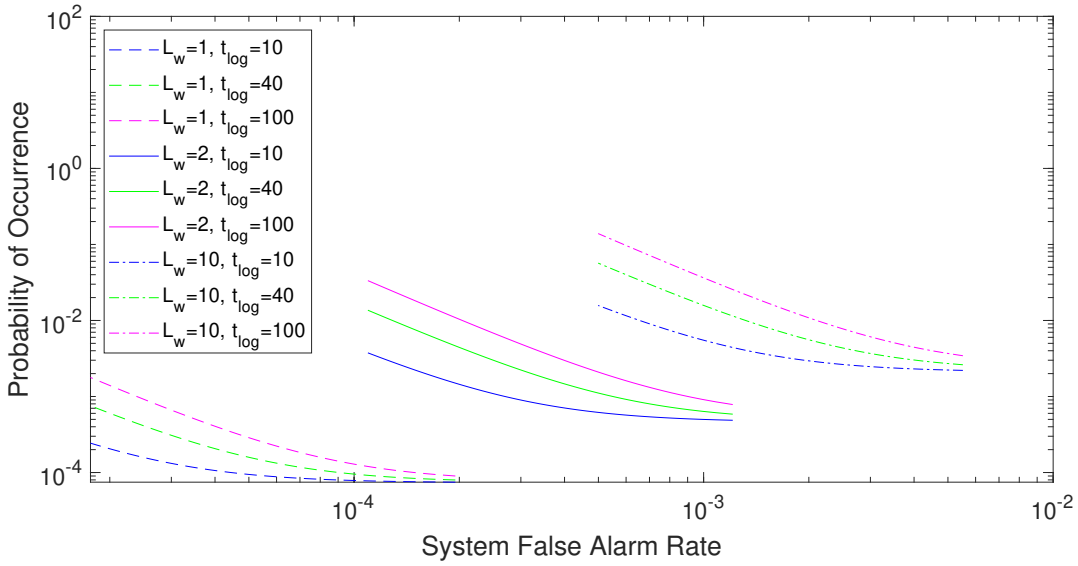
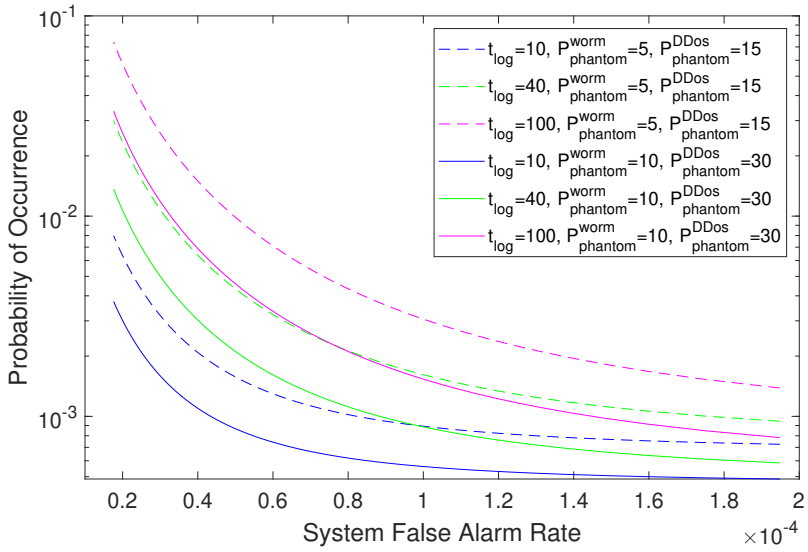
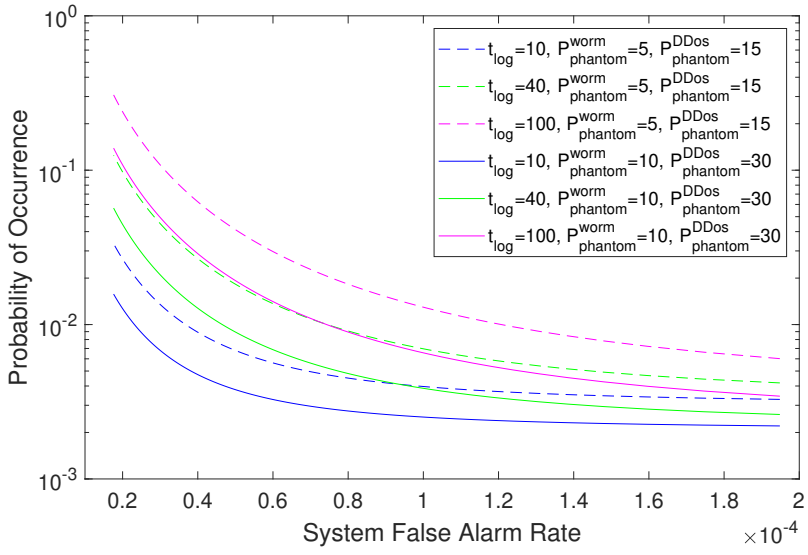


Figure B.1. Probability of a given system false alarm rate for various sets of external, input parameters, given $T^{worm}_{phantom}$ = 10, $T^{DDoS}_{phantom}$ = 30, and the units of L_w and t_{log} are seconds.



(a) $P(D^{sys}_{false})$ results for $L_w = 2$



(b) $P(D^{sys}_{false})$ results for $L_w = 10$

Figure B.2. Probability of a given system false alarm rate for various sets of external, input parameters, given $L_w = 2$ and $L_w = 10$ seconds, and the unit of t_{log} is seconds.

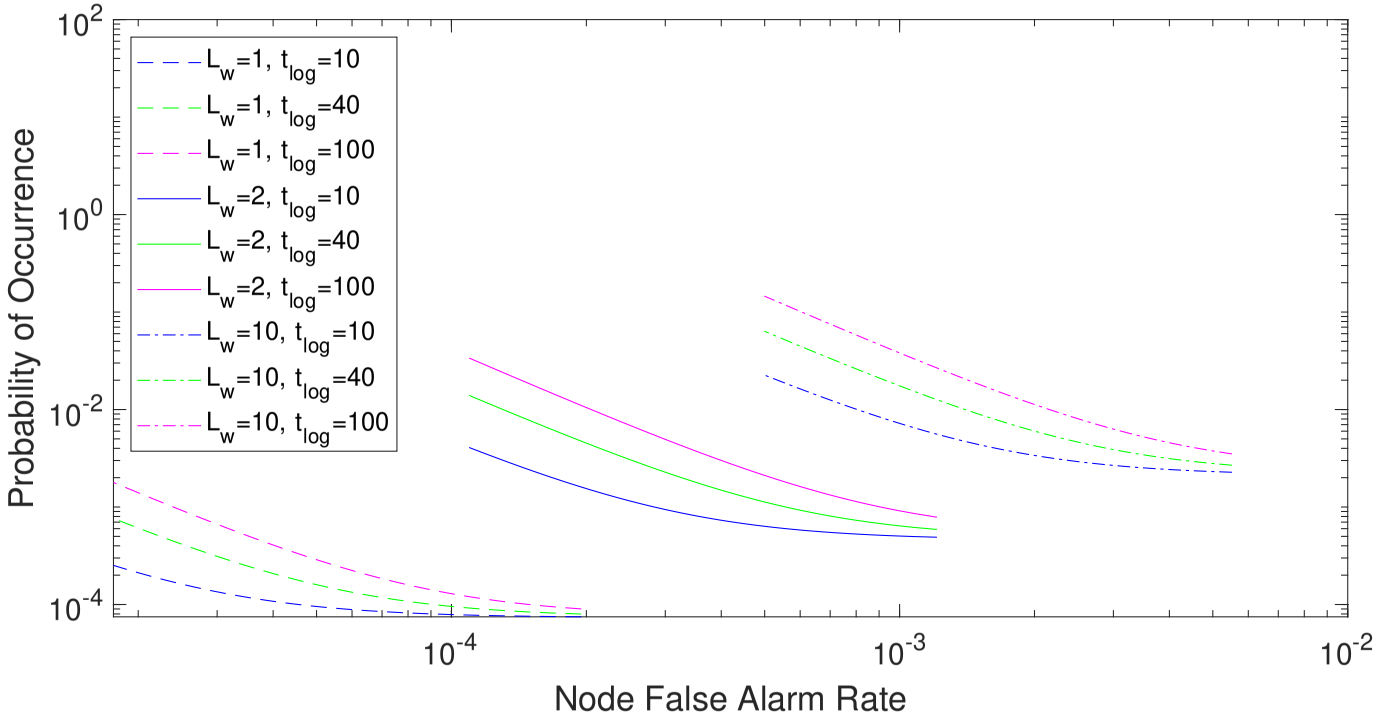
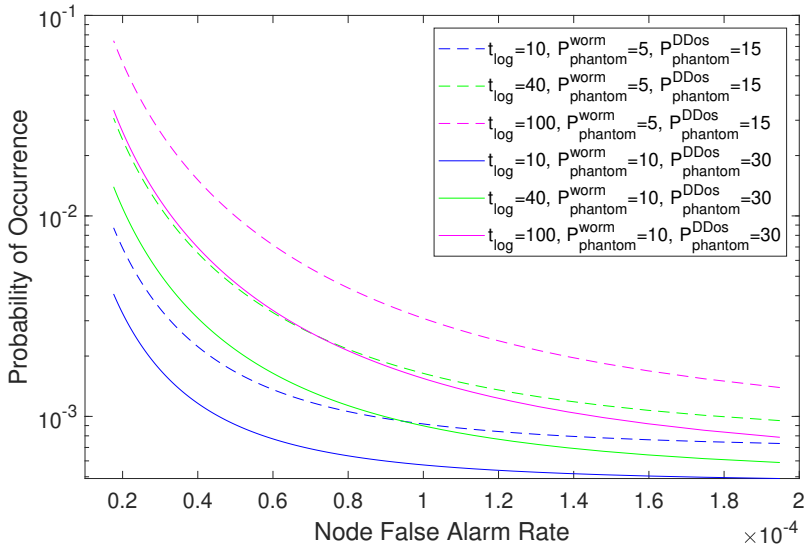
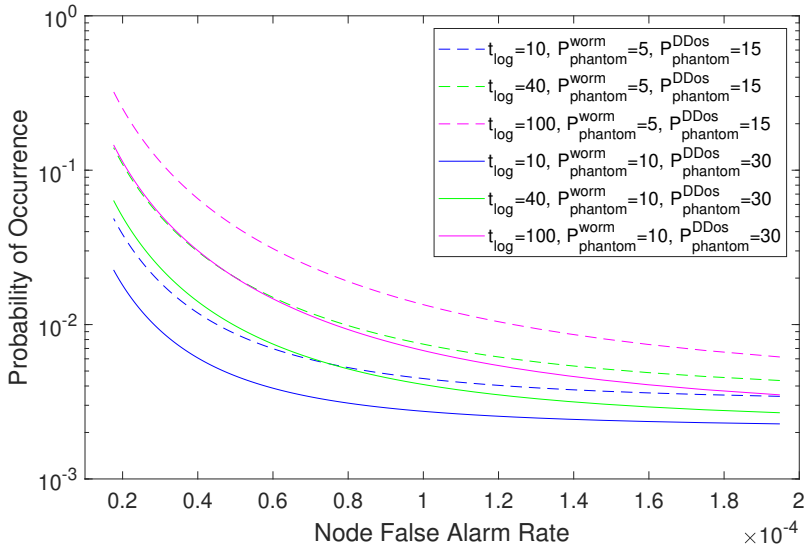


Figure B.3. Probability of a given nodal false alarm rate for various sets of external,input parameters, given $T_{phantom}^{worm} = 10$, $T_{phantom}^{DDoS} = 30$, and the units of L_w and t_{log} are seconds.



(a) $P(D^{node}_{false})$ results for $L_w = 2$



(b) $P(D^{node}_{false})$ results for $L_w = 10$

Figure B.4. Probability of a given nodal false alarm rate for various sets of external, input parameters, given $L_w = 2$ and $L_w = 10$ seconds, and the unit of t_{log} is seconds.

B.2 Eigenvalues of Perturbed Simple Graphs

In this section, additional results for $\Delta\lambda_j$ in simple graphs are provided for various $P(l_{i,j})$. For each simulation, 300 random simple graphs containing 50 nodes are formed using a static $P(l_{i,j})$ between each pair of nodes, one node in each graph is selected and edited to be completely connected (i.e., have links connecting it to every other node in the network), and $\Delta\lambda$ is evaluated as the links of the selected node are disconnected. The results for $P(l_{i,j}) = 0.01$ are shown in Figure B.5 and the results for $P(l_{i,j}) = 0.25$ and $P(l_{i,j}) = 0.9$ are shown in Figure B.6 where the blue asterisks are the data results from the simulation, the green dotted line is $\Delta\lambda_j^{UB}$, and the red dotted line is the equation $\Delta\lambda_j = l_{perturb}$. See Section 5.2.3 for additional information.

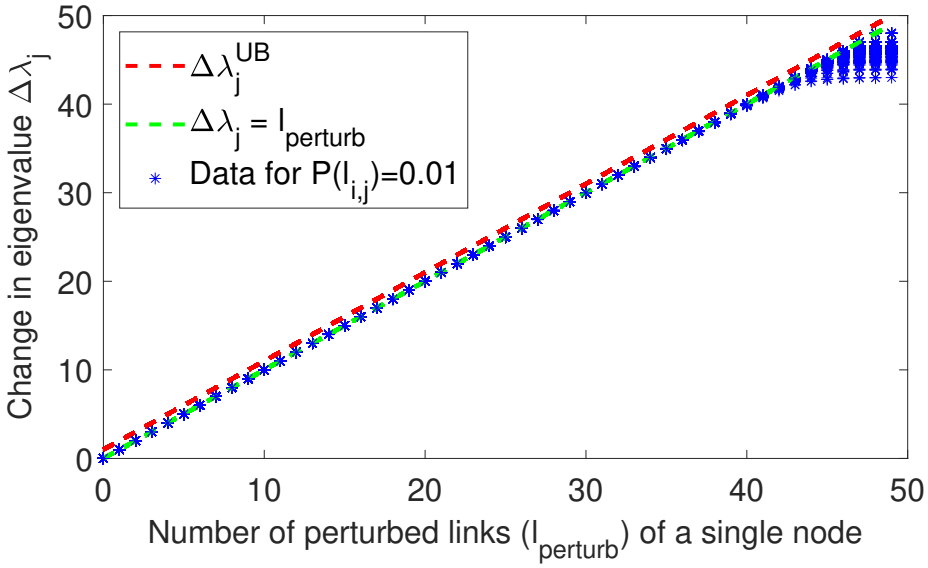
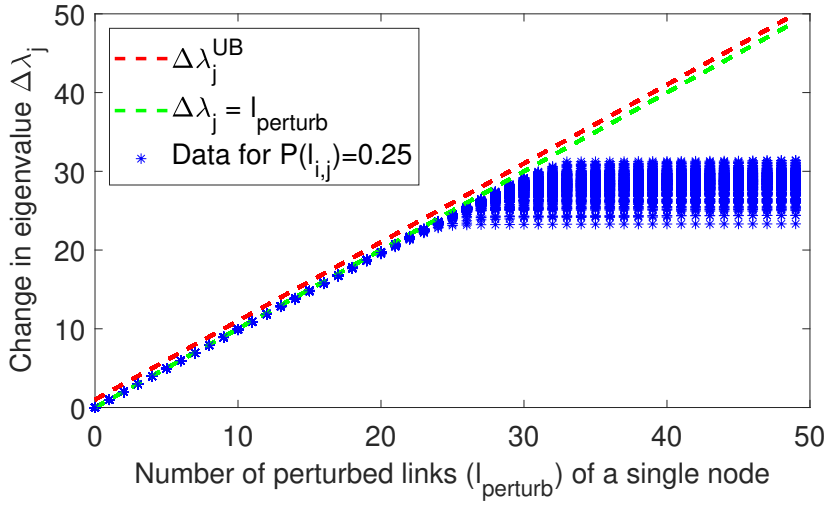
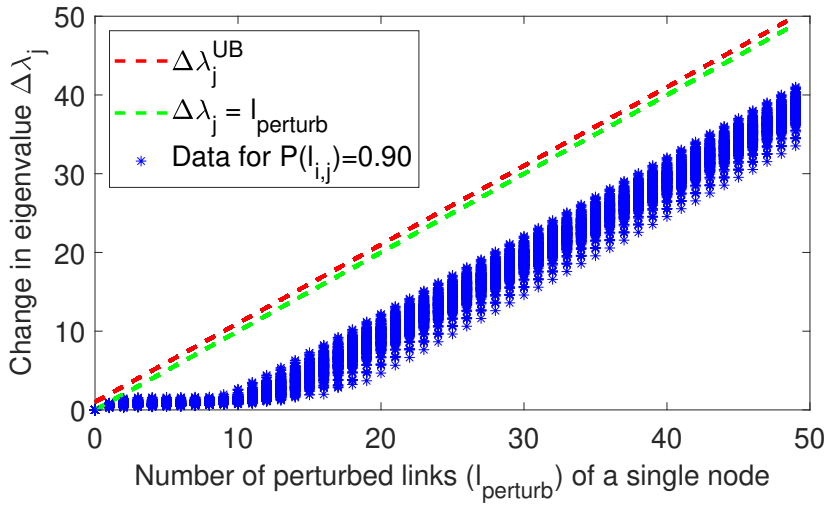


Figure B.5. Comparison of $\Delta\lambda_j^{UB}$ to the $\Delta\lambda_j$ simulation results of 300 random simple graphs containing 50 nodes and $P(l_{i,j}) = 0.1$ with link removals from a single, fully-connected node.



(a) $\Delta\lambda_j$ from graphs with $P(l_{i,j}) = 0.25$



(b) $\Delta\lambda_j$ from graphs with $P(l_{i,j}) = 0.9$

Figure B.6. Comparison of $\Delta\lambda_j^{UB}$ to the $\Delta\lambda_j$ simulation results of 300 random simple graphs containing 50 nodes and $P(l_{i,j}) = \{0.25, 0.9\}$ with link removals from a single, fully-connected node.

B.3 Eigenvectors of Perturbed Simple Graphs

In this section, additional results for $\Delta|v_{i,j}|$ in simple graphs are provided for various $P(l_{i,j})$. For each simulation, 300 random simple graphs containing 50 nodes are formed using a static $P(l_{i,j})$ between each pair of nodes, one node in each graph is selected and edited to be completely connected (i.e., have links connecting it to every other node in the network), and $\Delta|v_{i,j}|$ is evaluated as the links of the selected node are disconnected. The results for $P(l_{i,j}) = 0.01$ are shown in Figure B.7 and the results for $P(l_{i,j}) = 0.25$ and $P(l_{i,j}) = 0.9$ are shown in Figure B.8 where the blue asterisks are the data results from the simulation, and the red dotted line is $\Delta|v_{i,j}| = 1$. See Section 5.3.2 for additional information.

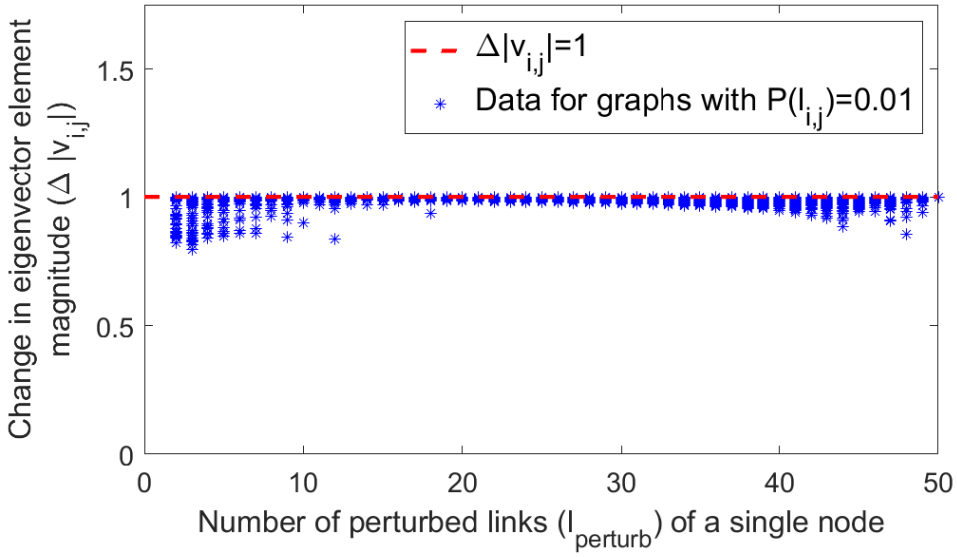
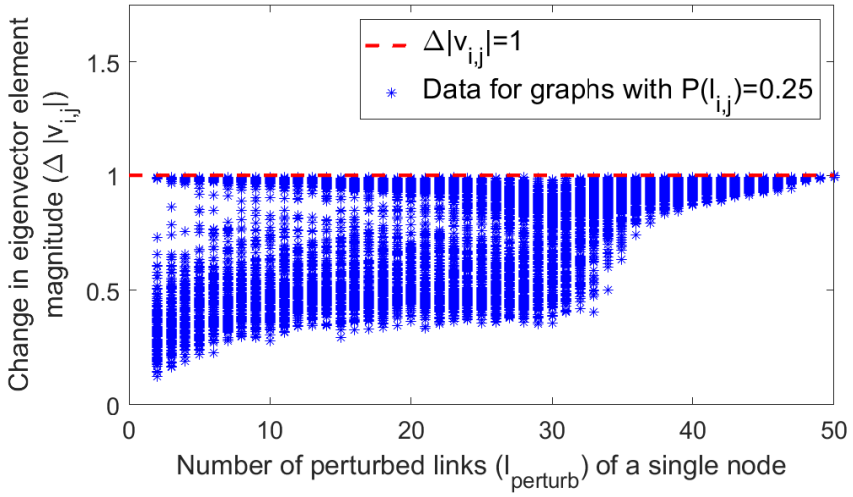
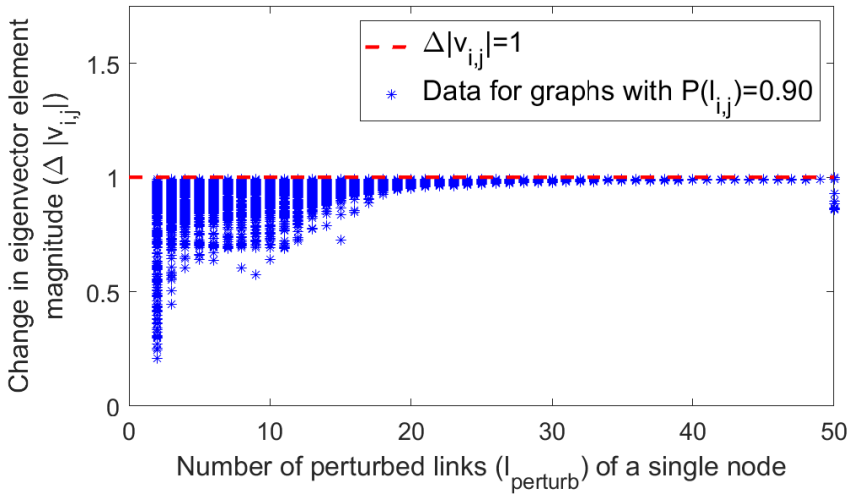


Figure B.7. Comparison of $\Delta|v_{i,j}^{UB}|$ to the $\Delta|v_{i,j}|$ simulation results of 300 random simple graphs containing 50 nodes and $P(l_{i,j}) = 0.01$ with link removals from a single, fully-connected node.



(a) $\Delta|v_{i,j}|$ from graphs with $P(l_{i,j}) = 0.25$



(b) $\Delta|v_{i,j}|$ from graphs with $P(l_{i,j}) = 0.9$

Figure B.8. Comparison of $\Delta|v_{i,j}^{UB}|$ to the $\Delta|v_{i,j}|$ simulation results of 300 random simple graphs containing 50 nodes and $P(l_{i,j}) = \{0.25, 0.9\}$ with link removals from a single, fully-connected node.

B.4 Eigenvector Distances of Perturbed Simple Graphs

In this section, additional results for D_{v_j} in simple graphs are provided for various $P(l_{i,j})$. For each simulation, 300 random simple graphs containing 50 nodes are formed using a static $P(l_{i,j})$ between each pair of nodes, one node in each graph is selected and edited to be completely connected (i.e., have links connecting it to every other node in the network), and the maximum D_{v_j} is evaluated as the links of the selected node are disconnected. The results for $P(l_{i,j}) = 0.01$, $P(l_{i,j}) = 0.25$, and $P(l_{i,j}) = 0.9$ are shown in FigureB.7 where the blue asterisks are the maximum eigenvector distance change results from the simulation, and the red dotted line is $D_{v_j} = 2$, and the green dotted line is $D_{v_j} = \sqrt{2}$. See Section 5.3.3 for additional information.

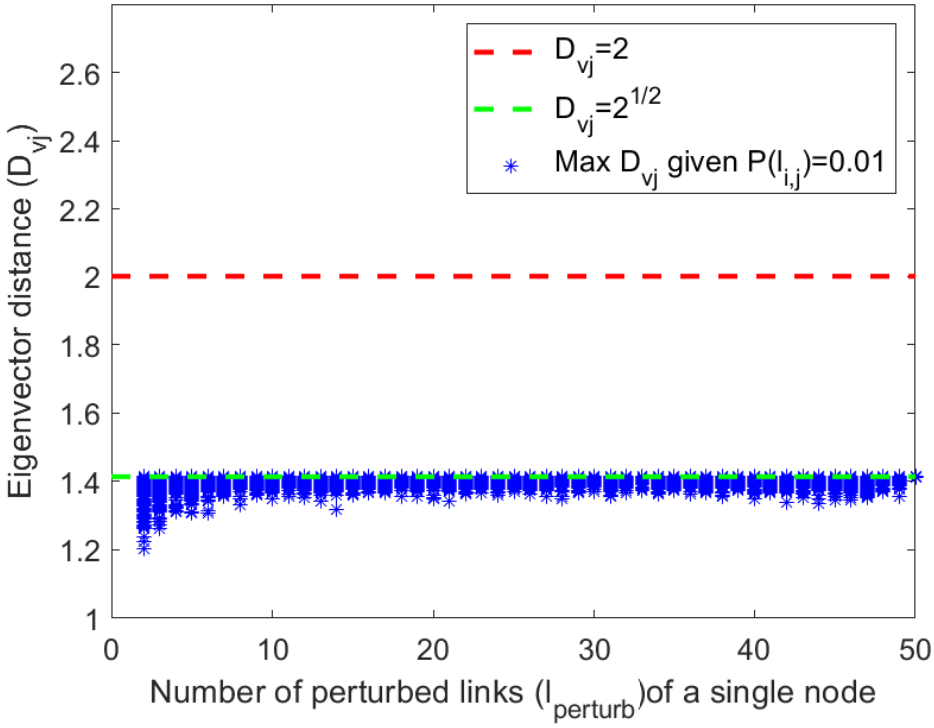
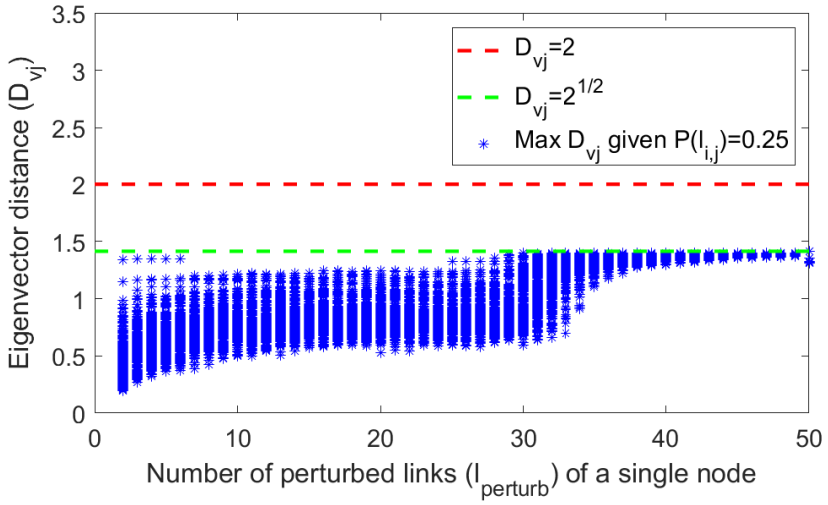
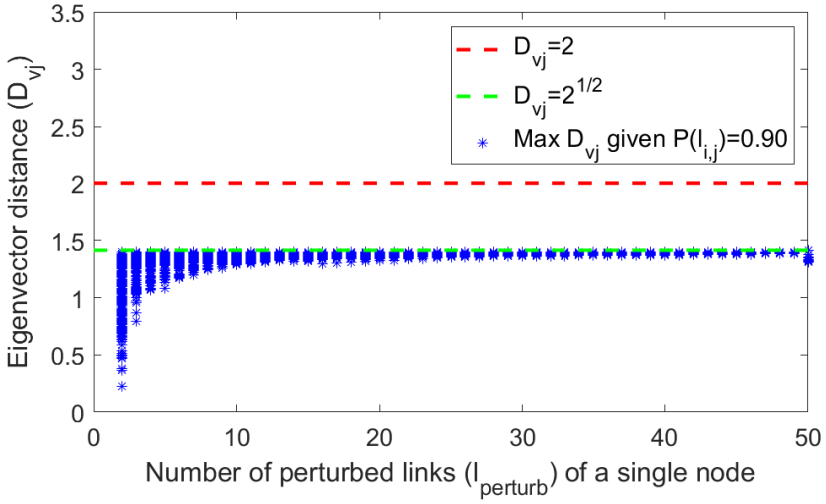


Figure B.9. Comparison of $D_{v_j}^{UB}$ (red dashed line) to the maximum D_{v_j} observed (blue asterisks) in 300 random simple graphs containing 50 nodes and $P(l_{i,j}) = 0.01$ with links removals from a single, fully-connected.



(a) Max D_{v_j} from graphs with $P(l_{i,j}) = 0.25$



(b) Max D_{v_j} from graphs with $P(l_{i,j}) = 0.9$

Figure B.10. Comparison of $D_{v_j}^{UB}$ (red dashed line) to the maximum D_{v_j} observed (blue asterisks) in 300 random simple graphs containing 50 nodes and $P(l_{i,j}) = \{0.25, 0.9\}$ with links removals from a single, fully-connected.

List of References

[1] K. Townsend. (2019, Mar. 21). Global security spend set to grow to $133.8 billion by 2022: IDC. *SecurityWeek*. [Online]. Available: https://www.securityweek.com/global-security-spend-set-grow-1338-billion-2022-idc

[2] NakedSecurity. (2018, Sep. 18). WannaCry – the worm that just won't die. *SOPHOS*. [Online]. Available: https://nakedsecurity.sophos.com/2019/09/18/wannacry-the-worm-that-just-wont-die/

[3] Symantec. (2019, Feb). Internet security threat report. *Symantec*. [Online]. Available: https://docs.broadcom.com/doc/istr-24-2019-en

[4] FireEye. (2019). M-trends 2019 special report. *FireEye Mandiant Services*. [Online]. Available: https://content.fireeye.com/m-trends

[5] SecurityMagazine. (2019, Oct. 22). First three quarters of 2019: 7.2 billion malware attacks, 151.9 million ransomware attacks. *Security Magazine*. [Online]. Available: https://www.securitymagazine.com/articles/91133-first-three-quarters-of-2019-72-billion-malware-attacks-1519-million-ransomware-attacks#:~:text=Million%20Ransomware%20Attacks-,First%20Three%20Quarters%20of%202019%3A%207.2%20Billion,Attacks%2C%20151.9%20Million%20Ransomware%20Attacks&text=In%20the%20first%20three%20quarters,over%2Dyear%20declines%2C%20respectively

[6] J. Davis. (2019, Jul. 24). Phishing attacks on the rise, 25% increase in threats evading security. *Health IT Security*. [Online]. Available: https://healthitsecurity.com/news/phishing-attacks-on-the-rise-25-increase-in-threats-evading-security#:~:text=July%2024%2C%202019%20%2D%20Security%20leaders,a%20new%20report%20from%20GreatHorn.&text=Respondents%20said%20they%20use%20an,than%20two%20email%20security%20solutions

[7] M. Boddy. (2019, Aug. 30). Report: cryptojacking campaigns up by 29%, ransomware attacks up 118%. *COINTELEGRAPH*. [Online]. Available: https://cointelegraph.com/news/report-cryptojacking-campaigns-up-by-29-ransomware-attacks-up-118

[8] MalwarebytesLabs. (2020, Feb). 2020 state of malware report. *MalwareBytes Labs*. [Online]. Available: https://resources.malwarebytes.com/files/2020/02/2020_State-of-Malware-Report.pdf

[9] H. Okhravi, J. Riordan, and K. Carter, "Quantitative evaluation of dynamic platform techniques as a defensive mechanism," in *Research in Attacks, Intrusions, and Defenses*, A. Stavrou, H. Bos, and G. Portokalidis, Eds. Cham: Springer International Publishing, 2014, pp. 405–425.

[10] J. M. Benyus, *Biomimicry: Innovation Inspired by Nature*. New York, NY: Perennial, 2002.

[11] J. Perno, "Virus: a comparative analysis between nature and machine," Master's thesis, Utica College, Utica, NY, 2016.

[12] K. Knapp, F. Morris, R. K. R. Jr., and T. A. Byrd, "Defense mechanism of biological cells: a framework for network security thinking," *Communications of the Association for Information Systems*, vol. 12, no. 47, pp. 701–719, 2003.

[13] J. Raiyn, "A survey of cyber attack detection strategies," *International Journal of Security and Its Applications*, vol. 8, no. 1, pp. 247–255, 2014.

[14] J. O. Kephart, "A biologically inspired immune system for computers," in *Artificial Life IV: Proc. of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*. Cambridge, MA: MIT Press, 1994, pp. 130–139.

[15] P. Biancaniello, G. Holness, J. Darvill, M. Craven, and P. Lardieri, "AIR: a framework for adaptive immune response for cyber defense," Lockheed Martin Advanced Technology Laboratories, 3 Executive Campus, Cherry Hill, NJ 08002, Tech. Rep., Dec. 2011.

[16] J. Jang-Jaccard and S. Nepal, "A survey of emerging threats in cybersecurity," *Journal of Computer and System Sciences*, vol. 80, no. 5, pp. 973–993, 2014.

[17] Y. Zhang, L. Wang, W. Sun, R. C. G. II, and M. Alam, "Artificial immune system based intrusion detection in a distributed hierarchical network architecture of smart grid," *2011 IEEE Power and Energy Society General Meeting*, pp. 1–8, 2011.

[18] E. K. Lua and R. Chen, "A holistic immune system against active P2P worms," in *The International Conference on Information Networking 2013 (ICOIN)*, 2013, pp. 24–29.

[19] D. Musliner, J. M. Rye, D. Thomsen, and D. Mcdonald, "FUZZBUSTER: a system for self-adaptive immunity from cyber threats," in *Proc. of 8th International Conference on Autonomic and Autonomous Systems (ICAS-12)*, 2012, pp. 118–123.

[20] S. A. Hofmeyr and S. Forrest, "Architecture for an artificial immune system," *Evolutionary Computation*, vol. 8, no. 4, pp. 443–473, 2000.

[21] D. Q. Le, T. Jeong, H. E. Roman, and J. W. Hong, "Traffic dispersion graph based anomaly detection," in *Proc. of the Second Symposium on Information and Communication Technology*, 2011, pp. 36–41.

[22] D. R. Ellis, J. G. Aiken, A. M. McLeod, and D. R. Keppler, "Graph-based worm detection on operational enterprise networks," MITRE, Tech. Rep. MTR-06W0000035, Apr. 2006.

[23] J. Johnson, "Software defined network monitoring scheme using spectral graph theory and phantom nodes," M.S. thesis, Dept. of Electrical and Computer Engineering, Naval Postgraduate School, Monterey, CA, 2014.

[24] J. Wu, S. Vangala, L. Gao, and K. Kwiat, "An effective architecture and algorithm for detecting worms with various scan," in *Proc. of the Network and Distributed System Security Symposium*, 2004.

[25] G. Gu, M. Sharif, X. Qin, D. Dagon, W. Lee, and G. Riley, "Worm detection, early warning and response based on local victim information," in *20th Annual Computer Security Applications Conference*, 2004, pp. 136–145.

[26] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," in *Computer Communication Review*, 10 2005, vol. 35, pp. 217–228.

[27] D. Nicol, "The impact of stochastic variance on worm propagation and detection," in *WORM '06: Proc. of the 4th ACM Workshop on Recurring Malcode*, New York, NY, 2006, pp. 57–64.

[28] W. Yu, X. Wang, P. Calyam, D. Xuan, and W. Zhao, "On detecting camouflaging worm," in *2006 22nd Annual Computer Security Applications Conference (ACSAC'06)*, 2006, pp. 235–244.

[29] X.-F. Chen and S.-Z. Yu, "CIPA: A collaborative intrusion prevention architecture for programmable network and sdn," *Computers Security*, vol. 58, pp. 1–19, May 2016.

[30] L. Ertoz, E. Eilertson, A. Lazarevic, P.-n. Tan, V. Kumar, and J. Srivastava, "MINDS – minnesota intrusion detection system," in *NextGeneration Data Mining*, H. Kargupta, J. Han, P. S. Yu, R. Motwani, and V. Kumar, Eds. Cambridge, MA: MIT Press, 2004, ch. 3, pp. 199–218.

[31] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. W. Chang, "A novel anomaly detection scheme based on principal component classifier," in *Proc. of ICDM Foundation and New Direction of Data Mining Workshop*, 2003, pp. 172–179.

[32] Y. Li, B.-X. Fang, L. Guo, and Y. Chen, "TCM-KNN algorithm for supervised network intrusion detection," in *Proc. of the 2007 Pacific Asia Conference on Intelligence and Security Informatics*, 2007.

[33] M. Panda, A. Abrahamb, and M. R. Patrac, "A hybrid intelligent approach for network intrusion detection," in *Procedia Engineering*, 2012, pp. 1–9.

[34] G. V. Nadiammai and M. Hemalatha, "Effective approach toward intrusion detection system using data mining techniques," *Egyptian Informatics Journal*, vol. 15, no. 1, pp. 37–50, 2014.

[35] K. Wang and S. J. Stolfo, "Anomalous payload-based network intrusion detection," in *Recent Advances in Intrusion Detection. RAID 2004. Lecture Notes in Computer Science*, E. Johnsson, A. Valdes, and M. Almgren, Eds. Berlin, Germany: Springer, 2004, vol. 3224.

[36] W. Willinger, V. Pason, and M. S. Taqqu, "Self-similarity and heavy tails: structural modeling of network traffic," *A Practical Guide to Heavy Tails: Statistical Techniques and Applications*, vol. 23, pp. 27–53, 1998.

[37] A. Karasaridis and D. Hatzinakos, "Network heavy traffic modeling using α-stable self-similar processes," *IEEE Trans. Communications*, vol. 49, no. 7, pp. 1203–1214, 2001.

[38] A. Sherrer, N. Larrieu, P. Owezarski, P. Borgnant, and P. Abry, "Nongaussian and long memory statistical characterizations for internet traffic with anomalies," *IEEE Trans. on Dependable and Secure Computing*, vol. 4, no. 1, pp. 56–70, 2007.

[39] C. Bollmann, M. Tummala, and J. McEachen, "Representation of positive alpha-stable network traffic through levy mixtures," in *2017 51st Asilomar Conference on Signals, Systems, and Computers*, 2017, pp. 1460–1464.

[40] F. Simmross-Wattenberg, A. Tristan-Vega, P. Casaseca-de-la Higuera, J. I. Asensio-Perez, M. Martin-Fernandez, Y. A. Dimitriadis, and C. Alberola-Lopez, "Modelling network traffic as α-stable stochastic processes: an approach towards anomaly detection," in *Proc. VII Jornadas de Ingenieria Telematica (JITEL)*, 2008, pp. 25–32.

[41] Simmross-Wattenberg, J. I. Asensio-Perez, P. Casaseca-de-la Higuera, M. Martin-Fernandez, Y. A. Dimitriadis, and C. Alberola-Lopez, "Anomaly detection in network traffic based on statistical inference and alpha-stable modeling," in *IEEE Trans. on Dependable and Secure Computing*, no. 4, 2011, vol. 8, pp. 494–504.

[42] E. E. Kuruoglu, "Analytical representation for positive/spl alpha/-stable densities," in *Acoustics, Speech, and Signal Processing, 2003. Proc.. (ICASSP'03). 2003 IEEE International Conference on*, 2003, vol. 6, pp. VI–729.

[43] S. T. Rachev and S. Mittnik, *Stable Paretian Models in Finance*, Jun. 2000, vol. 7.

[44] C. Menn and S. Rachev, "Smoothly truncated stable distributions, garch-models, and option pricing," *Mathematical Methods of Operations Research*, vol. 69, pp. 411–438, Jul. 2009.

[45] A. Chernobai, K. Burnecki, S. Rachev, S. Tr uck, and R. Weron, "Modelling catastrophe claims with left-truncated severity distributions," *Computational Statistics*, vol. 21, pp. 537–555, 2006.

[46] S. Hosseinzadeh, M. Amirmazlaghani, and M. Shajari, "An aggregated statistical approach for network flood detection using gamma-normal mixture modeling," *Computer Communications*, vol. 152, pp. 137–148, 2020.

[47] C. Davis and W. Kahan, "The rotation of eigenvectors by perturbation. III," *SIAM Journal on Numerical Analysis*, vol. 7, no. 1, pp. 1–46, 1970.

[48] J. Eldridge, M. Belkin, and Y. Wang, "Unperturbed: spectral analysis beyond Davis-Kahan," in *Proc. of Machine Learning Research*, F. Janoos, M. Mohri, and K. Sridharan, Eds. PMLR, 07–09 Apr 2018, vol. 83, pp. 321–358. Available: http://proceedings.mlr.press/v83/eldridge18a.html

[49] Y. Chen. (2018). Spectral methods. Princeton University, ELE 538B: Mathematics of High-Dimensional Data [Online]. Available: https://www.princeton.edu/~yc5/ele538_math_data/lectures/spectral_method.pdf

[50] Y. Yu, T. Wang, and R. J. Samworth, "A useful variant of the Davis–Kahan theorem for statisticians," *Biometrika*, vol. 102, no. 2, pp. 315–323, 2015.

[51] J. Moro and F. Dopico, "First order eigenvalue perturbation theory and the newton diagram," in *Applied Mathematics and Scientific Computing*, Z. D. c, V. Hari, L. Sopta, Z. Tutek, and K. V. c, Eds. Boston, MA: Springer, Jan. 2002.

[52] S. C. Eisenstat and I. C. F. Ipsen, "Relative perturbation bounds for eigenspaces and singular vector subspaces," in *Proc. of the Fifth SIAM Conference on Applied Linear Algebra*, J. G. Lewis, Ed., 1994, pp. 62–66.

[53] R.-C. Li, "On perturbations of matrix pencils with real spectra," *Mathematics of Computation*, vol. 62, pp. 231–265, 1994.

[54] D. B. West, *Introduction to Graph Theory*, 2nd ed. Nioda, India: Pearson, 2015.

[55] P. V. Mieghem, *Graph spectra for complex networks*. Cambridge, NY: Cambridge University Press, 2011.

[56] V. K. Balakrishnan, *Schaum's Outlines of Theory and Problems of Graph Theory*. New York: McGraw Hill, 1997.

[57] H. N. Djidjev, G. Sandine, C. Storlie, and S. V. Wiel, "Graph based statistical analysis of network traffic," in *MLG'11*, 2011.

[58] P. V. Mieghem. (n.d.). *Graph Eigenvectors, Fundamental Weights and Centrality Metrics for nodes in networks*. arXiv. [Online]. Available: http://arxiv.org/abs/1401.4580

[59] G. Strang, *Linear Algebra and Its Applications*. Belmont, CA: Brooks/Cole, Cengage Learning, 2006.

[60] T. Parker, "Spectral graph theory analysis of software-defined networks to improve performance and security," Ph.D. dissertation, Naval Postgraduate School, Monterey, CA, 2015.

[61] Y. Wang, S. Wen, Y. Xiang, and W. Zhou, "Modeling the propagation of worms in networks: a survey," *IEEE Communications Surveys Tutorials*, vol. 16, no. 2, pp. 942–960, 2013.

[62] G. E. Riley, M. L. Sharif, and Wenke Lee, "Simulating internet worms," in *The IEEE Computer Society's 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems, 2004. (MASCOTS 2004). Proceedings.*, 2004, pp. 268–274.

[63] S. Staniford, V. Paxson, and N. Weaver, "How to 0wn the internet in your spare time," *USENIX Security Symposium*, pp. 149–167, 2002.

[64] D. Moore, C. Shannon, and J. Brown, "Code-Red: a case study on the spread and victims of internet worm," in *2nd ACM SIGCOMM Workshop on Internet Measurement*, 2002.

[65] Z. Tan, A. Jamdagni, X. He, P. Nanda, and R. P. Lui, "A system for denial of service attack detection based on multivariate correlation analysis," *IEEE Transactions on Parallell and Distributed Systems*, vol. 25, no. 2, pp. 447–456, 2014.

[66] G. Oke, G. Loukas, and E. Gelenbe, "Detecting denial of service attacks with bayesian classifiers and random neural network," *2007 IEEE International Fuzzy Systems Conference*, 2007.

[67] Imperva. (2019). Distributed denial of service attack (DDoS) definition. *Imperva*. [Online]. Available: https://www.imperva.com/learn/application-security/ddos-attacks/

[68] C. Shalizi. Mixture Models, Latent Variables and the EM Algorithm. *CMU, Statistics 36-350, Fall 2009 Lecture Notes*. [Online].

[69] S. Borak, H. Wolfgang, and R. Weron, "Stable distributions," Berlin, Germany, Center for Applied Statistics and Economics, Humboldt-Universit at zu Berlin SFB 649 Discussion Paper 2005-009, 2005.

[70] A. Fiche, J.-C. Cexus, A. Martin, and A. Khenchaf, "Features modeling with an α-stable distribution: application to pattern recognition based on continuous belief functions," *Information Fusion*, vol. 14, no. 4, pp. 504–520, 2013.

[71] G. Samoradnitsky and M. S. Taqqu, *Stable Non-Gaussian Random Processes: Stochastic Models with Infinite Variance*. CRC press, 1994, vol. 1.

[72] J. P. Nolan, *Stable Distributions - Models for Heavy Tailed Data*. Boston, MA: Birkhauser, 2018, in progress, Chapter 1 online at http://fs2.american.edu/jpnolan/www/stable/stable.html.

[73] C. D. Hardin, "Skewed stable variables and processes," University of North Carolina, Chapel Hill, North Carolina, USA, Center for Stochastic Processes Technical Report no. 79, 1984.

[74] M. Rouse. (2014). Noise. *TechTarget*. [Online]. Available: https://whatis.techtarget.com/definition/noise#:~:text=Noise%20is%20unwanted%20electrical%20or, images%2C%20audio%2C%20and%20telemetry.

[75] T. Tao. (2010, Jan.). 254A, notes 3a: eigenvalues and sums of Hermitian matrices. *Wordpress*. [Online]. Available: https://terrytao.wordpress.com/2010/01/12/254a-notes-3a-eigenvalues-and-sums-of-hermitian-matrices/

[76] A. M. del Rey, "Mathematical modeling of the propagation of malware: a review," *Security and Communication Networks*, vol. 8, pp. 2561–2579, 2015.

[77] P. Lee, "Passivity framework for modeling, composing and mitigating cyber attacks," Ph.D. dissertation, Univ. of Washington, 2016. Available: https://digital.lib.washington.edu/researchworks/handle/1773/37102

[78] A. Prajapati and B. K. Mishra, "Cyber attack and control techniques," *Information Systems Design and Intelligent Applications*, vol. 339, pp. 157–166, 2015.

[79] M. Roberts and J. Heesterbeek, "Mathematical models in epidemiology," *Mathematical Models*, 2003.

[80] M. Kermark and G. McKendrick, "Contributions to the mathematical theory of epidemics," in *Proc. of Royal Society London Series A*, no. 5, 1927, vol. 115.

[81] IDM. (2020, Aug.). SIR and SIRS models. *Bill Melinda Gates Foundation*. [Online]. Available: https://idmod.org/docs/emod/generic/model-sir.html

[82] B. K. Mishra and A. Prajapati, "Cyber warfare: worm's transmission model," *International Journal of Advanced Science and Technology*, vol. 63, pp. 83–94, 2014.

[83] L. Feng, X. Liao, Q. Han, and H. Li, "Dynamical analysis and control strategies on malware propagation model," *Applied Mathematical Modelling*, vol. 16-17, pp. 8225–8236, 2013.

[84] D. R. Ellis, J. G. Aiken, K. S. Attwood, and S. D. Tenaglia, "A behavioral approach to worm detection," in *Proc. of the 2004 ACM Workshop on Rapid Malcode*, 2004, pp. 43–53.

[85] K. C. Das, "The Laplacian spectrum of a graph," *Computers and Mathematics with Applications*, vol. 48, pp. 715–724, 2004.

[86] J. Safar, M. Tummala, and J. McEachen, "Spectral graph-based cyber worm detection using phantom components and strong node concept," *54th Hawaii International Conference on System Sciences*, 2021.

[87] A. Jamakovic and S. Uhlig, "On the relationship between the algebraic connectivity and graph's robustness to node and link failures," in *Proc. of 3rd IEEE EURO-NGI Conference on Next Generation Internet Networks*, 2007.

[88] H. Wang and P. V. Mieghem, "Algebraic connectivity optimization via link addition," in *Proc. of IEEE/ACM Bionetics*, 2008.

[89] J. G. Kemeny and J. L. Snell, *Finite Markov Chains*. New York: Van Nostrand, 1959.

[90] S. Abrahamand 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.

[91] N. Kalanjee. (2016, July). 54 days to recovery: the impact of cyber crime. *Hewlett Packard*. [Online]. Available: https://blog.ext.hp.com/t5/BusinessBlog-en/54-daysto-recovery-The-impact-of-cyber-crime/ba-p/6720

[92] J. Safar, C. Bollmann, M. Tummala, and J. McEachen, "A novel Lévy-impulse mixture based connection model for computer network traffic," *14th International Conference on Signal Processing and Communication Systems*, 2020.

[93] F. d. Castro. (2020, July). Fitmethis version 1.5.2. *MathWorks File Exchange*. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/40167-fitmethis

[94] "Matlab version 9.7.0.1261785 (r2019b) update 3," Natick, Massachusetts: The Mathworks Inc., 2019.

[95] MathWorks. (2020). Polyfit. *MathWorks*. [Online].

[96] MAWILab. MAWI working group traffic archive. *MAWI Working Group of the WIDE Project*. [Online]. Available: http://mawi.wide.ad.jp/mawi/

[97] DARPA scalable network monitoring (SNM) program traffic, IMPACT ID: USC-LANDER/DARPA_2009_DDoS_attack-20091105/rev4383. *USC/LANDER Project*. [Online]. Available: http://www.isi.edu/ant/lander

[98] C. Bollmann, "Network anomaly detection with stable distributions," Ph.D. dissertation, Dept. of Electrical and Computer Engineering, Naval Postgraduate School, Monterey, CA, 2018.

[99] LANDER:DARPA 2009 DDoS attack-20091105. *PREDICT wiki*. [Online]. Available: https://wiki.isi.edu/predict/index.php?title=LANDER:DARPA_2009_DDoS_attack-20091105

[100] C. W. Therrien and M. Tummala, *Probability and Random Processes for Electrical and Computer Engineers*, 2nd ed. Boca Raton: CRC Press, 2012.

[101] D. A. Spielman. (2009, Sep.). Spectral graph theory lesson 02: the Laplacian. *Yale University*. [Online]. Available: https://www.cs.yale.edu/homes/spielman/561/2009/lect02-09.pdf

[102] B. Halabisky. Classification of neuronal data: 'n'-dimensional measurements. *Stanford University*. [Online]. Available: https://hlab.stanford.edu/brian/euclidean_distance_in.html

[103] StratosphereLab. (2017, July). Index of /publicDatasets/CTU-Malware-Capture-Botnet-284-1. *Virus Total*. [Online]. Available: https://mcfp.felk.cvut.cz/publicDatasets/CTU-Malware-Capture-Botnet-284-1/

[104] StratosphereLab. (2017, July). Index of /publicDatasets/CTU-Malware -Capture-Botnet-285-1. *Virus Total*. [Online]. Available: https://mcfp.felk.cvut.cz/publicDatasets/CTU-Malware-Capture-Botnet-285-1/

[105] StratosphereLab. (2017, July). Index of /publicDatasets/CTU-Malware -Capture-Botnet-253-1. *Virus Total*. [Online]. Available: https://mcfp.felk.cvut.cz/publicDatasets/CTU-Malware-Capture-Botnet-253-1/

[106] J. V. Hoogstraten. (2003). Malware FAQ: what is W32/Blaster worm? *SANS*. [Online]. [Online]. Available: https://www.sans.org/security-resources/malwarefaq/w32-blasterworm

[107] Milliways. Comparing TCP and UDP speed and packet loss over LAN and WAN. *Milliways*. [Online]. [Online]. Available: http://milliways.bcit.ca/res/report.pdf

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California