Chapter 5
Characterization of Evolving Networks for Cybersecurity

Josephine M. Namayanja and Vandana P. Janeja

5.1 Introduction

Computer networks are vulnerable to varying cyber attacks that alter the structure and activity of the network. Hence, in order to define and understand the vulnerabilities associated to the network, one must have an understanding of the overall structure and nature of communication patterns within the network as well as the potential points of vulnerability. Network analytics provides the basis for how network structures are modeled, measured, and compared such that a network is modeled as a graph, which describes a collection of nodes or vertices and the communications between them, indicated by edges.

This chapter discusses approaches to change detection where the objective is studying how the network evolves over time and how these changes can be attributed to potential cyber attacks. Techniques such as change detection play a role in network characterization mainly because they detect shifts in network behavior over time. Changes in network behavior can be defined as sudden downtime of key points, for example, servers on the network during peak hours, existence of new or unidentified connections to the network, and specific time periods associated with shifts in network behavior. Such shifts in network behavior may come as a result of a cyber threat. This chapter discusses graph theory concepts to model network behavior and then utilizing analytics to understand the dynamics

J.M. Namayanja (✉)
University of Massachusetts Boston, Boston, MA, USA
e-mail: josephine.namayanja@umb.edu

V.P. Janeja
University of Maryland, Baltimore County, Baltimore, MD, USA
e-mail: vjaneja@umbc.edu

© Springer International Publishing Switzerland 2017

I.M. Alsmadi et al. (eds.), *Information Fusion for Cyber-Security Analytics*, Studies in Computational Intelligence 691, DOI 10.1007/978-3-319-44257-0_5

of the network. Cyber attacks are becoming increasingly sophisticated. One of the key challenges is knowing whether there is even an attack on the network in the first place. Let us consider the following scenario:

5.1.1 Cyber Attacks Are Unrelenting

Large computer networks comprised of tens of thousands of machines generate terabytes of network traffic each day. This traffic typically consists of hundreds of millions of connection records and poses a big data problem. Such significant volume and diversity traffic presents a daunting challenge in the detection of cyber attacks, particularly when it comes to small amounts of malicious activity. Additionally, attacks are increasingly becoming sophisticated and are designed to be undetectable. The behavior of such cyber attacks is extremely dynamic and thus changes over time. Furthermore, the continuous evolution of network structures such as the Web creates complexity in the efficient analysis of computing environments.

In an effort to establish a state of continuous awareness of network behavior, the Supercomputing Enabled Transformational Analytics Capability (SETAC) project at Lawrence Livermore National Laboratory aims to increase the ability to detect, characterize, and combat malicious attacks on large computer networks [1].

Several major incidents of cyber attacks have reported delayed detection of attacks. This delay can take from months to even years before the threat on the network is discovered. In 2014, it took organizations a median of 205 days to detect attackers in their network environments [2]. Such delays in attack detection can be due to the complexity of networks both in scale and dynamism which makes it difficult to keep track of what is taking place. From a graph perspective, networks are comprised of multiple dimensions, which include, nodes, edges, and time, where such dimensionality poses a challenge in identifying a vulnerability, detecting an attack, and potentially preventing an attack. Certain attacks are usually targeted to specific points in the network and are used in conjunction with advanced persistent threats. Such targeted attacks are designed to exploit and cause harm on the network.

The process of characterizing networks through change detection can be potentially useful to understand and control the dynamics of the network [3]. This chapter discusses state-of-the-art techniques in change detection that may be geared toward modeling network behavior and detecting patterns, which can indicate potential cyber threats such as the onset of a massive cyber attack which changes the way a network appears.

The rest of the chapter is organized as follows: Sect. 5.2 presents a detailed background on concepts in graph theory. Section 5.3 discusses fundamental concepts in network evolution. Section 5.4 presents an extensive overview on the fundamentals of change detection in temporally evolving networks. Section 5.5 discusses key applications for change detection. Lastly, Sect. 5.6 presents conclusions and future work.

5.2 Graph Theory Concepts

Each network presents specific topological features that characterize a network and its connectivity. Several different network measures can be calculated from a given graph. Network measures can be calculated for the entire graph or for each individual node. Node-level measures in the form of node centrality enable modeling the network to determining the role of a node in a network which can be useful in threat detection [4]. According to [5], an assessment of network vulnerabilities indicates that an attacker is likely to exploit the weak points such as critical nodes whose corruption greatly affects network performance. Additionally, graph-level measures such as density and diameter provide an overall picture of the impact on threats on individual nodes to the entire network. Let us consider the fundamental concepts in graph theory as they are utilized in network analytics for cybersecurity.

5.2.1 Graph

A graph is made up of nodes or vertices and edges that connect them. It is defined as:

A graph $G = (n, e)$*, where* $n = \{n_1 \ldots n_v\}$ *is a set of nodes and* $e = \{e_1 \ldots e_w\}$ *is a set of edges, such that* (n_i, n_j) *is an edge between nodes* n_i *and* n_j*.*

A graph can be directed or undirected. A directed graph G identifies the direction of the edge between the source and destination nodes, respectively. For example, $n_i \rightarrow n_j$ indicates n_i as a source node and n_j as the destination node as shown in Fig. 5.1a. On the other hand, an undirected graph G does not identify the direction of the edge between the nodes as shown in Fig. 5.1b.

This chapter discusses concepts that are applicable to both directed and undirected networks.

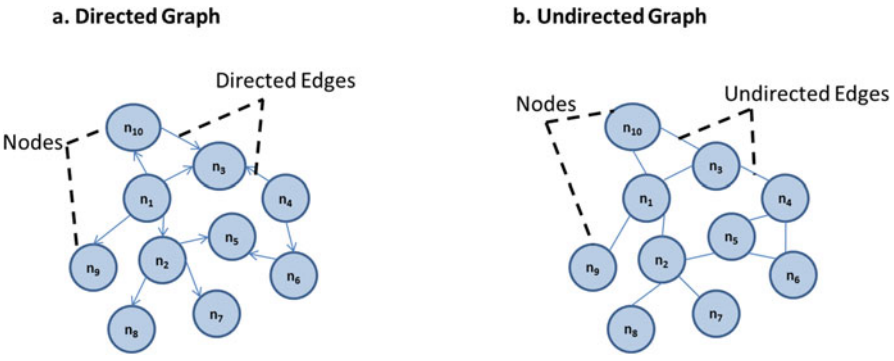

Fig. 5.1 Directed versus undirected graph

5.2.2 *Node Centrality*

The centrality of a node in a network determines a node's individual connectivity on the network. Here, we discuss selected centrality measures, namely, degree centrality [6] which is relative to the node, betweenness centrality [6], PageRank centrality [7], and eigenvector centrality [8, 9], which are individual node based but still relative to the rest of the network. Other measures include closeness centrality and Katz centrality to mention a few. These measures are applicable to both directed and undirected networks.

5.2.2.1 Degree Centrality

The degree of a node n_i is the number of edges incident on it. The degree centrality [6] is the most basic of all measures, and it counts how many times a node is involved in an interaction. It is defined, for a node n_i, as the number of edges that are incident on it.

*Given x number of nodes in the network, the connectivity $a_{ij} = 1$ if nodes i and j are connected by an edge and $a_{ij} = 0$ otherwise. Hence, the degree d_i of node n_i is the sum of all a_{ij}. The connectivity between nodes is represented through a v*v adjacency matrix A, where v is the number of nodes.*

If a node n_i is connected to a node n_j, then there exists an edge (n_i, n_j) between nodes n_i and n_j. We provide an example of an adjacency matrix in Fig. 5.2.

In Fig. 5.2, we see that if two nodes are adjacent or connected, then the row and column intersection is 1, else 0. For example, nodes n_1 and n_2 are connected and

	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8	n_9	n_{10}
n_1	0	1	1	0	0	0	0	0	1	1
n_2	1	0	0	0	1	0	1	1	0	0
n_3	1	0	0	1	0	0	0	0	0	1
n_4	0	0	1	0	1	1	0	0	0	0
n_5	0	1	0	1	0	1	0	0	0	0
n_6	0	0	0	1	1	0	0	0	0	0
n_7	0	1	0	0	0	0	0	0	0	0
n_8	0	1	0	0	0	0	0	0	0	0
n_9	1	0	0	0	0	0	0	0	0	0
n_{10}	1	0	1	0	0	0	0	0	0	0

Fig. 5.2 An adjacency matrix representing an undirected computer network

nodes n_2 and n_3. Using this adjacency matrix, one can determine the degree centrality of nodes in a network. For example, the degree centrality of node n_1 is 4 based on the sum of connectivity a_{ij}, for n_2 the degree centrality is 4, for n_3 is 3, and so on.

5.2.2.2 Betweenness Centrality

Betweenness centrality [6] is a measure of how often a node lies along the shortest path or geodesic path between the two other nodes for all nodes in a graph.

Given x nodes, g_{jk} is the number of geodesic paths between nodes n_j and n_k; the betweenness of node n_i is defined as $g_{jk(i)}$ which is the number of geodesic paths that pass through n_i among g_{jk}.

5.2.2.3 Eigenvector Centrality

The eigenvector centrality [8, 9] can be understood as a refined version of the degree centrality in the sense that it recursively takes into account how neighboring nodes are connected.

Given λ as the largest eigenvalue, the eigenvector centrality e_i for a node n_i is the ith component of the eigenvector associated with the largest eigenvalue λ of the network and is proportional to the sum of the eigenvector centrality of the nodes it is connected to. λ assures the centrality is nonnegative.

While the eigenvector centrality of a network can be calculated via the standard method using the adjacency matrix representation of the network, it can be also computed by an iterative degree calculation [10].

5.2.2.4 PageRank Centrality

PageRank [7] is used to measure the relative importance of nodes on the network by computing a ranking for every node based on the connectivity on the network. Let A be a square matrix with the rows and column corresponding to nodes. Let $A_{u,v} = 1/N_u$ if there is an edge from u to v and $A_{u,v} = 0$ if not. If we treat R as a vector over nodes, then we have $R = cAR$. So R is an eigenvector of A with eigenvalue c. In fact, we want the dominant eigenvector of A. It may be computed by repeatedly applying A to any nondegenerate start vector.

Overall, metrics for node centrality are considered individual or local network measures. However, these can also be translated into graph-level measures by averaging them out over the count of nodes in the graph [11–13].

5.2.3 Graph-Level Measures

Graph-level measures account for connections in the entire network and not just individual nodes in a network.

5.2.3.1 Density

A network is called dense if its number of edges is roughly quadratic to its number of nodes.

Density of the network is defined as the proportion of the actual number of edges to the potential number of edges.

Network structures with high density are well connected internally. This may work well for information sharing; however, as the size of the network increases, a high-density measure may be undesirable because the corresponding high number of links for each node could lead to information overload. According to [14, 15], networks densify over time. This means that the number of edges is increasing superlinearly with the number of nodes. This superlinear increase in the number of edges can be measured through an increase in the average degree of nodes in a network over time. Therefore, as the average degree increases over time, then a network is said to obey the densification power law. Densification power law is defined as a relation $e(t) \propto n(t)^{a}$ where $e(t)$ is number of edges at time t and $n(t)$ is the number of nodes at time t, while $\boldsymbol{a}$ is the densification exponent [14, 15]. When $\boldsymbol{a} = 1$, then the average degree of nodes is constant over time, whereas if $\boldsymbol{a} = 2$, then average degree is increasing over time; hence, the network is becoming denser with time [14, 15].

5.2.3.2 Diameter

The diameter of a graph G is the shortest maximum distance between any two nodes in G. In order to find the diameter of a computer network, we first determine all possible paths p in G where $p = \{p_1 \ldots p_n\}$. A path $p_i = (n^{pi}, e^{pi})$, where $n^{pi} = \{n_0, n_1, \ldots, n_k\}$ and $e^{pi} = \{n_0n_1, n_1n_2, \ldots, n_{k-1}n_k\}$ such that nodes n_o to n_k are linked by p_i, and the number of edges in p_i or $|p_i|$ is the length of p_i. Thus, p_i is a simple graph whose nodes can be arranged in a linear sequence in such a way that two nodes are adjacent if they are consecutive in the sequence and nonadjacent if otherwise. We show an example of a path between nodes in Fig. 5.3.

In Fig. 5.3, we show a path p_i from nodes n_6 to n_9 in a computer network represented as p_i (n_6, n_9). The length of p_i (n_6, n_9) represented as $|p_i$ (n_6, $n_9)|$equals to 4 based on the total number of edges between n_6 and n_9. Paths are used to determine the distance between nodes on the network defined as:

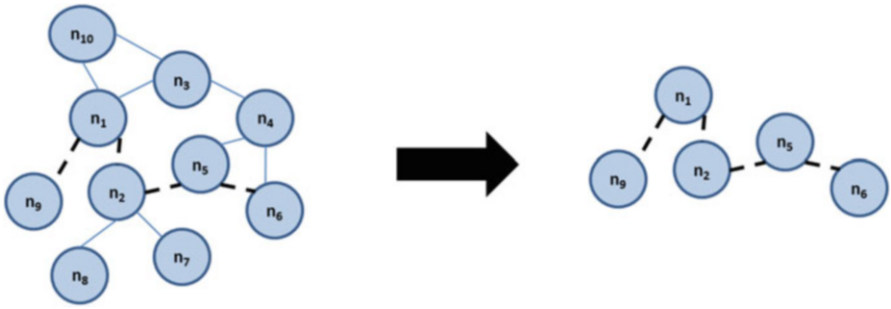

Fig. 5.3 Path between nodes

For any path p_i where $|p_i| = min|(n^{pi}, e^{pi})|$, then p_i is shortest path between each pair of nodes n_i and n_j, and p_i is also referred to as distance d where d is the distance $dist_G(n_i,n_j)$ between n_i and n_j.

This distance d is measured in terms of the number of edges between the nodes in question. In Fig. 5.3, the number of edges from nodes n_6 to n_9 is 4 such that $d = 4$. Hence, this is the shortest path between these two nodes and is thus the distance between these nodes. It should be noted that a computer network can have multiple distances since it is based on the shortest path between each pair of nodes on the network. However, the network can only have one diameter defined as:

For any path p_i where $|p_i| = min(max|(n^{pi}, e^{pi})|)$, then p_i is the diameter h of G represented as diam(G).

In order to determine the diameter of the network, we need to first determine all the shortest paths or distances *d* between each pair of nodes. The shortest maximum distance value between any pair of nodes is the diameter *h* of the overall network. According to Fig. 5.2, the distance between n_6 and n_9 is the shortest maximum distance between any pair of nodes in the network which makes it the diameter *h* of the network. The diameter of a network can be used to determine how dense or sparse a network is. Thus, if a network has a small diameter, then it is said to be well connected. On the other hand, if a network has a large diameter, then it is said to be sparse.

Both node centrality and graph-level metrics can be utilized to characterize how a network evolves over time.

5.3 Network Evolution

5.3.1 Node Evolution

The study of node evolution involves observing connections in a graph. From this, top central or influential nodes such as high-degree nodes as well as less popular nodes such as low-degree nodes can be identified [17–22], observed, and compared

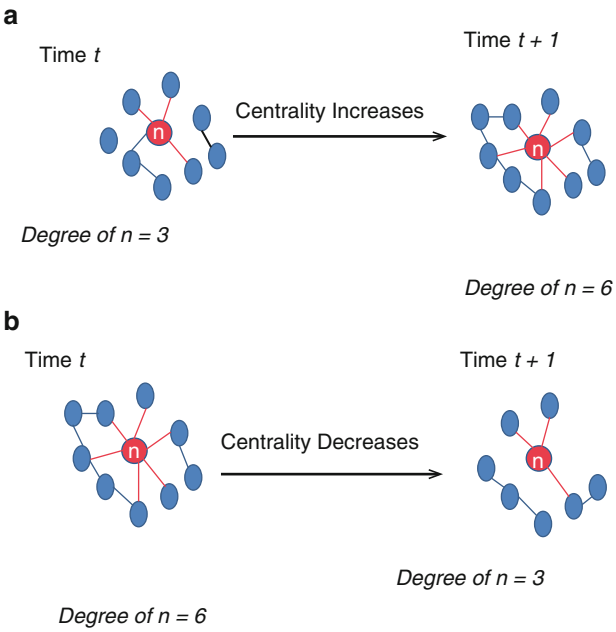

Fig. 5.4 (**a**) Centrality of a node increases over time. (**b**) Centrality of a node decreases over time

over time. Node evolution can also be observed in relation to neighborhoods as discussed in [23]. A certain set of numerical features of the neighborhood can be established for each node such as the number of neighbors (degree of a node) and the edges of the neighborhood, among others. Here it is possible that during network evolution, node centrality changes over time and that some nodes may disappear after sometime, or their centrality levels go higher and drop after a while for some, and that some nodes appear after a while and remain constantly present and maintain a high centrality level [19–22]. An example of changing node centrality is shown in Fig. 5.4.

5.3.2 Community Evolution

In order to detect community changes, [24–26] identify communities of nodes or communication patterns in the network and study how they evolve over time. For example, [25] study time-evolving networks where they analyze the evolution of network clusters through time to identify splits, merges, appearances, and disappearances of communities. On the other hand, [26] model the evolution of communities in heterogeneous networks where they study the size of communities to determine how they increase or decrease with time.

5.3.3 Graph Evolution

In graph evolution, [16–18, 27] observe key fundamental network properties to determine how networks grow and evolve over time. Particularly, such fundamental properties include densification power law, power-law degree, power-law eigenvector and eigenvalue distribution, edge-by-edge evolution, shrinking diameter, diameter, and radius. These properties are observed in relation to the degree of nodes.

For instance, [14, 15] clearly demonstrate that networks obey the densification power law where edges grow faster than nodes. First, the graph over time maintains a power-law degree distribution with a constant power law degree distribution exponent γ. If $\gamma < 2$ and is constant over time, then the graph is said to densify. An illustration is provided in Figs. 5.5, 5.6, and 5.7 for undirected and directed networks based on key subgraphs selected from network traffic data by the Center for Applied Internet Data Analysis (CAIDA) for the duration of December 2008 to January 2010 [37–39].

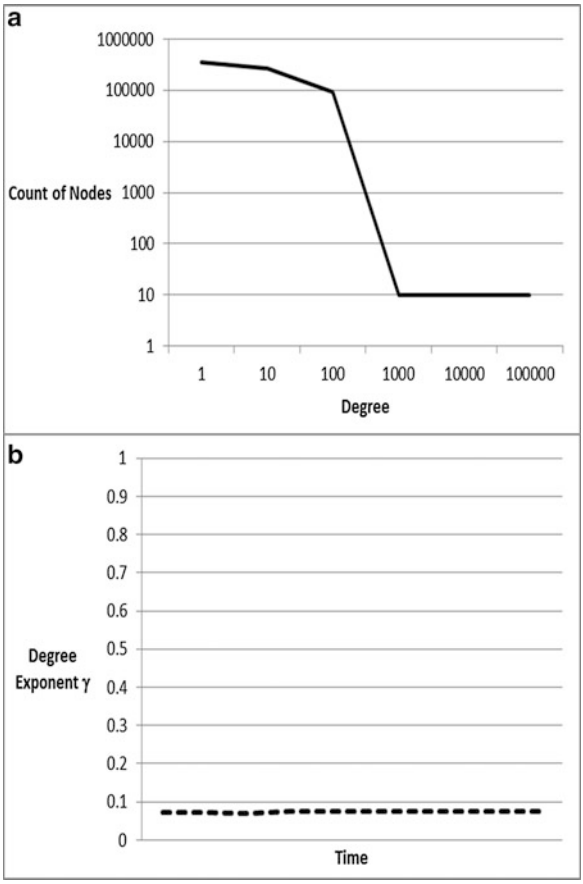

Fig. 5.5 (**a**) Example of a degree distribution in an undirected network. (**b**) Example of a degree exponent over time in an undirected network

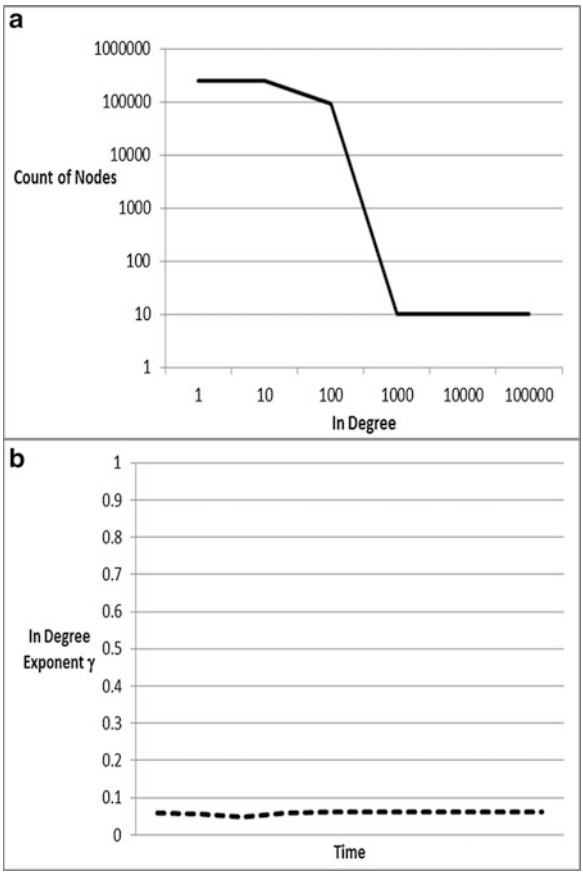

Fig. 5.6 (**a**) Example of an in-degree distribution in a directed network. (**b**) Example of an in-degree exponent over time in a directed network

Overall, Figs. 5.5, 5.6, and 5.7 show that the degree distribution has a long tailed distribution and thus follows a power-law distribution. Additionally, the power law degree distribution exponent $\gamma < 1$ in all cases and is constant over time. These [14, 15] also show that the diameter of the network shrinks over time such that as the network grows, the distances between nodes slowly decrease.

According to [15], in a temporally evolving graph, if the power law degree distribution exponent γ is constant over time, the densification exponent α is a function of γ such that $\alpha = 1$ if > 2, $\alpha = 2/\gamma$ if $1 < \gamma < 2$, and then $\alpha = 2$ if $\gamma < 1$. These properties can be used to clearly demonstrate how graphs densify over time.

5.4 Scientific Fundamentals for Change Detection

The study of network structures calls for an understanding of network structural features and fundamental network properties as described in graph concepts and network evolution, respectively. Such features and properties provide a basis for

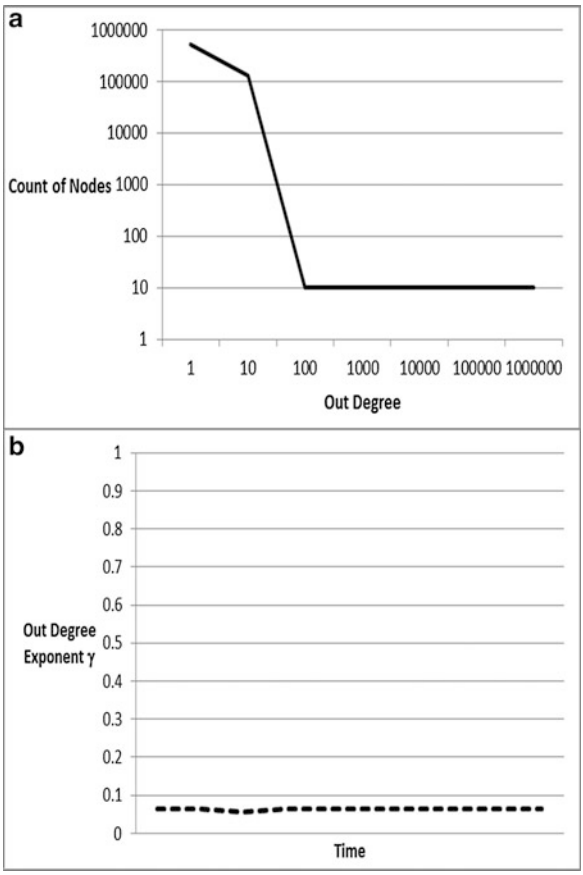

Fig. 5.7 (**a**) Example of an out-degree distribution in a directed network. (**b**) Example of an out-degree exponent over time in a directed network

analysis of network behavior associated to identifying patterns such as changes in the network over time. This section presents an overview on the concept of change detection in evolving networks. Here the focus of change lies on evaluating network behavior based on node features, network-level properties, or both. This chapter therefore discusses change detection in network structures from two perspectives: (1) uni-level change detection which focuses on detecting changes in either node-level behavior or network-level behavior, respectively, and (2) multilevel change detection which combines aspects of the network by observing both node-level and network-level behavior. A detailed description on each approach follows.

5.4.1 Uni-Level Change Detection

Uni-level change detection refers to the detection of change in a single network dimension where a single dimension is considered to be network level or node level,

respectively. The analysis of macroscopic behavior in network structures has been widely applied in detecting changes at the network level based on structural differences in network-level properties such as density, diameter, average degree, as well as other node centrality measures by translating them into network-level metrics [3, 11–13, 22, 28–30] study techniques to detect a change or disorder in the state of a time process, usually from normal to abnormal [24] propose GraphScope, an approach to discover communities of graphs and identify any changes in the community structure over time. Their approach identifies new graph segments which mark an abrupt change in the community structure and are thus considered to be discontinuities in time.

The concept of change detection has been explored in network analysis in relation to the application of Statistical Process Control (SPC) using techniques such as sequential probability ratio test (SPRT), the cumulative sum (CUSUM) chart, the exponentially weighted moving average (EWMA), and the Shiryaev–Roberts (SR) procedure [11, 29, 30]. However, SPC operates on the assumption that the data is sequential or time sequenced [31]. Additionally, such techniques may not be suitable to identify changes in non-sequential data such as variations between graph elements such as nodes within the same time period. Furthermore, there are differences between change-point analysis and control charting where the latter is generally better at detecting isolated abnormal points and at detecting a major change quickly, while change-point analysis can detect subtle changes frequently missed by control charts. Interestingly, the two methods can be used in a complementary fashion [32] given that changes usually cause shifts, minor or major, that can be viewed as abnormal. On the other hand, pattern recognition techniques, spectral theory, and mean/median of graphs have been discussed in graph change detection for macroscopic analysis [3]. Also, distance measures such as Hamming distance and Euclidean distance have been applied in change detection, although they do not provide the statistical distribution of the data and are suitable for static networks [11, 12].

5.4.2 Multilevel Change Detection

Multilevel change detection identifies multiple dimensions of change defined as micro- and macro-level changes in evolving networks. Here micro-level changes refer to changes with respect to structural characteristics in the behavior of nodes [20, 21] such as the centrality of nodes in the network, and macro-level changes refer to changes with respect to structural characteristics in the behavior of network-level properties such as density, average degree, and diameter [22]. Detection of multidimensional changes presents unique benefits to challenges associated to big data and dynamism of large complex network structures. As such, it can be used to detect phenomena that may not be evident from a single perspective, such as only micro level or macro level, respectively. More so, multilevel change detection can be used to identify correlated network behavior that may prove useful in detecting

cyber threats [22]. For example, changes at the macro level such as the diameter of the network may be associated to micro-level shifts in the behavior of key components within the network such as changes in the centrality level of nodes. Alternatively, changes in the centrality level of nodes such as a decrease in degree centrality indicates decrease in network connectivity which may thus lead to an increase in network diameter. In both micro- and macro-level changes, identifying time when such changes occurred indicates time points of change especially if they exist in a novel pattern [20–22].

Therefore, the studies described in [20–22] present a novel approach to characterizing large evolving networks and detecting changes in such evolving networks, which includes the following steps:

(a) **Selection of central nodes and subgraphs**: This utilizes a hybrid methodology that combines sampling, clustering, and stratified binning to select central nodes and key subgraphs associated to the central nodes from a network over time. This provides a selective analysis of large networks to reduce on the size and dimensionality. Most importantly, graph properties of selected subgraphs should emulate the established graph properties in the full graph. These properties as outlined by [15] specify that the networks are becoming denser over time and the average degree is increasing; hence densification follows a power-law distribution, and the diameter decreases as the network decreases in many cases.

(b) **Micro-level change detection:** For micro-level shifts in the network, the presence and centrality levels of the central nodes is observed to identify **Co**nsistent and **In**consistent (**CoIn**) central nodes where inconsistency marks changes in the presence and centrality of central nodes, respectively. Additionally, times associated to the changes in behavior of these central nodes are detected which are also referred to as CoIn **T**ime **P**eriods of **C**hange (**CoIn-TPC**). A node-level analysis drills down into the network and provides specifics on network activity that may be invisible on a larger scale.

(c) **Macro-level change detection:** Given that micro-level characteristics of the network do not relay information about the bigger picture in the overall network, the key subgraphs associated with the central nodes are used to identify times when the fundamental structural or network-level properties, particularly when significant changes in density, diameter, and average degree occur as a result of changes in the behavior of central nodes. These macro-level changes are referred to as Network Level CoIn (**NL-CoIn**) Change Points. Additionally, a correlation between CoIn central nodes and NL-CoIn is used to determine the impact of node-level changes on the network level as well as similarities between change points in CoIn-TPC and NL-CoIn. Here a network-level analysis describes a generic picture of underlying events in the network.

(d) **Validation based on real-world cyber events:** In order to ascertain that the changes identified are associated to real-world cyber events, CoIn-TPC and NL-CoIn are evaluated using ground truth in order to determine if the changes

are associated to existing cyber attacks [22]. The ground truth evaluation is based on real-world events from Internet-security reports by Akamai Technologies [35, 36]. Specifically, findings in [22] show high accuracy, precision, and recall levels in both node- and network-level changes associated to big cyber attacks such as the Conficker worm particularly during December 2008, January 2009, and February 2009.

5.5 Key Applications

The process of change detection to characterize network behavior can be potentially useful in the cybersecurity domain as discussed in the following sections.

5.5.1 Network-Intrusion Detection

Network-based intrusion detection attempts to identify unauthorized, illicit, and anomalous behavior based solely on network traffic. A network intrusion detection system (NIDS) is used to monitor traffic on a network looking for suspicious activity, which could be an attack or unauthorized activity. Change detection can be used to maintain a map of network activity by identifying and creating critical points on the network. For example, a large NIDS server can be set up on a backbone network, to monitor and audit all traffic; or smaller systems can be set up to monitor traffic and define a threshold on the behavior of central network elements, which can be a particular server, switch, gateway, or router. Specifically, a NIDS server can also detect changes in the connectivity levels of such central nodes on a network based on the number of connections at a particular time by looking for suspicious traffic or usage patterns that match a typical network compromise or threat. Such a server can also play a proactive role to identify potential exploits or for scanning live traffic to see what is actually going on within the network. The process of change detection can be used to develop a comprehensive list of network activity and structural organization in order to establish normal versus abnormal network activities.

5.5.2 Threat Mitigation

Security and technology teams must be ready for cyber attacks against critical infrastructure. With destructive cyber attacks on the rise, there is a need to practice troubleshooting processes for critical system restorations before outages occur [34]. Hence, it is important to know a system so well in order to quickly determine

what process caused the outage by identifying what went wrong and why. The motivation behind computer crime can be anything: financial gain, curiosity, revenge, boredom, "street cred," delusions of grandeur, and more. But what if it is a cyber attack? Change detection can be used to reduce on the complexity surrounding network analysis by identifying vulnerable points on the network. Here, an attack profile can be developed to control and minimize the impact of an attack on the network. For example, taking down a highly connected node such as a server could put network communications on a halt. This essentially affects the connectivity on the network which is determined by density on the network, as well as the distance from one network point to another which is determined by the diameter. Hence, change detection can be used to identify the potential source of the problem and use it to trace any changes in network behavior.

5.5.3 Network Design

Computer attacks have been graphically modeled since the late 1980s by the US DoD [33]. With the support of advanced tools, network risks can be modeled based on an attack graph where each node in the graph represents an attack state and the edges represented a transition or a change of state caused by an action of the attacker. Such models can be used for network security evaluation. Preventing cyber attacks poses several challenges considering the complexities surrounding large evolving network structures. In order to alleviate such challenges, a wide range of strategies may require testing to identify network vulnerabilities and determine resource allocation on the network. Particularly, change detection can be used to ensure risk management on the network during network design. Similar to threat detection, it can be utilized in identifying vulnerable points such as central nodes that can be targeted to cripple the network. Based on this, network redundancy can be created where such central nodes are duplicated to maintain consistent network activity by redirecting communications in case of an attack.

5.6 Future Directions

This chapter has reviewed state-of-the-art techniques in change detection and network characterization utilizing essential graph-based knowledge. The future directions for this work include addressing challenges associated with sampling big data contained in large graphs by predicting the samples from a given range data in large evolving graphs while at the same time preserving the fundamental network properties. On the other hand, the process of change detection can be extended into predictive network modeling where change points detected as well as non-change points can be used as feature vectors for prediction of network behavior in order to determine if a persistent pattern exists in the micro- and macro-level changes.

Furthermore, given that change detection has been mainly explored in the context of time, it creates an interesting opportunity to adapt such techniques into the spatio-temporal paradigm particularly by identifying spatial regions associated with network changes as well as potential cyber threats.

References

1. Matarazzo, C.: Defending computer networks against attack. Lawrence Livermore National Laboratory. Research Highlights. https://str.llnl.gov/JanFeb10/pdfs/1.10.3.pdf. (2010)
2. Aleksandrova, D.: Detecting cyber attacks. How long does it take? IT Governance. http://www.itgovernance.co.uk/blog/detecting-cyber-attackers-how-long-does-it-take/. (2015)
3. Gaston, M., Kraetzl, M., Wallis, W.: Using graph diameter for change detection in dynamic networks. Australas. J. Combin. **35**, 299–311 (2006)
4. Scripps, J., Tan, P.-N., Esfahanian, A.-H.: Node roles and community structure in networks. Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis, pp. 26–35. New York, NY, USA, ACM, (2007).
5. Shen Y., Nguyen, N.P., Xuan, Y., Thai, M.T.: On the discovery of critical links and nodes for assessing network vulnerability. IEEE Trans. Network. (2012)
6. Freeman, L.C.: Centrality in social networks: conceptual clarification. Soc. Network. **1**(3), 215–239 (1979)
7. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: bringing order to the Web. Technical Report. Stanford InfoLab. (1999)
8. Bonacich, P.: Technique for analyzing overlapping memberships. Sociol. Methodol. **4**, 176–185 (1972)
9. Bonacich, P.: Power and centrality: a family of measures. Am. J. Soc. **92**, 1170–1182 (1987)
10. Lee, C.-Y. Correlations among centrality measures in complex networks. arXiv:physics/0605220 [physics.soc-ph]. (2006)
11. McCulloh, I., Carley, K.M., Horn, D.B.: Change detection in social networks. United States Army Research Institute for the Behavioral and Social Sciences. (2008)
12. McCulloh, I.: Detecting changes in a dynamic social network. Institute for Software Research School of Computer Science Carnegie Mellon University. Thesis (2009)
13. McCulloh, I., Carley, K.M.: Detecting change in longitudinal social networks. J. Soc. Struct. **12**(2011) (2011)
14. Leskovec, J., Kleinberg, J., and Faloutsos, C. Graphs over time: densification laws, shrinking diameters and possible explanations. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD) (2005)
15. Leskovec, J., Kleinberg, J., Faloutsos, C.: Graph evolution: densification and shrinking diameters. In: ACM Transactions on Knowledge Discovery from Data (TKDD), vol 1 (2007).
16. Leskovec, J., Faloutsos, C.: Scalable modeling of real graphs using kronecker multiplication. In International Conference on Machine Learning (ICML) (2007)
17. Leskovec, J.: Dynamics of large networks (2008)
18. Kang, U., Tsourakakis, C., Appel, A., Faloutsos, C., Leskovec, J.: Radius plots for mining tera-byte scale graphs: algorithms, patterns, and observations. In: SIAM International Conference on Data Mining (SDM) (2010)
19. Tong, H., Papadimitriou, S., Yu, P.S., Faloutsos, C.: Proximity tracking on time-evolving bipartite graphs. In: SDM (2008)
20. Namayanja, J.M., Janeja, V.P.: Discovery of persistent threat structures through temporal and geo-spatial characterization in evolving networks. ISI 191–196 (2013)

21. Namayanja, J.M., Janeja, V.P.: Change detection in temporally evolving computer networks: a big data framework. First International Workshop on High Performance Big Graph Data Management, Analysis, and Mining, Co-located with the IEEE BigData 2014 21. J. M. (2013)
22. Namayanja, J.M., Janeja, V.P.: Change detection in temporally evolving computer networks: changes in densification and diameter over time. ISI (2015)
23. Akoglu, L., McGlohon, M., Faloutsos, C.: Oddball: Spotting anomalies in weighted graphs. In: PAKDD, pp. 21–24 (2010)
24. Sun, J., Faloutsos, C., Papadimitriou, S., Yu, P.S.: GraphScope: parameter-free mining of large time-evolving graphs. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 687–696 (2007a)
25. Ferlez, J., Faloutsos, C., Leskovec, J. J., Mladenic, D., Grobelnik, M.: Monitoring network evolution using mdl. In IEEE International Conference on Data Engineering (ICDE) (2008)
26. Han, J., Sun, Y., Yan, X., Yu, P.S.: Mining heterogeneous information networks. In: KDD (2010)
27. Fabrikant, A., Koutsoupias, E., Papadimitriou, C.H.: Heuristically optimized trade-offs: a new paradigm for power laws in the Internet, volume 2380 of Automata, Languages and Programming, p. 781. Springer (2002)
28. Opsahl, T., Agneessens, F., Skvoretz, J.: Node centrality in weighted networks: generalizing degree and shortest paths. Soc. Network. **32**(3), 245–251 (2010)
29. Akoglu, L., Faloutsos, C. Anomaly, event, and fraud detection in large network datasets. In: Proceedings of the Sixth ACM International Conference on Web Search and Data Mining, pp. 773–774. ACM. (2013)
30. Tartakovsky, A.G., Polunchenko, A.S., Sokolov, G.: Efficient computer network anomaly detection by changepoint detection methods. IEEE J. Selected Topics Signal Process. **7**(1), 7–11 (2013)
31. Slavin, V.: Improper use of control charts: traps to avoid. (2006)
32. Taylor, W.A.: Change-point analysis: a powerful new tool for detecting changes, WEB: http://www.variation.com/cpa/tech/changepoint.html. (2000)
33. Abraham, S., Nair, S.: Cyber security analytics: a stochastic model for security quantification using absorbing markov chains. J. Commun. (2014)
34. Lohrmann, D.: Hacking critical infrastructure is accelerating and more destructive. http://www.govtech.com/blogs/lohrmann-on-cybersecurity/Hacking-Critical-Infrastructure-is-Accelerating-and-More-Destructive.html. (2015)
35. Akamai Technologies.: 4th Quarter 2008: The State of the Internet. **1**(4). https://www.stateoftheinternet.com/resources-connectivity-2008-q4-state-of-the-internet-report.html. (2009)
36. Akamai Technologies.: 1st Quarter 2009: The State of the Internet. **2**(1). https://www.stateoftheinternet.com/resources-connectivity-2009-q1-state-of-the-internet-report.html. (2009)
37. The CAIDA UCSD [Anonymized Internet Traces 2008]–[April 2011–December 2013], http://www.caida.org/data/[/passive-2008/
38. The CAIDA UCSD [Anonymized Internet Traces 2009]–[April 2011–December 2013], http://www.caida.org/data/[/passive-2009/
39. The CAIDA UCSD [Anonymized Internet Traces 2010]–[April 2011–December 2013], http://www.caida.org/data/[/passive-2010/