2019 IEEE International Conference on Big Data (Big Data)

The Effectiveness of Edge Centrality Measures for Anomaly Detection

Candice Mitchell*, Rajeev Agrawal, PhD†, Joshua Parker, PhD‡
*School of Mathematics and Natural Science
University of Southern Mississippi, Hattiesburg, MS
Email: candice.bardwell@usm.edu
† U.S. Army Engineer Research and Development Center, Vicksburg, MS
† Email: rajeev.k.agrawal@erdc.dren.mil
‡ Email: joshua.m.parker@erdc.dren.mil

***Abstract*—Anomalies in network traffic are often detected using machine learning techniques, such a Artificial Neural Networks, Self-Organizing Maps, k-Nearest Neighbors, or Principal Component Analysis. These techniques are built upon certain predetermined features that are believed to be useful in detecting anomalies. Many researchers are using graph-based features, such as betweenness centrality or eigenvector centrality. The choice of these particular features is due to the assumption that they can be used to accurately predict an anomaly in the flow of traffic. However, there appears to be no solid foundation for these assumptions.**

This work investigates edge centralities and how accurately they predict anomalies using netflow data. We propose to use known traits of different network interactions to identify how information will flow. We will then predict which measures of centrality should be most applicable to these particular flows. Finally, using public cybersecurity data sets, we will investigate which measures of edge centrality accurately identify anomalies as outliers then make comparisons with our predictions. Ideally, this will allow us to choose graph-based features that are highly efficient in anomaly detection.

***Index Terms*—cybersecurity, anomaly detection, centrality**

I. Introduction

In our increasingly connected world, we have access to services, people, and devices at the click of a button. Cybersecurity improves upon information security to ensure that all of our assets, from money to personal information, are kept secure while we are connected [1]. Unfortunately, as our technological abilities and interconnections increase, so do the abilities of those who wish to gain access to our information. Anomaly detection on a network is one method for detecting possible vulnerabilities.

How can we use router data about network traffic to determine if there is an intrusion? On any network, we expect traffic to behave in a similar way at any given time, so we should be able to use this information to determine if something anomalous is occurring on the network. In looking for changes in traffic patterns, we can investigate changes in the level of influence of certain connections.

Graph-based features like centrality have proven to be an effective method for anomaly detection. However, in regards to computer networks, there are still some gaps. Many researchers look only at node centrality measures, specifically betweenness and eigenvector centrality without investigating edge centrality measures. Other often overlooked aspects are the underlying assumptions made by many of the measures, like how the information moves and replicates as it traverses the network.

This work intends to provide a firm understanding of edge centralities and their usefulness for network data analytics and event detection. We investigate three centralities: edge betweenness centrality, edge load centrality, and edge random walk betweenness centrality. We look at the underlying assumptions for each of these measures and make predictions about their usefulness in anomaly detection. We then analyze their actual performance using a labeled cybersecurity data set and make comparisons with our predictions.

The organization of the paper is as follows: Section II discusses previous work into this area. The edge centrality measures are defined and analyzed in Section III. A description of the cybersecurity dataset is given in Section IV. Section V discusses the algorithm used for anomaly detection, and Section VI discusses the results we obtained. Finally, we give our conclusions and possible directions for future work in Section VII.

II. Related Work

The use of graph-based measures for anomaly detection is a growing area of research due to its promising results. Botnets are a network anomaly that consists of interconnected bots which exist to cause disruptions on the network. Botnet detection using graph-based measures like node betweenness and eigenvector centrality was investigated by [2]. They were able to detect botnets using a limited number of nodes. This is important when looking at network traffic, because the infrastructure is often very large and burdensome to work with. [3] investigated a hybrid approach which uses graph-based features like betweenness centrality and node degree along with a flow-based analysis for botnet detection. This hybrid method also yielded good results in detecting botnets.

978-1-7281-0858-2/19/$31.00 ©2019 IEEE

Graphlets are induced subgraphs that describe local topography. [4] uses graphlet counts on netflow data as a highly efficient means of detecting network outliers in almost real-time. [5] investigates using background knowledge in the form of rule coverage. They use this background information as a way to guide the outlier detection process and were able to decrease computational time as well as memory usage.

In social networks, the uses and applications of centrality measures have been studied extensively. A lot of insight was gained from [6] and their in depth discussion of the underlying assumptions made by many centrality measures. [7] gives a thorough history and breakdown of different centrality measures along with their formal definitions. They also discuss their applications in many real-world scenarios.

For anomaly detection, researchers are mainly looking at node centralities. Traffic betweenness centrality [8] weights sources and destinations based on the amount of traffic that travels between them. It thins out any nodes that cannot be sources or destinations, making the graph more manageable. It was introduced as a method for optimizing malware placement filters. They also discuss the use of an adjacency matrix for vulnerability visualization. Routing betweenness centrality [9] rates the probability that one node will proceed to a given node from its current location. It combines betweenness centrality, load centrality, and flow betweenness centrality. Both of these are new node centrality measures created in an attempt to better detect anomalies on a network.

A thorough overview of the accuracy of many graph-based features in anomaly detection can be found in [10]. They discuss node-based centralities but overlook edge-based centralities. They correlate their work to computer networks and discuss the importance of not just finding but also classifying anomalies. Due to difficulties with scalability, [11] has created a distributed algorithm for calculating load centrality on nodes. It exploits routing protocol information to created a distributed algorithm. They chose load centrality, because it converges to betweenness centrality on large graphs. There is also research by [12] into the effectiveness of combining multiple node centralities on many different types of networks, for example football and airport data. Their results are very interesting, but they claim that the reason these centralities correlate is unknown.

It seems that many researchers feel that anomalies can be detected based on nodes. However, netflow data is generally obtained in the form of a connection, which represents an edge on the network graph. [13] investigates an eigenvector approach to finding edge centralities. They also compare different edge centralities based on their entropy and describe the correlation between different entropies. [14] discusses an improvement to spanning edge centrality which looks at the portion of spanning trees that contain a given edge. Both of these are methods for scaling edge centralities to large networks. However, there is still little work into which edge centralities would be most effective and, more importantly for extending the work, why.

	Parallel Duplication	Serial Duplication	Transfer
Geodesics	<No process>	Mitotic reproduction	Package delivery
Paths	Internet name-server	Viral infection	Mooch
Trails	E-mail broadcast	Gossip	Used goods
Walks	Attitude Influencing	Emotional support	Money exchange

TABLE I: Typology of flow processes.

III. Centrality Measure Analysis

Previous work [6] in social networks has shown that there are implied assumptions for many measures of centrality. For example, betweenness centrality assumes that information only travels along the geodesic (shortest path). It then becomes important to consider how network traffic realistically flows before assuming that we can utilize a particular measure of centrality effectively. The goal of this research is to adapt these ideas from social networks and apply them to computer networks.

To classify information flow on the network, we will reference the classification used by [6]. First, we look at how information is diffused. Can it be replicated, like an email, or does it need to be transferred, like a package? If it can be replicated, does this happen one at a time (serial duplication), like a word document undergoing edits, or simultaneously (parallel duplication), like a group email? Next, we look at how information travels. Will it take a walk where edges (connections) and nodes (IP addresses) can be repeated? Will it follow a trail that does not allow edges to be traversed twice? Will it follow a path where no edges or nodes are repeated, and, if so, will it take the shortest of these, called the geodesic? Examples of each type from [6] can be seen in Table I.

Again, since we are talking about network *connections*, we will look at edge centralities. We investigate three different edge centralities, which are defined below. Given a graph G with edge e,

- **Edge betweenness centrality** measures the proportion of shortest paths between pairs of vertices that contain e,

$$C_b = \sum_{s \neq t} \frac{\sigma_{st}(e)}{\sigma_{st}}$$

where σ_{st} is the total number of shortest paths and $\sigma_{st}(e)$ is the total number of those paths that pass through e. Betweenness centrality of nodes is one of the most commonly used centrality measures for anomaly detection. Edge betweenness centrality assumes that we are traveling along geodesics and that information cannot be duplicated. Intruders are likely to travel along a geodesic; however, it is unlikely that they will not be replicated. For example, a Distributed Denial of Service (DDoS) needs

to infect many machines, so it must replicate at each new machine.

- **Edge load centrality**, C_l, is derived from load centrality of a node which sends information to the node closest to the target. If there are two or more such nodes, the information is split equally between them. Similarly, edge load centrality sends information along the *edge* closest to the target and splits information equally between edges with the same (closest) distance from the target. An example can be seen in Figure 1. In the example, we are passing 10 units in at node A and watching how they travel to node D. Since the shortest path length is three edges, the information will not flow along C-G or G-E at all. In an undirected graph, node load centrality converges to betweenness centrality; however, this does not appear to be the case for edge load centrality. Load centrality assumes a geodesic, like betweenness centrality; however, it allows for parallel duplication of information. We would assume that this would be a more accurate measure for intruder detection.

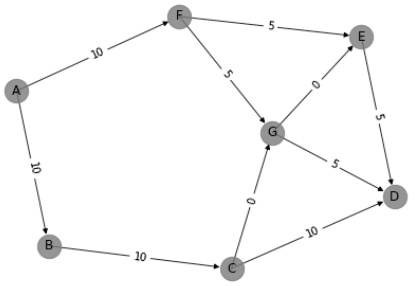


Fig. 1: Edge Load Centrality on 10 Units Flowing from A to D

- **Edge random walk betweenness centrality** measures the fraction of random walks between pairs of vertices that pass through e

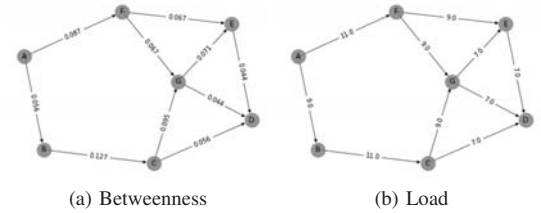
$$C_r = \sum_{s \neq t} \frac{\omega_{st}(e)}{\omega_{st}}$$

where ω_{st} is the total number of random walks and $\omega_{st}(e)$ is the number of those paths that pass through an edge e. In this final centrality, we are allowed to traverse nodes and edges multiple times, but we assume that information cannot be duplicated.

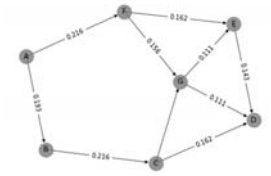
An example of all three centrality measures on a toy network can be seen in Figure 2.

IV. Cybersecurity Dataset Description

For this work, we chose to use the publicly available UNSW-NB15 dataset which was created by the IXIA Perfect-Storm tool in the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS) for generating a hybrid of real modern normal activities and synthetic contemporary attack behaviors



(a) Betweenness (b) Load



(c) Random Walk Betweenness

Fig. 2: Centralities on a Toy Network

This dataset is broken down into four parts, each of which have 700,000 interactions (except the fourth which only has 440,000). Among these interactions between 3 and 22.5% are anomalous depending on which you are looking at. There are a total of 49 features and nine types of attacks, namely, Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms. [15]

V. Methodology

To determine their actual effectiveness at anomaly detection, we designed an algorithm that uses Pandas to read in source and destination IP addresses and ports from a cybersecurity data set. Originally, we were only using IP addresses, but we quickly realized that this did not provide us with enough information about the network. In an attempt increase the amount of information, we concatenated the source IP address with the source port and similarly concatenated destination information. For example, if we read in an IP address of 175.45.176.1 using port 2396, we reduce this to 175.45.176.1:2396. However, this turned out to be too much information, and we were unable to process the data. In order to reduce the number of edges while still maintaining some level of individuality, we adapted the algorithm to concatenate the IP address with how well-known its port is. As is conventional, we define a port to be well-known if its value is between 0 and 1023. Well-known ports are assigned a value of 1. Otherwise, the value assigned is 2. The example from above is now reduced to 175.45.176.1:2.

Our algorithm then creates a graph which is broken into subgraphs. We calculate the chosen centrality measure of each edge and return an updated list of edges that includes centralities. Using a predetermined cutoff percentage, we determine which edges have centralities above this cutoff. Specifically, if our cutoff is 10%, we should be finding the edges in the largest 10% of all found centralities. Figure 3 shows the edges found above a 10% cutoff for edge load centrality. Every edge above the red line has an edge load centrality greater than or equal to 70.2257, so they are in the top 10% of centrality scores.

Unfortunately, labeling the graph makes it too cluttered. In order to see which edges where identified, we added a hovering feature when the graph is output that allows you to see the corresponding edge by hovering over it on the plot.

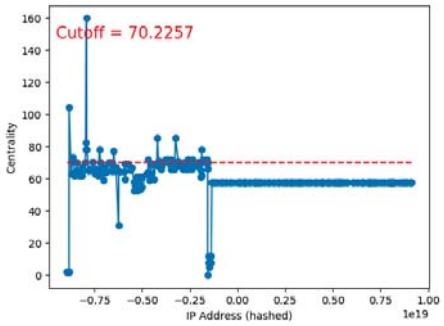


Fig. 3: Edges Found above a 10% Cutoff for Edge Load Centrality

Finally, since we are using a labeled dataset, we compare our findings to the actual anomalous edges. We calculate the following four statistics based the comparisons. In the following definitions, 't,' 'f,' 'p,' and 'n' represent 'true,' 'false,' 'positive,' and 'negative,' respectively. If our algorithm labeled an edge anomalous, but it was actually normal, it would be a false positive, 'fp'.

- **Accuracy** is the ratio of correctly labeled edges out of all of the edges,

$$acc = \frac{tp + tn}{tp + fp + tn + fn}$$

- **Precision** is the ratio of correctly labeled anomalous edges out of all edges labeled anomalous,

$$prc = \frac{tp}{tp + fp}$$

- **Recall** is the ratio of actual anomalous edges that were correctly predicted out of all true anomalous edges,

$$rec = \frac{tp}{tp + fn}$$

- **F1 Score** is the weighted mean of precision and recall,

$$f1 = \frac{2 \star prc \star rec}{prc + rec}$$

In our original data set, we see that some edges can be both normal and anomalous due to the way we are labeling them with IP address and 1 or 2 depending on how well-known the port is. This led to the obvious question of how we should label this edge. Should we label it anomalous if it ever occurs as an anomalous edge or should we take a ratio of the times it is anomalous to the total number of occurrences? Initially, we tried the first method; however, to improve accuracy, we ultimately went with the second method. In the algorithm, this works in the following way. Suppose we label an edge anomalous, and, in the ground truth, this edge occur 24 times as anomalous and 6 times as normal. We add 0.8 $\left(\frac{24}{30}\right)$ to the true positive count and and 0.2 $\left(\frac{6}{30}\right)$ to the false positive count.

VI. Results

Some of the findings can be seen in Figure 4, which shows the varying effectiveness of anomaly detection for all three measures using different cutoffs. We notice that at a 40% cutoff edge load centrality's effectiveness is drastically reduced. A similar reduction happens for the other two at 45%. However, before this reduction, load centrality (C_l) consistently does a better job of detecting anomalies than the other two measures and has a recall of 93.8% at a 35% cutoff. We can also see that edge betweenness centrality (C_b) is not bad with recall just under 70% using a 35% cutoff and, starting at a 25% cutoff, edge random walk betweenness centrality (C_r) gives almost exactly the same results.

Once the algorithm labels the edges, we see some normal edges misclassified as anomalous and vice versa. If the edge is anomalous but our algorithm misclassifies it as normal, we may avoid detecting an intruder on the network. Therefore, we will place more value on recall than precision. It is perhaps easier to see what we mean in Figure 5 which shows the comparisons between the actual network and what our algorithm finds. Ideally, we want the found graphs to be identical to the actual (first) graph. However, our goal is to

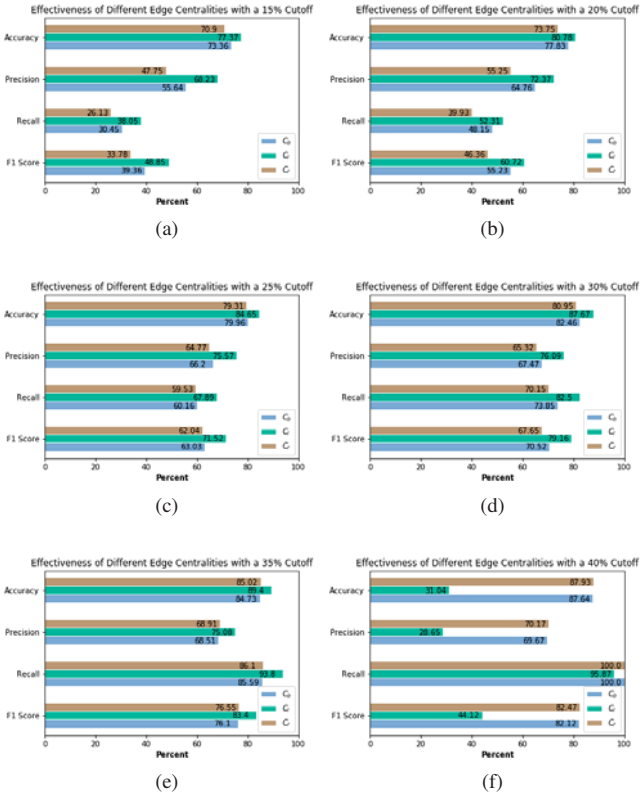
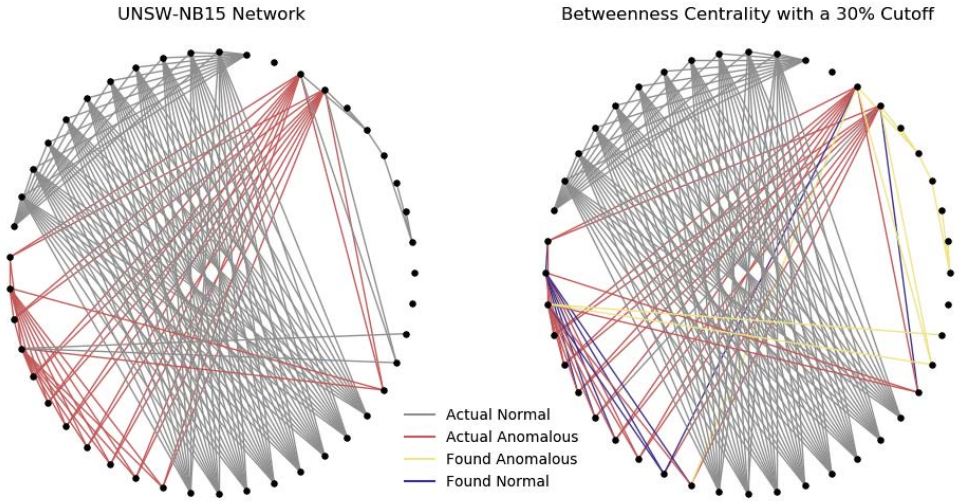
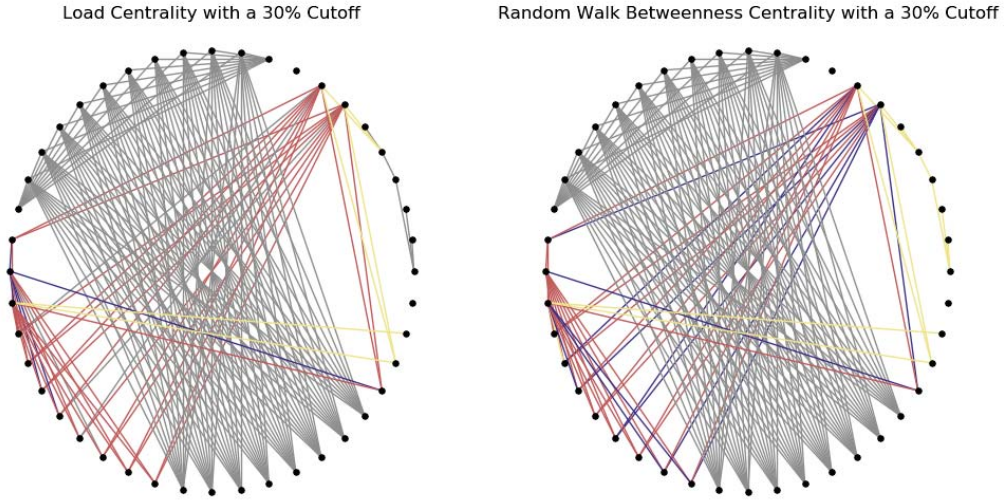


Fig. 4: Varying Cutoffs and Their Effectiveness using Individual Centralities



(a)



(b)

Fig. 5: Comparisons of Actual Network Activity and Found Network Activity

reduce the number of blue edges which represent anomalous edges labeled as normal.

After calculating the centralities as individual measures, we combine multiple measures to check if performance was better using two or three measures combined. Figure 6 shows that combining two centralities does in fact lead to better performance. At a 30% cutoff, we are already getting recall of 100% with the combination of edge load and random walk betweenness centralities as well as with the combination of all three centralities. At the 40% cutoff, we again see a drastic reduction in effectiveness which is carried over from the individual measures. We also note that at cutoffs above 30% adding the third centrality leads to virtually no change.

VII. Conclusions and Future Work

Our investigations into edge centralities have shown that they have the potential to be effective for anomaly detection

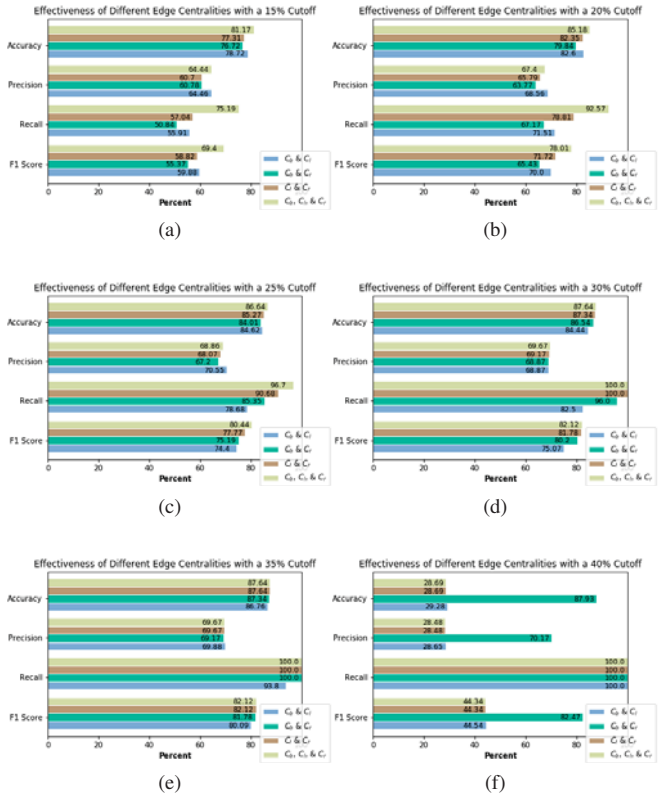


Fig. 6: Varying Cutoffs and Their Effectiveness using Combined Centralities

when we keep in mind their underlying assumptions and use them properly. Some future plans for this work include testing our methods on a variety of datasets and adapting the methods to directed graphs in an attempt to better represent information flow on the network. We wish to investigate adding other graph-based features, like node degree, and well as using more netflow information, such as duration and number of packets sent. Finally, we wish to test the method on time steps, or small portions of time, to see if anomalies are more easily detected on a slice of the data.

REFERENCES

[1] R. von Solms and J. van Niekerk, "From information security to cyber security," *Computers & Security*, vol. 38, pp. 97 – 102, 2013. Cybercrime in the Digital Economy.

[2] S. Chowdhury, M. Khanzadeh, R. Akula, F. Zhang, S. Zhang, H. Medal, M. Marufuzzaman, and L. Bian, "Botnet detection using graph-based feature clustering," *Journal of Big Data*, vol. 4, p. 14, May 2017.

[3] Y. Shang, S. Yang, and W. Wang, "Botnet detection with hybrid analysis on flow based and graph based features of network traffic," in *Cloud Computing and Security* (X. Sun, Z. Pan, and E. Bertino, eds.), (Cham), pp. 612–621, Springer International Publishing, 2018.

[4] C. R. Harshaw, R. A. Bridges, M. D. Iannacone, J. W. Reed, and J. R. Goodall, "Graphprints: Towards a graph analytic method for network anomaly detection," in *Proceedings of the 11th Annual Cyber and Information Security Research Conference*, CISRC '16, (New York, NY, USA), pp. 15:1–15:4, ACM, 2016.

[5] S. Velampalli and W. Eberle, "Novel graph based anomaly detection using background knowledge.," in *Proceedings of the Thirtietth International Florida Artificial Intelligence Research Society Conference*, pp. 538–543, 2017.

[6] S. P. Borgatti, "Centrality and network flow," *Social Networks*, vol. 27, no. 1, pp. 55 – 71, 2005.

[7] K. Das, S. Samanta, and M. Pal, "Study on centrality measures in social networks: a survey," *Social Network Analysis and Mining*, vol. 8, p. 13, Feb 2018.

[8] A. S. Cheema, J. Kohli, K. Arora, S. Gupta, and S. S. Ahmed, "Network security using graph theory," 2013.

[9] S. Dolev, Y. Elovici, and R. Puzis, "Routing betweenness centrality," *J. ACM*, vol. 57, pp. 25:1–25:27, May 2010.

[10] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: a survey," *Data Mining and Knowledge Discovery*, vol. 29, pp. 626–688, May 2015.

[11] L. Maccari, L. Ghiro, A. Guerrieri, A. Montresor, and R. L. Cigno, "On the distributed computation of load centrality and its application to dv routing," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pp. 2582–2590, April 2018.

[12] S. Oldham, B. D. Fulcher, L. Parkes, A. Arnatkeviciute, C. Suo, and A. Fornito, "Consistency and differences between centrality metrics across distinct classes of networks," *CoRR*, vol. abs/1805.02375, 2018.

[13] X. Huang, M. Huang, and W. Huang, "A novel metric for edge centrality," in *Proceedings of the International Conference on Big Data and Internet of Thing*, BDIOT2017, (New York, NY, USA), pp. 16–20, ACM, 2017.

[14] C. Mavroforakis, R. Garcia-Lebron, I. Koutis, and E. Terzi, "Spanning edge centrality: Large-scale computation and applications," in *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, (Republic and Canton of Geneva, Switzerland), pp. 732–742, International World Wide Web Conferences Steering Committee, 2015.

[15] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, pp. 1–6, Nov 2015.