

# Measuring and Analyzing the Significance of Nodes via Various Methods in a Diffusion Network with Bridge Detection

Yunho Song

Chadwick International Sophomore, Incheon, South Korea

[ysong2023@naver.com](mailto:ysong2023@naver.com)

DOI: 10.26821/IJSHRE.9.5.2021.9108

## ABSTRACT

*Diffusion is a process of spread through networks through connections between individuals. These networks will show how one disease can spread from one to another creating a pattern between nodes and with their significance of nodes. In these networks, the centrality of nodes can vary. In this paper, the various centrality measurement methods and bridge detection algorithms were introduced with examples. Additionally, each of the method's capabilities were measured from the test results and bridge detection is shown as most effective. Finally, the most effective and accurate method for analyzing the diffusion network is suggested.*

**Keywords: Diffusion Network, Bridge Detection, Centrality, Eigenvector, Pagerank, Time Complexity**

## 1. INTRODUCTION

Diffusion is a process of spread through networks through connections between individuals. These spreads can involve information, viruses, and other transmissible vectors. In diffusion, a network graph can be created from connections between individuals as nodes, and their transmissions as edges. These networks will show how one disease can spread from one to another creating a pattern between nodes and with their significance of nodes.

Specifically for a particular disease, a diffusion network graph can effectively project the human interactions and the causes for spreadings of that disease throughout a community.[1] Patterns can be investigated through the movements between nodes

and edges, and future actions can be made to prevent these diseases.

Since not every node would have the same amount of edges, there can be certain nodes that hold more significance compared to others. The degree of significance will be measured as centrality, where there can also be multiple methods of measurement. A node with a higher centrality is more likely to spread diseases to more individuals when infected.

To make sure that the time is efficiently used, the big graph will be divided into few sub-graphs with bridge detection, where the bridge is a specific edge that divides the whole graph into two sub-graphs, then the centrality calculations will be applied, identifying the central nodes in each subgraph. In the process, we will perceive the sub-graphs as a single community, since in a diffusion network, the nodes only connected by a single edge would have a weak connection. Starting in chapter 2, the contributions will be discussed. In chapter 3, the related works consisting of the centrality measurements will be explained. Chapter 4 will provide an explanation of bridge detection with chapter 5 with the evaluation of the work. chapter 6 will have the conclusion, and finally, chapter 7 will include the future research.

## CONTRIBUTION

The contribution points of this paper are the following:

a) The analysis of appropriate types of centrality measurement for diffusion graphs

b) The analysis of effects of bridge detection

Since there are different time and results for different methods of measuring centrality, it is important to find out which one would most resemble the results from an actual diffusion network. Additionally, bridge detection will be used to enhance efficiency of time as mentioned in the introduction.

$$C_d^{norm}(v_i) = \frac{d_i}{n-1} \quad \text{(Equation 3)}$$

Using the maximum degree placing it as a denominator will instead divide the selected degree by the maximum degree.

**2. Related Works**

The important nodes will be calculated in a diffusion pattern. Calculating node centrality had methods of degree centrality, betweenness centrality, closeness centrality, eigenvector centrality and pagerank centrality.[2],[4] and [6]

$$C_d^{max}(V_i) = \frac{d_i}{\max_j d_j} \quad \text{(Equation 4)}$$

Ultimately, the degree sum can be divided with the total number of degrees added from each node.

**2.1 Degree Centrality**

Degree centrality calculates the degree of importance through measuring the number of edges for each node. This is similar to a real-life situation where one can consider another with more social connections to be more socially active. The degree centrality will order each node depending on their degrees to how significant they are. In terms of disease spreading, the higher-ranking nodes would likely spread their diseases to more individuals when infected.

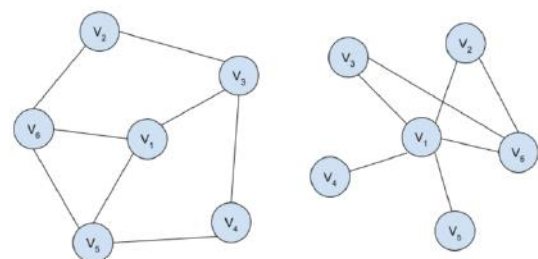
$$C_d^{sum}(v_i) = \frac{d_i}{\sum_j d_j} = \frac{d_i}{2|E|} = \frac{d_i}{2m} \quad \text{(Equation 5)}$$

**Example 1**

Considering figure 1 and 2, normalized degree centrality is equal to the degree of each node divided by the total number of nodes. For figure 1, the total number of nodes is  $n = 5$ . The denominator will be the corresponding degree to each node. When applied to equation 3, the results will be the following.

$$C_d(V_i) = d_i \quad \text{(Equation 1)}$$

In this context, outputs of each node will hold more significance since the contamination will happen through the output of each node. Resultantly, our formula of focus will be measuring the gregariousness of the disease. Also, the output holds more importance since it is the result value of the patterns.



<figure 1>

<figure 2>

- (1)  $C_d(V_i) = d_i^{in}$  (prestige)
  - (2)  $C_d(V_i) = d_i^{out}$  (gregariousness)
  - (3)  $C_d(V_i) = d_i^{in} + d_i^{out}$
- (Equation 2)

$$V_1 = 3/5, V_2 = 2/5, V_3 = 3/5, V_4 = 2/5, V_5 = 3/5, V_6 = 3/5$$

Unlike equation 2, equation 3 divides the equation by the total number of nodes except the self. The number of nodes (n) measures excluding the self-node.

For figure 2, the results are obtained with the same method from figure 1.

$$V_1 = 5/5, V_2 = 2/5, V_3 = 2/5, V_4 = 1/5, V_5 = 1/5, V_6 = 3/5$$

Considering *figure 1* and *2*, the normalized maximum degree is equal to the degree of each node divided by the maximum degree. For *figure 1*, the maximum number of degrees is 5. The denominator will be the corresponding degree to each node. When applied to equation 4, the results will be the following.

$$V_1 = 3/3, V_2 = 2/3, V_3 = 3/3, V_4 = 2/3, V_5 = 3/3, V_6 = 3/3$$

For *figure 2*, the results are obtained with the same method from *figure 1*.

$$V_1 = 5/5, V_2 = 2/5, V_3 = 2/5, V_4 = 1/5, V_5 = 1/5, V_6 = 3/5$$

Normalization moves the centrality result to be 0 to 1 which gives a more suitable score to each of the nodes. When this is done by dividing the results by the maximum degree, a simpler comparison will be available between different graphs.

## 2.2 Betweenness Centrality

Another way of measuring centrality is by computing the shortest number of paths between nodes. The shortest path between *s* and *t* will be chosen, and the centrality will be measured through how central the node in is connecting *s* and *t*. Between node *s* to *t*, the number of paths including node *v<sub>i</sub>* will be calculated.

$$C_b(V_i) = \sum_{s \neq t \neq v_i} \frac{\sigma_{st}(V_i)}{\sigma_{st}} \quad \text{(Equation 6)}$$

Betweenness centrality also needs to be normalized by the maximum value of the node. Equation 7 will be the denominator in equation 8 as the number of total paths. 2 is multiplied because the pathway can go both ways.

$$\sum_{s \neq t \neq v_i} 1 = 2 \binom{n-1}{2} = (n-1)(n-2) \quad \text{(Equation 7)}$$

The normalized betweenness then can be calculated by dividing the pathways by its maximum value.

This equation will become the denominator in equation 8.

$$c_b^{norm}(v_i) = \frac{c_b(v_i)}{2 \binom{n-1}{2}} \quad \text{(Equation 8)}$$

The same process will be completed for each node, so computing all nodes for their shortest paths from a single node will enable a comparison between each centrality.

### Example 2

Example betweenness centrality will be calculated using the same figures from above, and the total number of shortest paths including each node will be compiled after it is divided by the total combination of paths.

*figure 1*

$$V_1 = 2 * (0 + 0 + 0 + 0 + 0 + 1/2 + 1/2 + 0 + 0 + 0) / 20 = 0.1$$

$$V_2 = 2 * (0 + 0 + 0 + 0 + 0 + 0 + 1/2 + 0 + 0 + 0) / 20 = 0.05$$

$$V_3 = 2 * (1/2 + 1/2 + 0 + 0 + 1/1 + 0 + 0 + 0 + 0 + 0) / 20 = 0.2$$

$$V_4 = 2 * (0 + 0 + 0 + 0 + 0 + 0 + 1/2 + 0 + 0 + 0) / 20 = 0.05$$

$$V_5 = 2 * (0 + 0 + 1/2 + 0 + 0 + 0 + 0 + 0 + 0 + 1/2) / 20 = 0.1$$

$$V_6 = 2 * (0 + 0 + 1/2 + 0 + 0 + 0 + 0 + 0 + 0 + 1/2) / 20 = 0.1$$

*figure 2*

$$V_1 = 2 * (1/1 + 1/1 + 1/1 + 0 + 1/1 + 1/1 + 0 + 1/1 + 1/1) / 20 = 0.7$$

$$V_2 = 2 * (0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0) / 20 = 0$$

$$V_3 = 2 * (0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0) / 20 = 0$$

$$V_4 = 2 * (0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0) / 20 = 0$$

$$V_4 = 2 * (0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0) / 20 = 0$$

$$V_6 = 2 * (0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0) / 20 = 0$$

Compared to *figure 1*, *figure 2* has a more centralized pattern of nodes since  $v_1$  is the only node that doesn't have a centrality value of 0.

### 3.3. Closeness Centrality

Closeness centrality portrays that the degree of shortness to other nodes will measure the nodes' centrality. In other words, a node with a shorter distance (fewer edges in between) with other nodes will likely be a central node.

$$C_c(V_i) = \frac{1}{\bar{l}_{v_i}} \quad \text{(Equation 9)}$$

The denominator of the formula is the average of the smallest distances between nodes with a single node-set as a measuring standard. With one node measuring, the distance between those nodes and other nodes will be summed up and will be divided by  $n-1$ , which is to exclude the original node itself.

$$\bar{l}_{v_i} = \frac{1}{n-1} \sum_{v_j \neq v_i} l_{i,j} \quad \text{(Equation 10)}$$

#### Example 3

Example closeness centrality using the same figures from above, and the values will be obtained by adding all of the distance between nodes from a single node and will be divided by the number of nodes - 1.

*figure 1* 1

$$1/V_1 = (2+1+2+1+1)/5 = 1.4, V_1 = 0.71$$

$$1/V_2 = (2+1+2+2+1)/5 = 1.6, V_2 = 0.625$$

$$1/V_3 = (1+1+2+1+2)/5 = 1.4, V_3 = 0.71$$

$$1/V_4 = (2+2+1+1+2)/5 = 1.6, V_4 = 0.625$$

$$1/V_5 = (1+2+2+1+1)/5 = 1.4, V_5 = 0.71$$

$$1/V_6 = (1+1+2+2+1)/5 = 1.4, V_6 = 0.71$$

*figure 2* 2

$$1/V_1 = (1+1+1+1+1)/5 = 1.0, V_1 = 1$$

$$1/V_2 = (1+2+2+2+1)/5 = 1.6, V_2 = 0.625$$

$$1/V_3 = (1+2+2+2+1)/5 = 1.6, V_3 = 0.625$$

$$1/V_4 = (1+2+2+2+2)/5 = 1.8, V_4 = 0.56$$

$$1/V_5 = (1+2+2+2+2)/5 = 1.8, V_5 = 0.56$$

$$1/V_6 = (1+1+1+2+2)/5 = 1.4, V_6 = 0.71$$

For these values in figures, the actual centrality values will be the reciprocal of these above values since these values would be higher with a shorter distance. For *figure 1*, node 2 and 3 would have the greatest closeness centrality, and for *figure 2*, node 1 would have the greatest closeness centrality value since it can reach all other nodes with only passing a single pathway.

### 3.4. Eigenvector Centrality

Apart from degree centrality which considers the connections to a node to hold more significance, eigenvector centrality takes into consideration how much centrality does the nodes that are connected to the target node each have. The centrality of the target node ( $V_i$ ) will be shown through the recordings of matrix A. Since the function of its neighbors' centralities is proportional to the summation of their centralities, the following equation will be used.

$$C_e(V_i) = \frac{1}{\lambda} \sum_{j=1}^n A_{j,i} C_e(V_j) \quad \text{(Equation 11)}$$

In a particular node from matrix A, the values of  $e$  and  $j$  will be each placed into comparison measuring the node  $j$ 's significance through measuring the other two nodes. The value of  $1/\lambda$  is

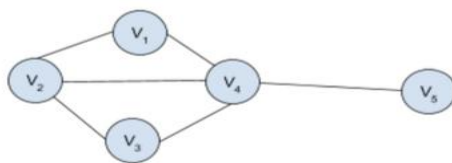
multiplied to measure when the matrix becomes 0, as the solution would be the eigenvalue of that particular value of  $\lambda$ .  $\lambda$  will be a fixed constant Assuming  $C_e = (C_e(v_1), C_e(v_2), \dots, C_e(v_n))^T$  is the centrality vectors for all nodes, the above equation can be rewritten as

$$\lambda C_e = A^T C_e \quad \text{(Equation 12)}$$

The previous equation can be changed into the following equation, as  $V_i$  and  $V_j$  will have corresponding values in the matrix, making it possible to assume that they are identical. Additionally,  $C_e$  will be the eigenvalue of the matrix  $A^T$ , as shown in the above equation.

$$(A - \lambda I) C_e = 0 \quad \text{(Equation 13)}$$

This means that  $C_e$  is an eigenvector of the adjacency matrix  $A^T$  and  $\lambda$  is the corresponding eigenvalue. Therefore, to have positive centrality values, the eigenvalues of  $A$  will be computed then selected with the largest eigenvalue. An identity matrix will be multiplied to allow comparison.



<Figure 3>

**Example 4**

The figure 3 matrix's adjacency will be the following:

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad \text{(Equation 14)}$$

While solving  $(A - \lambda I)C_e = 0$ ,

Since  $C_e$  is not equal to  $[0 \ 0 \ 0 \ 0 \ 0]^T$ , the characteristic equation is

$$C_e = [u_1 \ u_2 \ u_3 \ u_4 \ u_5]^T \quad \text{(Equation 15)}$$

Or equivalently,

$$\begin{bmatrix} 1-\lambda & 1 & 0 & 1 & 0 \\ 1 & 1-\lambda & 1 & 1 & 0 \\ 0 & 1 & 1-\lambda & 1 & 0 \\ 1 & 0 & 1 & 1-\lambda & 1 \\ 0 & 0 & 0 & 1 & 1-\lambda \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} = 0 \quad \text{(Equation 16)}$$

$$-\lambda^5 + 5\lambda^3 + 2\lambda^2 - 2\lambda = -\lambda(\lambda^4 - 5\lambda^2 - 2\lambda + 2) = -\lambda(\lambda + 1)(\lambda^3 - \lambda^2 - 4\lambda + 2) \quad \text{(Equation 17)}$$

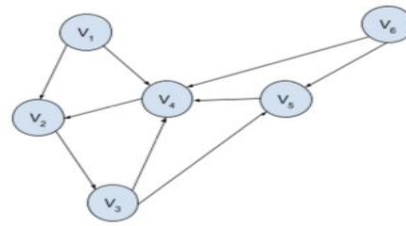
Thus,  $\lambda_1=0$ ;  $\lambda_2=-1$ ;  $\lambda_3 \approx -1.814$ ;  $\lambda_4 \approx 0.471$ ;  $\lambda_5 \approx 2.343$

These would be the possible values of  $\lambda$  and we would select  $\lambda_5$  since it has the greatest value. After that, we would be able to get the following values by placing  $\lambda$  back into the matrix.

Because the determinant of the matrix will be calculated giving a degree corresponding to the size

of the matrix. Assuming  $C_e$  has norm 1, its solutions would be

$$\begin{bmatrix} 1-\lambda & 1 & 0 & 1 & 0 \\ 1 & 1-\lambda & 1 & 1 & 0 \\ 0 & 1 & 1-\lambda & 1 & 0 \\ 1 & 0 & 1 & 1-\lambda & 1 \\ 0 & 0 & 0 & 1 & 1-\lambda \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} = 0 \quad (\text{Equation 18})$$



<Figure 4>

### Example 5

The iterations, the number of the repetitive movements, will be different for each recording and will record each probability to reach a certain node after making n times of iterations. For example, Iter 2 will present the chances of reaching a certain node after it moves a single time in the diagram. In iteration 1, since the starting point is the destination, they will all be the same probability. For iteration 2, the probability of reaching a certain node after it starts off from iteration 1 will be recorded.

Where node 2 would be considered the most centralized node, since the eigenvalue measures the significance of neighboring nodes, even though node 4 would have the highest degree centrality, node 2 that has node 4 as a neighbor would have the highest eigenvalue centrality.

$$C_e = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} = \begin{bmatrix} 2.245 \\ 2.916 \\ 2.245 \\ 2.343 \\ 1 \end{bmatrix} \quad (\text{Equation 19})$$

### 3.5. Pagerank Centrality

Pagerank Centrality is measured similarly as eigenvector centrality[5], but a controlled value is incorporated since there are situations where a node can not be ensure to have superiority over another. (In figure 3, it is unclear whether node 2 or 4 has more significance)



Iter	1:
V1.	1/6
V2.	1/6
V3.	1/6
V4.	1/6
V5.	1/6
V6.	1/6

Iter	2:
V1.	0, V6. 0
V2.	$1/6 * 1/2 + 1/6 * 1 = 0.25$

V3.  $1/6 * 1 = 1/6$

V4.  $1/6 * 1/2 + 1/6 * 1/2 + 1/6 + 1/6 * 1/2 = 0.416$

V5.  $1/6 * 1/2 + 1/6 * 1/2 = 1/6$

Iter	3:
V1.	0, V6. 0

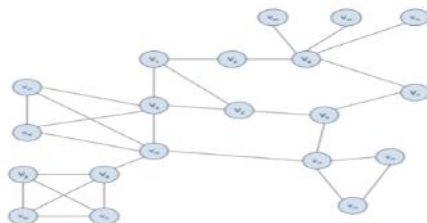
V2.  $0 * 1/2 + (0.416) * 1 = 0.416$  V3.  $0.25 * 1$

V4.  $1/6 + 1/6 * 1/2 = 0.25$  V5.  $1/6 * 1/2$

Iter			4:
V1.	0,	V6.	0
V2.			0.25
V3.			0.416
V4.	$1/8 + 1/6 * 1/2 = 0.208$		V5. 0.125
Iter			5
V1.	0,	V6.	0
V2.			0.208
V3.			0.25
V4.	$0.208 + 0.125 = 0.333$		V5. 0.208

As the iteration steps increase, what started off as equal seems to approach node number 4 as the node with the maximum degree, which is suitable from the diagram with node 4 in the center. However, in an actual diffusion network, the direction of diffusion should be understood as the opposite direction.

### 3.6. Final Rankings



<Figure 5>

	First Node	Second Node	Third Node	Fourth Node
Degree Centrality	$v_6, v_{14}$	$v_{15}, v_8$	$v_{14}$	$v_5$
Eigenvector Centrality	$v_{15}$	$v_8$	$v_{14}$	$v_{11}$
Closeness Centrality	$v_{14}$	$v_5$	$v_{15}$	$v_{17}$
Betweenness Centrality	$v_{14}$	$v_{15}$	$v_5$	$v_6$
Katz Centrality	$v_{14}$	$v_8$	$v_{15}$	$v_8$

<Table 1>

This table represents the centrality ranking of each node among the 20 nodes from figure 5. In the table, nodes 14 and 15 both rank in the first three for all centrality measurements, which proves that they are significant to a certain degree. There are also other nodes that show up in the first three ranks, but none of them show a constant appearance throughout the table.

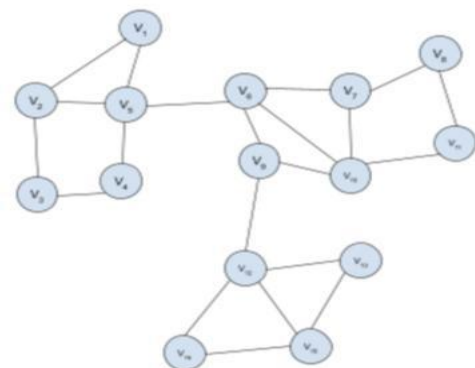
## 4. BRIDGE DETECTION

Since community detection takes too much time, bridge detection allows the division of the groups, which will then lead to a smaller group with faster detection of communities. Inside these communities, node centrality will be calculated for each node finding each of their significance.[3]

Bridges are certain edges that connect two communities in a diffusion network. If there are two big communities and there is a bridge in between, diffusion won't happen from one community to the other without these edges(bridges). The algorithm used in the bridge detection is the following:

For visit and low order, the parent's visit order and the node with the smallest visit order while connected to the currently visiting node will be compared to be validated. If there are no more nodes to visit, then the direction will return to the parent node. If the visiting node is not connected with the parent but with a different visited node, the low order will be altered to the visiting node of the visited node. If the node has to return to its parent node, there is a possibility that the edge between them is a bridge, so we validate by comparing the low order of the current node to the visit order of the parent node. If the validating node is not a part of a bridge, the re-visited parent node's lower order will be changed to the current node's low order. Repeat the above process until the pattern revisits the node with the visit order of 1.

### Example 6



<figure 6>

Visit order	node	Visit order	node
1	v1	8	v8
2	v2	9	v11
3	v3	10	v10
4	v4	11	v9
5	v5	12	v12
6	v6	13	v13
7	v7	14	v15
		15	v14

<Table 2>

Current node	Visit order	Low order	Validate	Update
v1	1	1		
...				
v5	5	1		
...				
v10	10	6		
v9	11	6		
...				
v15	14	12		
v14	15	12		
v15	14	12	Since the child node's low order (12) is smaller than the visit order of the current node (14), the edge between the two is not a bridge.	

v13	13	12	Since the child node's low order (12) is smaller than the visit order of the current node (13), the edge between the two is not a bridge.	
v12	12	12		
v9	10	6	Since the child node's low order (12) is greater than the visit order of the current node (10), the edge between the two is a bridge.	
v10	9	6	Since the child node's low order (6) is smaller than the visit order of the current node (9), the edge between the two is not a bridge.	
v11	9	9-->6	Since the child node's low order (6) is smaller than the visit order of the current node (9), the edge between the two is not a bridge.	Since the nodes v11 and v10 are not part of a bridge, the low value of v11 will be replaced with 6 from the low value of v10.
v8	8	8-->6	Since the child node's low order (6) is smaller	Since the nodes v11 and v8 are not part of a

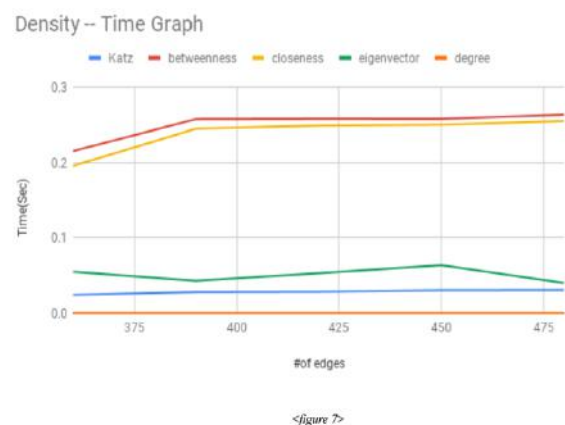
			than the visit order of the current node (8), the edge between the two is not a bridge.	bridge, the low value of v11 will be replaced with 6 from the low value of v11.
v6	6	6		
v5	5	1	Since the child node's low order (6) is greater than the visit order of the current node (1), the edge between the two is a bridge.	
v4	4	4-->1	Since the child node's low order (1) is smaller than the visit order of the current node (4), the edge between the two is not a bridge.	Since the edge between v4 and v5 is not a bridge, the low order of v4 will be updated from 4 to 1.
v3	3	3	Since the child node's low order (1) is smaller than the visit order of the current node (3), the edge between the two is not a bridge.	Since the edge between v4 and v3 is not a bridge, the low order of v3 will be updated from 3 to 1.
v2	2	2	Since the child node's low order (1) is smaller than the visit order of the current node (2), the edge between the two is not a	Since the edge between v3 and v2 is not a bridge, the low order of v2 will be updated from 2 to 1.

			bridge.	
v1	1	1	Since the child node's low order (1) is equal to the visit order of the current node (1), the edge between the two is not a bridge.	

<Table 3>

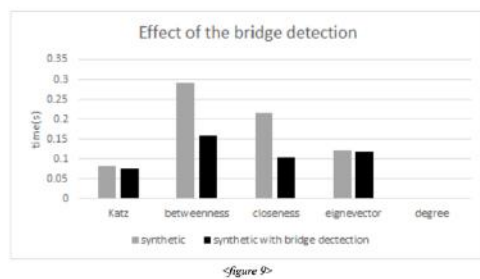
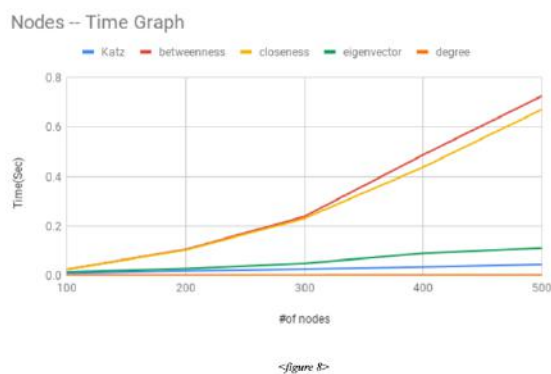
### 5. EVALUATION

The following is the environment where the measurement calculations were completed. The processor used was 3.1 GHz Intel Core i5 and the memory was 8 GB 2133 MHz LPDDR3. The experiments were conducted through one measure with the same ratio(1.3) between the nodes and edges ( $|V|=[100, 200, 300, 400, 500], |E|=[130, 260, 390, 520, 650]$ ). Another with a sole increase on the edges, the sole increase in the edges was to find the appropriate number of node to edge ratio, where 1.3 was the most adequate ratio for experiment. The last experiment was the comparison between measurement time without the bridge and the bridge, where I measured the effectiveness of figuring out the bridge when measuring a centrality.



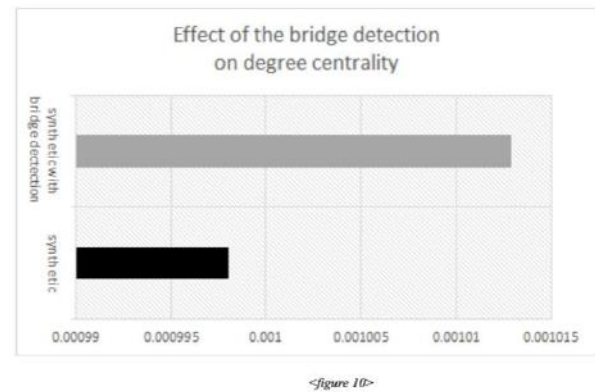
When the number of nodes is 300 and the number of edges is 360 to 480, there is a continuous increase except for when the ratio between the nodes and edges is 1.3 (figure 7). The methods will return their answers in a constant time except for specific ratios.

When changing the node, we chose the ratio of 1.3 from the previous graph. When the nodes increase, closeness and degree centralities have an exponential growth, while katz, eigenvector and betweenness centrality has a linear growth (figure 8). Considering equation 6, betweenness centrality considers  $n-1$  nodes for a single node, and since there are  $n$  nodes, the time complexity becomes  $O(n^2)$ . Similarly, closeness centrality has the time complexity equal to betweenness centrality since it is the reciprocal of the values from equation 6. Degree centrality should have a linear growth, but the computing power alters it. For Eigenvector and Katz centrality, since the calculation would involve the matrix calculation with  $n$  equations, the resulting time complexity will be  $O(n)$ .



The experiment was conducted using a synthetic graph as shown in figure 9, a graph created for referencing the effecting of bridge detection with 800 nodes and 130 edges. Since there is a higher

value of node compared to edges, the edges can easily be bridges making betweenness and closeness centrality highly effective with bridge detection. The time complexity that is originally  $O(n^2)$  would be decreased by  $O(n^2/\# \text{ of bridges}^2)$ . However, for Katz and Eigenvector centrality, the time complexity that is  $n$  will not decrease so much and will cancel out with the time taken from bridge detection.



The As in degree centrality, the time complexity  $O(n)$  wouldn't decrease much from bridge detection, and rather the time complexity will increase from the time it takes from bridge detection.

## 6. CONCLUSION

In a diffusion network, one might assume that eigenvector centrality is more effective, but degree centrality more effectively shows the "important" node. The most significant node in the one who has the most influence by itself. Degree centrality will be hard to implement in real-life situations, since bridge detection wouldn't decrease the time it takes to measure the centrality.

## 7. FUTURE WORKS

For future work, the calculation time can be prolonged even with the use of bridge detection. In this sense, it would be effective to also incorporate community detection with bridge detection, searching the community in the pre-created sub-graphs. Since community detection has multiple methods, there would be methods that maximize the efficiency and accuracy of measurement in each

graph. This will be a good topic for extra research later on.

## 8. REFERENCES

1. X. Zhao, S. Guo, and Y. Wang, "The Node Influence Analysis in Social Networks Based on Structural Holes and Degree Centrality," in IEEE International Conference on CSE and IEEE International Conference on EUC, 2017, pp. 708-711.
2. Freeman, L.C., "A set of measures of centrality based on betweenness", *Sociometry*, Vol. 40, pages 35-41, 1977.
3. Akram, V.K.; Dagdeviren, O. "Breadth-First Search-Based Single-Phase Algorithms for Bridge Detection in Wireless Sensor Networks." in *Sensors* 2013, pp. 8786-8813.
4. F. Cadini, E. Zio, and C. Petrescu, "Using centrality measures to rank the importance of the components of a complex network infrastructure," in *CRITIS*, 2008, pp. 155-167.
5. M.E.J. Newman, "A measure of betweenness centrality based on random walks," in *Soc. Netw.* 27 (1), 2003, pp. 39- 54.
6. P. Bonacich, "Power and centrality: a family of measures," in *Am. J. Sociol.* 92 (5), 1987, pp. 1170-1182.

*urnals*