

Estimating the Expected Influence Capacities of Nodes in Complex Networks under the Susceptible-Infectious-Recovered (SIR) Model

Aybike ŞİMŞEK^{a,1}

^a Düzce University Department of Computer Engineering, Faculty of Engineering, Düzce
81620, Turkey

Email: aybikesimsek@duzce.edu.tr

Abstract

In recent years, epidemic modeling in complex networks has found many applications, including modeling of information or gossip spread in online social networks, modeling of malware spread in communication networks, and the most recent model of the COVID-19 pandemic. If the information disseminated is accurate, for example, maximizing its distribution is desirable, whereas if it is a rumor or a virus, its spread should be minimized. In this context, it is very important to identify super-spreaders that maximize or minimize propagation. Lately, studies for detecting super-spreaders have gained momentum. Most of the studies carried out aim to distinguish the influences of nodes under a specific propagation model (such as SIR) using network centrality measures and subsequently, to rank the nodes accordingly. However, in this study, we developed an algorithm that approximates the expected influence of nodes under the popular SIR model. By considering the behavior of the SIR model and only the shortest paths between nodes, the algorithm ranks the nodes according to this approximated value. Our developed algorithm is named the Expected Value Estimation (EVE). We compared the performance of EVE, using different SIR settings on real datasets, with that of many current well-known centrality measures. The experimental studies demonstrated that the solution quality (ranking capability) of EVE is superior to that of its competitors.

Keywords: Complex networks; Susceptible-Infectious-Recovered model; Epidemic modeling; Expected influence.

1. INTRODUCTION

Complex networks are highly suitable tools for modeling the real world. They have applications in many different fields such as natural sciences [1], health [2], cyber security [3], economics [4], and social networks [5]–[7]. Moreover, epidemic modeling in complex networks has attracted attention in recent years for its many practical benefits. The spread of a virus outbreak (such as Covid-19) can be estimated and precautions can be taken based on this [8]. By modeling the spread of gossip on the social network, the spread can be prevented [9], [10]. Or, the desired information may reach the maximum number of people [11]. Whether you want to minimize the spread of gossip or maximize the spread of information, in any case, in order to do so, the set having the smallest number of the most influential individuals should be identified

¹ Corresponding author

[12], [13]. The influences of these individuals under certain epidemic models (such as SIR) should be calculated in order to identify the smallest number of the most influential individuals (i.e., key players). For this, it is necessary to model the propagation by selecting each node individually as the seed. Since propagation models are stochastic models, they must be repeated many times (e.g., about 10,000 iterations) and the average value taken. This operation requires very high processing power. On the other hand, researchers have noticed a correlation between the influence capacity of the nodes and network centrality measures, which have been used for a long time to determine the importance of nodes in complex networks. The basic expectation here is that as a centrality measure increases, the influence capacity increases, and as the centrality measure decreases, the influence capacity decreases. Since the calculation of centrality measures requires much less processing power than modeling the propagation thousands of times, studies have turned to this area. For this purpose, basic centrality measures such as Degree, Closeness, Betweenness [14], Katz [15], PageRank [16] were used and new centrality measures were developed. However, many of the measures developed only considered the local and global impacts of the nodes [17]–[21] or network communities [18], [19], [22]–[24]. Recently, another approach has been adopted that combines multiple centrality measures to develop new hybrid centrality measures [25]–[32]. However, many of these studies ignore the dynamics of the propagation model.

In this study, we developed an algorithm that ranks nodes according to their influence capacity, taking into account the propagation behavior in the Susceptible-Infectious-Recovered (SIR) model. We named our developed algorithm the Expected Value Estimation (EVE) because it is based on approximating the expected influence of each node. It is worth mentioning here that the EVE algorithm does not calculate the importance of nodes contrary to the centrality measures. Instead, it calculates the approximate expected influence of the nodes under the SIR model and ranks the nodes accordingly.

Motivation

Under certain epidemic models (such as SIR), it is necessary to perform heavy Monte-Carlo simulations to distinguish the influence of nodes. However, if the dynamics of the SIR propagation model are taken into account, the process can be simplified by ignoring some of the behaviors of this model. Thus, the approximate expected influence of nodes can be calculated and used to rank nodes (similar to a centrality measures). Generally speaking, in the SIR model, a node affects its neighbor nodes with a probability β . If not its direct neighbor, it is likely to affect its neighbors' neighbors with probability $(\beta \times \beta)$. If the network is a tree, the probability of a node influencing another l -hop away node can be calculated as β^l since there can be only one path between each pair of nodes. Thus, the expected influence of a node can be calculated using its distance to all other reachable nodes by this node as the sum of β^l values. However, real networks rarely exhibit tree structures. Hence, there can be many different paths of different lengths between any two nodes. It is also costly to use all paths to all other nodes to calculate the expected influence of a node. However, the probability of one node influencing another node decreases exponentially with the distance between them, although in practice, the value of β is much less than 1. The natural consequence of this is $\beta^n \gg \beta^{n+1}$, where $n \in \mathbb{N}^+$. Based on this information, the expected probability of a node influencing another node can only

be approximated using the shortest path between these two nodes. This is because the probability of influence calculated for routes other than the shortest path will be much lower. These calculated values can be used to distinguish the influence capacities of the nodes (similar to a centrality measure).

2. PRELIMINARIES

Before discussing the details of EVE, it would be useful to give some preliminary information.

Let $G = (V, E)$ be an undirected unweighted graph (network). Here, V is the set of nodes (vertices), and E is the set of edges (links).

Definition 1 (*Susceptible-Infectious-Recovered Model*): The Susceptible-Infectious-Recovered (SIR) model is a well-known model used for population-based epidemic modeling. In recent years, due to their popularity, SIR and SIR variations have been applied to network topologies [33]. In the SIR model, nodes are found in one of three states: Susceptible, Infected, and Recovered. The transition of nodes between states occurs according to certain probabilities. Susceptible nodes are more likely to be infected by neighbors who are already infected with probability β . Infected nodes are also likely to go into a recovered state with probability γ . Initially, all other nodes are in a susceptible state, except for nodes that carry the disease (i.e., those that are infected). Starting from the nodes that are initially infected (called ‘seed nodes’), the disease spreads over the network. After a certain period of time, there are no remaining infected nodes on the network and thus, the model is terminated.

Definition 2 (*Kendall’s tau Ranking Correlation Coefficient*) [34]: Let (a_i, b_i) and (a_j, b_j) be tuples of joint A and B ranking lists. If $a_i > a_j$ and $b_i > b_j$ or $a_i < a_j$ and $b_i < b_j$, then the tuples are concordant. If $a_i > a_j$ and $b_i < b_j$ or $a_i < a_j$ and $b_i > b_j$, then the tuples are discordant. If $a_i = a_j$ or $b_i = b_j$, then the tuples are neither concordant nor discordant. Finally, τ is defined as in Equation (1).

$$\tau = \frac{N_c - N_d}{0.5N(N - 1)} \quad (1)$$

Here, N_c is the number of concordant pairs, N_d is the number of discordant pairs, and N is the number of all combinations. Positive τ values indicate a positive correlation, and negative τ values indicate a negative correlation.

Definition 3 (*Ranking Monotonicity*) [35]: Monotony is a metric of how well the centrality measure assigns each node to different rank levels. The ranking monotonicity (RM) will be ‘1’ if all nodes are assigned to a different ranking level. If all nodes are assigned to the same ranking level, the RM will be ‘0’. Of course, for a centrality measure, the closer it is to RM 1, the better. The RM is calculated as follows:

$$RM(L) = \left(1 - \frac{\sum_{r \in L} n_r(n_r - 1)}{n(n-1)}\right)^2 \quad (2)$$

Here, n is the length of the L-ranking list and n_r is the number of elements assigned to the same r rank.

3. EVE

The working principle of EVE is based on expected value calculation. Therefore, it is useful to first look into the details of how a node infects its neighbor nodes in SIR and how this node recovers. This situation is shown for one iteration in Algorithm 1 [36]. The node u in the algorithm was initially selected as the infected node or one infected at any point in time.

Algorithm 1. Infection and Recover States of SIR

```

1 sn = susceptible neighbors of node u
2 for each v in sn
3     rnd = random number in [0.0,1.0)
4     if rnd <  $\beta$  then
5         mark v as infected
6 end for
7 rnd = random number in [0.0,1.0)
8 if rnd <  $\gamma$  then
9     mark u as recovered

```

According to Algorithm 1, the node u infects its neighbors with probability β . After the node u infects its neighbors, this node is recovered with probability γ . If $\gamma = 1$, the node u has absolutely only one attempt to infect its neighbors since it will not be in the Infected state in the next iteration. If $\gamma = 0.5$, roughly speaking, the node u has two attempts to infect its neighbors since it will be in the Infected state in the next iteration with probability 0.5. If we generalize, the node u has at least $1/\gamma$ attempts to infect its neighbors. Since the probability of the node u infecting its neighbors is β , the expected value of infecting a neighbor by node u would be $1/\gamma$ times β ; that is, β/γ .

Let us explain the situation in Figure 1, where different topologies are shown. Notice that Figure 1 a, b, and c are trees. Therefore, there is only one path between all nodes.

In Figure 1-a, let the node u initially be selected as a seed (infected). The expected influence value (ev) of the node u becomes $ev(u) = 1 + \beta/\gamma$. Here, 1 has been added as node u is already infected.

Figure 1-b shows the expected influence value (ev) of the node as $ev(u) = 1 + \beta/\gamma +$ (probability of u infecting y). In order to infect the node y , the node u must infect the node x . Next, the node x must infect the node y . The probability of these two events happening together can be obtained by multiplying the probabilities of their respective occurrence. Thus, the expected value of u infecting the node y is $(\beta/\gamma \times \beta/\gamma)$, i.e., $(\beta/\gamma)^2$. Thus, the expected influence value (ev) of the node u becomes $ev(u) = 1 + \beta/\gamma + (\beta/\gamma)^2$.

For Figure 1-c, the expected influence value (ev) of the node u is $ev(u) = 1 + 2 \times \left(\frac{\beta}{\gamma}\right) + 2 \times \left(\frac{\beta}{\gamma}\right)^2$.

The expected value of a node infecting another node decreases exponentially with the distance between them. If we generalize the ev calculation, we get Equation (3).

$$ev(u) = 1 + nn_1 \times \left(\frac{\beta}{\gamma}\right) + nn_2 \times \left(\frac{\beta}{\gamma}\right)^2 + \dots + nn_h \times \left(\frac{\beta}{\gamma}\right)^h \quad (3)$$

Here, nn is the size of the set of node u's neighbors at h-hop distance. The situation is a little different in Figure 1-d. The node y is both a 1-hop and a 2-hop neighbor of the node u. Therefore, the node u can infect the node y directly, as well as through the node x. Thus, the expected value of node u infecting the node y is the sum of these two possibilities, or 1 at most.

Ultimately, the expected influence of the node u becomes $ev(u) = 1 + \left(\frac{\beta}{\gamma}\right) + \max\left\{1, \left(\frac{\beta}{\gamma}\right) + \left(\frac{\beta}{\gamma}\right)^2\right\}$. Let us explain why we use the *max* function here. For example, if $\frac{\beta}{\gamma} = 1$, the expected value of node u infecting the node y would be 2. However, this value can be at most 1, since once a node is infected, it cannot be infected again.

In large and complex networks, there can be many different paths having different lengths from one node to another. It is quite costly to consider all paths. Instead, only the shortest paths can be considered. Thus, as in Figure 1-e, the (x, y) edge is ignored and the approximate ev can be calculated using Equation (3). However, instead of changing the structure of the graph, only neighbors with h-shortest path hop distance can be included when creating nn_h sets. Thus, it is guaranteed that $nn_a \cap nn_b = \emptyset$; here $a \neq b$ ve $a, b \in \{1 \dots h\}$. If we named as spn_h to the sets created by selecting only neighbors with h-shortest path hop distance, we can calculate the measure we call EVE as in Equation (4).

$$EVE(u) = 1 + spn_1 \times \left(\frac{\beta}{\gamma}\right) + spn_2 \times \left(\frac{\beta}{\gamma}\right)^2 + \dots + spn_h \times \left(\frac{\beta}{\gamma}\right)^h \quad (4)$$

Equation (4) does not take into account paths other than the shortest paths. In the literature, β is usually taken as very small (e.g., ≤ 0.1) and γ as large (e.g., $= 1$). The corollary of this is $\left(\frac{\beta}{\gamma}\right)^l \gg \left(\frac{\beta}{\gamma}\right)^{l+1}$, where $l \in \mathbb{N}^+$. Thus, it can be considered reasonable to ignore paths other than the shortest paths.

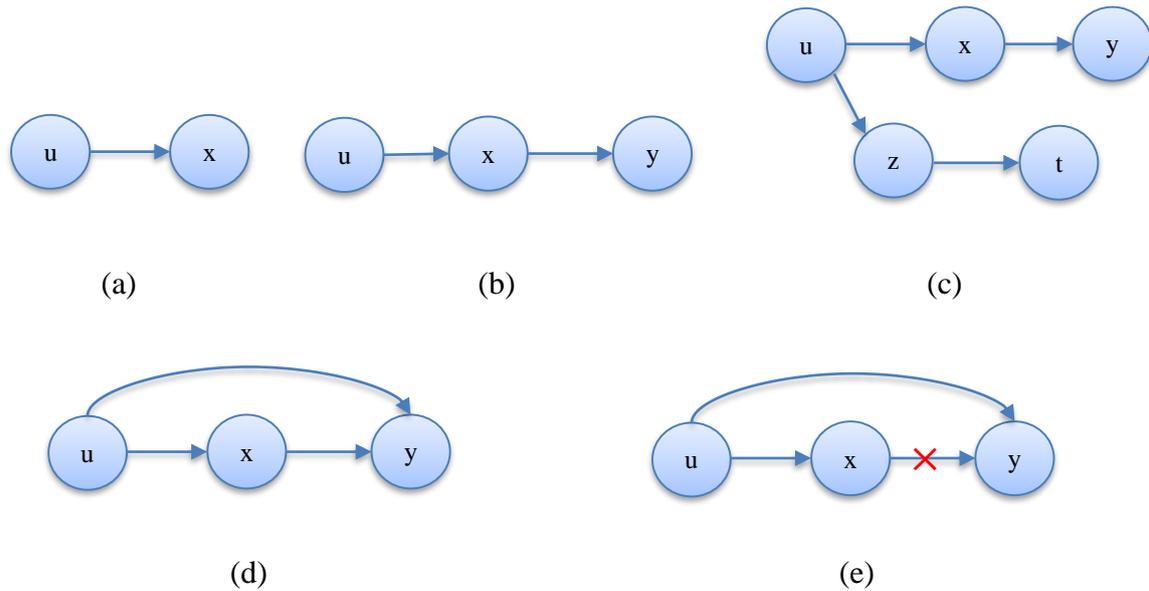


Figure 1. Sample graphs for EVE calculation: (a), (b), (c) every node belongs to only one h-hop neighborhood, (d), (e) node-y belongs to different h-hop neighborhoods.

In practice, EVE can be calculated as in Algorithm 2.

Algorithm 2. EVE

FunctionEVE(G: Graph, β , γ)

Begin

$L = \{ \}$ // L is a (key,value) dictionary as $L[node]=EVE$

SP = dictionary of all pairs shortest path of G.

//dictionary[source,destination]=length.

// If there is at least one path between two nodes then $SP[node,node]$ is a number.

// Otherwise, it is ∞ .

$V = G$'s set of nodes

for each u **in** V

EVE = 0

for each v **in** V

if $SP[u,v] \neq \infty$ **then**

EVE = EVE + **Power**(β/γ , $SP[u,v]$)

$L[node]=EVE$

Sort L descending order by value

return key list of L

End

The Sort function in Algorithm 2 sorts the dictionary entries according to their values in descending order. The Power function takes two parameters such as x and y and returns the value x^y . As a result, Function EVE returns the list of nodes sorted in descending order according to their EVE values.

Let us calculate the time complexity of EVE.

The EVE needs the shortest path information for all pairs. If the Floyd–Warshall algorithm is used, its time complexity is $O(|V|^3)$. The algorithm has two nested “for” loops and each works

exactly with step $|V|$. So, the time complexity is $O(|V|^2)$. Finally, the resulting list is sorted. If an algorithm with a time complexity of $n \log n$ is used for this, the time complexity will be $O(|V| \log |V|)$. As each these processes must follow one another, the time complexity is $O(|V|^3 + |V|^2 + |V| \log |V|)$, that is, $O(|V|^3)$. As a result, the time complexity of EVE is dominated by the shortest path calculation.

4. EXPERIMENTS

To evaluate the performance of EVE, we determined five competitor centrality measures and experimented with different SIR settings over four real-world datasets. First, let us look at the competing centrality measures and datasets.

4.1. Centrality measures

DC (*Degree Centrality*) is calculated by dividing the degree of the node by the total number of nodes in the graph minus one [37].

EC (*Eigenvector Centrality*) is used to determine the importance of a node in the network. The basic logic of EC is that the more adjacent a node is to the important nodes, the more important it is [38].

CC (*Closeness Centrality*) is a measure of how close a node is to other nodes [39]. The closer the node is to other nodes, the larger the CC.

BC (*Betweenness Centrality*) is the proportional information on how many of the shortest paths between all pairs are through a node [14].

GC (*Gravitational Centrality*) is a recent centrality measure inspired by Newton's gravitational formula [29]. Instead of the mass in the original formula, it uses the k-shell values of the nodes and instead of the distance, it uses the length of the shortest path between nodes. Its formula is as follows:

$$GC_i = \frac{ks_i \times ks_j}{\sum_{j \in N} d(j, i)} \quad (5)$$

Here, $d(\cdot)$ is the length of the shortest path between nodes i and j ; N is the set of 3-hop neighbors of node i .

The GC was chosen as a competitor because it is a recent centrality measure that gives successful results. It is also similar to EVE because it is calculated using the shortest path length between nodes.

4.2. Datasets

We used four real-world networks for the experiments. We experimented with each network for three different β and γ values: $\beta = 0.1$ and $\gamma = 1$, $\beta = 0.05$ and $\gamma = 1$, and $\beta = 0.05$ and $\gamma = 0.25$. Thus, we experimented with a total of 12 different situations. The properties of the networks are given in Table 1.

Karate: This network consists of 34 nodes and 78 edges. The nodes denote members of the karate club, and the edges denote the friendship between them [40]. This dataset is taken from <http://konect.cc/networks/ucidata-zachary>.

Email-Enron: This network consists of 143 nodes and 623 edges [41]. This dataset is taken from <http://networkrepository.com>.

Email-Univ: This network consists of 1133 nodes and 5451 edges [42]. This dataset is taken from <http://konect.cc/networks/arenas-email>.

CS-PhD: This network consists of 1882 nodes and 1740 edges [43]. This dataset is taken from <http://networkrepository.com>.

Table 1. Network dataset features

Dataset	 V 	 E 	$\langle K \rangle$	K_{max}	Density
<i>Karate</i>	34	78	4.588	17	0.1390374
<i>Email-Enron</i>	143	623	8	42	0.0613612
<i>Email-Univ</i>	1133	5451	9.62	71	0.0085002
<i>CS-PhD</i>	1882	1740	1.849	46	0.0009830

4.3. Comparison of Performances of the Measures

We evaluated the performance of EVE and the competitor centrality measures from different angles. First, we looked at the Kendall ranking performances. Next, we evaluated the "rank index vs. SIR score" graphics created by the measures. We then compared their Monotonicity performances. Finally, we looked at how many of the nodes in the top 5% of the ranking lists created by the measures corresponded to the ranking lists created according to the SIR simulations.

In the SIR simulations, we set each node as the only infected node in the network. We ended the simulations when there were no infected nodes left in the network. At the end of each simulation, we took the number of recovered nodes in the network as the influence of the node selected as the single infected node at the beginning of that simulation. We repeated the simulation for each node 1000 times and took the average of their influences as the final SIR score. For the simulations we used Python and NetworkX [44].

Kendall ranking

The ranking performances of EVE and the competitor centrality measures are shown in Figures 2-4. Ranking performances were calculated using Definition 2, as the Kendall's tau ranking correlation coefficient. The ranking list created by the measure and the list created by SIR

simulations were used in the calculation. We were inspired by [17], [31] to use this type of graphic to compare the methods.

The best results were given by EVE in six experiments, by GC in four experiments, and by EC in two experiments. In addition, the EVE τ values in all experiments are very close to 0.8 or higher.

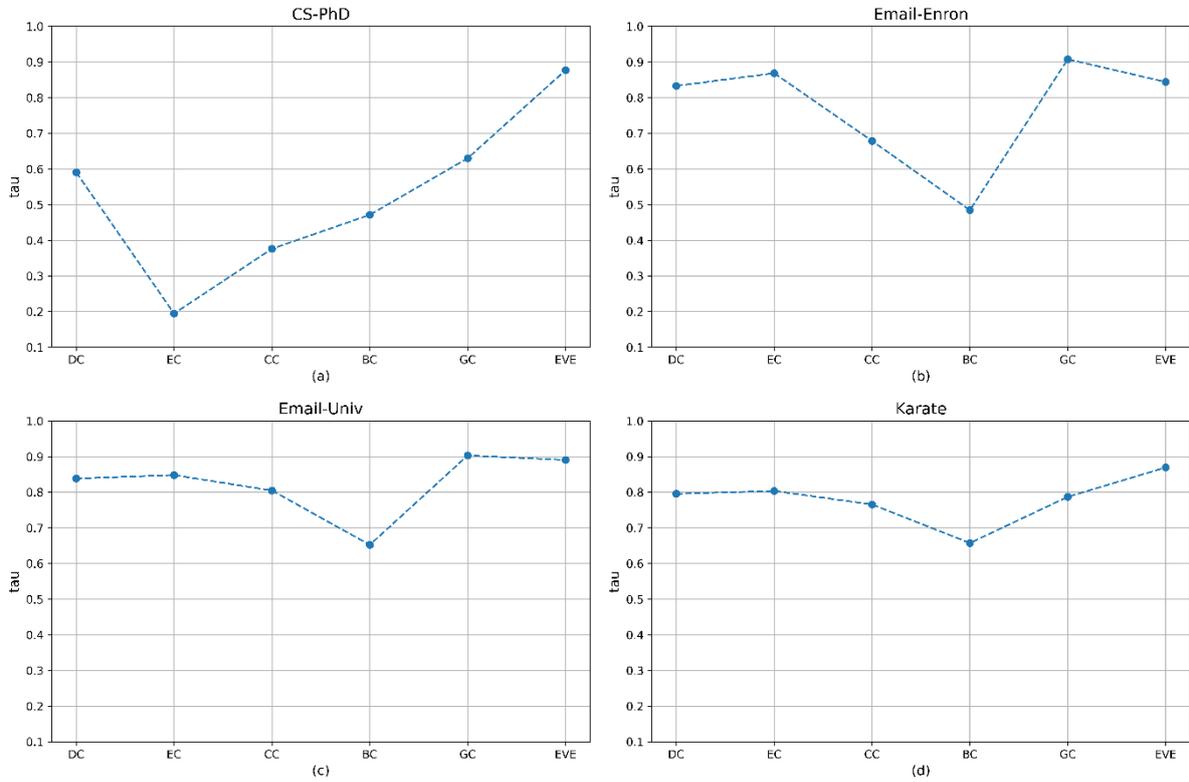


Fig. 1. Kendall's τ correlation coefficient values of different centrality measures and EVE. Infection rate: $\beta = 0.1$ and Recovery rate: $\gamma = 1$ for all experiments.

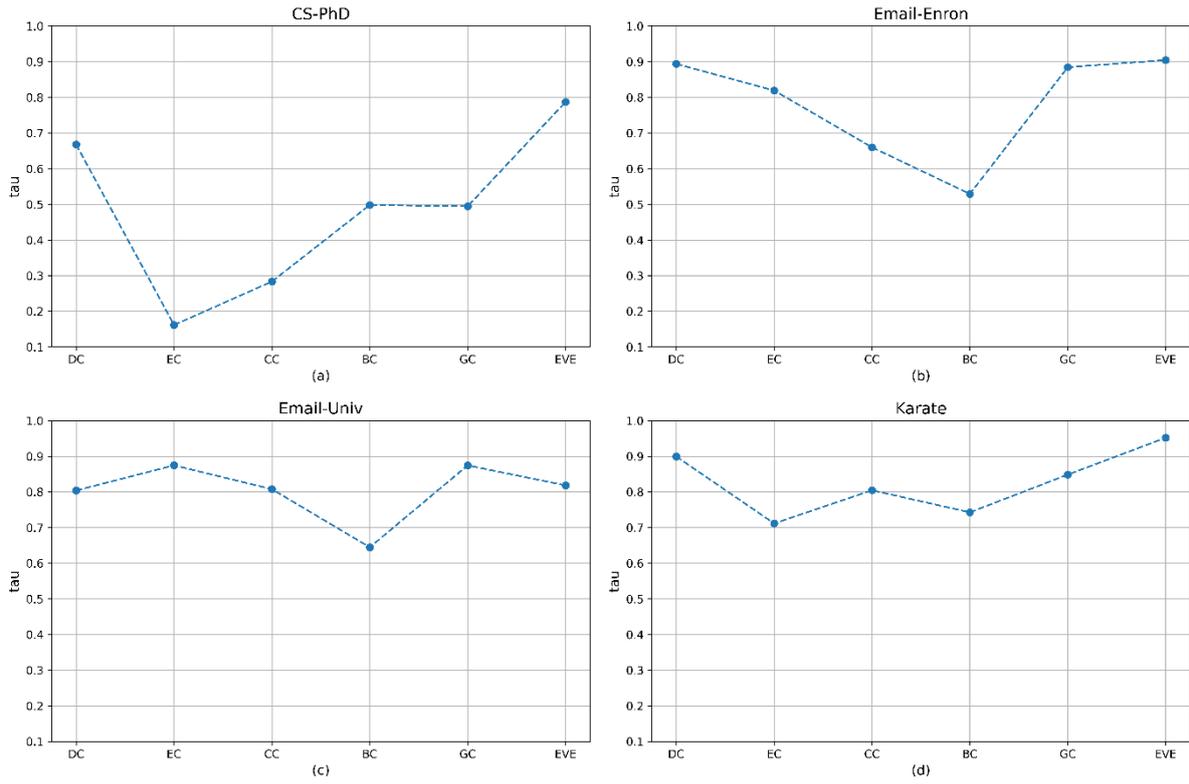


Fig. 3. Kendall's τ correlation coefficient values of different centrality measures and EVE. Infection rate: $\beta = 0.05$ and Recovery rate: $\gamma = 1$ for all experiments.

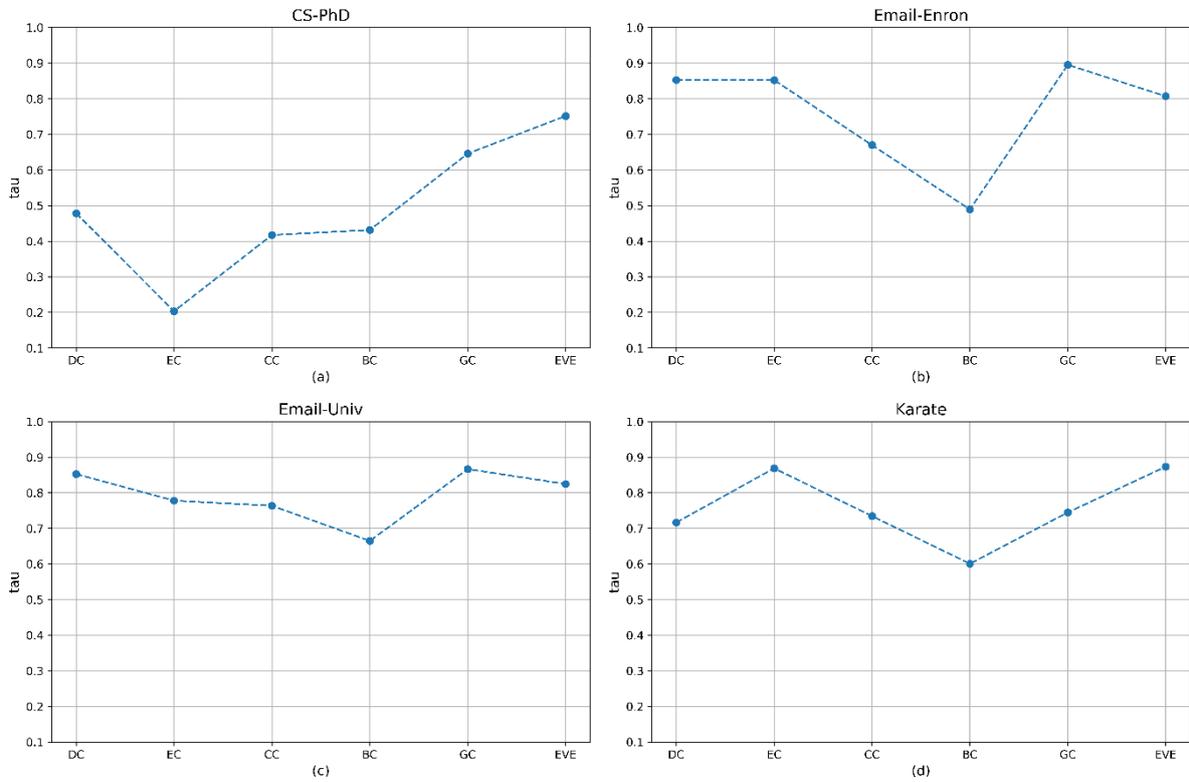


Fig. 4. Kendall's τ correlation coefficient values of different centrality measures and EVE. Infection rate: $\beta = 0.05$ and Recovery rate: $\gamma = 0.25$ for all experiments.

Rank index vs. SIR score

The graphics of the ranking indices created by the measures vs. the SIR scores are shown in Figures 5-7. As the index increases (i.e., as the centrality decreases), the SIR score is expected to decrease. This is an indication that nodes have been assigned the correct rank level. We were inspired by [17], [31] to use this type of graphic to compare the methods. In most experiments, it can be said that EVE created a more uniformly decreasing graphic than the other methods.

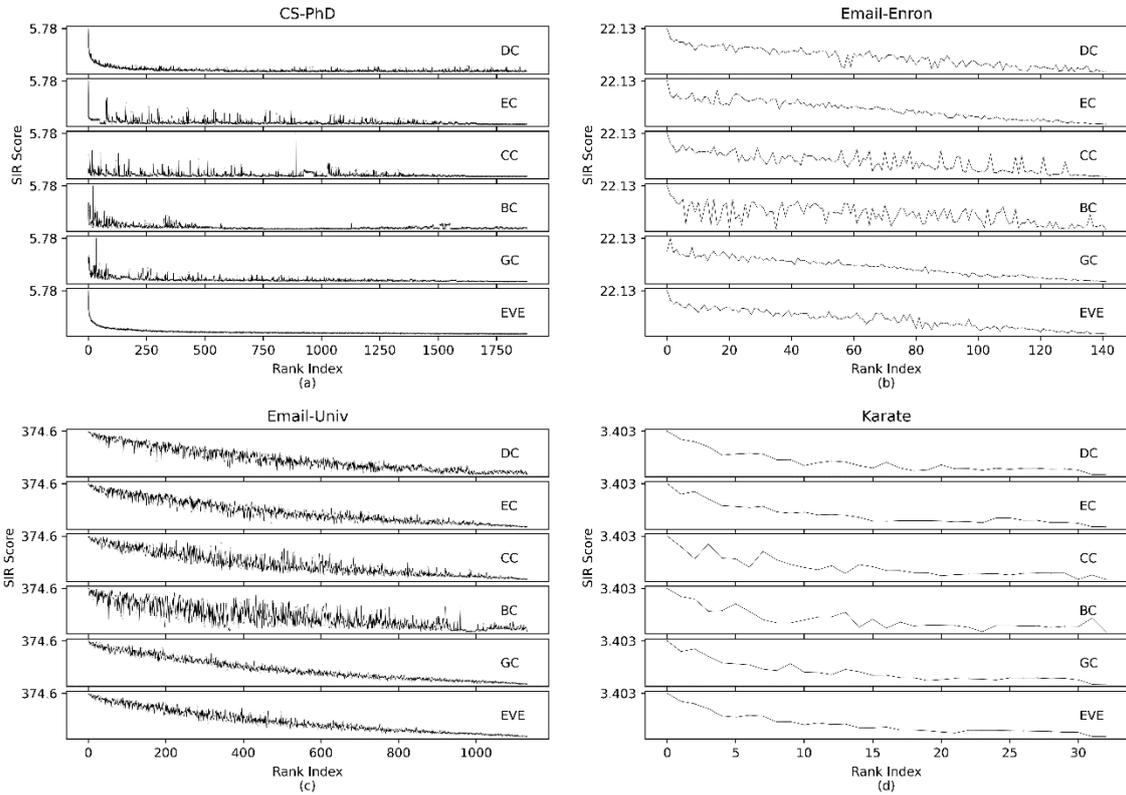


Fig. 5. SIR score trends of nodes ranked from large to small according to different centrality measures and EVE. Infection rate: $\beta = 0.1$ and Recovery rate: $\gamma = 1$ for all experiments.

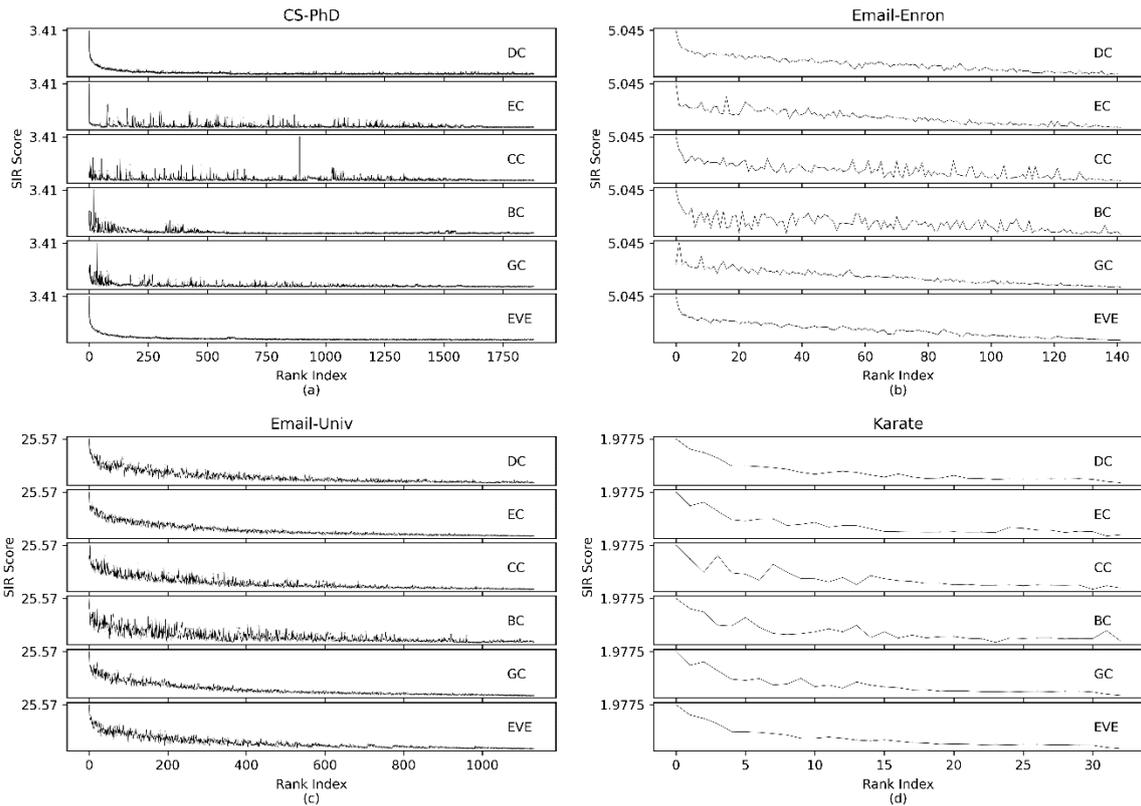


Fig. 6. SIR score trends of nodes ranked from large to small according to different centrality measures and EVE. Infection rate: $\beta = 0.05$ and Recovery rate: $\gamma = 1$ for all experiments.

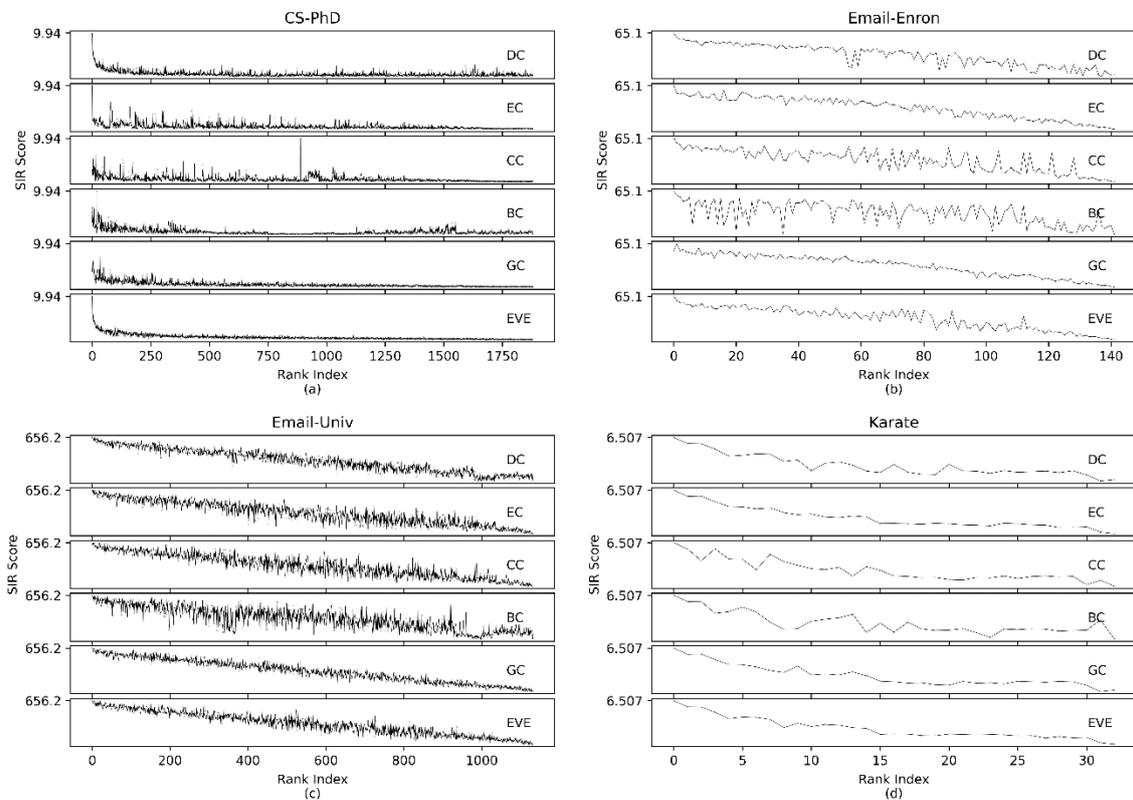


Fig. 7. SIR score trends of nodes ranked from large to small according to different centrality measures and EVE. Infection rate: $\beta = 0.05$ and Recovery rate: $\gamma = 0.25$ for all experiments.

Ranking Monotonicity

The monotonicity values of the ranking lists created by EVE and the competitor centrality measures are shown in Tables 2. The values were calculated using Definition 3. Since the ranking lists produced by the centrality measures depends only on the network structure, their monotonicity values were calculated only once for each data set. Since the ranking list produced by EVE is dependent on SIR, its monotony values were calculated separately for different SIR settings. The monotonicity values calculated for EVE were 1 in seven experiments and very close to 1 in the other experiment. This means that EVE assigns a different rank to all nodes in seven experiments. Meanwhile, the EC, CC, and GC also yielded successful results.

Table 2. Monotonicity values of the measures.

	DC	EC	CC	BC	GC	EVE		
						$\beta = 0.1$ $\gamma = 1$	$\beta = 0.05$ $\gamma = 1$	$\beta = 0.05$ $\gamma = 0.25$
CS-PhD	0.5223	0.9938	0.9934	0.5718	0.9935	0.9939	0.9983	0.9953
Email-Enron	0.8910	1	1	0.9944	1	1	1	1
Email-Univ	0.9073	1	0.9999	0.9778	1	1	1	1
Karate	0.8025	0.9439	0.9220	0.8682	1	1	0.9439	0.9886

Finally, we examined how many of the nodes in the top 5% of the ranking lists created by the measure coincided with the nodes in the top 5% of the ranking list created according to the SIR simulations. The results are shown in Tables 3-5. Nodes in the top rank levels formed by the measure are expected to be more influential nodes. Therefore, it is important that the nodes at the top of the list and those at the top of the ranking list created according to the SIR simulations are the same. We were inspired by [31] to use this type of graphic to compare the methods. According to the calculated values, EVE gave the best results in eight experiments.

Table 3. Number of overlapping nodes in the top 5% of the ranking list created according to SIR simulations with the ranking lists created by the measures.

$\beta = 0.1$ $\gamma = 1$	DC	EC	CC	BC	GC	EVE
CS-PhD	76	16	28	46	54	88
Email-Enron	5	3	4	4	4	5
Email-Univ	41	32	38	37	40	43
Karate	1	1	1	1	1	1

Table 4. Number of overlapping nodes in the top 5% of the ranking list created according to SIR simulations with the ranking lists created by the measures.

$\beta = 0.05$ $\gamma = 1$	DC	EC	CC	BC	GC	EVE
CS-PhD	86	8	25	42	50	89
Email-Enron	7	3	4	4	4	6
Email-Univ	38	45	37	32	44	39
Karate	1	1	1	1	1	1

Table 5. Number of overlapping nodes in the top 5% of the ranking list created according to SIR simulations with the ranking lists created by the measures.

$\beta = 0.05$ $\gamma = 0.25$	DC	EC	CC	BC	GC	EVE
CS-PhD	59	26	26	41	51	71
Email-Enron	6	3	4	4	4	5
Email-Univ	38	40	37	30	43	40
Karate	1	1	1	1	1	1

6. Discussion and Conclusions

In this study, we proposed an approach that approximates the influences of nodes in complex networks under the SIR propagation model using the shortest paths between nodes and then applies this to rank the nodes. The EVE is similar to a centrality measure in that it is used for ranking nodes. However, EVE is not a centrality measure, but a metric specific to the SIR model. As a result of 12 simulations we made with four different real-world datasets and three different SIR settings, EVE performed better than state-of-the-art and well-known centrality measures.

We compared EVE with well-known centrality measures as well as with a state-of-the-art measure such as Gravitational Centrality, which is successful and innovative method. The EVE demonstrated that the expected influences of nodes could be better distinguished by using the parameters of the propagation model and the shortest paths (without using the centrality measures of the nodes).

The EVE is calculated using the shortest paths between nodes. This means that all other paths are ignored. In dense networks, there can be many different paths other than the shortest path between two nodes. Therefore, ignoring these paths increase the difference (error) between EVE and the actual expected influence. In future studies, we plan to develop approaches that produce more precise results without increasing the time complexity.

Biography

Aybike ŞİMŞEK received her BS degree from the Selçuk University Department of Computer Engineering in 2003, her MS degree from the Gazi University Department of Computer Engineering in 2010, and her PhD from the Düzce University Department of Electrical-Electronics and Computer Engineering in 2018. She was a postdoctoral researcher in the Department of Computer Science at the Humboldt University of Berlin from August 2018 to July 2019. She was a lecturer in the Department of Computer Programming at Düzce University from January 2015 to December 2020. She has been working as assistant professor in the

Department of Computer Engineering at Düzce University since December 2020. Her current research interests include social network analysis, complex networks, and epidemic modeling.

References

- [1] J. Gao, B. Barzel, and A.-L. Barabási, “Universal resilience patterns in complex networks,” *Nature*, vol. 530, no. 7590, pp. 307–312, Feb. 2016, doi: 10.1038/nature16948.
- [2] A.-L. Barabási, N. Gulbahce, and J. Loscalzo, “Network medicine: a network-based approach to human disease,” *Nat. Rev. Genet.*, vol. 12, no. 1, pp. 56–68, Jan. 2011, doi: 10.1038/nrg2918.
- [3] H. Alasmary *et al.*, “Analyzing and Detecting Emerging Internet of Things Malware: A Graph-Based Approach,” *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8977–8988, 2019, doi: 10.1109/JIOT.2019.2925929.
- [4] J. Yang, C. Yao, W. Ma, and G. Chen, “A study of the spreading scheme for viral marketing based on a complex network model,” *Phys. A Stat. Mech. its Appl.*, vol. 389, no. 4, pp. 859–870, Feb. 2010, doi: 10.1016/j.physa.2009.10.034.
- [5] İ. Tuğal and A. Karcı, “Comparisons of Karcı and Shannon entropies and their effects on centrality of social networks,” *Phys. A Stat. Mech. its Appl.*, vol. 523, pp. 352–363, Jun. 2019, doi: 10.1016/j.physa.2019.02.026.
- [6] J. Leskovec, D. Huttenlocher, and J. Kleinberg, “Predicting positive and negative links in online social networks,” in *Proceedings of the 19th international conference on World wide web - WWW '10*, 2010, p. 641, doi: 10.1145/1772690.1772756.
- [7] S. P. Borgatti, A. Mehra, D. J. Brass, and G. Labianca, “Network Analysis in the Social Sciences,” *Science (80-.)*, vol. 323, no. 5916, pp. 892–895, Feb. 2009, doi: 10.1126/science.1165821.
- [8] S. Chang *et al.*, “Mobility network models of COVID-19 explain inequities and inform reopening,” *Nature*, vol. 589, no. 7840, pp. 82–87, Jan. 2021, doi: 10.1038/s41586-020-2923-3.
- [9] Y. Yang, X. Wang, Y. Chen, M. Hu, and C. Ruan, “A Novel Centrality of Influential Nodes Identification in Complex Networks,” *IEEE Access*, vol. 8, pp. 58742–58751, 2020, doi: 10.1109/ACCESS.2020.2983053.
- [10] J. Zhang, C. Yang, Z. Jin, and J. Li, “Dynamics analysis of SIR epidemic model with correlation coefficients and clustering coefficient in networks,” *J. Theor. Biol.*, vol. 449, pp. 1–13, 2018, doi: 10.1016/j.jtbi.2018.04.007.
- [11] S. Banerjee, M. Jenamani, and D. K. Pratihar, “A survey on influence maximization in a social network,” *Knowl. Inf. Syst.*, 2020, doi: 10.1007/s10115-020-01461-4.
- [12] D. Kempe, J. Kleinberg, and É. Tardos, “Maximizing the spread of influence through a social network,” in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '03*, 2003, p. 137, doi:

10.1145/956750.956769.

- [13] S. P. Borgatti, “Identifying sets of key players in a social network,” *Comput. Math. Organ. Theory*, vol. 12, no. 1, pp. 21–34, Apr. 2006, doi: 10.1007/s10588-006-7084-x.
- [14] L. C. Freeman, “A Set of Measures of Centrality Based on Betweenness,” *Sociometry*, vol. 40, no. 1, p. 35, Mar. 1977, doi: 10.2307/3033543.
- [15] L. Katz, “A new status index derived from sociometric analysis,” *Psychometrika*, vol. 18, no. 1, pp. 39–43, Mar. 1953, doi: 10.1007/BF02289026.
- [16] L. Page, S. Brin, R. Motwani, and T. Winograd, “The PageRank Citation Ranking: Bringing Order to the Web.,” Stanford InfoLab, 1999. [Online]. Available: <http://ilpubs.stanford.edu:8090/422/>.
- [17] J. Sheng *et al.*, “Identifying influential nodes in complex networks based on global and local structure,” *Phys. A Stat. Mech. its Appl.*, vol. 541, p. 123262, 2020, doi: 10.1016/j.physa.2019.123262.
- [18] C. Salavati, A. Abdollahpouri, and Z. Manbari, “Ranking nodes in complex networks based on local structure and improving closeness centrality,” *Neurocomputing*, vol. 336, pp. 36–45, 2019, doi: 10.1016/j.neucom.2018.04.086.
- [19] Z. Ghalmane, M. El Hassouni, C. Cherifi, and H. Cherifi, “Centrality in modular networks,” *EPJ Data Sci.*, vol. 8, no. 1, 2019, doi: 10.1140/epjds/s13688-019-0195-7.
- [20] J. Zhao, Y. Song, and Y. Deng, “A novel model to identify the influential nodes: Evidence theory centrality,” *IEEE Access*, vol. 8, pp. 46773–46780, 2020, doi: 10.1109/ACCESS.2020.2978142.
- [21] Z. Lv, N. Zhao, F. Xiong, and N. Chen, “A novel measure of identifying influential nodes in complex networks,” *Phys. A Stat. Mech. its Appl.*, vol. 523, pp. 488–497, 2019, doi: 10.1016/j.physa.2019.01.136.
- [22] Z. Zhao, X. Wang, W. Zhang, and Z. Zhu, “A community-based approach to identifying influential spreaders,” *Entropy*, vol. 17, no. 4, pp. 2228–2252, 2015, doi: 10.3390/e17042228.
- [23] Z. Ghalmane, C. Cherifi, H. Cherifi, and M. El Hassouni, “Centrality in Complex Networks with Overlapping Community Structure,” *Sci. Rep.*, vol. 9, no. 1, pp. 1–29, 2019, doi: 10.1038/s41598-019-46507-y.
- [24] Y. Zhao, S. Li, and F. Jin, “Identification of influential nodes in social networks with community structure based on label propagation,” *Neurocomputing*, vol. 210, pp. 34–44, 2016, doi: 10.1016/j.neucom.2015.11.125.
- [25] Y. Y. Keng, K. H. Kwa, and C. McClain, “Convex combinations of centrality measures,” *J. Math. Sociol.*, 2020, doi: 10.1080/0022250X.2020.1765776.
- [26] S. S. Ali, T. Anwar, and S. A. M. Rizvi, “A Revisit to the Infection Source Identification Problem under Classical Graph Centrality Measures,” *Online Soc. Networks Media*, vol. 17, no. xxxx, p. 100061, May 2020, doi: 10.1016/j.osnem.2020.100061.
- [27] M. Alshahrani, Z. Fuxi, A. Sameh, S. Mekouar, and S. Huang, “Efficient algorithms

- based on centrality measures for identification of top-K influential users in social networks,” *Inf. Sci. (Ny)*, vol. 527, pp. 88–107, Jul. 2020, doi: 10.1016/j.ins.2020.03.060.
- [28] M. Şimşek and H. Meyerhenke, “Combined centrality measures for an improved characterization of influence spread in social networks,” *J. Complex Networks*, vol. 8, no. 1, Feb. 2020, doi: 10.1093/comnet/cnz048.
- [29] L. Ma, C. Ma, H. Zhang, and B. Wang, “Identifying influential spreaders in complex networks based on gravity formula,” *Phys. A Stat. Mech. its Appl.*, vol. 451, pp. 205–212, Jun. 2016, doi: 10.1016/j.physa.2015.12.162.
- [30] X.-L. Yan, Y.-P. Cui, and S.-J. Ni, “Identifying influential spreaders in complex networks based on entropy weight method and gravity law,” *Chinese Phys. B*, vol. 29, no. 4, p. 048902, Apr. 2020, doi: 10.1088/1674-1056/ab77fe.
- [31] A. Simsek, “Lexical Sorting Centrality to Distinguish Spreading Abilities of Nodes in Complex Networks under the Susceptible-Infectious-Recovered (SIR) Model,” Jan. 2021, [Online]. Available: <http://arxiv.org/abs/2101.10975>.
- [32] X. Wen, C. Tu, M. Wu, and X. Jiang, “Fast ranking nodes importance in complex networks based on LS-SVM method,” *Phys. A Stat. Mech. its Appl.*, vol. 506, pp. 11–23, Sep. 2018, doi: 10.1016/j.physa.2018.03.076.
- [33] D. Tolić, K.-K. Kleineberg, and N. Antulov-Fantulin, “Simulating SIR processes on networks using weighted shortest paths,” *Sci. Rep.*, vol. 8, no. 1, p. 6562, Dec. 2018, doi: 10.1038/s41598-018-24648-w.
- [34] M. G. KENDALL, “A NEW MEASURE OF RANK CORRELATION,” *Biometrika*, vol. 30, no. 1–2, pp. 81–93, Jun. 1938, doi: 10.1093/biomet/30.1-2.81.
- [35] J. Bae and S. Kim, “Identifying and ranking influential spreaders in complex networks by neighborhood coreness,” *Phys. A Stat. Mech. its Appl.*, vol. 395, pp. 549–559, Feb. 2014, doi: 10.1016/j.physa.2013.10.047.
- [36] G. Rossetti, L. Milli, S. Rinzivillo, A. Sirbu, D. Pedreschi, and F. Giannotti, “NDlib: a python library to model and analyze diffusion processes over complex networks,” *Int. J. Data Sci. Anal.*, vol. 5, no. 1, pp. 61–79, Feb. 2018, doi: 10.1007/s41060-017-0086-6.
- [37] M. Newman, *Networks*, Second. Oxford, UK: Oxford University Press, 2018.
- [38] P. Bonacich, “Power and Centrality: A Family of Measures,” *Am. J. Sociol.*, vol. 92, no. 5, pp. 1170–1182, Mar. 1987, doi: 10.1086/228631.
- [39] G. Sabidussi, “The centrality index of a graph,” *Psychometrika*, vol. 31, no. 4, pp. 581–603, Dec. 1966, doi: 10.1007/BF02289527.
- [40] W. Zachary, “An Information Flow Model for Conflict and Fission in Small Groups,” *J. Anthropol. Res.*, vol. 33, pp. 452–473, 1977.
- [41] R. A. Rossi and N. K. Ahmed, “The Network Data Repository with Interactive Graph Analytics and Visualization,” 2015, [Online]. Available: <http://networkrepository.com>.
- [42] R. Guimerà, L. Danon, A. Díaz-Guilera, F. Giralt, and A. Arenas, “Self-similar

- Community Structure in a Network of Human Interactions,” *Phys. Rev. E*, vol. 68, no. 6, p. 65103, 2003.
- [43] W. De Nooy, A. Mrvar, and V. Batagelj, *Exploratory social network analysis with Pajek*, vol. 27. Cambridge University Press, 2011.
- [44] A. A. Hagberg, D. A. Schult, and P. J. Swart, “Exploring Network Structure, Dynamics, and Function using NetworkX,” in *Proceedings of the 7th Python in Science Conference*, 2008, pp. 11–15.