Journal of Theoretical and Applied Information Technology
15th April 2021. Vol.99. No 7


ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

# IDENTIFYING INFLUENTIAL SPREADERS IN COMPLEX NETWORKS BY WEIGHTED VOTE RANKING AND HYBRID METHODS

[1] RAAMAKIRTINAN S, [2] JENILA LIVINGSTON L M
[1,2]School of Computer Science and Engineering, Vellore Institute of Technology (VIT), Chennai Campus, Tamil Nadu, India.
E-mail: [1]raamakirtinan.s2013@vit.ac.in, [2]jenila.lm@vit.ac.in

## ABSTRACT

Identifying influential users in complex networks has become the need of the hour in digital era and it has emerged as an important and interesting research topic. There are many methods like centrality measure, page rank, k-shell etc., to rank influential users, but each has its own shortfalls. In this paper we propose weighted vote ranking method and a hybrid ranking method as an extension of weighted vote ranking Method. The spreading ability of a node in a network mainly depend upon three key factors such as the support (vote) that it receives from neighbor nodes, the relationship strength (weight) and the nodes position in the network (coreness). So, to identify potential super spreaders we propose weighted vote ranking (WVR) method with all these three metrics such as vote, weight and coreness with a dynamic weight parameter control. Also, to improvise further we extended this proposed method by combining with weighted mixed degree decomposition (WMDD) method which has ability to consider removed and remaining nodes in decomposition technique. This hybrid method is obtained by combining these WVR and WMDD methods in 70:30 ratio which gives better results, compared to other ratio combinations. To study the spreading dynamics, Susceptible Infected Recovered (SIR) epidemic model is used. The average Kendall Tau of proposed weighted vote ranking and hybrid methods across various networks taken for experiment is 0.86 and 0.83 respectively whereas the highest average Tau from other methods is 0.79 (1 is maximum Tau value). Also, when we compared the average infection scale with other methods that is taken for experiment, we get 5.1%-14.5% higher infection scale for weighted vote ranking method and 8.8%-18.2% higher infection scale for hybrid method. Similar various other experiments like varying number of seed nodes and infection probability also show that proposed methods perform better when compared to recent other methods.

**Keywords:** *Complex Network, Influential Spreaders, Network Decomposition, Weighted Vote Rank, Information diffusion*

## 1. INTRODUCTION

In today's world social networking plays a major role in every one's life. There are millions of active users in popular social networking platforms like Facebook, Twitter etc. The major problem these social networking platforms face is rumors. These rumors are started by very few people or community which becomes viral in no matter of time. So, to combat the rumors its high time to identify users from whom it originates and these users are known as influential users. It is to be noted that identifying these users is beneficial not only to arrest rumors but it has other use cases like cost effective business advertisement, community finding, promoting government messages and schemes and so on. And how to identify these influential users has gained up a momentum because of its practical and theoretical significance [1]. This knowledge of spreading gives ability to find social leaders [2].

Till date various methods of analyzing this information spreading has been proposed and out of those centrality measures are the classic ones to rank the influential nodes and these can give better result on simple networks and not scalable to complex real time networks due to its computational complexity [3]. Page rank [4] which was proposed to rank web pages was later used to identify influential users, but it was efficient in ranking web pages alone and was

Journal of Theoretical and Applied Information Technology
15th April 2021. Vol.99. No 7
© 2021 Little Lion Scientific


not much suitable in complex networks like social networks where real users are connected. So latter after many ranking methods, k-shell [5] indexing method has emerged as successful one to rank users. And it was further improved by adding various parameters and other components to consider lower shell nodes [6]. But it was still lacking to identify influential users uniquely as it by default was placing the nodes based on nodes core number and there are high number of nodes with same ranks. In the meantime Zhang et al et al proposed a vote based ranking method named vote rank [7], but it was ranking users purely based on neighbors vote and networks coreness which was a major factor for k-shell based algorithms was not considered.

So, most of the existing methods to identify influential users are computed just considering the network structure alone. Even the recent proposed methods proposed by Wang et al[8], Giridhar Maji [9], Namtirtha et al [10] etc., try to refine the k-shell method and improvise it. The basic concept of k-shell method is to prune the nodes and rank the nodes are per their cores that it is placed. But in real world scenario the influential capacity of node just doesn't depended on structure alone and depend on numerous factors such as relationship strength, neighbors' position, relationship age etc. For instance, in a social network like Facebook, Twitter etc, even though a person is connected to a page or group, that doesn't mean that he agrees with all the content of page/group. The intention to share a message from a particular page/group (node) mainly depend on the content of the feed. Also, many a times a person X might have become a friend to Y just as he was his colleague/student of same class, in real world X may have negative sentiments with Y and will not spread any information from Y. i.e, nodes with high degrees might have small weight and vice versa. So, in case of using algorithms which just considering structural information of a network will not result in finding right influential users. Hence, we hypothesis that both structural and other factors like sentiments (weights) associated with node must be considered for computing influential users.

Based on above explained hypothesis, we propose an algorithm which primarily works by support (vote) [7] that it receives from neighbor nodes, the relationship strength (weight) and the nodes position in the network (coreness). Later we also extend the proposed algorithm with degree decomposition technique [11] which further adds coreness power and identities influential users more precise and distinct.

Few key highlights of this method are that, the dynamic parameter controls like alpha, beta, gamma in proposed method gives the end users to give flexibility to give appropriate weightage to each network accordingly. In real world all networks are not the same kind. (for example, social network is different from email network). So, this algorithm has flexibility to vary the parameter network to network. Also, in the experiment (section 4) you can notice that for some points the proposed Weighted Vote Rank (WVR) performs lower than bench mark algorithms, but the extended hybrid version (HYB) performed well in that case. This means having goodness of two algorithm helped hybrid method to perform better in those areas. Moreover, adding multiple features like vote, weight, coreness and blending helped to distinguish users uniquely unlike k-shell methods, where users are grouped based on core number and many are categorized in same group. Hence, we could see the infection scale is higher in the various experiments.

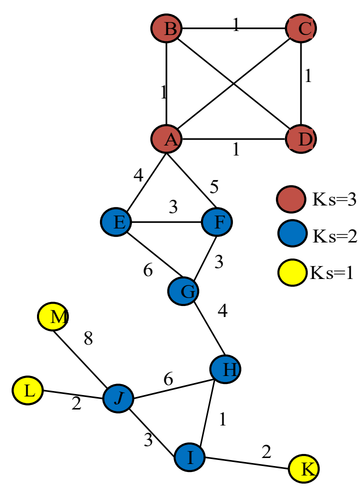


*Figure 1: Toy network to show proposed methods benefit over others.*

To further demonstrate the effectiveness of proposed method, consider above toy network in Fig 1, here incase if some popular methods like k-shell or any other similar methods would have been used, the nodes in Ks =3 (brown) would be in the top list among influential users as those are placed at shell/core number 3. In case of using the proposed method, we would get some nodes in Ks=2 (blue) or even node M in top influential user list, though they are in core 2. The reason is that, though brown nodes are in shell three they are weekly connected in terms of weight and neighborhood coreness. But blue nodes have more weights and neighborhood coreness. Even when we compute rank via SIR ranking benchmark, we get lower shell nodes have higher spreading capability than the core nodes b, c, and d. This shows that the proposed Weighted Vote Rank method and its extension Hybrid Method have better ranking ability.

The rest of this paper is organized as follows. Section 2 provides a review of the state of the art work. Section 3 discusses the proposed approach. Section 4 provides the experiment and result. Finally, the conclusion is summarized in Section 5.

## 2. RELATED WORKS

Out of many works that has emerged in this field of identifying influential users, centrality measure is the early one and over the years much this measure has been used widely. In the following we will discuss about these including state of the art methods which uses k-shell methodology.

The degree centrality (DC) [12] is based on number of links connecting to the node. When n nodes are connected to a node then degree centrality will be n, its direct 1:1 relationship between number of connected nodes and degree. S. Gao et al. [13] represented it as shown below,

$$C_D(v) = \sum_{u=1}^{n} p_{uv} = |\Gamma_h(v)| \quad (1)$$

Here u, v are the connected nodes and $\Gamma_h(v)$ is used to denote the set of neighbors within h-hops from node v and h is 1 for this case.

The betweenness centrality (BC) [14] is a measure which denotes fraction of shortest path between the node pairs that pass through the given node. It is calculated as,

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{shrt}(v)}{\sigma_{shrt}} \quad (2)$$

Here $\sigma_{shrt}$ represents overall number of shortest paths from node s to node v. $\sigma_{shrt}(v)$ represents number of shortest paths from node s to t which pass through the node v.

The Closeness centrality (CC) [15] is defined as the reciprocal of the sum of the shortest distance to all nodes in the network. It is defined as,

$$C_c(v) = \left[\sum_{u \in V \setminus v} d_G(v,u)\right]^{-1} \quad (3)$$

Here $d_G(v,u)$ represents geodesic distance between v and u and the computational complexity is $O(n^3)$.

Following these centrality measures Page rank [4] which was used to rank webpages and rank influential nodes and Leader rank [2] a ranking methodology specially used for user networks was proposed. But both were not successful as Page rank was performing well in ranking webpages alone and not suited for real user networks and Leader rank was not working well for undirected networks. Some of the other ranking methods like HITS [16] were seemed to be promising, but they were not performing well for all the given networks. Hill-climbing based greedy algorithm [17] was proposed by Kempe et al, which can find a group of important nodes but it was much time consuming in large scale networks. Chen et al. proposed semi-local centrality measure by trading of few other costly centrality measures [3]. By considering neighbor number and clustering coefficient, ClusterRank [18] was proposed. This ClusterRank was working by making use of local cluster information rather than considering the whole network for computing ranks. This raking method was having low computational complexity comparted to others. However, if top ranked nodes are selected as a group to spread, the result is not so good. Wang et al. proposed fast ranking method [19] to rank nodes by using a k-shell iteration to differentiate between the spreading capability within the same core. This approach solved monotonicity issue a bit, but still the spreaders are not ranked in decentralized way. Later Kitsak et al. defended that nodes global position is more important than local structure. Chen et al. proposed degree discount heuristics method [20] which improves influence spread rate and runs in minimal time. Sanjay Kumar et al proposed Neighborhood Coreness based voting approach by including neighborhood coreness metrics [21]. Wang et al. came up with improved k-shell method

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

which works by taking account of shortest distance from target node to core node. [8] Giridhar Maji proposed potential edge weight based k-shell method which works by considering edge weight based on connecting nodes [9] Xiao-Li Yan et al proposed a method based on entropy weight method and gravity law which works by combining traditional centrality metrics [22]. Namtirtha et al [10] proposed a weighted kshell method based neighbors degree which is intron a combination of k-shell and degree of nodes. One major pain point for k-shell method is that, in k-shell pruning process only residual links degree were taken into account and exhausted links during pruning was not considered and this was addressed in mixed degree decomposition method [11], but still even this decomposition worked on same principle of k-shell algorithm which focuses on coreness of the network.

Later, Zhang et al proposed vote rank [7] which suggested to rank nodes by getting voted by its neighbors. The neighbor nodes which voted and elected leader get their voting abilities reduced in subsequent iterations. The key advantage of this method is that it was able to locate spreaders in decentralized positions, which paved way to increase spreading rate in short time. However, it has a limitation that this method can applied only to an unweighted network. In real world situations most of the complex networks are weighted, where the strength of edges indicates the interactions between nodes. So, in summary most of the latest methods [8] listed in this section uses primarily network's coreness metric (k-shell) alone. Though some of them were using weights for computation, it was in turn used in same k-shell method whose exhausted links are not considered in pruning process [9][10]. Also, key important factor like neighbors coreness was not considered. And in case of neighbors coreness metric is considered the weight is not considered [21]. In some of the other popular methods which uses different technique like vote rank [7] or using k-shell methods which considers exhausted node link [11] do not consider neither of these important metrics like network coreness or weight.

So to address these issue, inspired by Vote rank algorithm [7], we propose a Weighted Vote Raking (WVR) algorithm by considering important network metric like vote, weight and coreness and then extend the method by blending the top ranked node set with Weighted Mixed Degree Decomposition (WMDD) method's node set, which was built based upon Mixed Degree Decomposition (MDD) [11] method in appropriate proposition and build a hybrid method which effectively identifies influential spreaders.

## 3. PROPOSED METHODOLOGY

This section discusses about our proposed methodology Weighted Vote Raking algorithm (WVR) and list of the set of processes involved in it. Later in this section we also cover the Weighted Mixed Degree Decomposition (WMDD) and its blending technique with weighted vote raking algorithm.

As discussed in above section, vote rank has a limitation that, it cannot be applied on to a weighted network. Also, another major factor is that the voting ability of each node is considered the same across network at initial stage and the same is taken for vote score and nodes are ranked. This causes both core nodes and leaf nodes to be treated alike. For instance, take the network shown in Fig 2, where nodes are colored ranging from one core (violet) to six core nodes (red). If traditional vote rank is applied to this network, thought it gets vote score from neighbors, it treats all nodes as the same. I.e., six core nodes as well as one core node get similar kind of vote score and ranked. Also, the relationship strength between nodes (weight) is also being considered. Contrary to coreness example, an outer core node may have high weight than compared to inner core. So, in case if we rank the nodes as such without considering these factors, even though the nodes will be ranked from one to N, in real world case when an information is conveyed to selected spreader, the selected node will not propagate the info to its adjacent nodes as it has poor relationship strength (weight) between nodes. So, we argue that it is vitally important to consider vote score, underlying weight between nodes and its coreness in appropriate proposition and rank them accordingly.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

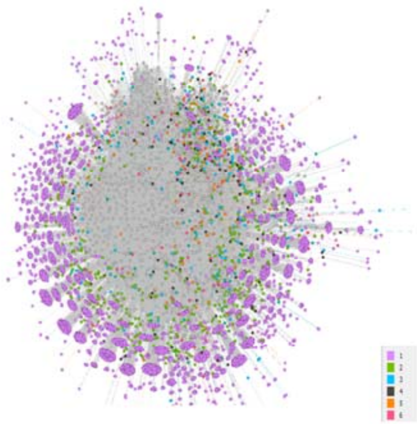

*Figure 2. Visualization of bitcoin Facebook forum network in Gephi tool using Force Atlas algorithm and coloring using network coreness.*

Consider a weighted complex network G = (V, E), where V is the vertices/nodes and E is edges/links between the nodes and $w_{uv}$ represents weight between the nodes u and v. The neighborhood coreness is represented as NC(V). Now to implement the core principle of vote rank, each node v ∈ V is associated with a tuple ($V_s$, $V_a$) where $V_s$ is the voting score and $V_a$ is the voting ability of the node. Voting ability means the maximum score that a node can vote its immediate neighbor. Vote score is the total voting score received from its neighbors. It can be computed by $V_s = \sum_{i \in N(V)} V_a$ where N(v) is the neighbor nodes set. Listed below are the detailed step of this weighted vote rank algorithm.

**Step I (Initialization):** In this step we initialize all nodes and following parameters necessary for computing weighted vote rank.

a. Tuples of the nodes are set to (0,1) where first element represents the total vote score and second element represents voting ability.
b. Extract weight of each edge, normalize between 0 to 1 value as the values may have scaling issues due to varying magnitude as per Eq. (4). In case the weights are spread across nodes, take average value between node u & node v and make it as its edge weight.
c. Compute coreness of the given network's nodes by k-shell algorithm and assign coreness value to each node in the network.

$$\text{Normalized weight of Node (Ni)} = \frac{Weight\ of\ N_i - \min(Weight)}{\max(Weight) - \min(Weight)} \quad (4)$$

**Step 2 (Voting & Key Metric computation):** Each node vote for their neighbors and at the same time, it also get voted by its neighbors. Then the final value is computed using the three metrics vote score ($V_s$), edge weight ($W_i$) and node coreness ($N_c$) then raised to the power α, β and γ respectively. The same is represented in Eq (5) below.

$$V_S := |N(Neigh)| * \left( \sum_{i \in N(v)} \left( (v_{ai})^{\alpha} * (w_i)^{\beta} * (NC_i)^{\gamma} \right)^{\frac{1}{(\alpha+\beta+\gamma)}} \right) \quad (5)$$

Here

- $v_{ai}$ → Vote score, voted by immediate neighbors
- $w_i$ → Edge weight between nodes U and Vi
- $NC_i$ → Neighborhood Coreness (NC) I.e., K-shell value of its neighbors
- $N(Neigh)$ → Number of neighbor nodes
- α, β & γ → Adjustable parameters for weightage and optimization

As per above Eq. (5), after each node's votes to a node N along with edge weight and the coreness, the total vote score is aggregated. For instance, if node A gets vote score $V_{s1}$, $V_{s2}$ and $V_{s3}$ from its adjacent neighbors along with weight and coreness, it is then summed up as $V_S = V_{s1} + V_{s2} + V_{s3}$, where $V_{si}$ is product of voting ability, weight between nodes and neighborhood coreness value.

Here we give weightage to three key metric voting ability value, weight and neighborhood coreness as 0.5, 0.25 and 0.25 respectively giving preference to vote score. So, in that case all three parameters are given same values as its constant value. These makes the Eq. (5) to get simplified as below.

$$V_S := |N(Neigh)| * \left( \sum_{i \in N(v)} \left( (v_{ai})^{0.5} * (w_i)^{0.25} * (NC_i)^{0.25} \right) \right) \quad (6)$$

This same proposed voting score computation can be extended for an unweighted network by treating all edges to have equal weight of 1 and apply zero to weight parameter in Eq (5). In such a scenario Eq. (5), would get further simplified as below.

$$V_S := |N(Neigh)| * \left( \sum_{i \in N(v)} ((v_{ai})^{0.75} * (NC_i)^{0.25}) \right) \quad (7)$$

So, using this of mechanism to compute vote score means that vote score can vary any value between voting ability of neighbors and weight and coreness value product. The node which gets maximum vote score will be selected as spreader in this iteration and will not be participating in subsequent voting rounds. This is done by updating its voting ability to zero. Also, to ensure this same node is not elected again its vote is also set to zero after marking the node it to leaders list.

**Step 3 (Weaken voting ability):** After voting and scoring step, we must weaken the voting ability of the nodes which earlier voted for the node which was elected as to spreader. This is to ensure the core principle of vote rank that diverse nodes are selected. i.e., only if we weaken the voting ability of neighbor nodes, in subsequent iteration nodes from diverse positions would be elected as spreaders which will help to improve the overall spreading processing in real world scenario. A factor *f* is set to be reduce the voting ability of neighbors. Below equation explains the voting ability reducing factor.

$$V_{ai} = \begin{cases} V_{ai} - f & if\ V_{ai} > 0 \\ 0 & Otherwise \end{cases} \quad (8)$$

Where $f = \frac{1}{<k>}$

<k> is the average degree of the network. This voting ability updation is done up to one level of neighbors alone as per Eq. (8). The subsequent levels voting ability is not updated. The step of updating (reducing) voting ability ensures to get spreaders from a diverse position of network instead of concentrated code.

**Step 4 (Iterate):** Repeat steps 3 and 4 until C number of nodes are selected, where C is a constant of any desired number. $C \in 1 \dots N$.

### 3.1 Time Complexity:

To compute the time complexity of this proposed weighted vote rank, we can split the entire process into three main categories. 1. Voting, 2. Selecting the node with highest vote score and 3. Updating the voting ability and vote scores. To compute these, complexity we represent entire given network as G = (V, E), where |E|= m is the total number of edges and |V|=n is the total number of nodes.

For the first step phase, it needs O(n) steps to initialize voting ability of n number of nodes. As the vote score is calculated by voting ability ($V_{ai}$), weight ($W_i$) and k-shell coreness ($NC_i$), O(($V_{ai}$)*($W_i$)*($NC_i$)) = O(m) is needed. Then to compute for all its neighbors' nodes we need $\sum deg_i$ = O(n+m). Therefore, for first phase we can simplify it as O(n+m).

In second phase of section of spreaders node getting highest vote score would be selected as top nodes. The number of node selection is user preference constant C. Hence, we can define as O(c*n) where C is constant. The third phase consist of updating voting ability of its immediate neighbors is performed by in O(c⟨*f*⟩) = O(c ∗ m/n) where ⟨*f*⟩ is the average degree of the nodes in the given network.

Hence, the total time complexity is given as O (n+ m + cn +c ∗m/n) where c is user preference constant which is generally very low value (c<<n). If the network is very sparse like, O(n) = O(m) the time complexity of the entire process can be even simplified to O(n).

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

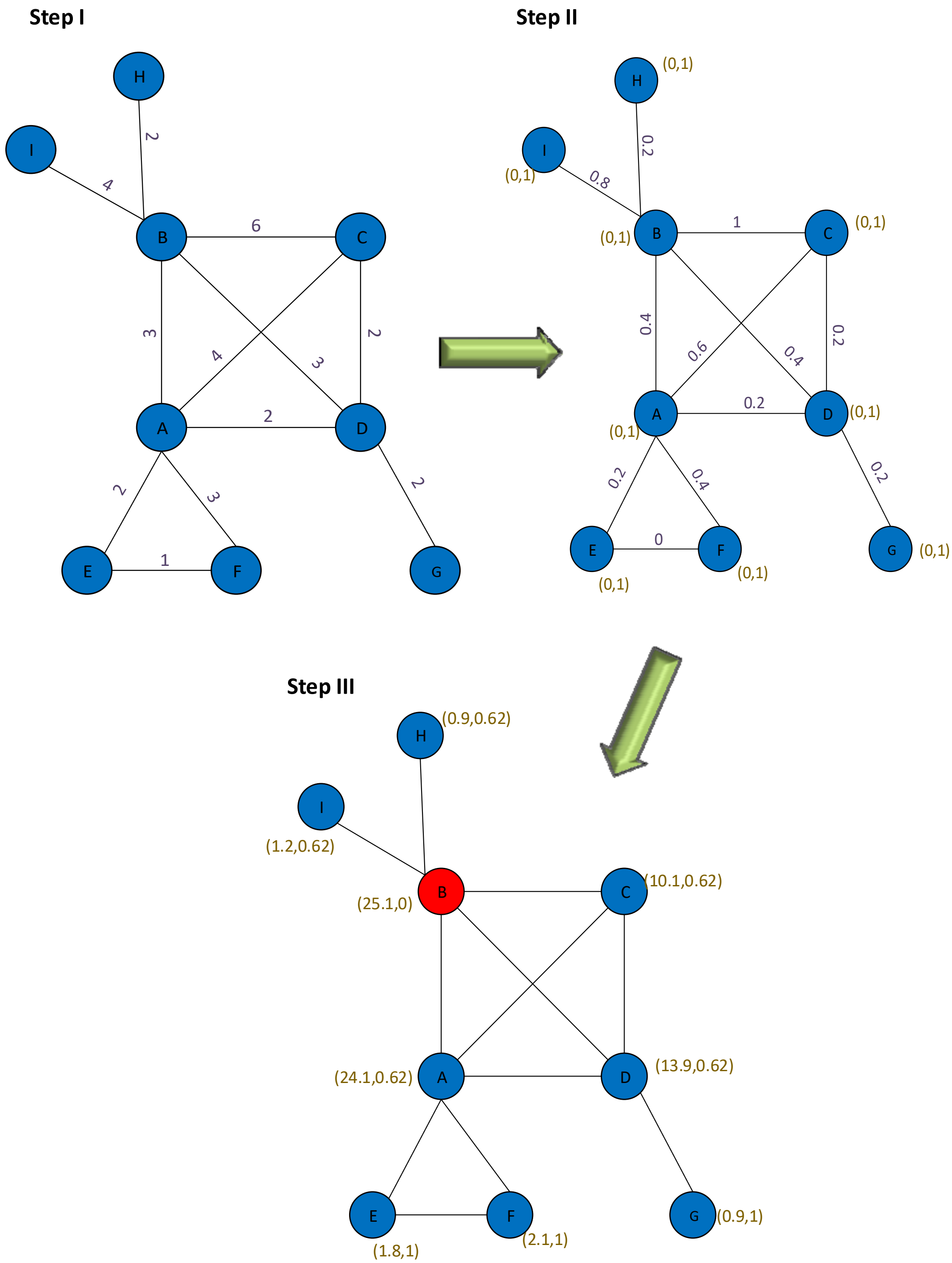


*Figure 3: A sample toy network illustrating weighted vote rank process for one iteration.*

### 3.2 A Running Example with Toy Network:

In order to give an illustrative explanation, we use this proposed weighted ranking methodology on a toy network as shown in Fig 3. Considering varying weightage parameter as below, we get Eq (9) for first iteration

Varying weightage parameters:

$\alpha = 0.5$
$\beta = 0.25$
$\gamma = 0.25$

$$V_S := 5 * \left( \sum_{i \in N(v)} \left( \begin{array}{c} (v_{ai})^{0.5} \\ * (w_i)^{0.25} \\ * (NC_i)^{0.25} \end{array} \right)^{\frac{1}{(0.5 + 0.25 + 0.25)}} \right) \quad (9)$$

Substituting values for the node values like voting ability, weight and neighborhood coreness after normalizing the weights as per Eq (6), we get below result for node A.

$$= 5 * \sum((1)^{0.5} * (0.4)^{0.25} * (3)^{0.25}) + ((1)^{0.5} * (0.6)^{0.25} * (0.3)^{0.25}) + ((1)^{0.5} * (0.2)^{0.25} * (3)^{0.25}) + ((1)^{0.5} * (0.2)^{0.25} * (2)^{0.25}) + ((1)^{0.5} * (0.4)^{0.25} * (2)^{0.25})$$

So as per above computation, we get value simplified as per Table 1. Now the same operation has to be performed for all nodes in the network G. Once all nodes vote score is computed and voted, we get below vote score tuple as per Fig 3 (Step III). The second value in the tuple is computed by using average degree of network with the formula $f = \frac{1}{<k>}$. <K> value for this graph network is 2.66. So, we get $f = \frac{1}{2.66} = 0.38$. Now as per algorithm reduce the voting ability by 0.38 for its neighbors, after which we get $V_{ai} = 0.62$. It is to be noted that this above computation illustration is done only for one node 'A' for first iteration alone. This has to be repeated for all nodes till C number of times, where C is the desired number of influential spreaders to be obtained.

*Table 1: Values obtained for various metrics for node A in first iteration.*

| **Node $V_i$** | **Node U** | **Vote ability ($V_{ai}$)** | **Weight ($W_i$)** | **Neighborhood coreness ($NC_i$)** | **Total** |
|---|---|---|---|---|---|
| a | b | 1 | 0.795271 | 1.316074013 | 1.046635139 |
| a | c | 1 | 0.880112 | 1.316074013 | 1.158292185 |
| a | d | 1 | 0.66874 | 1.316074013 | 0.880111737 |
| a | e | 1 | 0.66874 | 1.189207115 | 0.795270729 |
| a | f | 1 | 0.795271 | 1.189207115 | 0.945741609 |
| | | | | **A** | **=5*(4.82) = 24.1** |

*Algorithm 1: Pseudocode of Weighted Vote Rank Algorithm*

**Input**: Weighted graph G, whose edges contain a weight W
**Output**: Return ranked list of nodes
Step 1: $V_a$ ← 1; //1. Initialization
W_of_edge ← get(G, 'normalized weight');
Coreness ← core_number(G)

Step 2 **While** number_of_leaders < total_nodes **do**

Step 3: **for each** $node_i$ **in** graph_G_not_in_leaders_list **do** //2. Vote step
$V_s$ ← 0 and $N_c$ ← 0 //re-setting every time
**for each** $node_j$ **in** neighbour_list($node_i$) //Computing Score using all its neighbors
wt ← wt($node_i$, $node_j$)
$V_s$ ← $V_s$ + (node_tuple($node_j$)[1] * α + wt* β + coreness($node_j$)* γ)
$N_c$ ← $N_c$ +1
$V_s$ ← ($N_c$ * $V_s$)* (1/ (α + β + γ))
node_tuple($node_i$) ← ($V_s$, node_tuple($node_j$)[1])

Step 4: **for each** node **in** graph_G_not_in_leaders_list **do** //Get current leader
    score_list[node]←node_tuple[node][0]
  max_value ← max(score_list.values())
  current_leader←[k for k,v in score_list.items() if v == max_value]

Step 5: **for each** element **in** current_leader **do** //3. Decrease voting ability
    node_tuple[element]←(0,0)
    **for each** node$_k$ in neighbour(node$_k$) **do**
      updated_va←node_tuple[element][1] – (1/average_degree(**G**))
      **if** updated_va<0:
        updated_va←0
      node_tuple[node$_k$]←(node_tuple[element][0], updated_va) //Assigning updated tuple

Step 6: **for each** k **in** current_leader **do**:
    leaders[k] ← max_value //assign or update each node value for each iteration

Step 7: **return** leaders

*Algorithm 2: Pseudocode of hybrid ranking method*

**Input**: Graph G, Rank_List_1, Rank_list_2, Weightage%_1, Weightage%_2
**Output**: Hybrid values //Lower the better
Step 1: **for each** node, rank_val **in** rank_list_1.items() **do**:
    temp_rank_1[node]← rank_val* Weightage%_1
Step 2: **for each** node, rank_val **in** rank_list_2.items() **do**:
    temp_rank_2[node]← rank_val* Weightage%_2
Step 3: **for each** node **in** graph G **do**:
    hybrid_vaues(node)← temp_rank_1[node] + temp_rank_2[node]
Step 4: **return** hybrid_values

This above proposed algorithm 1, Weighted Vote rank algorithm (WVR) itself performs better in most cases. But to harness the power weighted mixed ranking method (WMDD), which was built based upon Mixed Degree Decomposition (MDD) [11] we blend the proposed technique with WMDD method and propose a hybrid model. This WMDD method considers both remaining and exhausted node information in most prevalent k-shell decomposition technique. We blend these Weighted Vote rank algorithms (WVR) and weighted mixed ranking method (WMDD) in a ratio of 70:30 respectively as per below algorithm 2. We take the final returned 'hybrid values' from algorithm 2 and order them in ascending order to get final ranked list.

## 4. EXPERIMENTS AND RESULTS

We evaluated our proposed method by applying it to synthetic and real-world networks and checked its effectiveness by comparing with other ranking methods. Below sections describes in detail about network details, spreading model used and type of comparisons.

### 4.1 Experimental Setup

We have used Compute Optimized instance EC2 instance from Amazon Web Service (AWS) whose model is c5.9xlarge, vCPU is 36 and Memory is 72 GB and EBS disk volume of 1000 GB.

#### 4.1.1 Dataset description

To study the robustness in all type of networks we initially take two synthetic scale-free weighted networks generated by LFR model [23]. The difference between these networks are the parameters used to generated the networks. I.e., we input the number of nodes required for each network and the system generated its own network in random manner with random weights. This gives us unique synthetic networks to test our modes. We also take a real-world dataset for evaluation. All the networks considered here are positive edge weighted ones which indicate relationship strength. The networks taken for consideration are as follows

1. LFR1, is a random synthetic network generated with number of nodes fixed as 100.
2. LFR2, is a random synthetic network generated with number of nodes fixed as 400.
3. Facebook Forum [24], as the name says it is extracted from a popular online social network based on users activity in a forum.
4. Scientific Collaboration [25], this network is from a co-authorship network extracted based on arXiv e-print archive.
5. Airport [26], it is based on US airport network. The nodes in this network represent the number of airport and the weights are based on the number of seats.
6. Health [27], it was constructed based on a questionnaire from school. Based on their response the metrics are summed up and valued network was created. It is basically a kind of school friendship network.
7. USAir [28], It yet another kind of airport network but it was captured in late 90's and created as a network for analysis.
8. Moreno Beach [29], it's a windsurfers network dataset created to analyze social intelligence.

All network properties as listed in table 2.

***Table 2:*** *Synthetic & Real-world network properties of taken for evaluation*

| Network Name | Nodes | Edges | Diameter | Avg Degree | Graph Density | Avg Path Length | Avg Clustering Coefficient |
|---|---|---|---|---|---|---|---|
| **LFR1** | 100 | 350 | 8 | 3.5 | 0.035 | 3.655 | 0.028 |
| **LFR2** | 400 | 1000 | 14 | 2.5 | 0.006 | 6.119 | 0.005 |
| **Facebook Forum** | 897 | 142760 | 4 | 159.153 | 0.178 | 1.88 | 0.696 |
| **Scientific Collaboration** | 16264 | 95188 | 18 | 5.85 | 0.12 | 6.628 | 0.638 |
| **Airport** | 500 | 5960 | 7 | 11.92 | 0.024 | 2.991 | 0.617 |
| **Health** | 1569 | 4794 | 27 | 3.055 | 0.002 | 7.204 | 0.123 |
| **USAir** | 332 | 2127 | 6 | 6.407 | 0.019 | 2.564 | 0.314 |
| **Moreno Beach** | 43 | 336 | 5 | 7.814 | 0.186 | 1.74 | 0.327 |

### 4.1.2 SIR simulator

We use Susceptible–Infected–Recovered (SIR) model to study the effectiveness of the proposed method. In this model susceptible node means the nodes which are about to get information from its neighboring nodes and infected nodes means that it carries information. At start of the process all nodes in the network are set to susceptible once except selected N number of infected nodes. A node which is marked as susceptible which is getting contacted with an infected neighbor becomes an infected node with the infection probability β. In this model recovery probability is denoted as γ and it's is set to one as general case. This means that at any given timestamp t, after spreading the information the infected nodes will move to recovered state at t+1 and they do not participate in spreading process again. The infection probability β is decided based upon the chosen network epidemic threshold $\beta th \simeq \langle k \rangle / \langle k^2 \rangle$. And the β value is chosen above the obtained threshold limit. So, the process starts as at time t=0 selected seed nodes of influential spreaders are considered infected and with successive timestamps its random neighbors are infected. As the neighbors are infected based on randomness, to get robust/consistent value, the same process of random infection is iterated over many times and its average value is taken for final infection of nodes.

### 4.2 Results and Analysis

In this section, we examine both our proposed methods Weighted Vote rank algorithm (WVR) and the hybrid (HYB) method comparing against popular ranking like Weighted Page Rank (WPR), Weighted K-shell (WKS), Neighborhood cores (NC), Weighted MDD (WMDD), Vote Rank (VR) and with centrality measures like Degree centrality (DC). Standard Susceptible–Infected–Recovered (SIR) model is used to simulate the influence spreading in the networks taken for experimentation. The effectiveness is verified using below three different types of experiments.

**4.2.1 Experiment 1: Time series of infection**

The number of nodes infected in a network changes for every time step (t). The measure of infection is the number of nodes in the system which is infected at given timestamp. The infected nodes as per SIR model will continuously infect its neighboring nodes and at next step, they themselves get into recovered state. This process of infection increases rapidly and attain a peak in after time period (t+N). Effective spreaders can quick active this peak. The below equation describes the infection scale F(t).

$$\mathrm{F(t)} = \frac{n_{i\,(t)} + n_{r\,(t)}}{n} \tag{10}$$

Here in above Eq (10), $n_{i\,(t)}$ denotes the number of infected nodes at time (t) and $n_{r\,(t)}$ denotes the number of recovered nodes at time (t).

As we can notice in Fig 4, we have plotted this infection scale F(t) vs time (t) for all the networks. I.e., for synthetic as well as real networks which were taken into consideration. Here while computing these metrics for our proposed method of Weighted Vote ranking algorithm Eq (5) we have used 0.5, 0.25, 0.25 as α, β & γ parameter values respectively and for hybrid ranking we have used 0.7:0.3 ratio as ranking proposition mixing ratio. Also, it has to be noted that for all the networks initial seed nodes are set and this seed node will trigger the spreading process. The infected scale mentioned here is the final snapshot of the network at time (t). I.e., when the process become stable and the number of infected nodes doesn't increase no matter how much ever time is given. We can notice in the Fig 4, that all the networks attain a stable state after time (t). So, this signifies that no more infection propagates beyond that time. Now taking a deeper look into Fig 4 graph, we can notice in almost all graphs our proposed methodology of Weighted Vote rank (WVR) is infecting more nodes than compared to other ranking algorithms. Also, we could see that the hybrid method performs sightly much better than Weighted Vote rank (WVR) and many times better than other traditional algorithms and centrality measures. We can also observe that for networks like 'LFR1' and 'Moreno Beach' the infection rate is comparatively higher than other networks. This is due to that fact that these two networks are relatively small and have many inter connected vertices which makes spreading to larger extent of nodes. In some network like 'Airport' we may notice that all ranking method are infecting lower number of nodes and our proposed methods also act the same, but don't fall low in number of infections. In other networks some traditional ranking methods are infecting higher number of nodes and at the same time our hybrid model and Weighted Vote rank (WVR) also performs equivalent or higher. This shows our proposed method is better in terms of infection scale. Also, table 3 shows maximum infection scale for all networks. The last column shows its average and when we compare the average infection scale with recent proposed methods that is taken for experiment, we get 5.1%-14.5% higher infection scale for Weighted Vote rank (WVR) method and 8.8%-18.2% higher infection scale for Hybrid (HYB) method.

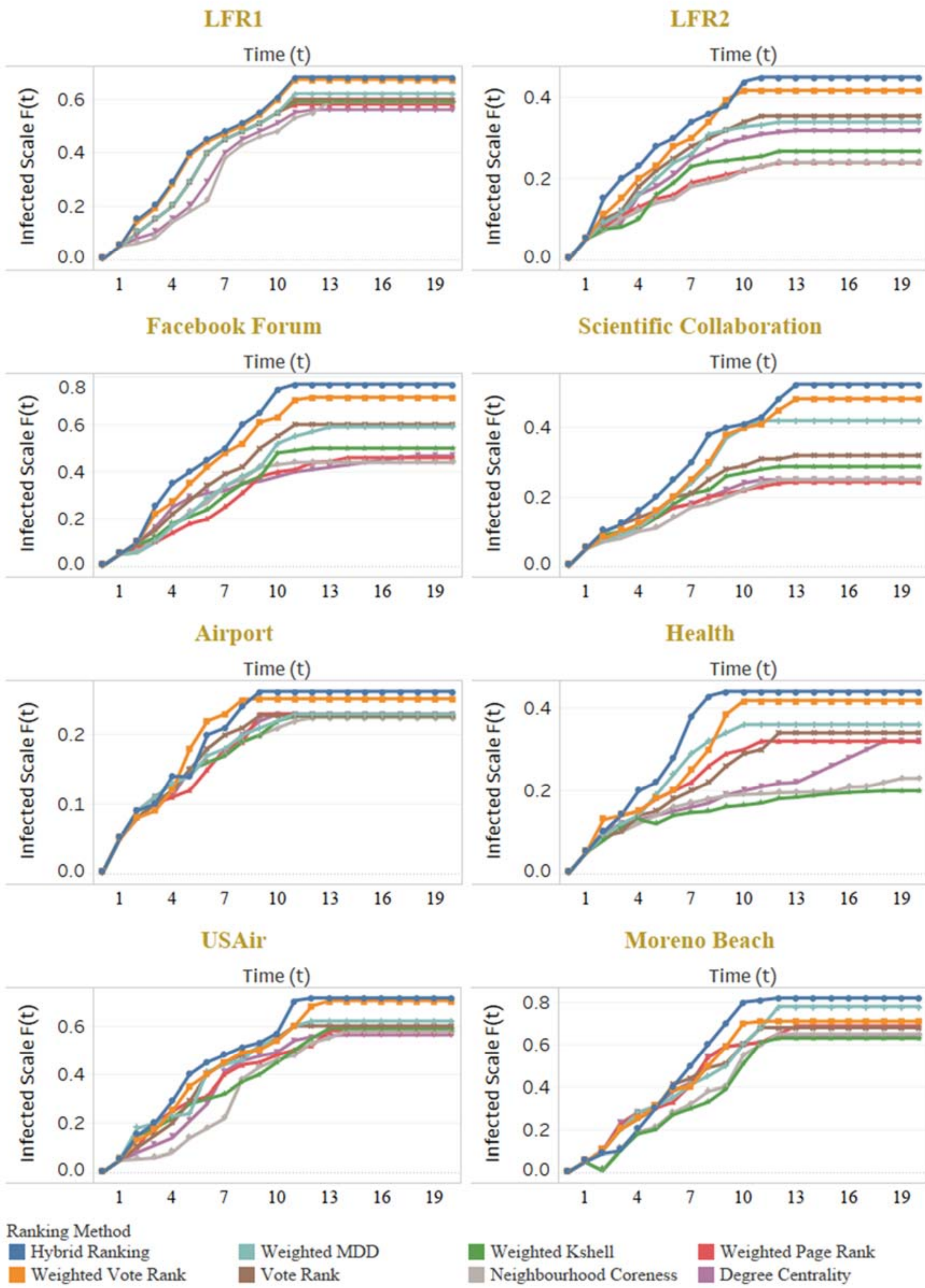


*Figure 4: Infected scale F(t) vs time (t) of all taken networks under the same percentage of seeding*

***Table 3: Maximum infection scale and its average per network for each ranking method based on experiment 1.***

| Ranking Methods | LFR1 | LFR2 | Facebook Forum | Scientific Collab-ration | Airport | Health | USAir | Moreno Beach | Average Infection Scale |
|---|---|---|---|---|---|---|---|---|---|
| **HYB** | 0.683 | 0.451 | 0.77 | 0.525 | 0.263 | 0.44 | 0.715 | 0.82 | **0.5834** |
| **WVR** | 0.672 | 0.418 | 0.715 | 0.483 | 0.252 | 0.418 | 0.704 | 0.71 | **0.5465** |
| **WMDD** | 0.62 | 0.34 | 0.59 | 0.42 | 0.23 | 0.36 | 0.62 | 0.78 | **0.495** |
| **VR** | 0.6 | 0.355 | 0.6 | 0.32 | 0.228 | 0.34 | 0.6 | 0.68 | **0.4654** |
| **WKS** | 0.59 | 0.268 | 0.5 | 0.288 | 0.227 | 0.2 | 0.59 | 0.63 | **0.4116** |
| **NC** | 0.59 | 0.24 | 0.44 | 0.25 | 0.225 | 0.23 | 0.58 | 0.65 | **0.4006** |
| **WPR** | 0.58 | 0.24 | 0.46 | 0.244 | 0.23 | 0.32 | 0.58 | 0.688 | **0.4178** |
| **DC** | 0.56 | 0.319 | 0.467 | 0.25 | 0.23 | 0.32 | 0.563 | 0.64 | **0.4186** |

**4.2.2 Experiment 2: Infected scale vs spreader seed nodes**

Here in this experiment, we try to measure the final infected scale by varying initial spreader seed nodes. As mentioned above SIR (Susceptible–Infected–Recovered) model which takes c as initial seed node, and β as infection probability and γ as recovery probability is used. Here γ defines the rate at which infected nodes get into recovered state. In this entire spreading process, we set γ as 1, which means that any nodes that is moves into infected state from susceptible state will definitely get moved to recovered state at time t+1. Hence the final number of infected scale or final recovered scale can be defined as the list of nodes which was initially infected and then moved to a recovered state.

$$\mathrm{F(t)} = \frac{n_{r\,(t)}}{n} \qquad (11)$$

Where $n_{r\,(t)}$ represents the number of recovered nodes at the end of infection process and n is the total number of nodes of the taken network for evaluation. It is to be noted that if SIR infection process ends at time 't', the final scale of count are the nodes are computed as the nodes which become infected and get recovered at time 't+1'. Similar to previous experiment while computing these metrics for our proposed method of Weighted Vote ranking algorithm Eq (5) we have used 0.5, 0.25, 0.25 as α, β & γ parameter values respectively and for hybrid ranking we have used 0.7:0.3 ratio as ranking proposition mixing ratio.

Fig 5 shows the effect of increase/decrease in spreader fraction to the infection scale which was computed using Eq (5) for weighted vote ranking algorithm. As like previous experiment we notice that the infection scale is higher for small networks irrespective of ranking algorithm. We can notice in this Fig 5 that, some algorithm method like weighted k-shell perform inconsistently over various networks. For instance, its shows good progress in LFR2 network and is well below the other ranking methods in health network. Similarly, centrality measure like degree centrality generally shows that the infection scale is consistently lower in all networks. Now when we consider of Weighted Vote rank method it's in almost all networks it performs consistently higher than other predominant ranking methodologies. We can notice that this proposed Weighted Vote rank method performs sometimes equal to other traditional ranking methods at smaller seed node fraction in networks like Airport, but later when spreader fraction increases the infection/recovery scale also increase up higher than other networks. This is almost the case in all other networks too. Also, when we check the hybrid ranking method it has slightly better infection and recovery scale than Weighted Vote rank method. In some cases, like Health network, it is almost equal to Weighted Vote rank method. So, based on all these observations we can conclude that both the proposed Weighted Vote rank method and hybrid ranking method performs better in spreading information over varying initial seed node fraction when compared to other ranking methods.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

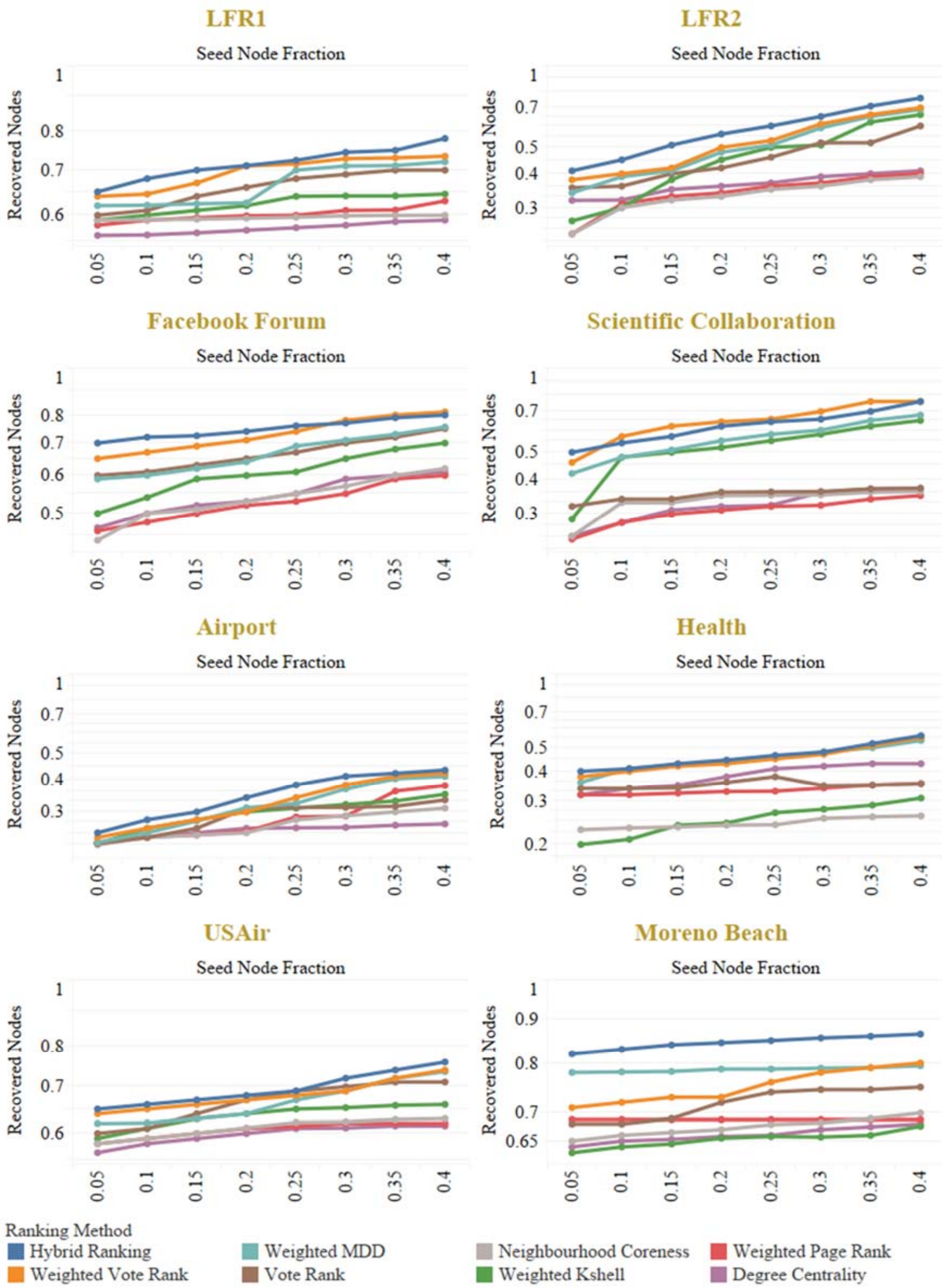


*Figure 5: Recovered scale by varying the number of seed nodes.*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

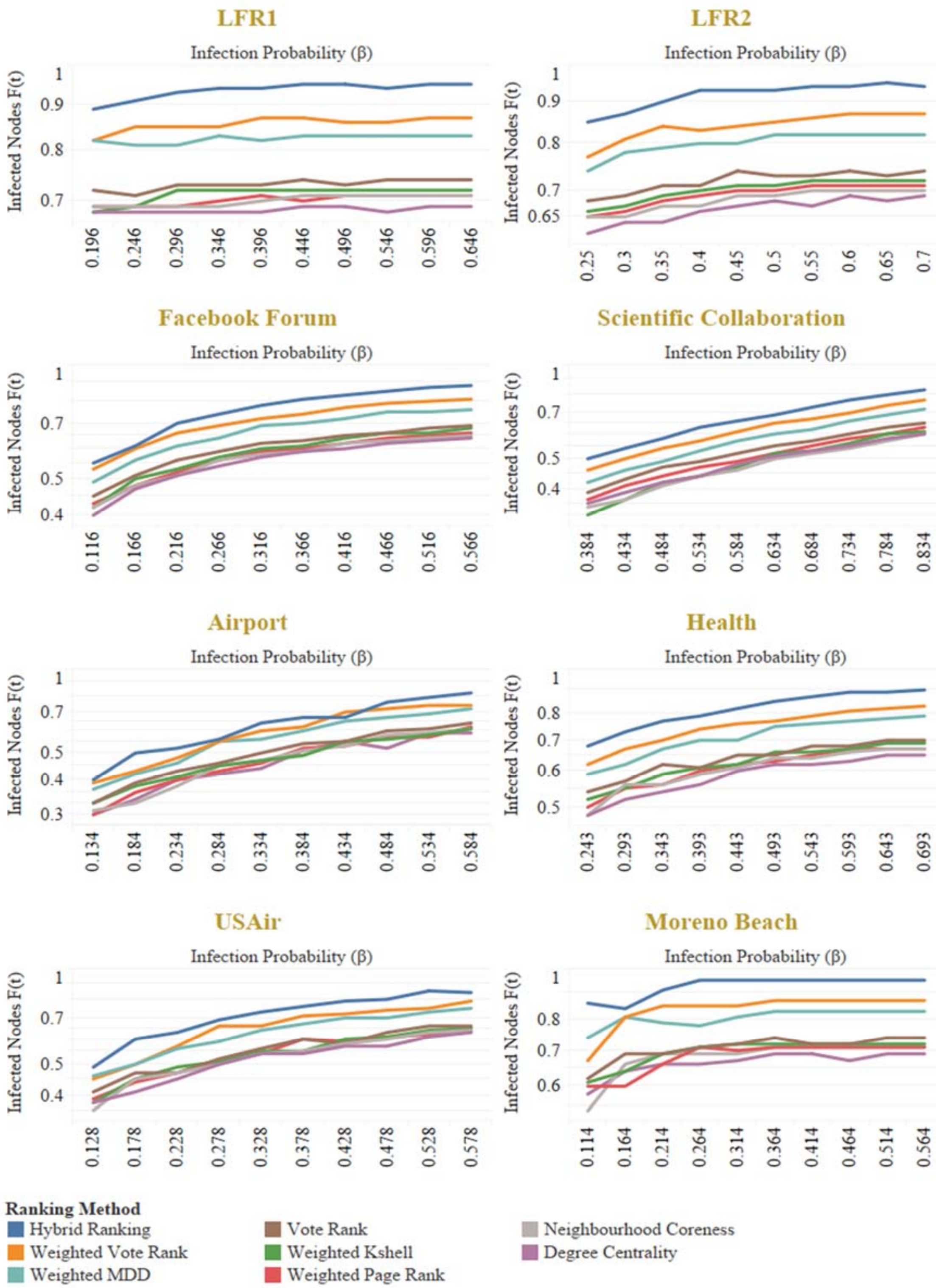


*Figure 6: The final infected number F(t) of nodes for various infection probability Beta (β)*

#### 4.2.3 Experiment 3: Infected scale over various infection probability

A node which is marked susceptible passes the acquired information to its neighbors with the infection probability β. This β value for the network in the SIR model is defined as using the networks epidemic threshold value which is calculated as Eq (12). Now in this experiment we are going to vary this βth value (*above the calculated threshold limit*) and check the scale of infection against various networks for all the ranking methods. This will show the effectiveness of the proposed methods against the other ranking methods.

$$\beta\text{th} \simeq \frac{\text{k}}{k^2} \quad (12)$$

Where k is the average degree of the network and $k^2$ second order average degree.

Fig 6 show the such infection scale obtained over varying β values. In this in almost all networks we can notice that the spreading rate is uniform (i.e., gradually increasing) across all ranking methods for varying infection probability. Specially we could notice that two synthetic networks LFR1 and LFR2 exhibit very similar trend. Since both are synthetic networks their node interconnection property might have led to this trend. But ignoring these trend similarities and when we notice the infection scale over ranking methods, we can see proposed Weighted Vote raking method and hybrid methods performs much higher infection in both these networks. When we take a look at scientific coloration and airport network infection scale is not too higher but performs better comparatively with other ranking methods. We could also notice in Airport network that for few β values infection scale of hybrid ranking method is almost equal to Weighted Vote rank and weighted MDD methods. But in Moreno beach network which is relatively small when compared to all other network, both proposed methods hybrid and Weighted Vote rank method's infection scale is clearly higher than other ranking methods. So, when we take an overall looks the both proposed methods hybrid and Weighted Vote rank methods perform equal or better than traditional ranking methods for varying infection probability (β).

#### 4.2.4 Experiment 4: Kendall rank correlation coefficient

Kendall's Tau (τ ) correlation coefficient [30], [31] measures the correlation between two raking list by considering the position at which the nodes are listed in the ranking list. It is calculated by counting the number of concordant and discordant pairs between the two ranked lists. For example, let's consider two rank list named List1 and List2. Let (x1, y1) and (x2, y2) be a set of ranks from this list. If x1>x2 and y1>y2 or x1<x2 and y1<y2 then the above rank pairs (x1, y1) and (x2, y2) are marked as concordant. In case if x1>x2 and y1<y2 or x1<x2 and y1<y2, they are marked as discordant. In some cases, x1=x2 or y1=y2 then it is neither concordant or discordant. The Kendall's Tau (τ) is defined as below.

$$\tau = \frac{2 * (N1 - N2)}{N * (N - 1)} \quad (13)$$

Where N1 and N2 are the number of concordant and discordant pairs and N is the network size. It is to be noted that τ value ranges from -1 to +1. Higher the value signifies greater similarity and lower signifies dissimilarity. Based on this above defined Eq 13, Kendall's Tau (τ) correlation coefficient for various ranking methods against Susceptible Infected Recovered (SIR) epidemic model has been listed in Table 4 along with proposed method (WVR) and its extended hybrid method (HYB). In this table 4 we can notice that the correlation coefficient of proposed methods across various networks is comparatively higher than other methods like Weighted Mixed Degree Decomposition (WMDD) and Neighborhood coreness (NC). And also, much better than other traditional ranking methods like Degree Centrality (DC) and Weighted Page Ranking (WPR) where the ranks are generally computed primarily based on edges alone.

We can also notice that τ (WMDD) is better when compared to proposed method τ(WVR) in 'LFR1' and 'Moreno Beach' networks, but the extended hybrid τ(HYB) method has better correlation coefficient than τ (WMDD) in those cases. On computing an average correlation coefficient across networks, we get 0.861 and 0.836 for proposed Hybrid (HYB) and Weighted Vote Rank (WVR) methods respectively, where as the average Kendal Tau for other methods are lower than it.

*Table 4: Kendall's Tau (τ) values for various datasets*

| Ranking Methods | LFR1 | LFR2 | Face-book Forum | Scientific Collab-ration | Airport | Health | USAir | Moreno Beach | Average τ |
|---|---|---|---|---|---|---|---|---|---|
| τ(HYB) | 0.82021 | 0.84274 | 0.89046 | 0.85393 | 0.85891 | 0.8702 | 0.87172 | 0.88517 | **0.86167** |
| τ(WVR) | 0.80916 | 0.83158 | 0.85406 | 0.81193 | 0.83992 | 0.84076 | 0.84621 | 0.85885 | **0.83656** |
| τ(VR) | 0.6944 | 0.73701 | 0.71492 | 0.72485 | 0.70531 | 0.62348 | 0.69818 | 0.76362 | **0.70772** |
| τ(WKS) | 0.51299 | 0.65545 | 0.55063 | 0.59652 | 0.58744 | 0.59604 | 0.61373 | 0.55141 | **0.58303** |
| τ(WPR) | 0.4133 | 0.41385 | 0.39536 | 0.50713 | 0.38015 | 0.38118 | 0.35727 | 0.49488 | **0.41789** |
| τ(DC) | 0.4225 | 0.32201 | 0.35867 | 0.40939 | 0.43753 | 0.45066 | 0.41099 | 0.4948 | **0.41332** |
| τ(WMDD) | 0.81394 | 0.78726 | 0.75711 | 0.76304 | 0.79519 | 0.77754 | 0.81383 | 0.87674 | **0.79808** |
| τ(NC) | 0.7234 | 0.71481 | 0.75749 | 0.70775 | 0.70589 | 0.74567 | 0.72761 | 0.77894 | **0.7327** |

### 4.3 Limitations of the Study

Here in this paper, we have studied the problem of effectively identifying influential users and addressed it with our proposed methodologies. While the proposed methods were performing well in above experiments, we would like to put forth, some of the key study limitations.

As like most of the research works in the domain of identifying influential users, we also used Susceptible Infected Recovered (SIR) epidemic model as bench mark. Though this SIR is the most common scenario in real world cases, there are few instances in networks like were we need users to perform repeated infections. i.e., Even after the infection is completed in first round, the same infected users instead of going to recovered state, they become infected again and in turn infect their neighbors and this spreading process goes till time period t. In those kinds of scenarios/networks, we would a ranking algorithm which was evaluated by Susceptible Infected Susceptible (SIS) epidemic model and this ranking algorithm may not be best suit for those cases.

Though the tunable parameters and the hybrid method proposition give the flexibility for end users to configure the weightages dynamically, one needs to be cautious in choosing right values between (0-1) to achieve best performance of the proposed algorithm for those variables, as it can't be a static one for all kinds of use cases.

One another key factor we see here is, all these above computations are performed on snapshot of the network. It is no doubt that the proposed methodologies will work on given same structure of the network. But in real world cases, the links and other parameters changes time to time. So, applying/simulating information dissimilation on a highly modified network will not yield desired result of influence maximization. This is not a limiting factor to proposed method alone, instead applies to all most all methods proposed in this domain of influence maximization problem.

### 4.4 Results Summary

The proposed weighted vote rank method is designed in such a way that it does not take the network structure alone for influential user identification, but in turn use network's other metrics like neighbor coreness and weights with appropriate parameter control. This design helped to identify influential users not only from the core of the network, but also lower/outer shell nodes too. Generally other existing methods like WKS and NC identified nodes from core nodes only, which is why the spreading ability of other ranking methods is comparatively lower than proposed once. The experiment 4 results also show that hybrid method has high ranking co-relation with the benchmark SIR compared to existing ranking methods VR, WKS, WPR, DC, WMDD and NC. Also, we could notice that the extension, hybrid method performs better than the base weighted vote rank method as it has features of both weighted vote rank and degree decomposition method. One more key reason for our proposed method to perform better in terms of infection scale is due to the basic characteristics of vote rank. The vote rank works based on voting principle from its neighbors and hence the top ranked nodes are by default not located in near to each other and are sparsely located. Adding weights and neighborhood coreness still effectively identifies vital nodes even located at outer ring of the network. So, when an infection process is initiated from top ranked nodes of weighted vote rank method, the infection happens from both core of the network as well as from strongly weighted and connected outer nodes. Whereas in most of the other method infection process is sparkled only from core nodes. This tactical location of influential nodes is a greater advantage of weighted vote rank compared to other method. We can also observe that our proposed method performs better on varying the number of seed nodes too. From these above results of all the experiments, we summarize that proposed weighted page rank identifies influential users effectively.

## 5. CONCLUSION

The key objective of this research is to propose a better ranking method which effectively identifies top influential users who will be able to achieve maximum information propagation in a given network. We hypothesized that usual ranking methods which just considers structural information of the network will not be sufficient enough to narrow down appropriate influential users. To build a more robust process of identifying influential users, we would need to consider networks other key metrics like weight, neighborhood coreness in appropriate proposition. Also, we mentioned that like other prominent k-shell algorithms having the top ranked nodes in core of the network alone will not be sufficient and we would need identify influential users even from outer shell of the network too. Based on these above hypotheses, we have built a Weighted Vote Rank and a Hybrid version using Weighted Mixed Degree Decommission technique of it as an extension of it. This Weighted Vote Rank was clearly helpful in achieving the objective of the research. I.e., as we hypothesized this algorithm took all three key factors of a network such as support (vote), relationship strength (weight) and nodes position in the network (coreness) and ranked influential users. Only on considering these additional metrics, it was able to locate influential users much co-related to our benchmark Susceptible Infected Recovered (SIR) epidemic model and this could be seen in experiment 4. Also, it was able to locate users in inner core like most algorithms as well as located outer shell nodes, only this paved way to have better infection scale when compared to other ranking methods in experiments 1,2 and 3. Also the blending with Weighted Mixed Degree Decomposition method (WMDD) was acting as a support factor for our prosed Weighted Vote Rank (WVR) algorithm in Hybrid (HYB) method. This could be noticed in few instances of experiment 1 and 2 data points, wherever WVR was lacking, WMDD helped to find right influential users. HYB method made sure that networks coreness importance is never lost, as it considered WMDD in appropriate proposition.

It is also to be noted that the proposed weighted vote ranking methods mainly considers three key values such as voting ability score, weight and the neighborhood coreness. All these three values weightages are controlled by its respective parameters α, β and γ. These tunable parameters give the algorithm a flexibility to control the importance given to respective values. And the hybrid ranking method is an extension of our proposed Weighted Vote ranking algorithm works by combining the ranks that are generated by weighted vote ranking algorithm and weighted mixed degree decomposition method. This gave the hybrid method to extract the goodness of both these methods and yield better result.

We have evaluated our both the proposed methods against various prominent ranking and traditional centrality measures. The various experiment results such as infection scale vs time, recovery scale vs number of seed nodes and infected scale with varying infection probability shows that both our proposed methods better in synthetic and real-world networks. So, we re-iterate that combination of key properties of networks nodes yield us better ranking method. Being said that we also believe that inclusion of other key properties like node creation time, relationship history over etc., also may play a major role. So, in future we would need to experiment and extend our work based on these factors too.

## REFERENCES:

[1] R. Albert and A.-L. Barabási, “Statistical mechanics of complex networks,” *Rev. Mod. Phys.*, vol. 74, no. 1, pp. 47–97, Jan. 2002, doi: 10.1103/RevModPhys.74.47.

[2] L. Lü, Y. Zhang, C. H. Yeung, and T. Zhou, “Leaders in social networks, the Delicious

case.," *PLoS One*, vol. 6, no. 6, p. e21202, Jan. 2011, doi: 10.1371/journal.pone.0021202.

[3] D. Chen, L. Lü, M. Shang, Y. Zhang, and T. Zhou, "Identifying influential nodes in complex networks," *Phys. A Stat. Mech. its Appl.*, vol. 391, no. 4, pp. 1777–1787, Feb. 2012, doi: 10.1016/j.physa.2011.09.017.

[4] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Comput. networks ISDN Syst.*, vol. 30, no. 1–7, pp. 107–117, 1998.

[5] S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes, "K-core organization of complex networks," *Phys. Rev. Lett.*, vol. 96, no. 4, pp. 3–6, 2006, doi: 10.1103/PhysRevLett.96.040601.

[6] A. Namtirtha, A. Dutta, and B. Dutta, "Identifying influential spreaders in complex networks based on kshell hybrid method," *Phys. A Stat. Mech. its Appl.*, vol. 499, pp. 310–324, 2018, doi: 10.1016/j.physa.2018.02.016.

[7] J. X. Zhang, D. B. Chen, Q. Dong, and Z. D. Zhao, "Identifying a set of influential spreaders in complex networks," *Sci. Rep.*, vol. 6, no. February, pp. 1–10, 2016, doi: 10.1038/srep27823.

[8] M. Wang, W. Li, Y. Guo, X. Peng, and Y. Li, "Identifying influential spreaders in complex networks based on improved k-shell method," *Phys. A Stat. Mech. its Appl.*, vol. 554, p. 124229, 2020, doi: 10.1016/j.physa.2020.124229.

[9] G. Maji, "Influential spreaders identification in complex networks with potential edge weight based k-shell degree neighborhood method," *J. Comput. Sci.*, vol. 39, p. 101055, 2020, doi: 10.1016/j.jocs.2019.101055.

[10] A. Namtirtha, A. Dutta, and B. Dutta, "Weighted kshell degree neighborhood: A new method for identifying the influential spreaders from a variety of complex network connectivity structures," *Expert Syst. Appl.*, vol. 139, p. 112859, 2020, doi: 10.1016/j.eswa.2019.112859.

[11] A. Zeng and C. J. Zhang, "Ranking spreaders by decomposing complex networks," *Phys. Lett. Sect. A Gen. At. Solid State Phys.*, vol. 377, no. 14, pp. 1031–1035, 2013, doi: 10.1016/j.physleta.2013.02.039.

[12] P. Bonacich, "Factoring and weighting approaches to status scores and clique identification," *J. Math. Sociol.*, vol. 2, no. 1, pp. 113–120, Jan. 1972, doi: 10.1080/0022250X.1972.9989806.

[13] S. Gao, J. Ma, Z. Chen, G. Wang, and C. Xing, "Ranking the spreading ability of nodes in complex networks based on local structure," *Phys. A Stat. Mech. its Appl.*, vol. 403, pp. 130–147, 2014.

[14] L. C. Freeman, "Centrality in social networks conceptual clarification," *Soc. Networks*, vol. 1, no. 3, pp. 215–239, 1978.

[15] G. Sabidussi, "The centrality index of a graph," *Psychometrika*, vol. 31, no. 4, pp. 581–603, 1966.

[16] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *J. ACM*, vol. 46, no. 5, pp. 604–632, 1999, doi: 10.1145/324133.324140.

[17] 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*, 2003, pp. 137--146, doi: 10.1145/956755.956769.

[18] D.-B. Chen, H. Gao, L. Lü, and T. Zhou, "Identifying Influential Nodes in Large-Scale Directed Networks: The Role of Clustering," *PLoS One*, vol. 8, no. 10, pp. 1–10, 2013, doi: 10.1371/journal.pone.0077455.

[19] Z. Wang, Y. Zhao, J. Xi, and C. Du, "Fast ranking influential nodes in complex networks using a k-shell iteration factor," *Phys. A Stat. Mech. its Appl.*, vol. 461, pp. 171–181, 2016, doi: https://doi.org/10.1016/j.physa.2016.05.048 .

[20] W. Chen, Y. Wang, and S. Yang, "Efficient Influence Maximization in Social Networks," *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. 67, no. 1, p. 199, 2009, doi: 10.1145/1557019.1557047.

[21] S. Kumar and B. S. Panda, "Identifying influential nodes in Social Networks: Neighborhood Coreness based voting approach," *Phys. A Stat. Mech. its Appl.*, vol. 553, p. 124215, 2020, doi: 10.1016/j.physa.2020.124215.

[22] S.-J. N. Xiao-Li Yan, Ya-Peng Cui, 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. 48902, Apr. 2020, doi: 10.1088/1674-1056/ab77fe.

[23] A. Lancichinetti and S. Fortunato, “Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities,” *Phys. Rev. E*, vol. 80, no. 1, p. 16118, Jul. 2009, doi: 10.1103/PhysRevE.80.016118.

[24] T. Opsahl, “Triadic closure in two-mode networks: Redefining the global and local clustering coefficients,” *Soc. Networks*, vol. 35, no. 2, pp. 159–167, 2013.

[25] M. E. J. Newman, “The structure of scientific collaboration networks,” *Proc. Natl. Acad. Sci.*, vol. 98, no. 2, pp. 404–409, 2001.

[26] V. Colizza, R. Pastor-Satorras, and A. Vespignani, “Reaction–diffusion processes and metapopulation models in heterogeneous networks,” *Nat. Phys.*, vol. 3, no. 4, pp. 276–282, 2007.

[27] J. Moody, “Peer Influence Groups: Identifying Dense Clusters in Large Networks,” *Soc. Networks*, vol. 23, no. 4, pp. 261–283, 2001.

[28] R. Rossi and N. Ahmed, “The network data repository with interactive graph analytics and visualization,” in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015, pp. 4292–4293.

[29] L. C. Freeman, S. C. Freeman, and A. G. Michaelson, “On human social intelligence,” *J. Soc. Biol. Struct.*, vol. 11, no. 4, pp. 415–425, 1988.

[30] W. R. R. Knight, “A Computer Method for Calculating Kendall’s Tau with Ungrouped Data,” *J. Am. Stat. Assoc.*, vol. 61, no. 314, pp. 436–439, 1966, doi: 10.1080/01621459.1966.10480879.

[31] M. G. Kendall, “The treatment of ties in ranking problems,” *Biometrika*, vol. 33, no. 3, pp. 239–251, Nov. 1945, doi: 10.1093/biomet/33.3.239.