Enhancement of Voting Scores with Multiple Attributes Based on VoteRank++ to Identify Influential Nodes in Social Networks

Pham Van Duong[1,3], Tuan Minh Dang[2,3], Le Hoang Son[4], and Pham Van Hai[1](✉)

[1] School of Information and Communication Technology, Hanoi University of Science and Technology, Hanoi, Vietnam
haipv@soict.edu.vn

[2] Posts and Telecommunications Institute of Technology, Km10, Nguyen Trai Street, Hanoi, Vietnam
tuandm@ptit.edu.vn

[3] CMC Institute of Science Technology, No. 11 Duy Tan Street, Hanoi, Vietnam
dmtuan1@cmc.com.vn

[4] VNU University of Science, Vietnam National University, Hanoi, Vietnam
sonlh@vnu.edu.vn

Abstract. With the prosperity of social networks, Influence maximization is a crucial analysis drawback within the field of network science due to its business value. In this regard, we propose the EAVoteRank++, inspired by VoteRank++, to iteratively select the influential node. It is commonly recognized that degree is a well-known centrality metric for identifying prominent nodes, and neighbors' contributions should also be considered. Furthermore, topological connections between neighbors have an impact on node spreading ability; the more connections between neighbors, the higher the risk of infection. Therefore, EAVoterank++ algorithm identify nodes's voting ability by considering degree, position in network by improve k-shell decomposition and clustering coefficient as well as neighbors. The weights of attribute are calculated by entropy technology. Furthermore, based on VoteRank++, EAVoteRank++ minimizes the voting ability of 2-hop neighbors of the selected nodes to decrease the overlapping of influential regions of spreaders. To demonstrate the effectiveness of the proposed method, we employ both the SIR model and LS model to simulate the spreading progress, then calculate the accuracy of the proposed algorithm, compare with other methods. The experimental results with 2 propagation simulation models on 6 datasets demonstrate the good performance of the proposed method on discrimination capability and accuracy.

Keywords: Influence maximization · Influential node · Social networks · Voting approach · Spreading model · Multiple attribute · Information entropy

1 Introduction

The modalities of communication between individuals and the methods in which information is generated are fast evolving, due to the continued growth of the Internet. In

A. L. Pinto and R. Arencibia-Jorge (Eds.): DIONE 2022, LNICST 452, pp. 242–257, 2022.
https://doi.org/10.1007/978-3-031-22324-2_19

particular, social media platforms like Facebook, Twitter, Tiktok, as well as a burgeoning landscape of microblogs, have aided in the consumption and sharing of information [1]. As a result, studies on social networks have received a lot of interest from academic and industrial groups. Researchers regconized information may be transferred through the links in a social network, potentially causing a chain reaction [2]. Many algorithms have been studied and proposed with different approaches to solve influence maximation. One of these approaches is to rank the nodes in the graph according to their importance to determine the set of important nodes in the network. The researchers employ two methodologies: network embedding and node centrality.

Early approaches for learning node representations mostly relied on matrix decomposition. These solutions, however, come at a high expense in terms of computing. DeepWalk [3] is a pioneer of network embedding methods, which uses word2vec to generate low-dimensional representation vectors of nodes using random walk sequences. Node2vec improves on the DeepWalk algorithm by integrating breadth-first and depth-first search into the random walk sequence generation. Besides, SDNE [4] uses deep neural networks to model the nonlinearity between nodes. For node centrality method, this kind of method usually express the significance of one node with a numerical value and finally outputs a ranking list. Degree centrality [5] (DC) is the simplest method to determine node influence, this metric focuses only on the number of neighbors of the node. In addition, there are CC and BC algorithms, which use the local information of each node to calculate the influence of that node. K-shell and some improved methods are representative methods based on local network topology to determine the importance of nodes according to the position of nodes in the network. Furthermore, algorithms such as eigenvector centrality (EC) [6] and HITS [7] have been explored and proposed to enhance node ranking. Recently, several academics suggested a novel perspective on the influence index of network nodes. Ma et al. [8] suggested two methods based on the law of gravity, which take into account both neighborhood influence information and graph connectivity information.

In general, ranking algorithms assume that a set of important nodes will solve the IM problem, but some problems. These algorithms are likely to choose a set of influential nodes in the same locations, for example. The distances between multiple spreaders have a crucial role in maximizing the impact, according to Kitsak et al. As a result, selecting influential nodes throughout a network is a wise idea. The main concept is to identify a set of influential nodes, known as seed nodes, that are distributed across the network. To identify influential nodes in the network, Zhang et al. [9] presents the VoteRank algorithm, which is based on the voting mechanism. Guo et al. [10], based on the VoteRank algorithm, proposed the EnRenew algorithm, which selects seed nodes based on the information entropy of the nodes and achieves a significant improvement over the original technique. VoteRank is one of these algorithms that has gotten a lot of interest since it gives a quick and easy solution to the IM issue. Then a proposed VoteRank++ [11], an improvement to the VoteRank algorithm. This method is improved in two ways: (1) a node's voting ability is connected to its relevance in the initiation process, and (2) each node can vote for its neighbors differently in the voting process. Also, to reduce the computation time of the algorithm, we update only part of the nodes in each iteration instead of all the nodes. Based on an improved idea inspired by VoteRank++, this paper proposes a new algorithm named EAVoteRank++, which to repeatedly select influential nodes. The influence of neighboring nodes is ignored by the centrality measurements. It is generally recognized that magnitude is a well-known central metric for identifying prominent nodes and that the contributions of neighboring countries should also be considered. Furthermore, topological connections between neighbors have an impact on the propagation capacity of a node. The more connections there are between neighbors, the higher the risk of infection. Therefore, the EAVoterank++ algorithm determines the voting ability of the nodes, considering the degree, the position in the network by improving the k-shell decomposition and clustering coefficient as well as the neighborhoods. Attribute weights are calculated using entropy technology. Furthermore, based on VoteRank++, EAVoteRank++ minimizes the voting capacity of the 2-step neighbors of the selected nodes to reduce the overlap of the affected regions of the spreader. To demonstrate the effectiveness of the proposed method, we use both SIR to simulate the progression of the spread. The test results on 6 datasets prove the good performance of the proposed method in terms of discriminant ability and accuracy.

The rest of this paper is written out as follows. Section 2 discusses the concentration measuring algorithm, whereas Sect. 3 discusses the proposed algorithm and Sect. 4 discusses its performance evolution. Finally, in Sect. 5, we discuss our results and future improvements.

2 Backgrounds

Given a network $G = \langle V, E\rangle$ with G is an undirected and unweighted network, where V and E represent nodes and edges. We denote $A = (a_{ij})_{N\times N}$ is the adjacency matrix of G. If there is an edge between node i and node j then $a_{ij} = 1$, otherwise $a_{ij} = 0$.

2.1 Centrality Measures

Degree Centrality (DC) [4] is defined as the number of edges occurring on a node is known as the number of edges node. $DC(i)$ of node i can be calculated by:

$$DC(i) = k(i) = \sum_j a_{ij} \tag{1}$$

where $k(i)$ is the degree of node i.

K-Shell Decomposition: [12] This approach can be considered as a node degree-based coarse graining sorting algorithm. The following is the specific decomposition procedure: The initial stage in KS is to remove all nodes in the network with a degree of 1 from the network. Then, after one round of removal, it removes nodes with a degree of $k \leq 1$ since this step may cause the degree values to be reduced throughout the removal process. All nodes deleted in this stage generate 1-shell and their k-shell values are equal to one until there are no nodes in the network with degree $k \leq 1$. Then repeat the process to get two shells, three shells, and so on. Finally, all nodes are separated into distinct shells, and each node's k-shell value may be calculated.

Improve K-Shell Decomposition
[13] proposed he improved k-shell index of node i, denoted by $k_s^*(i)$, can be calculated by:

$$k_s^*(i) = k_s(i) + \frac{p(i)}{q(k) + 1} \tag{2}$$

where $p(i)$ is the stage in a k-shell layer decay for which node i is removed from the graph, $q(k)$ is the set of the number of stages in the k-shell decay.

Local Clustering Coefficient: Other measurements can be computed for each node and used to create a distribution function, but the most common is degree. A node's clustering coefficient [14] estimates how many vertices will more often than not be grouped together. There are two types of clustering coefficients, global and local. In this paper, we use the local clustering coefficient. Specifically, given a vertex v and và $d_v = |N_v|$ its degree. The local clustering coefficient C_v of node v can be calculated as follows:

$$C_v = \frac{2.\left|\left\{v', v''\right\} \in E(G) : v', v'' \in N_v\right|}{d_v(d_v - 1)} \tag{3}$$

H-Index
The H-index [15] of a node is defined to be the maximum value h such that there exists at least h neighbours of degree no less than h. H is a function that works on a finite number of reals $(x_1, x_2, \ldots, x_n)$ and returns an integer $y = H(x_1, x_2, \ldots, x_n)$, where y is the largest integer such that $(x_1, x_2, \ldots, x_n)$ has at least y items, each of which is greater than y. $x_1, x_2, .., x_n$ is the number of citations to these articles for a scholar with n publications, and $H(x_1, x_2, \ldots, x_n)$ the scholar's H-index.

$$h_i = H(k_{j1}, k_{j2}, \ldots, k_{ji}) \tag{4}$$

We define $h_i^{(0)} = k_i$ to be the zero-order H-index of node i, and define the n-order H-index iteratively as

$$h_i^{(n)} = H(h_{j1}^{(n-1)}, h_{j2}^{(n-1)}, \ldots, h_{jk_i}^{(n-1)}) \tag{5}$$

The H-index of node i is equal to the first-order H-index, namely$h_i^{(1)} = h_i$.

MCDE
The MCDE [16] algorithm considers a combination of the indices of the node location in the network, the node's order coefficient, and that node's entropy. This algorithm is used to rank the nodes in the network. MCDE of node v is calculated by the formula

$$MCDE(v) = \alpha KS(v) + \beta DC(v) + \gamma Entropy(v) \tag{6}$$

$$Entropy(v) = -\sum_{i=0}^{KS_{max}} (p_i * log_2 p_i) \tag{7}$$

$$p_i = \frac{the\ number\ of\ neighbors\ of\ node\ v\ in\ the\ i - th\ KS}{DC(v)} \tag{8}$$

ECRM

The ECRM [17] algorithm is an improved cluster ranking approach presented taking into account the common hierarchy of nodes and their neighborhoods. The influence score of a node in the network applying the ECRM algorithm is calculated as follows:

$$ECRM_i = \sum_{v_j \in N_i} CRM_j \tag{9}$$

$$CRM_i = \sum_{v_j \in N_i} SCC_j \tag{10}$$

$$SCC_i = \sum_{v_j \in N_i} \left(\left(2 - C_{i,j}\right) + \left(2 \frac{d_j}{max(d)} + 1\right) \right) \tag{11}$$

$$C_{i,j} = \frac{\sum_{k=1}^{f} \left(SV_i^{(k)} - \overline{SV_i}\right)\left(SV_j^{(k)} - \overline{SV_j}\right)}{\sqrt{\sum_{k=1}^{f} \left(SV_i^{(k)} - \overline{SV_i}\right)^2} \sqrt{\sum_{k=1}^{f} \left(SV_j^{(k)} - \overline{SV_j}\right)^2}} \tag{12}$$

$$SV_i = \left\{ \left|N_i^{(1)}\right|, \left|N_i^{(2)}\right|, \left|N_i^{(3)}\right|, \ldots \left|N_i^{(f)}\right| \right\} \tag{13}$$

where $ECRM_i$ is the influence score of node i calculated by ECRM algorithm, CRM_i is the coefficient of cluster coefficient ranking measure determined by the cascade clustering coefficients of the neighbors SCC_i, $C_{i,j}$ is Pearson's correlation coefficient among class vectors SV_i.

EnRenew

These two concepts are fed into the complex network in order to calculate node importance [10]. The information entropy of any node v can be calculated in

$$E_v = \sum_{u \in \Gamma_v} H_{uv} = \sum_{u \in \Gamma_v} -p_{uv} log p_{uv} \tag{14}$$

$p_{uv} = \frac{d_u}{\sum_{l \in \Gamma_v} d_l}$, $\sum_{l \in \Gamma_v} p_{lv} = 1$ and Γ_v is the set of neighbors of node$v$, d_u is the degree centrality of the node u, H_{uv} is the ability to propagate information from node u to node$v$. After each high impact node is selected, the algorithm improves the information entropy of all nodes in its local range according to the formula

$$H_{u^{l-1}u^l} - = \frac{1}{2^{l-1}} \frac{H_{u^{l-1}u^l}}{E_{\langle k \rangle}} \tag{15}$$

where $E_{\langle k \rangle} = -\langle k \rangle . \frac{1}{\langle k \rangle} . log \frac{1}{\langle k \rangle}$ where $\langle k \rangle$ is the average of the degree centrality of the entire network, $\frac{1}{2^{l-1}}$ is the coefficient decreases.

VoteRank

VoteRank algorithm [9] used to identify influential nodes in an unweighted network using the idea of voting process. A pair of values (vs_v, va_v) representing each node $v \in V$, where vs_v and va_v correspond to the influence score voted on by the neighbors and the node's voting power score. v. The voted influence score vs_v of a node v is equal to the sum of the voting possibility scores of all its direct neighbors, which is determined by the formula:

$$vs_v = \sum_{u \in \Gamma_v} va_u \tag{16}$$

where Γ_v is the set of neighbors of node v. VoteRank algorithm is divided into 4 main stages.

Initialization: The influence score vs_v of each node v is initialized by default to 0, and the voting ability of each node va_v is set to 1.

Voting: During this phase, voting is conducted. Each node v gets a vote equal to the sum of the voting ability scores of its immediate neighbors. This is considered as the point of influence of that node. The node with the highest influence score will be considered as the node that propagates the influence. Furthermore, the selected node will not be allowed to participate in subsequent voting rounds by setting that node's voting ability score to 0.

Update: TO find influence nodes from different locations in the graph network, thus the voting probability score of the neighbors of a node was chosen to perform influence propagation in the next iteration will be $\delta = 1/\langle k \rangle$, where $\langle k \rangle$ is the average of the degree of the nodes.

Iteration: The voting and update step will be repeated until l influence nodes are selected where l is a predefined constant.

VoteRank++

With the VoteRank++ approach [11], nodes with different degree centrality will have influence scores voted on by different node. Like VoteRank algorithm, VoteRank++ is divided into 4 main stages.

Initialization: The influence score vs_v of each node v is initialized to 0 by default, and the voting ability score of each node va_v is calculated by the formula:

$$va_v = log\left(1 + \frac{k_v}{k_{max}}\right) \tag{17}$$

Voting: During this phase, voting is conducted. Each node v gets a vote equal to the sum of the voting ability scores of its immediate neighbors. This is considered as the score of influence of that node.

$$VP_{uv} = \frac{k_v}{\sum_{w \in \Gamma_u} k_w} \tag{18}$$

$$vs_v = \sqrt{|\Gamma_v| \sum_{u \in \Gamma_v} VP_{uv}.va_u} \tag{19}$$

The node with the highest influence score will be considered as the node that propagates the influence. Furthermore, the selected node will not be allowed to participate in subsequent voting rounds by setting that node's voting ability score to 0.

Update: The VoteRank++ algorithm improves the reduction of the voting ability score to two-hop influence on its neighbors by the fomula:

$$va_v = \begin{cases} \lambda.va_v & v\ is\ a\ first - order\ neighbor \\ \sqrt{\lambda}.va_v, & v\ is\ a\ second - order\ neighbor \end{cases} \tag{20}$$

Iteration: The voting and update step will be repeated until l influence nodes are selected where l is a predefined constant.

Information Entropy

Information entropy [18] is a well-known notion in information theory. In general, the more unpredictable or random an occurrence is, the more data it contains.

The information entropy is defined as follows:

$$H(X) = H(x_1, x_2, \ldots, x_n) = -\sum_{i=1}^{n} p(x_i) log p(x_i) \tag{21}$$

where $X = \{x_1, x_2, \ldots, x_n\}$ is set of possible events and $p(x_i)$ is the probability of event x_i

One of the most important uses of information entropy in the field of social science is the entropy weighting technique. The entropy weighting approach is frequently used to compute the weights of distinct qualities in multiattribute decision-making situations, as well as to evaluate the relevance of nodes in complex networks, due to its good performance. Assume there are n qualities to take into account. The weight of attribute i abbreviated w_i, is computed as follows:

$$w_i = \frac{1 - H_i}{\sum_1^n (1 - H_i)} \tag{22}$$

where $H_i (i = 1, 2, .., n)$ is the information entropy of each attribute.

2.2 Performances Metrics

Susceptible Infected Recovered (SIR) Model and Susceptible Infected (SI) Model
A classic infectious disease model, the susceptible infected recovered (SIR) [19] model may also be used to abstractly represent information transmission. The SIR model assumes all nodes to be susceptible (S) at first, except for the source node, which has been infected (I). Each infected node has a possibility of infecting its susceptible neighbors with probability β. Each infected node recovers to become a recovered node with a chance of μ at each time step. The infection process will continue until there are no more infected nodes [18]. The influence of node i could be calculated using the formula:

$$P(i) = \frac{R(t^*)}{N} \tag{23}$$

where $\mathrm{R}(t^*)$ is the number of recovered nodes when the dynamic process achieves steady-state and N is the total number of nodes in the network. The corresponding epidemic threshold is:

$$\beta_c \approx \frac{\langle k \rangle}{\langle k^2 \rangle - \langle k \rangle} \tag{24}$$

The susceptible infected (SI) model is a special case of the susceptible infected (SIR) model. The SI model is based on the SIR model and assumes that the recovery rate $\mu = 0$, meaning that once a node has been infected, it cannot be recovered. The number of experiments is K. $F(t)$ represents the average number of infected nodes in the SI model at time t.

Average Shortest Path Length
L_s Is the average shortest path length of initial infection set S [9]. Usually, with larger L_s, the initial spreaders are more dispersed and can influence a larger range. This can be caculated by fomula:

$$L_S = \frac{1}{|S|(|S|-1)} \sum_{u,v} l_{u,v} \tag{25}$$

where S is the set of spreaders that have been identified, and $l_{u,v}$ denotes the length of the shortest path from node u to node v. If there is no path between u and v, $l_{u,v} = \delta + 1$, where δ is the enormous linked component's diameter.

3 Proposed Algorithm

The proposed algorithm, EAVoterank++ researches and improves the VoteRank++ [11] algorithm to determine the set of influential nodes in the graph. Algorithm divided into 4 main steps.

Step 1: Calculate Voting Ability Score
In EAVoterank++, we make the appropriate modifications in the method's voting and update stages. Each node's state is represented as a tuple (vs_v, va_v), where vs_v is the voting score of node v as determined by its neighbors, and va_v is the voting score of node v as determined by its neighbors. We consider that a node's voting ability is proportional to its importance. So, the voting ability score (va_v) is obtained by considering the order coefficient, information about the position of the node in the network is based on the k-shell decomposition as well as the clustering coefficient of the node along with its neighbors. The weight of each index is determined by the entropy information [18]. The voting ability of node v is calculated mathematically as

$$va_v = log\left(1 + \frac{score_v}{score_{max}}\right) \tag{26}$$

with $score_v$ of each node v is caculated by fomula:

$$score_v = w_1 I_{DC}(v) + w_2 I_{KS\_im}(v) + w_3 I_C(v) \tag{27}$$

where $I_{DC}(v)$ indicates the influence of the node's degree and the degree of its neighbors v, respectively, $I_{KS\_im}(v)$ denotes the influence of the node's k-shell and the k-shell of its neighbors v and $I_C(v)$ represents the influence of clustering coefficient and second-level neighbors' clustering coefficient of node v is determined according to the formula (28), (29), (30).

$$I_{DC}(v) = DC(v) + \sum_{u \in \Gamma_v} DC(u) \tag{28}$$

$$I_{KS\_im}(v) = KS\_im(v) + \sum_{u \in \Gamma_v} KS\_im(u) \tag{29}$$

$$I_C(v) = e^{-C_v} \sum_{u \in \Gamma_v^2} C_u \tag{30}$$

$w_1, w_2, w_3$are weights corresponding to the index and are calculated using entropy information as follow:

Firstly, set up a decision matrix with the values $I_{DC}(v)$, $I_{KS\_im}(v)$ and $I_C(v)$ with all nodes present in the network

$$D = \begin{bmatrix} I_{DC}(1) & \ldots & I_{DC}(n) \\ I_{KS\_im}(1) & \ldots & I_{KS\_im}(n) \\ I_C(1) & \ldots & I_C(n) \end{bmatrix} \tag{31}$$

The next step is to normalize the multi-attribute decision matrix. Since each index has a different dimension, the matrix D needs to be normalized to a normalized matrix R

$$R = \begin{bmatrix} r_{11} \ldots r_{1n} \\ r_{21} \ldots r_{2n} \\ r_{31} \ldots r_{3n} \end{bmatrix}, \quad r_{ij} = d_{ij} / \sqrt{\sum_{j=1}^{n} (d\_ij)^2} \tag{32}$$

Then, calculate the information entropy of each rating metric. According to Eq. (21), the information entropy of index j can be calculated by:

$$E_i = -\frac{1}{\ln n} \sum_{j=1}^{n} r_{ij} \ln r_{ij}, \qquad i = 1, 2, 3; j = 1, 2, \ldots, n \tag{33}$$

Finally, determining the weight of each metric. According to Eq. (9), the weight of index j can be calculated by:

$$w_i = \frac{1 - E_i}{2 - \sum_{i=1}^{3} E_i}, \qquad i = 1, 2, 3 \tag{34}$$

Step 2: Voting

We not only consider neighbor nodes as participating to the vote to elect the source spreader because nodes other than immediate neighbors in the system may play a crucial role in the selection of source spreaders, but also the voting ability of the node itself to determine the influence score. We expanded VoteRank from unweighted networks to weighted networks to determine the voting score of a node, also known as node influence score, and defined the voting score as

$$vs_v = \sqrt{|\Gamma_v|.va_v + \sum_{u \in \Gamma_v} w_{uv}.va_u} \tag{35}$$

where is w_{uv} is the weight between nodes u and v in weighted networks, $w_{uv} = 1$ in unweighted networks and Γ_v is the neighbor set of node v.

If this node was not elected before, the node with the most votes gets elected as an influential node in the current round. The same node will not participate in the voting process in later rounds, and its voting ability will be set to 0.

Step 3: Update

The voting ability of the neighbors of the selected spreaders is inhibited to guarantee that the identified influential nodes are dispersed throughout the network. As a consequence, when a node is voted as a spreader, it could have a two-hop influence on its neighbors. A node's voting ability is reduced by a factor of when it has voted and is up to two hops from the selected spreader. The updated voting ability of the neighbor nodes is represented by the following Eq. 20.

The voting ability of a node's first-order and second-order neighbors is reduced if it is determined as influential. As a result, only nodes with shortest distances of three or less from the selected node need to update their voting scores.

Step 4: Interation
Steps 2 and 3 are iterated until l influential nodes are selected, where l is a constant.

4 Experimental Results

We used complex network datasets of various types and abilities to verify the proposed approach as well as other existing methods. The datasets chosen comprise a variety of graph features displayed by the networks, which may be used to properly assess the algorithms' performance. The following are the brief descriptions for these networks. (1) Jazz: the network of collaborations between a group of jazz musicians. (2) Email: a network of email exchanges among members of the University Rovira I Virgili. (3) USAir: a network of the US air transportation system in 2010. (4) Hamster: a friendship network between users of the website hamsterster.com. (5) Condmat: a co-authorship network of scientists working on Condense Matter Physics. (6) Facebook: an anonymized social network extracted from Facebook consists of 'circles' (or friends lists) from Facebook. Facebook data was collected from survey participants using this Facebook app. Table 1 describes basic features of 6 experimental networks.

The experiment findings of infection scale $F(t)$ against time t are displayed in Fig. 1 using the SIR model. With $\beta = 1.5$ and $\rho = 0.03$ correspondingly the results are averaged across 1000 independent runs.

Table 1. Basic features of 6 experimental networks

Networks	N	M	$\langle k \rangle$	$\langle k_{max} \rangle$
Jazz	198	2,742	27.69	100
Email	1,133	5,451	9.62	71
USAir	1,574	17,215	21.87	314
Hamster	2,426	16,631	13.71	273
Facebook	4,039	88,234	43.69	1045
Condmat	23,133	93,497	25.64	281

The purpose of this experiment is to examine the diffusion rates of various ways after the number of initial spreaders is set. In all 6 experimental networks, EAVoteRank++ may infect a higher number of nodes than baseline approaches with the same number of initial spreaders, as seen in Fig. 1. More specifically, in the stable stage, VoteRank++ outperforms rival algorithms in the the infection scale on 6 experimental networks.

Figure 2 shows the final effect scales $F(t_c)$ for different ratios of infected nodes at the initiation. With an infection rate of $\beta = 1.5$, the results are averaged or 1000 independent experiments. The goal of this experiment is to see how different strategies with different numbers of initial spreaders effect the final ranges. More initial spreaders, without a doubt, reach a larger area. EAVoteRank++ achieves an impressive result on these networks, as evidenced by the fact that its ultimate influence grows under the same ratio of initial spreaders, especially when the ratio is enormous, are always ranked first. When is large, on the other hand, information can flow rapidly across the network.

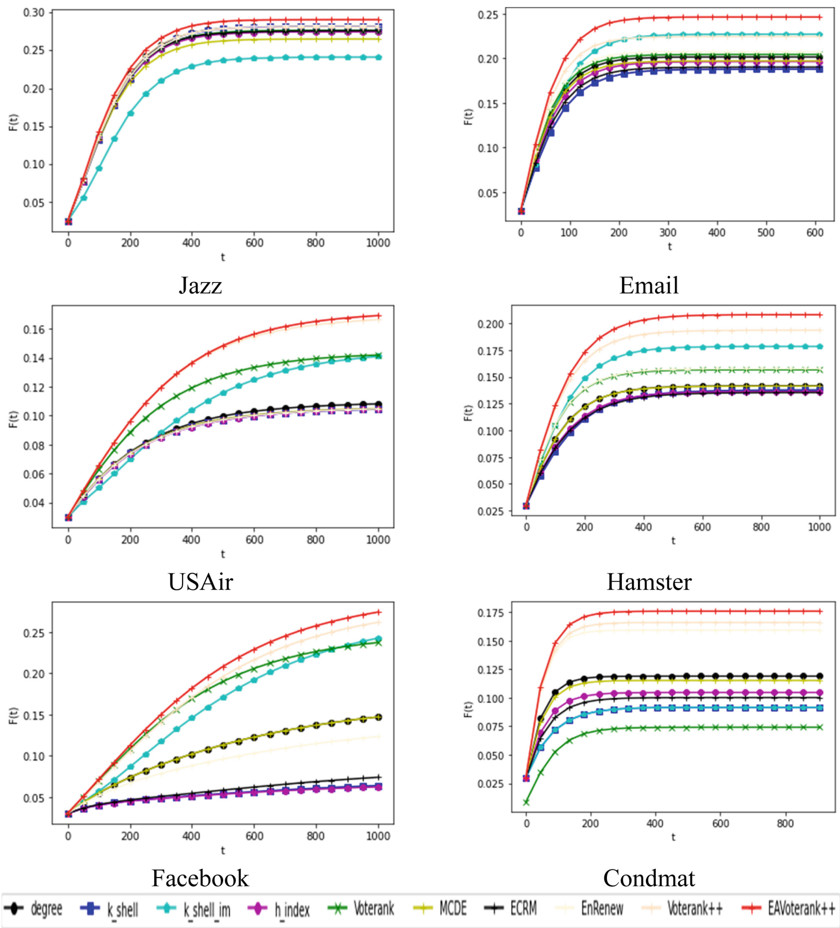

Fig. 1. The infection scales $F(t)$ against time t

Figure 3 shows the final effect scaling $F(t_c)$ for several approaches with varied infection rates. EAVoteRank++ can obtain the maximum or near to the largest spread scale in most situations under varied conditions, as shown in Fig. 3, notably on the Jazz, Email, USAir, Hamster, Facebook and Condmat networks. Based on the results in Fig. 3, we can conclude that EAVoteRank++ has a better generalization ability than baseline approaches.

The distances between spreaders are critical in increasing the impact. On 6 networks, Fig. 4 illustrates the L_s of spreaders found using various approaches. Except for the USAir network, EAVoteRank++ outperforms the baseline approaches because it always has the greatest or near to the largest L_s. In other words, the EAVoteRank++ technique identifies influential nodes that are dispersed over a network. The proposed algorithm outperforms all baseline approaches for the two largest networks, Condmat and Facebook. Implying that the prominent nodes picked by EAVoteRank + + are more decentralized in these networks. This demonstrates that the proposed algorithm is capable of identifying nodes with a high spreading capacity.

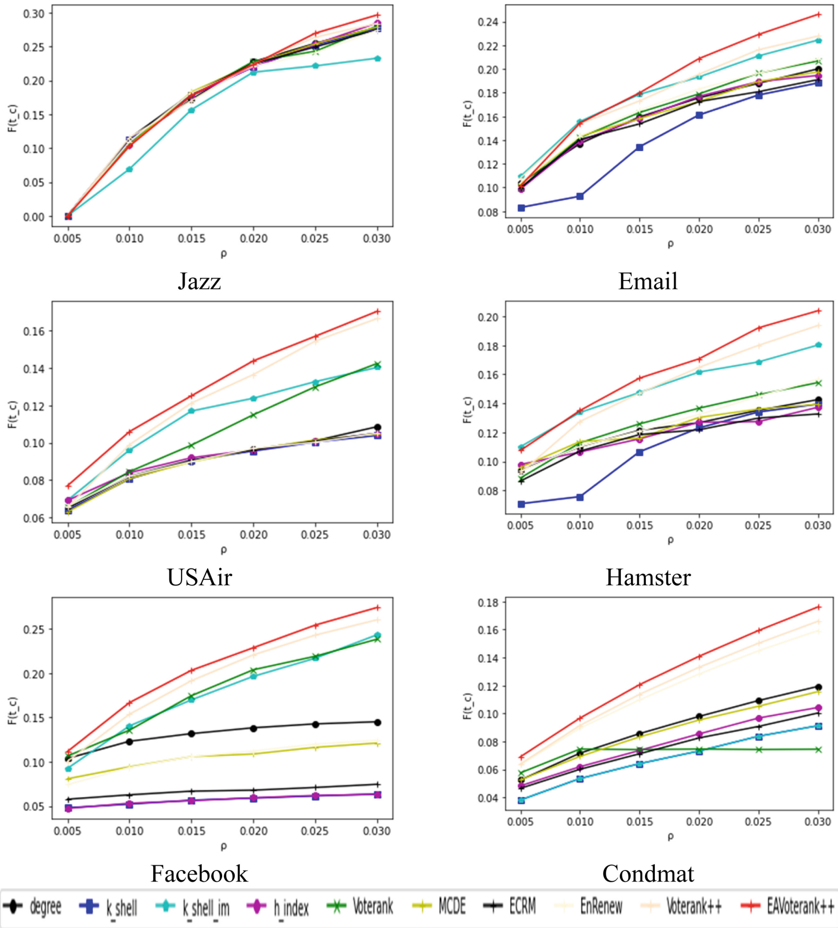

Fig. 2. The final infection scales $F(t_c)$ with different ratio of initial infected nodes ρ

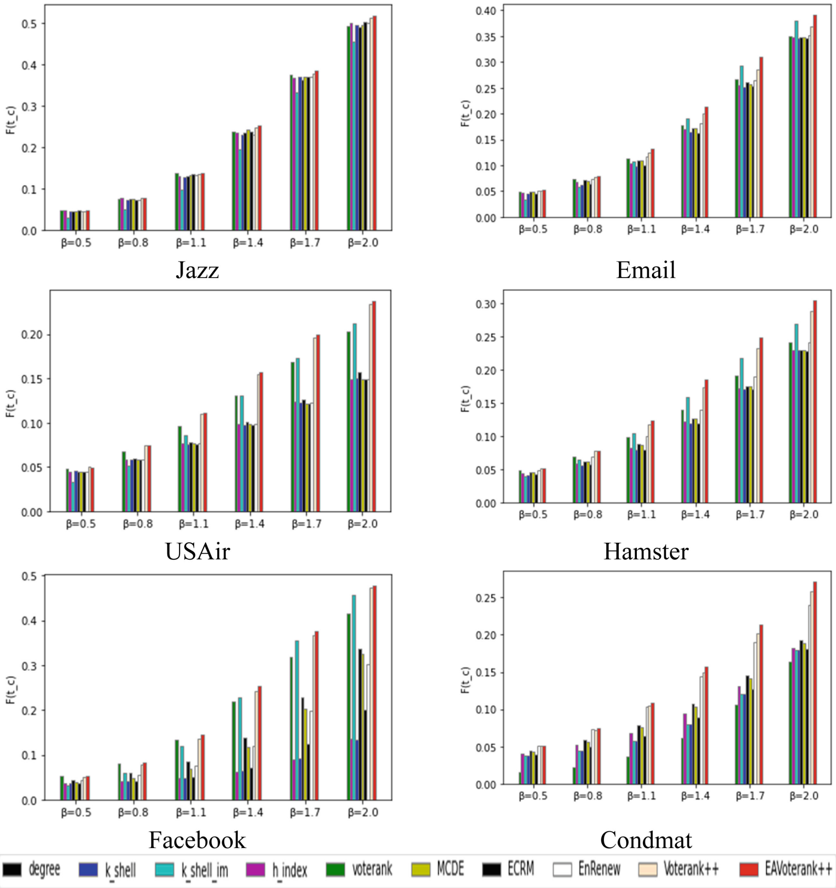

Fig. 3. The final infection scale $F(t_c)$ with different infect rate β

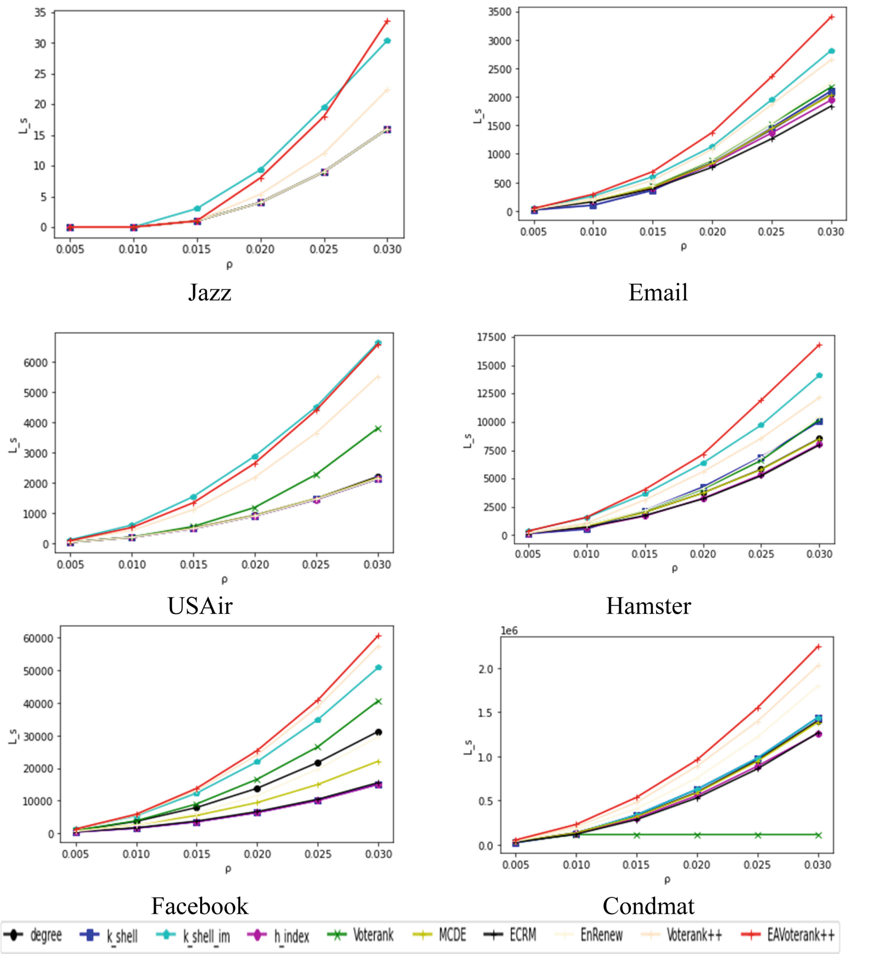

Fig. 4. The average shortest path length L_s between spreaders selected by different methods

5 Conclusions and Future Work

In this paper, we propose EAVoteRank++ for identifying influential nodes in social networks by using nodes's voting ability by considering degree, position in network by improve k-shell decomposition and clustering coefficient as well as neighbors. The weights of attribute are calculated by entropy technology. In experiment, The SIR model is employed to simulate the spreading progress. The experimental results on 6 networks show that in terms of infection scale, in most circumstances, EAVoteRank++ outperforms state-of-the-art baselines. However, there are still certain challenges to overcome in order to improve our work. Firstly, social networks, on the other hand, can evolve dynamically in real life. As a result, we will continue to strive to improve the algorithm's performance and investigate dynamic network features in future study. Firsly, we will consider using network embedding to extract the node feature information in the network, then we can apply Deep Learning model to use these features. Secondary, we will minimize the cost of further operations to accommodate a network of more nodes and relationships.

Acknowledgment. This research is funded by CMC Institute of Science and Technology (CIST), CMC Corporation, Vietnam.

References

1. Bond, R.M., et al.: A 61-million-person experiment in social influence and political mobilization. Nature **489**(7415), 295–298 (2012)
2. Perozzi, B., Al-Rfou, R., Skiena, S.S.: DeepWalk: online learning of social representations. In: ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2014)
3. Dinh, X.T., Van Pham, H.: Social network analysis based on combining probabilistic models with graph deep learning. In: Sharma, H., Gupta, M.K., Tomar, G.S., Lipo, W. (eds.) Communication and Intelligent Systems. LNNS, vol. 204, pp. 975–986. Springer, Singapore (2021). https://doi.org/10.1007/978-981-16-1089-9_76
4. Wang, D., Cui, P., Zhu, W.W.: Structural deep network embedding. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1225–1234 (2016)
5. Mikolov, T., Sutskever, I., Chen, K., et al.: Distributed representations of words and phrases and their compositionality. Adv. Neural Inf. Process. Syst. **26**, 3111–3119 (2013)
6. Gao, Y.-C., Fu, C.-J., Cai, S.-M., Yang, C., Eugene Stanley, H.: Repulsive synchroni-zation in complex networks. Chaos **29**, 053130 (2019)
7. Brin, S., Page, L.: Reprint of: the anatomy of a large-scale hypertextual web search engine. Comput. Netw. **56**(18), 3825–3833 (2012)
8. Ma, L.L., Ma, C., Zhang, H.F., Wang, B.H.: Identifying influential spreaders in complex networks based on gravity formula. Phys. A **451**, 205–212 (2016)
9. Zhang, J.-X., Chen, D.-B., Dong, Q., Zhao, Z.-D.: Identifying a set of influential spreaders in complex networks. Sci. Rep. **6**(1), 27823 (2016)
10. Guo, C., Yang, L., Chen, X., Chen, D., Gao, H., Ma, J.: Influential nodes identification in complex networks via information entropy. Entropy **22**(2), 1–19 (2020)
11. Liu, P., et al.: Identifying influential nodes in social networks: a voting approach. Chaos Solitons Fractals **152**, 111309 (2021)
12. Sabidussi, G.: Thee centrality index of a graph. Psychometrika **31**(4), 581–603 (1966)
13. Li, Z., Huang, X.: Identifying influential spreaders in complex networks by an improved gravity model. Sci. Rep. **11**(1), 1–10 (2021)
14. Eguiluz, V.M., Klemm, K.: Epidemic threshold in structured scale-free networks. Phys. Rev. Lett. **89**(10), 108701 (2002)
15. Lü, L., Zhou, T., Zhang, Q.-M., Stanley, H.E.: The H-index of a network node and its relation to degree and coreness. Nat. Commun. **7**(1), 10168 (2016)
16. Sheikhahmadi, A., Nematbakhsh, M.A.: Identification of multi-spreader users in social networks for viral marketing. J. Inf. Sci. **43**(3), 412–423 (2017)
17. Zareie, A., Sheikhahmadi, A., Jalili, M., Fasaei, M.S.K.: Finding influential nodes in social networks based on neighborhood correlation coefficient. Knowl Based Syst **194**, 105580 (2020)
18. Yang, Y., et al.: A novel centrality of influential nodes identification in complex networks. IEEE Access **8**, 58742–58751 (2020)
19. Shulgin, B., Stone, L., Agur, Z.: Pulse vaccination strategy in the SIR epidemic model. Bull. Math. Biol. **60**(6), 1123–1148 (1998)