Applied Intelligence
https://doi.org/10.1007/s10489-021-02266-w

IM-ELPR: Influence maximization in social networks using label propagation based community structure

Sanjay Kumar[1,2] · **Lakshay Singhla**[3] · **Kshitij Jindal**[4] · **Khyati Grover**[5] · **B. S. Panda**[1]

Accepted: 6 February 2021

Abstract

The popularity of social networks has grown manifolds in recent years because of various activities like fast propagation of ideas, publicity, and news. Influence maximization (IM) is one of the most highly studied problems in the field of social network analysis due to its business values. Influence maximization aims to identify influential nodes that can spread the information to the maximum number of nodes in the network through diffusion cascade. Traditional methods for IM include centrality based and greedy based measures. However, each method has some limitations. Recently, some methods of IM are introduced, which consider the presence of community structure in networks. Community structure has a significant impact on information diffusion, because of dense connections between the nodes in the community. In this paper, we propose a novel influence maximization algorithm using node seeding, label propagation, and community detection. We first use extended h-index centrality to detect the seed nodes and then use the label propagation technique to detect communities. Further, we merge smaller and related communities to a larger community with the help of a relationship matrix. Finally, top-k influential nodes from these communities are identified. These ideas lead to our proposed algorithm: Influence Maximization using Extended h-index, and Label Propagation with Relationship matrix (IM-ELPR). We adopt Independent Cascade (IC) information diffusion model to spread the information originating from chosen influential nodes. The proposed algorithm intends to identify influential nodes from different communities globally and does not depend on the specific community's antecedent structural information. Experimental results performed on several real-life data sets reveal that the proposed algorithm performs better than many other existing popular algorithms.

Keywords Complex networks · Community structure · Influence Maximization · Independent cascade model · Label propagation

1 Introduction

Many real-life networks, like online social networks, information networks, and biological networks, are complex networks [1, 2] that consist of numerous connected nodes with complicated interactions between them. An online social network (OSN) is an abstract view of our virtual social system depicting a large number of users and social acquaintances between them [3]. A social network can be modeled as graph $G = (V, E)$, where V represents a set of people or entities, and E denotes the set of edges. An edge exists between two nodes if they are socially connected like friendship, follow-followee, and colleagues. As a result of the boom of internet and electronics services, online social networks have become very popular, which connects millions of people worldwide to share ideas, information and perform many online activities [4]. These days, online social networks like Twitter, Facebook, Instagram, and Wechat play a pivotal role in shaping public perceptions and creating awareness [5]. These networks act as an excellent platform for the fast diffusion of information and viral marketing by exploiting the trust relationship between the users. Influence maximization (IM) is a hot research topic in the field of network science to measure the user's influence in spreading the information in a network [6]. It seeks to identify some seed nodes with high spreading capability such that information originating from these nodes reaches a maximum number of people in the network, and such nodes are called influential nodes [7]. Formally, the influence maximization (IM) problem is defined as "given a social

✉ Sanjay Kumar
sanjay.kumar@dtu.ac.in

Extended author information available on the last page of the article.

Published online: 15 March 2021

network represented as a graph $G(V, E)$ with $|V| = n$, $|E| = m$, and a positive number k where $k << n$, determine a set of k seed nodes, such that initially triggering them, the influence spread in the network can be maximized under a specific information dissemination model." Kempe et al. [8] proved that obtaining an optimal solution to the IM problem is an NP-hard optimization problem under the classical diffusion model. Due to mutual trust between the users, influential nodes are capable of accelerating the information spread by triggering a cascade in the diffusion process through word of mouth strategy [9, 10]. Many e-commerce companies are targeting these influential users to promote their products and to achieve optimal publicity [11].

Numerous standard node centrality based algorithms [12] based on local, semi-local, and global features of nodes are employed to solve IM problems with certain limitations and scalability issues. Degree centrality [13], betweenness centrality [14], PageRank [16], and k-shell centrality [17] are the commonly used centrality to rank the nodes based on its structural and topological position in the network. Many classical greedy based algorithms are proposed for influence maximization [8, 18, 19], which employ sub-modularity and monotony of the independent cascade (IC) model along with Monte Carlo simulations to improve the information propagation. Such algorithms are slow due to a large number of simulations and require exponential time complexity. Recently, many community structures based influence maximization algorithms are introduced with considerable improvement in the results [20, 38].

对于IM的研究现状介绍

The presence of community structures in complex networks is ubiquitous [21, 22]. Finding community structure can lead to the understanding of the functional organization of the system, and community detection is one of the prominent research topics among the researchers of different fields [23, 24]. A community can be referred to as a group of nodes having more connections within themselves and fewer connections to the rest of the network. In a social network, a community can correspond to a group of people in a fan club of a particular celebrity, employees working in the same project, researchers working in a particular field, and many more. Due to a dense connection, the information can spread rapidly in the community. Hence, the underlying community structure present in social networks can be utilized to perform the influence maximization task [25].

社区结构简介，两者的关系

In this paper, we introduce a novel method of IM named Influence Maximization using Extended h-index and Label Propagation with Relationship matrix (IM-ELPR) by utilizing the community structure using label propagation. The proposed algorithm is comprised of four phases, namely the seeding phase, the label propagation phase, merging communities using relationship matrix, and finally, finding k influential nodes. There is a possibility that a node acquires multiple labels signifying its presence in multiple communities during the label propagation phase. Therefore, we consider the overlapping community detection through label propagation in our algorithm. We use the popular independent cascade (IC) information diffusion model to spread the influence originating from seed nodes. Our contributions are summarised as; 本文主要贡献（创新点、研究内容）

1. We propose a novel algorithm to maximize the information propagation by employing seeding phase, label propagation, and community spread in the network.
2. We employ extended h-index centrality in the seeding phase, and also take care that the selected seed nodes are not adjacent to each other, rather far apart so that the entire network can be adequately covered.
3. The proposed algorithm detects the community by recognizing that a node can have multiple labels to map real-life scenarios and performs the merging of relatively smaller and related communities into a larger community.
4. The experimental results obtained on eight real-life datasets of various sizes and applications exhibit improved performance of the proposed algorithm over many contemporary IM algorithms.

The rest of the paper is organized as follows. Section 2, presents the related works of IM using various node centralities and other sophisticated methods. Section 3, describes the proposed work in detail, the time complexity of the proposed algorithm, and a toy network simulation. Section 4, presents the information diffusion model, datasets used, and performance matrices used in computing the results of the IM algorithms. Section 5, reports the experimental results generated by our algorithm and parallel analysis with some previous notable research works. Finally, Section 6 concludes the paper. 文章结构说明

2 Related work

In complex networks, many centrality measures have been employed to estimate the significance of nodes. These centrality measures order the nodes based on their topological positions and many other critical parameters. It is assumed that nodes with higher centrality values have more spreading capability than normal nodes. Majorly, centrality based methods are classified into three categories, namely, local, semi-local, and global measures. The local measure focuses on the strength of the node and its adjacent neighbors, like degree centrality [13]. Degree centrality

is the number of direct links a node can have with the other nodes. Usually, local measures are simple and have linear time, but they ignore the topological position of other nodes and may not produce good results in a large-scale network. For example, in the degree centrality, a node with a high degree but situated at the periphery of the network may not be suitable for becoming seed nodes. Contrary, global measures consider the ranking of nodes based on the strength of a particular node like the shortest path, information flow in comparison to all other nodes in the network, and hence take care of the global structure of the nodes. Betweenness centrality, Closeness, Eigenvector, Pagerank, and k-shell centralities are popular examples of global measures. Usually, these measures produce good results but require high time. Betweenness centrality [14] of a node depends on the number of shortest paths passing through that node. In a network, it signifies the control of a node for the information flow in the network. Eigenvector centrality [15] is based upon the fact that if a node is linked to important nodes, it is considered important too. PageRank (PR) [16] is an algorithm employed by Google Search to rank web pages in search engine results. It operates by counting the number and quality of incident edges to a node to estimate its importance. The underlying assumption is that more essential nodes (web pages) are likely to receive more edges from other nodes. k-shell centrality [17] argues that a node lying the core of the network can be a better candidate for information propagation, and a larger shell value signifies that the node is more centrally located in the system.

Semi-local measures intend to maintain the balance between insignificant local measures and expensive global measures. Typically, these measures consider the nodes up to 2 or 3-hops to assess the spreading ability of a node. Local centrality [26], h-index [27] are the example of semi-local measures. Recently many semi-local measures are proposed, and they came up with promising results that produced better results [28, 29]. h-index is a popular metric formulated by J.E. Hirsch [27] to measure the impact of a researcher based on the number of citations received. The same is applied in network [30], and h-index of a node is defined as the maximum value h such that it has at least h neighbors each with degree h or more. Mathematically, h-index can also be defined in the form of a cumulative function as follows [28]:

$$h - index(v) = \max_{h}(c_h(v)),\ where\, c_h(v) \geq h, \tag{1}$$

$$c_h(v) = |\{u|u \in N_v\, and\, d_u \geq h\}|. \tag{2}$$

Where $c_h(v)$ is the function that returns the number of neighbors of node v having degree greater than or equal to h. Hence, as per (1), the value of h-index of node v is a maximal value h such that h number of its neighbors have degrees greater than or equal to h. If a node has a high degree centrality lying in the periphery of the network, it means although the node has a large number of neighbors, those neighbors may have less degrees. But in the case of nodes with a high h-index, their neighbors also have a high degree. Thus, h-index centrality considers the neighborhood of a node up to 2-hops, unlike degree centrality. Therefore, h-index is a more appropriate semi-local measure to estimate the spreading capabilities of nodes. However, many nodes can have the same h-index, so it is difficult to differentiate in the spreading capability of those nodes. To overcome this, authors in [28, 31] suggested extended h-index measures. According to Liu et. al [28], extended h-index for a node (v) is formulated as follows:

$$Extended h-index(v) = h-index(v) + \sum_{u \in N(v)} h-index(u) \tag{3}$$

where $N(v)$ is the set of neighbours of v. Extended h-index of node v is equal to the h-index of v plus sum of h-indices of all neighbours of node v. Thus, an extended h-index takes neighbors up to 3-hops from a node in consideration and can distinguish between the nodes with the same value of h-index. Authors in [29] suggested a type of semi-local algorithm Fixed Neighbor Scale (FNS) by considering neighbors of a node up to multiple levels to determine its spreading influence. They determine the spreading capability of a node by adding multi-level neighbors' weights in the form of their distances from the source node.

By exploiting the community structures in networks, the information diffusion and influence spread can be magnified in the network. This is due to the dense connection between the members of a community. In the past few years, numerous community-based influence maximization algorithms are developed with considerably improved performance. Such algorithms consider independence between communities and perform parallel execution. There are many ways to detect communities in networks. One of the most efficient ways is to uncover community structure is by using the label propagation technique. Raghav et al. [32] proposed the idea of label propagation, which proves to be an efficient method to detect communities within a social network. A vital process in the field of network science, label propagation is a semi-supervised machine learning approach that takes into consideration the topology of the network. In the case of community detection, the process starts by allocating a unique label to every node. Every single node then adopts the label, which is having the highest rate of recurrence amidst its neighbors. Zhao et al. [33]

说明利用社区结构进行IM是有效的，且有先例的

proposed an IM algorithm named as IM-LPA based on community structure using label propagation. They practice degree centrality to find candidates for the seed nodes before performing label propagation. Salavati et al. [34] came up with the Global Local Ranking (GLR) method that uses the local critical node and gateway node of a community to rank every node in the graph using the closeness centrality measure. Berahmand et al. [35] have introduced a local ranking method named as DCL (Degree, Clustering coefficient, and Location method) that calculates the importance of node based upon the degree of a node, its clustering coefficient, and its location in the network. Rui et al. [36] proposed a Reversed Node Ranking (RNR) algorithm based on the reverse ranking of a node. The algorithm considers that a node is influential if it has a higher degree and if its neighbors are also influential. Wen et al. [37] proposed an approach for finding influential spreaders using the local information dimensionality (LID) method by considering the local structural properties around the central node(i). The method measures the information of the nodes in boxes through the Shannon entropy. The size of the box l around the central node(i) varies from one to ceil($d_i/2$), where d_i is the degree of central node. The objective of variation in the size of the box indicates that the method concentrates on the quasilocal structure about the central node and decreases the time complexity. Huang et al. [38] recommended a heuristic influence maximization technique, which combines community detection and topic awareness into influence diffusion modeling.

3 Proposed algorithm: IM-ELPR

In this section, we present the proposed algorithm named as IM-ELPR using the idea of label propagation and community detection. Information spreading through communities is the fastest way of disseminating information in the entire network because of the structural property and dense connection among the members of a community. A node in a community is more likely to circulate the information to nodes in the same community. Label Propagation is a novel procedure that finds communities in linear time complexity, which is efficient than many other community detection methods. Further, the Relationship matrix is used to refine the community detection process by infusing the modularity parameter of the algorithm. Finally, we select the top k influential nodes from the candidate seed nodes based on the number of other nodes who adopted the same label.

The proposed algorithm consists of four phases:

- Seeding Phase
- Label Propagation Phase
- Merging Communities by Relationship Matrix
- Finding k Influential Nodes

3.1 Seeding phase

In order to find influential nodes in a social network, we first select candidate seed nodes. We use the extended h-index centrality measure to determine the candidate seed nodes. From (3), extended h-index centrality for a node is calculated as the sum of h-index of itself and all of its neighboring nodes. The seeding phase should generate the best possible candidates for the seed nodes in less time. By using extended h-index centrality in the seeding phase, the neighbors up to 3-hops from the given node are taken care of. This covers the local neighborhood appropriately, thereby generating more prominent and effective seed nodes in linear time. We also discuss the reason for choosing the extended h-index over the degree and h-index centrality for the seeding phase with the help of a toy network, as shown in Fig. 1. Node 9 has the highest degree centrality of 7, but this node may not be an influential spreader because most of its neighbors are leaf nodes and are incapable of spreading the information further. Instead, nodes with a higher h-index such as nodes 1,2,3, and 4 can be better spreaders. However, since these four nodes have equal h-index values, the problem of choosing the most appropriate node from among them arises. This problem is solved by the extended h-index, which has different values for these nodes and selects node 3 as the most suitable candidate seed node. The benefit of considering extended h-index (or even h-index) is that it gets less affected by slight changes in the topology of the graph, such as the addition or deletion of some edges between the nodes, which frequently happens in the social networks. For example, the addition of a new node 16 and forming a connection with node 5 does not affect the h-index as well as the extended h-index values of the nodes 1 and 2. This is illustrated in Fig. 1.

The outline of the seeding phase algorithm is presented in Algorithm 1.

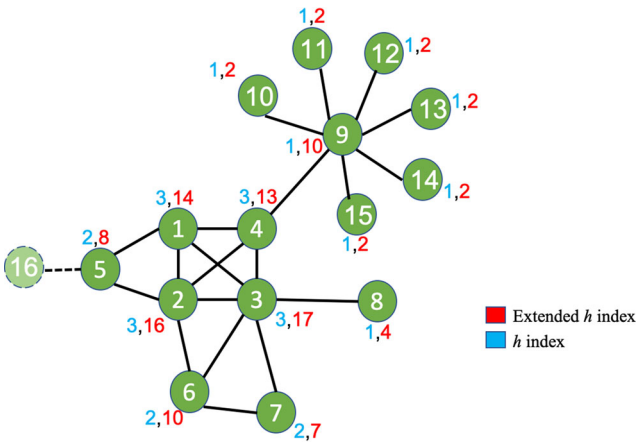


Fig. 1 Toy network for comparison of degree, h-index, and extended h-index centrality

Algorithm 1 Seeding phase.

Result: A set of seed nodes, S
1 Let V be the set of all nodes, W be the set of candidate nodes, and S be the set of seed nodes. Initialise, $W = V$ and $S = \phi$;
2 EH = calculate eh_index(W);
3 **while** $W \neq \phi$ **do**
4 Let u be the node with maximum centrality in W, i.e., $u = \max(EH(W))$;
5 Add u to S, $S = S \cup u$;
6 Remove u from W, $W = W \setminus u$;
7 **for** $v \in N(u)$ **do**
8 Remove v from W, $W = W \setminus v$;
9 **end**
10 **end**

The brief description of the seeding phase algorithm is as follows:

- Initially, we maintain two lists as discussed in line number 1. W represents the set of candidate nodes, which is initially equal to the set of all nodes, $W = V$, where V is the set of all nodes in the network. S denotes the set of seed nodes, which is initially empty, $S = \phi$.
- We calculate the extended h-index for every node and store the values in the list EH. This can be inferred from line number 2.
- Now in line number 4, we choose a node from W, which has the maximum value for extended h-index. If there is more than one node with the same maximum extended h-index value, we choose any one of them.
- As discussed in the line 5-9, we add u to S as a seed node and also remove u and its neighbors from the candidate set W. It ensures that the selected seed nodes are not adjacent to each other, rather far apart, so that the entire network can be adequately covered.
- We repeat lines 3-10 till the candidate node set gets empty, $W = \phi$ and we get the final set of seed nodes i.e. S.

3.2 Label propagation phase

In this phase, communities are formed with the help of seed nodes determined in the last phase in an asynchronous manner. Initially, every seed node is assigned a unique label, and the rest of the nodes have no label. Then in every iteration, the label of a particular node is updated to the label, which is most common among its neighbors. In case more than one label appears the maximum number of times in the neighboring nodes, both the labels are assigned to the node. Now, since each node can belong to more than one community simultaneously, we use a set of labels for each node instead of assigning one label to every node, which is done traditionally. Thus, it allows a single node to have more than one label signifying that a node may be a part of more than one community. The nodes having common labels belong to the same community. The outline of the label propagation phase algorithm is presented in Algorithm 2.

Algorithm 2 Label propagation phase.

Result: A list of label for each node, L and List of communities, C
1 Initialise a unique label for each node in seed node set S;
2 **for** $u \in S$ **do**
3 $LP(u) = random(label)$
4 **end**
5 **repeat**
6 Generate a random sequence of nodes, V;
7 **for** $u \in V$ **do**
8 Find the label set which appears maximum number of times in $N(u)$, $LP(u) = \max_{v \in N(u)} (Count(LP(v)))$;
9 **end**
10 **until** *at least 1 node is updated*;
11 C = list of communities;

The new label set acquired by each node is updated as follows:

$$LP'_u(t) = \max_{lp} \sum_{v \in N(u)} \delta(lp, LP_v(t-1)) \tag{4}$$

where $LP'_u(t)$ is the new label set for node u at time t, $LP_v(t-1)$ is the label set of node v at time $t-1$, lp denotes a particular label value, v is a neighbor of node u and $\delta(.)$ is the extended Kronecker delta function and has value $\delta(lp, LP_v) = 1$ when lp is in the set LP_v ; and $\delta(lp, LP_v) = 0$ when lp is not in the set LP_v.

The brief description of the label propagation phase algorithm 2 is as follows:

- In lines 1-4, each seed node, which has been selected from previous algorithm, is assigned a random label.
- We use the asynchronous label propagation method and generate a random sequence of nodes for each iteration of label propagation. As shown in lines 5-10, for each node, u from a randomly generated sequence of nodes, calculate the most frequent label set. That is, find the labels which occur the maximum number of times in the neighbor set of u. If more than one label has the same frequency, then select all of them and assign this label set to u.
- In line number 11, we generate a list of communities. For each community, we get to know all the nodes and their labels.

3.3 Merging communities by relationship matrix

Relationship matrix (R) is used to compute the similarity between communities and accordingly merge them [40].

We keep on merging the communities till the modularity of the network is maximized. The relationship matrix is a novel method for finding the similarity between obtained communities. It gives a quantitative measure for the relationship among all pairs of communities.

The R matrix is defined as follows:-

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} & \cdots & R_{1n} \\ R_{21} & R_{22} & R_{23} & \cdots & R_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R_{n1} & R_{n2} & R_{n3} & \cdots & R_{nn} \end{bmatrix}, \tag{5}$$

where $R_{ij} = \dfrac{E(c_i, c_j)}{\min(|c_i|, |c_j|)}$ subject to $i \in 1, 2, ..., |C|$ and $i \neq j$. The number of common edges between community c_i and community c_j is denoted by $E(c_i, c_j)$. $R_{ii} = m_i$, where m_i is the number of edges in community c_i. Each cell R_{ij} in the R matrix denotes the relationship between the community c_i and c_j. If there are n communities generated by label propagation, i.e. $n = |C|$, then the order of R matrix is $n \times n$. The resulting R matrix is always a sparse matrix because most of the communities are not inter-related to one another and hence, their similarity is 0.

The label propagation algorithm, which we use, forms a large number of smaller communities. Many smaller communities with large similarities can be merged to form a relatively large community for a better perspective. Having two communities c_i and c_j with many common edges can become a limitation for our use case. If both these communities contain an influential node, then their influence can be overlapped. On the contrary, if we assign only one influential node from new merged communities, then it may influence the members with no overlap. Thus, we adopt a method to merge smaller and related communities using a relationship matrix, leading to an overall less and a relevant number of communities. The Algorithm 3 depicting the merging of communities is as follows:

Algorithm 3 Merge communities using relationship matrix.

Result: Set of final community, C'

1 **repeat**
2 Calculate R matrix as per Eq. No. 5 ;
3 Choose maximum value of R_{ij} from the matrix ($i \neq j$);
4 Merge communities c_i and c_j;
5 Calculate the modularity for new communities using Eq. No. 6;
6 **until** *new modularity > previous modularity*;

We also handle the case that a node can have multiple labels to find communities using label propagation. After performing community detection, many smaller size communities arise that have overlapping features and can be a bottleneck for influence maximization purposes. In the algorithm, first, the similarity between every pair of communities is calculated to form the R matrix. We find the maximum value $R_{ij}(i \neq j)$ from the R matrix corresponding to i^{th} row and j^{th} column, and merge these two communities. The modularity for the new set of communities is calculated. The same process is repeated until the new modularity is no longer greater than the previously derived one. In this way, we merge smaller communities to form larger communities.

Modularity is the quantitative measure for detecting communities, which evaluates the structural properties of the communities [41]. The formula for modularity is given as:

$$Q = \frac{1}{2m} \sum_{u,v \in V} \delta(u, v)(A_{u,v} - \frac{d_u d_v}{2m}) \tag{6}$$

where m is total number of edges, $\delta(u, v)$ is the extended Kronecker delta function, which is equal to 1 when u and v belong to the same community and equal to 0 when they are in different communities, $A_{u,v}$ is the adjacency matrix element, d_u is the degree of node u.

3.4 Finding *k* influential nodes

After forming communities through label propagation and merging them, we get labels for each individual node. For selecting influential nodes from the initially selected seed nodes, we find a rank measure for every seed node and sort those values in descending order. Finally, we select the top k nodes as influential nodes. Let LP_v be the initial label for seed nodes. The rank measure $M(v)$ for every seed node can be calculated by the formula:

$$M(v) = \sum_{u \in V} L(u, LP_v) \tag{7}$$

where LP_v is the label of seed node v, V is set of all nodes and $L(u, LP_v)$ is a function which returns 1 when node u has acquired the label of node v, i.e., LP_v, otherwise 0. So, the rank measure $M(v)$ of seed node v is the number of nodes with the same label as that of v. In this manner, a top influential node is one whose label is adopted by the maximum nodes of the network.

Algorithm 4 Finding k influential nodes.

Result: Set of influential nodes

1 Calculate rank measure for each initially selected seed node using Eq. No. 7;
2 Select the top k influential nodes according to their rank values calculated in previous step.;

3.5 Time complexity of the proposed algorithm

We also analyze the time complexity of our proposed algorithm IM-ELPR. For this purpose we assume that our network has n nodes and m edges and $\hat{d}$ be the average degree of the nodes.

1. Seeding Phase: The extended h-index for all the nodes in the network can be calculated in $O(m)$ time. In this phase, finding the node with a maximum extended h-index to populate candidate seed nodes can be done in $O(n)$ time. Therefore, the time complexity of the seeding phase is $O(m+n) = O(m)$.
2. Label Propagation Phase: In this phase, we consider that each node can have $O(\hat{d})$ neighbors, and each of these neighbors can have minimum q labels where q is a constant. So, $O(\hat{d}q)$ time is needed to update the label of one node. For all the nodes, the time required is $O(n\hat{d}q)$ and for T iterations of label propagation, computational time is $O(Tn\hat{d}q)$.
3. Merging Communities: Let us consider that label propagation results in the formation of C communities. In each iteration of merging communities, $O(m)$ time is required to form the R matrix, which is a sparse matrix, and the size of the matrix is to be $C \times C$ leading to $O(C^2)$ time to find the maximum value in the matrix for merging. Therefore, for T' iterations where T' is a constant, total time taken is $O(T'C^2 + T'm)$.
4. Finding k influential nodes: In this phase, we calculate the rank measure for each candidate seed node based on the number of nodes that acquired its label and then finding the top k seed nodes as the influential nodes can be computed in $O(n)$ time.
5. Finally, the overall computational complexity of IM-ELPR is $O(m) + O(Tn\hat{d}q) + O(T'C^2 + T'm) + O(n) = O(T'm)$, which is linear in terms of input size.

3.6 Toy network simulation of the proposed algorithm

For a better understanding of the simulation of the proposed algorithm, we present a toy network depicting all four phases, as shown in Fig. 2. Figure 2a presents a toy network taken as an input containing 115 nodes and 613 edges where all the nodes are initially colored in red. Figure 2b displays the toy network after the seeding phase, where selected seed nodes are colored in green, and the rest nodes are colored in brown. The label propagation phase on the toy network is depicted in Fig. 2c, which generates a total of nine communities where nodes belong to the same community, are colored in the same color. Figure 2d presents the network after performing merging communities operation using R matrix, which leads to seven communities from nine communities. Finally, Fig. 2e depicts the top-10 influential nodes as selected by the proposed method where influential nodes are colored in green, and the rest nodes are colored in brown.

4 Information diffusion model, datasets, performance metrics

This section describes the information diffusion model, real-life network datasets, and performance metrics used in our study.

4.1 Information diffusion model

When a piece of information originates from seed users, predicting its overall spread in the network is essential for the influence maximization (IM). In this paper, the classical independent cascade (IC) model is used as the information diffusion [39]. In the IC model, initial seed nodes try to activate their inactive neighbors. The success of a node u in activating a neighboring node v depends on the propagation probability $p(u, v)$. Each edge $e = (u, v)$ has its own value of probability of propagation. When a set of seeds nodes is chosen, and activation probability $p(u, v)$ for each edge (u, v) is provided, a single simulation of the IC model is evolved in the following discrete steps:-

An active node (u) that gets activated at time t, tries to activate its inactive adjacent node (v), with probability of $p(u, v)$ at discrete time step $t + 1$. After every fulfilled attempt, a neighbor may get activated. In the IC model, every activated node is capable of activating its inactive neighbors only once. Once all the nodes are activated, or there are no remaining active nodes, the propagation ends. The influence spread of seed set S is referred to as the total number of nodes that got activated at the end of the spreading process. To get smooth results, we may need to run many simulations of the IC model. The resulting values are averaged over the total number of simulations of the IC model.

4.2 Datasets

The following used datasets are of different sizes and applications.

- Football [42]: The nodes represent football team and edges represent that two teams have played together.
- Email [43]: The nodes represent the users and the edges represent that email is exchanged between them.
- Jazz Musicians [44] : Network of Jazz musicians where node represents a musician and an edge denotes that two musicians have played together in an band.

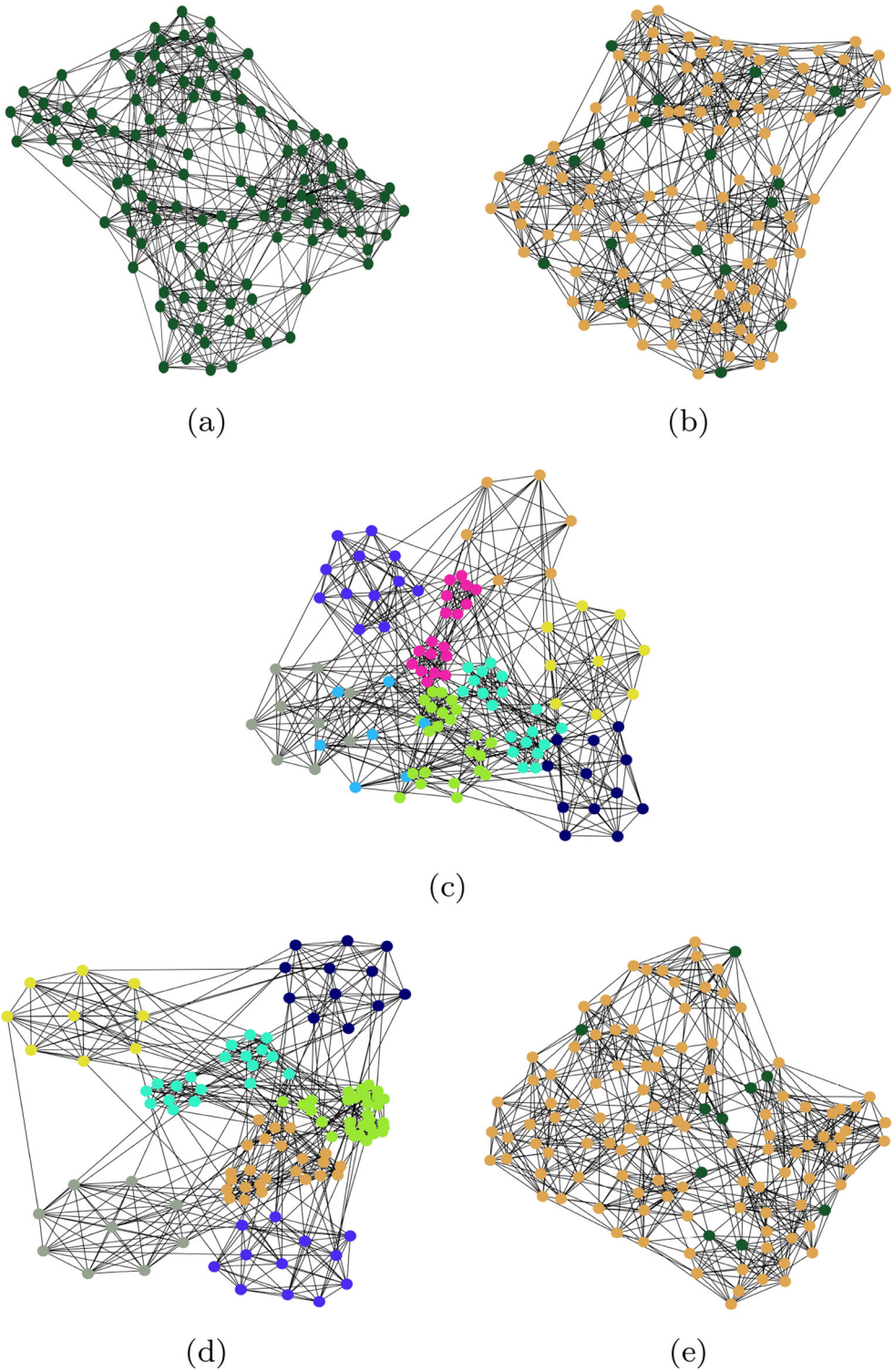


Fig. 2 Toy Network, which explains the working of our algorithm: **a** Input Toy Network **b** Seeding Phase **c** Label Propagation Phase **d** Merging communities using R matrix **e** Selected ten influential nodes colored in green

- Caenorhabditis Elegans [45]: Neural network of neurons and synapses in C. elegans, a type of worm.
- Facebook Pages [46]: This dataset represent blue verified Facebook page network of different categories. Nodes represent the pages and the edges are mutual likes among them.
- Wiki Vote [47]: The dataset contains the Wikipedia voting data from its inception till January 2008. Nodes represent Wikipedia users and the directed edges from node u to node v represent that user u voted on user v.
- Technological Routers [48]: Computer network of routers and their connectivity with one another.

- Gnutella P2P 08 [49]: This is a network of Gnutella hosts from 2002. The nodes represent Gnutella hosts, and the edges represent connections between them.

The brief statistical features of the network datasets used for conducting the experiments are summarized in Table 1 where n, m, D, d_{max}, and $\hat{d}$ represent the number of nodes, number of edges, density, the maximum degree of nodes, and average degree of the nodes respectively. Density (D) of the networks is the ratio of the number of all existing edges to the total possible number of edges in the network [50].

4.3 Performance metrics

The following performance metrics are considered in order to compute the performance of our devised algorithm along with other previously devised ones:

(i) Infected Scale ($F(t)$) The infected scale is the fraction of total nodes that become active at various time steps during the information diffusion process. The metric compares the nodes infected, relative to discrete time steps. The number of nodes getting infected continuously changes with every discrete time step, and the spread of infection peaks at a specific value of time. Infected Scale ($F(t)$) is computed using the following formula:

$$F(t) = \frac{n_a(t)}{n} \tag{8}$$

where $n_a(t)$ is the number of nodes at timestamp t and n is the total number of nodes in the system.

(ii) Final Infected Scale ($F(t_c)$) or Influence spread: The final infected scale or influence spread is the fraction of total nodes that become active at the end of the spreading process caused due to the triggering of the selected set of influential nodes. The number of initial influential nodes directly affects the final infected scale in the network. The fraction of total nodes that act as initial spreaders is called the spreader fraction. Usually, the high value of the spreader fraction leads to the high value of the final infected scale. Considering the real-life network scenarios, comprehending an optimal maximum value of the spreader fraction becomes an important task. The final infected scale ($F(t_c)$) is computed using the following formula:

$$F(t_c) = \frac{n_{\text{total active nodes}}}{n} \tag{9}$$

where n is the total number of nodes and $n_{\text{total active nodes}}$ is the number of nodes that are active after the whole process of information spreading is completed.

(iii) Kendall Tau Correlation (τ): Kendall tau correlation (τ) [51] is employed to determine the precise spreading influence of the selected nodes that act as initial spreaders. This method is to find the correlation between the rank list (L_1) generated by various algorithms with the natural ranking list (L_2) generated by the IC Model. For a pair of seed nodes, if $x_i > y_j$ in the list L_1, and $x_i > y_j$ in actual rank list L_2 then such pair is known as concordant pair otherwise it is called discordant pair. Kendall tau correlation (τ) between two ranked list L_1 and L_2 is computed using the following formula:

$$\tau(L_1, L_2) = \frac{n_{con} - n_{dis}}{\frac{1}{2}n(n-1)} \tag{10}$$

where n_{con} is the number of concordant pairs, n_{dis} is the number of discordant pairs and n is total number of nodes in both lists L_1 and L_2.

(iv) Average Shortest Path Length (L_s): The average shortest path length between selected spreaders is the average of the shortest path length between all possible pairs of selected influential nodes. It determines how efficient the information passing process in a social network is. Usually, a large value of the average shortest path length indicates that the selected influential nodes are evenly spread out across the network and are not concentrated in a small region of the network. This makes a larger value of the metric to be a desirable one as it facilitates the effective and efficient spread of information. The average shortest

Table 1 Brief statistics of datasets

Network	n	m	D	d_{max}	$\hat{d}$
Football	115	613	0.0935	12	10
Email	143	623	0.0613	42	8
Jazz Musicians	198	2742	0.1406	100	27
C. Elegans	297	2148	0.0533	134	15
Facebook Pages	620	2091	0.0109	132	6
Wiki Vote	889	2914	0.0074	102	6
Tech. Routers	2113	6632	0.0029	109	6
Gnutella P2P 08	6301	20777	0.0010	97	6

path length (L_s) is computed using the following formula:

$$L_s = \frac{1}{k \times (k-1)} \sum_{u,v \in S} l_{u,v} \tag{11}$$

where k is the number of influential nodes, S be the set of selected influential nodes, and $l_{u,v}$ is the shortest path length between influential nodes u and v, if any such path exists between them.

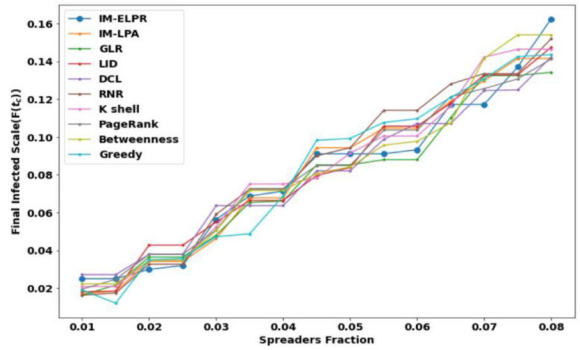
5 Experimental results

In this section, we present a comparative analysis of the proposed algorithm named as Influence Maximization using Extended h-index, and Label Propagation with Relationship matrix (IM-ELPR). For the performance comparison, we consider some well known classical algorithms such as Betweenness centrality [14], Greedy algorithm [8], PageRank [16], k-shell centrality [17] and some recently published influence maximization algorithms namely, IM-LPA [33], global local ranking (GLR) [34], DCL (Degree, Clustering coefficient, and location based method) [35], Reversed Node Ranking (RNR) [36], and local information dimensionality (LID) algorithm [37]. We adopt four different performance metrics, as mentioned in Section 4.3, to judge the performance of the proposed algorithm along with various algorithms stated above for the comparisons. We utilized eight different real-life datasets, as listed in Table 1, to perform the extensive analysis. The experimental results are compiled on a personal system with configuration as Intel(R) Core(TM) i7-7700T CPU @ 3.80 GHz processor and 4 Core(s), and 8GB RAM.

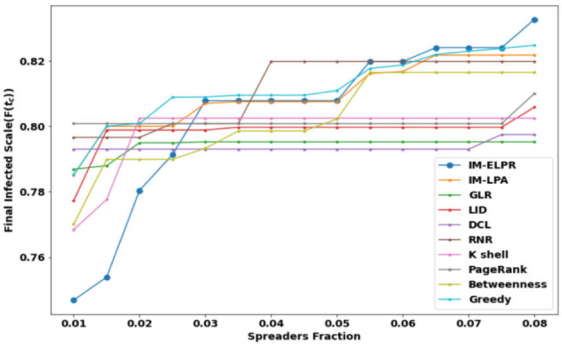
We present below the detailed comparisons of the experimental results obtained by the proposed method along with other methods based on each performance measure as listed in Section 4.3:

5.1 Final infected scale ($F(t_c)$) Vs. Spreader fraction

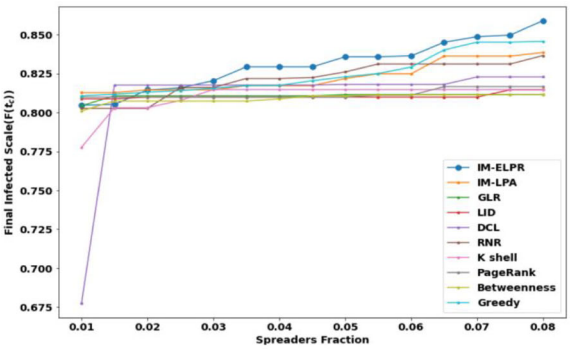
We plot the final infected scale ($F(t_c)$) or influence spread graph by taking various values of spreader fraction on the x-axis and the obtained value of the final infected scale on the y-axis. We keep spreader fraction values in the range 0.01 and 0.08 for the datasets having a total number of nodes less than 2000, and when total nodes are higher than 2000, the spreader fraction is taken in between 0.003 and 0.045. Different values of spreader fraction are used for smaller and bigger datasets because the final top k influential nodes selected may either be too small for smaller datasets or too large for larger datasets. Hence, we use relatively larger spreader fraction value for smaller datasets and relatively smaller spreader fractions for larger datasets. The $F(t_c)$ values in the resulting graphs are the average values produced by 100 independent simulations of independent cascade (IC) information diffusion model. Here, one simulation of the IC model means running the information diffusion model one time to propagate the information in the network and computing the influence spread originating from the selected seed noes. The obtained results for $F(t_c)$ are depicted in Fig. 3. The proposed algorithm, along with all the considered algorithms for comparisons, are simulated to calculate $F(t_c)$ values on all the datasets with different activation probability (p) in the IC model. This is because each dataset has a different number of nodes, density, and structural properties. Figure 3a depicts the result of the American Football dataset. We take activation probability (p) value as $p = 0.05$. This dataset is relatively small, and almost all algorithms perform similarly in the beginning. However, when the spreader fraction is more than 0.07, the proposed algorithm outperforms other algorithms. In the case of the Email dataset, we consider probability as $p = 0.30$. In this case, as the spreader fraction (number of influential nodes) increases, our approach slightly performs better as per Fig. 3b followed by the Greedy method. Figure 3c presents the result on Jazz Musicians with probability values $p = 0.1$. This network is relatively dense, and our approach performs pretty well. The second-best approach is the Greedy method. Results for C. Elegans are shown in Fig. 3d, with probability value, $p = 0.25$. For all spreader fractions, our method outperforms all other methods. Here, the second-best algorithm is LID. Figure 3e represents the results of Facebook Pages, which is a slightly larger network than previous networks. We have used probability value $p = 0.55$. This network has relatively less density, and thus the probability for activating neighboring nodes in the IC model is kept high as compared to other networks. IM-ELPR, IM-LPA, and RNR perform better than all other algorithms, but our algorithm outperforms all, including IMLPA and RNR. For the Wiki Vote network, the probability used is $p = 0.3$. As depicted in Fig. 3f, this network is also less dense. The proposed method performs better than IM-LPA, and both these methods surpass other methods. Figure 3g illustrate the result of Technological Routers, which is a medium-sized network. Probability values chosen are, $p = 0.35$. In this network, our algorithm carries out much better than others. This is because of the less density, as each node is connected to a very less number of nodes. Hence, when the probability is less, a situation may arise where a node doesn't activate any of its neighbors even when it has inactive neighbors. In the case of Gnutella P2P, we took activation probability, $p = 0.2$. For all the spreader fraction (and hence for all k), the proposed method is far better than any of the other algorithms, as depicted in Fig. 3h. From the results obtained in Fig. 3, it is observed that the



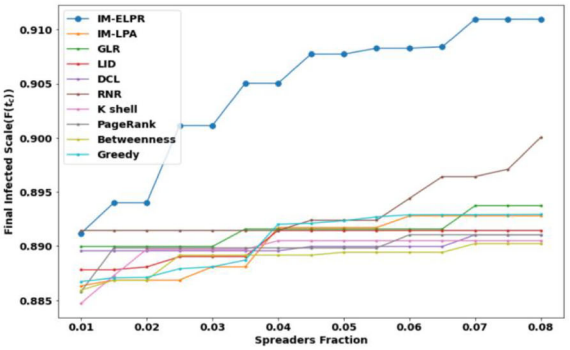
(a) Football , p=0.05



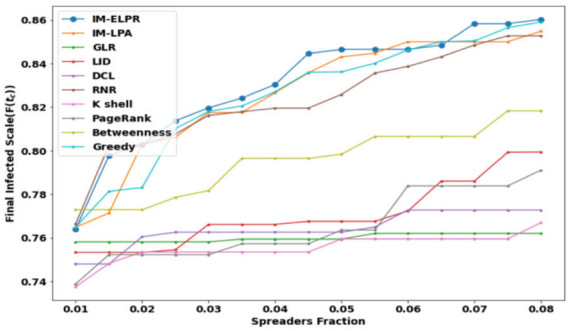
(b) Email , p=0.3



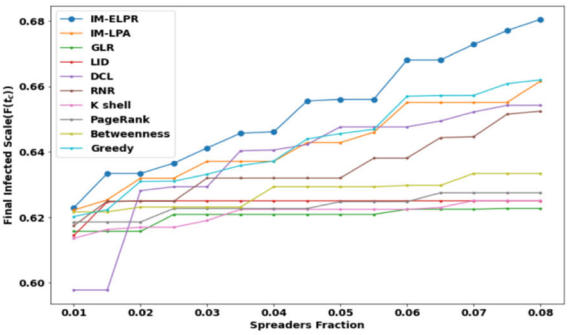
(c) Jazz , p=0.1



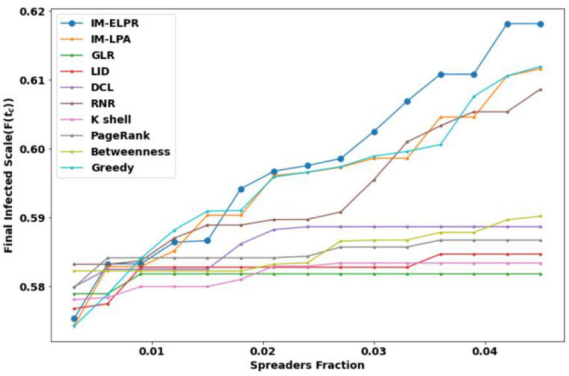
(d) C.Elegans ,p=0.25



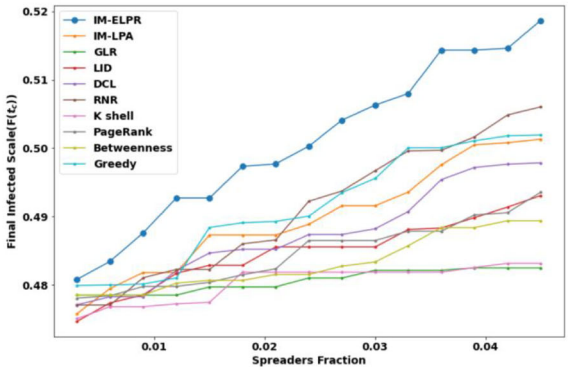
(e) Facebook Pages,p=0.55



(f) Wiki-Vote, p=0.3

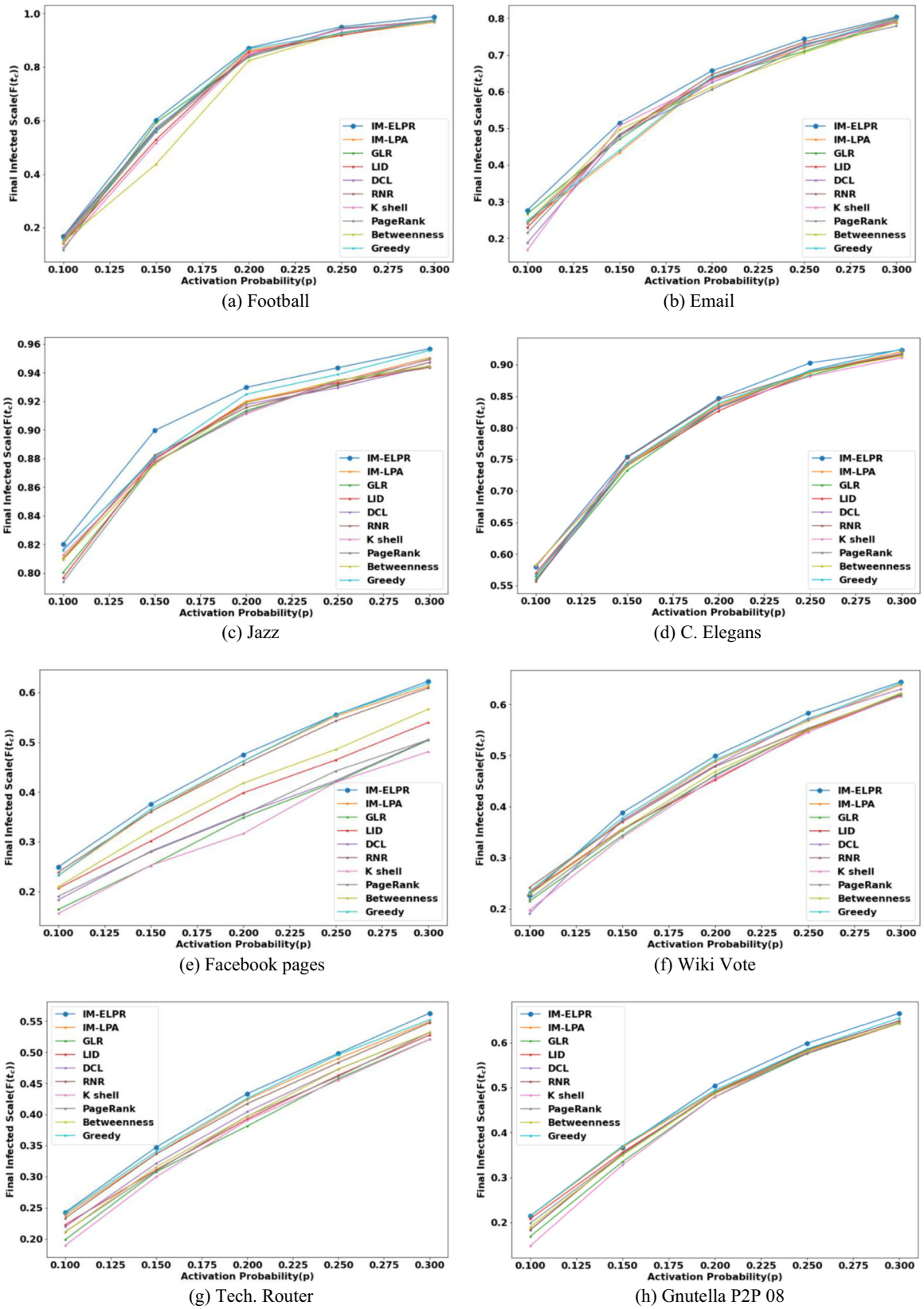


(g) Tech. Router, p=0.35



(h) Gnutella P2P 08, p=0.2

Fig. 3 **a**–**g** Final infected scale vs. spreader fraction values for different datasets- Football, Email, Jazz, C. Elegans, Facebook, Wiki Vote, Tech Router, and Gnutella P2P. The results are generated over 100 independent simulations of the IC model with given activation probability (p) value in the result of each dataset



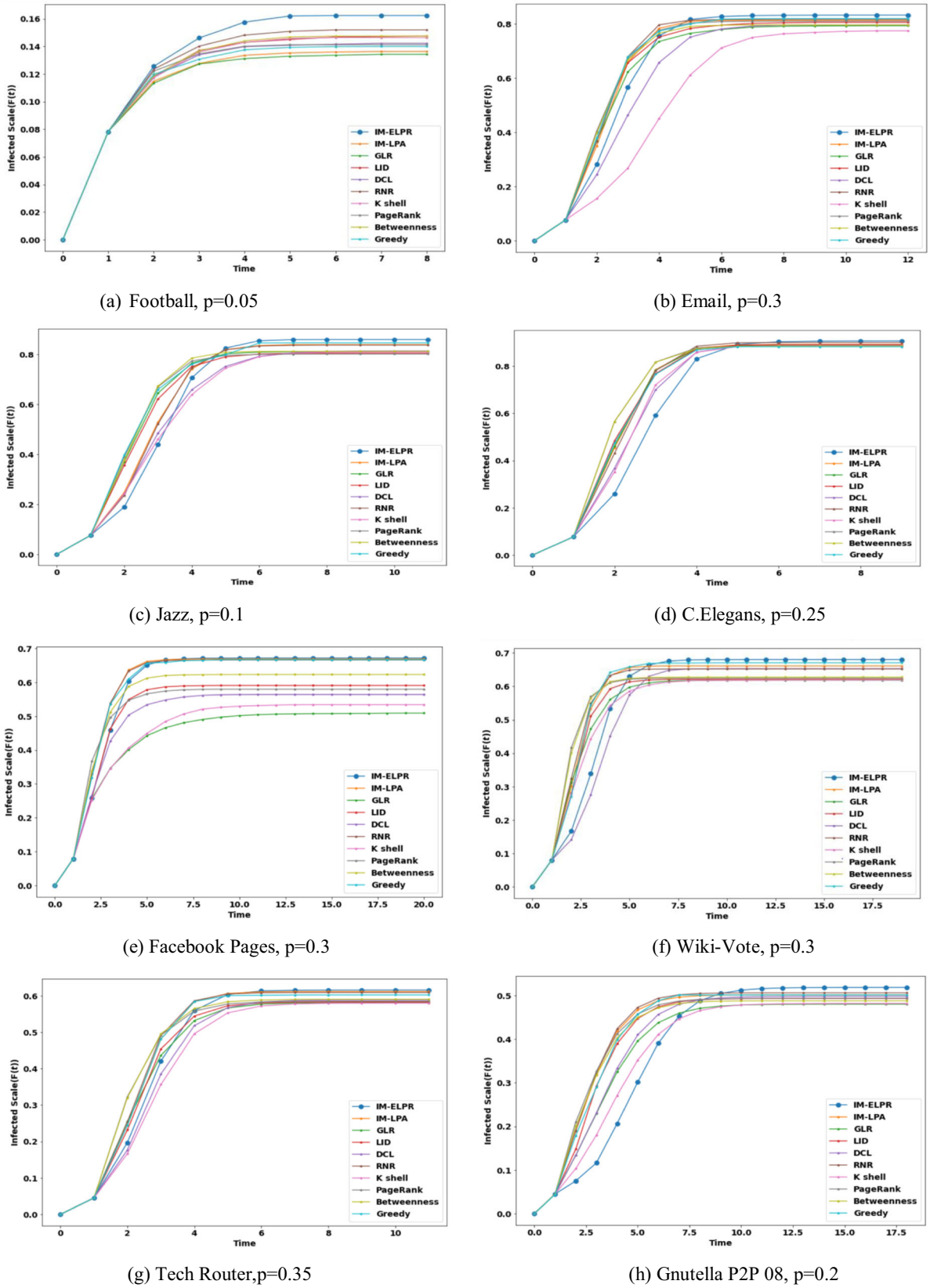
(a) Football (b) Email

(c) Jazz (d) C. Elegans

(e) Facebook pages (f) Wiki Vote

(g) Tech. Router (h) Gnutella P2P 08

Fig. 4 **a**–**g** Final infected scale vs. activation probability (p) values for different datasets- Football, Email, Jazz, C. Elegans, Facebook, Wiki Vote, Tech Router, and Gnutella P2P. The results are generated over 100 independent simulations of the IC model by taking the top 3% of nodes as the initial spreaders



(a) Football, p=0.05

(b) Email, p=0.3

(c) Jazz, p=0.1

(d) C.Elegans, p=0.25

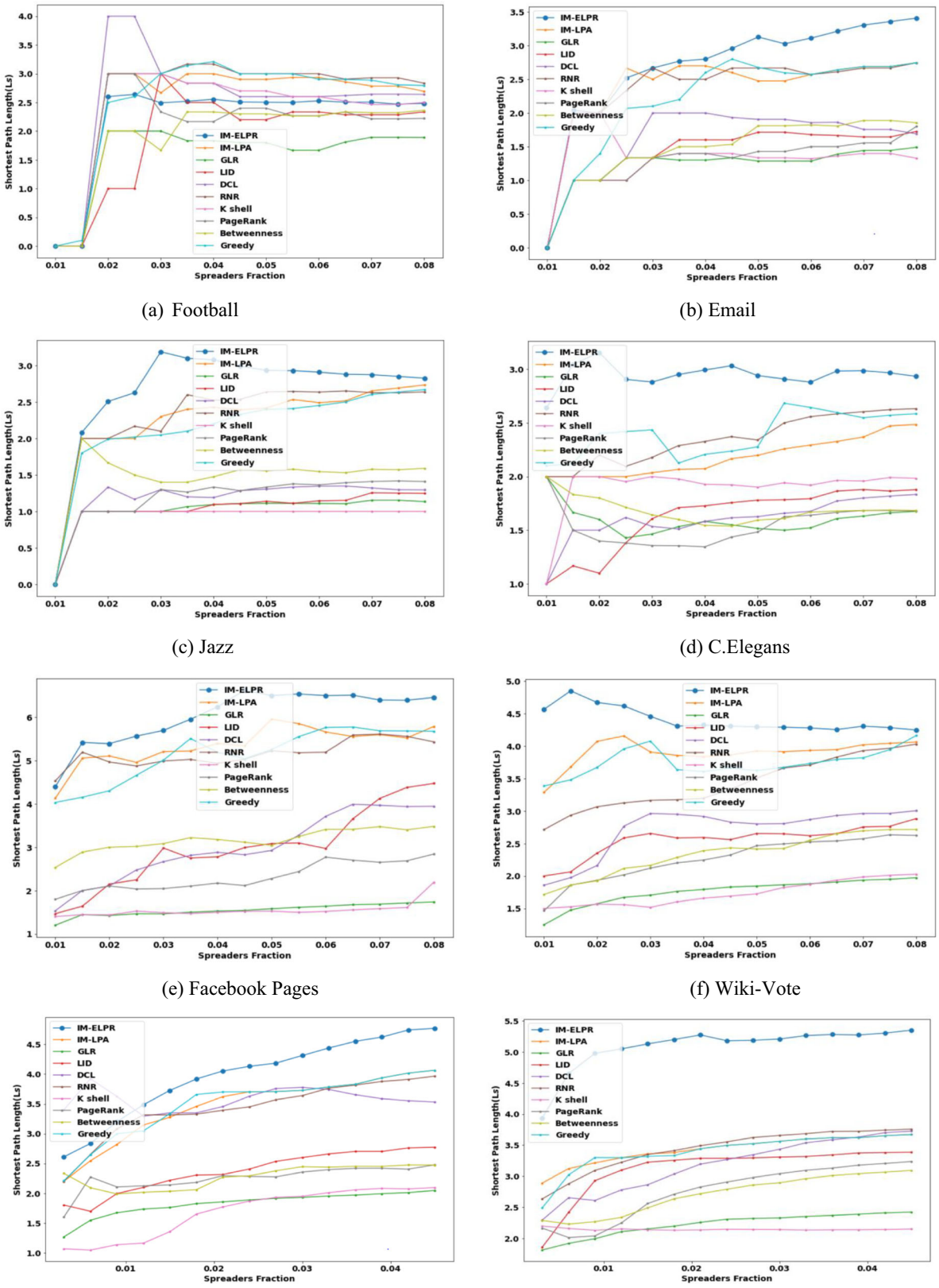
(e) Facebook Pages, p=0.3

(f) Wiki-Vote, p=0.3

(g) Tech Router,p=0.35

(h) Gnutella P2P 08, p=0.2

Fig. 5 **a**–**g** Infected Scale $F(t)$ vs. Time values for different datasets - Football, Email, Jazz, C. Elegans, Facebook Pages, Wiki Vote, Tech Routers, and Gnutella P2P. The results are generated over 100 independent simulations of the IC model with a given activation probability (p) value in the result of each dataset



(a) Football (b) Email

(c) Jazz (d) C.Elegans

(e) Facebook Pages (f) Wiki-Vote

(g) Tech. Router (h) Gnutella P2P 08

Fig. 6 **a**–**h** Average shortest path length (L_s) Vs. Spreader Fraction values for different datasets - Football, Email, Jazz, C. Elegans, Facebook Pages, Wiki Vote, Tech Routers, and Gnutella P2P 08

proposed algorithm performs better and consistently well than other algorithms considered in the comparison. In most of the cases, the second-best algorithm is the Greedy method followed by LID, RNR, or IM-LPA, and other techniques.

5.2 Final infected scale ($F(t_c)$) Vs. Activation probability (p)

We also perform the impact on the value of final infected scale by varying activation probabilities. Figure 4 shows the plot of final infected scale ($F(t_c)$) vs. activation probability (p) on all the datasets used in this study. To perform the analysis, we keep the values of activation probability (p) in the range of 0.10, 0.15, 0.20, 0.25 and 0.30. We consider the top 3% of the nodes as the initial spreaders nodes chosen by various methods to obtain the value of final infected scale against a particular value of activation probability (p). The results are obtained over 100 independent simulations of IC model. From the results, it is evident that as the value of activation probability increases the value of final infected scale or influence spread also increases in all the algorithms of influence maximization for each dataset. The obtained results depicts that our proposed algorithm (IM-ELPR) performs uniformly well and outperforms all other algorithms on this evaluation parameter.

5.3 Infected scale ($F(t)$) Vs. Time

We employ the infected Scale ($F(t)$) metric to compare the results of various algorithms along with the proposed algorithm, which compares the fraction of active nodes at each discrete timestamp during the process of information spreading. The results for $F(t)$ vs. Time are shown in Fig 5. Here, time on x-axis denotes the time-step of the independent cascade (IC) information diffusion model. In this paper, we utilize the IC model to disseminate the information originating from selected seed nodes. At the time, t=1, only selected seed nodes are infected. At $t = 1$, initial seed nodes try to activate their neighbors with activation probability (p). Hence, the active node that gets infected at time t tries to infect its inactive neighbors at time $t = t+1$, and this process continues. We run the IC model for 100 times. The value of the infected scale ($F(t)$) on y-axis is the fraction of total nodes that are active averaged over 100 iterations of the IC model for any particular time-step. To perform the Infected scale ($F(t)$) vs. Time analysis, we consider the number of spreader fractions as 0.08, i,e, top 8% of the nodes are chosen as influential spreader nodes under each algorithm. Considering the number of nodes in each network as mentioned in Table 1, the corresponding values of influential spreader nodes come out to be 9, 11, 15, 23, 49, 71, 169, and 504 for the datasets Football, Email, Jazz, C. Elegans, Facebook Pages, Wiki Vote, Tech Routers and Gnutella P2P, respectively. For example, the Football dataset has a total of 115 nodes whose 8% comes out to be is 9. The values of the fraction of active nodes in the results are the average values produced by 100 independent simulations of the IC model.

In majority of the networks, our approach starts at a lower pace, that is in the beginning the number of active nodes in our method are less. But, as the time progresses, our method outshines other methods and the active nodes are much greater. The slow start but higher performance at later time can be explained by the use of extended h-index in the seed phase as it identifies nodes which spread influence up to nodes that are 3-hops away. Hence, the influence is spread out to a relatively more substantial part of the network. In the case of Facebook and Tech Router datasets, the performance of the proposed algorithm is similar to the Greedy, M-LPA, and RNR algorithms but leads to all other approaches. On the datasets like Football, Jazz, C. Elegans, and Wiki Vote, the Greedy method performs second-best after the proposed algorithm (IM-ELPR).

Table 2 Kendall Tau values for different datasets calculated using various algorithms such as IM-EL (IM-ELPR), IM-LPA, GLR (Global Local Ranking), LID (Local Information Dimensionality), DCL (Degree, Clustering coefficient, and Location-based method), RNR (Reversed Node Ranking), KS (k-shell), PR (PageRank), BW (Betweenness) centrality and GD (Greedy method)

Dataset	IM-EL	IM-LP	GLR	LID	DCL	RNR	KS	PR	BW	GD
FT	0.77	0.66	0.51	0.75	0.57	**0.85**	0.45	0.71	0.48	0.72
Email	**0.88**	0.54	0.49	0.82	0.54	0.81	0.47	0.65	0.50	0.67
Jazz	0.69	0.53	0.47	**0.79**	0.53	0.77	0.56	0.61	0.59	0.46
CE	0.79	0.62	0.51	**0.87**	0.58	0.79	0.53	0.74	0.56	0.62
FB	**0.83**	0.73	0.60	0.80	0.48	0.78	0.44	0.70	0.49	0.57
WV	0.71	0.65	0.51	0.69	0.60	0.63	0.42	0.66	0.45	**0.75**
TR	**0.78**	0.61	0.57	0.74	0.54	0.69	0.48	0.71	0.44	0.70
GP2P	**0.85**	0.79	0.59	**0.85**	0.47	0.78	0.57	0.67	0.53	0.75

Bold entries signify the maximum value attained by a particular method in each dataset

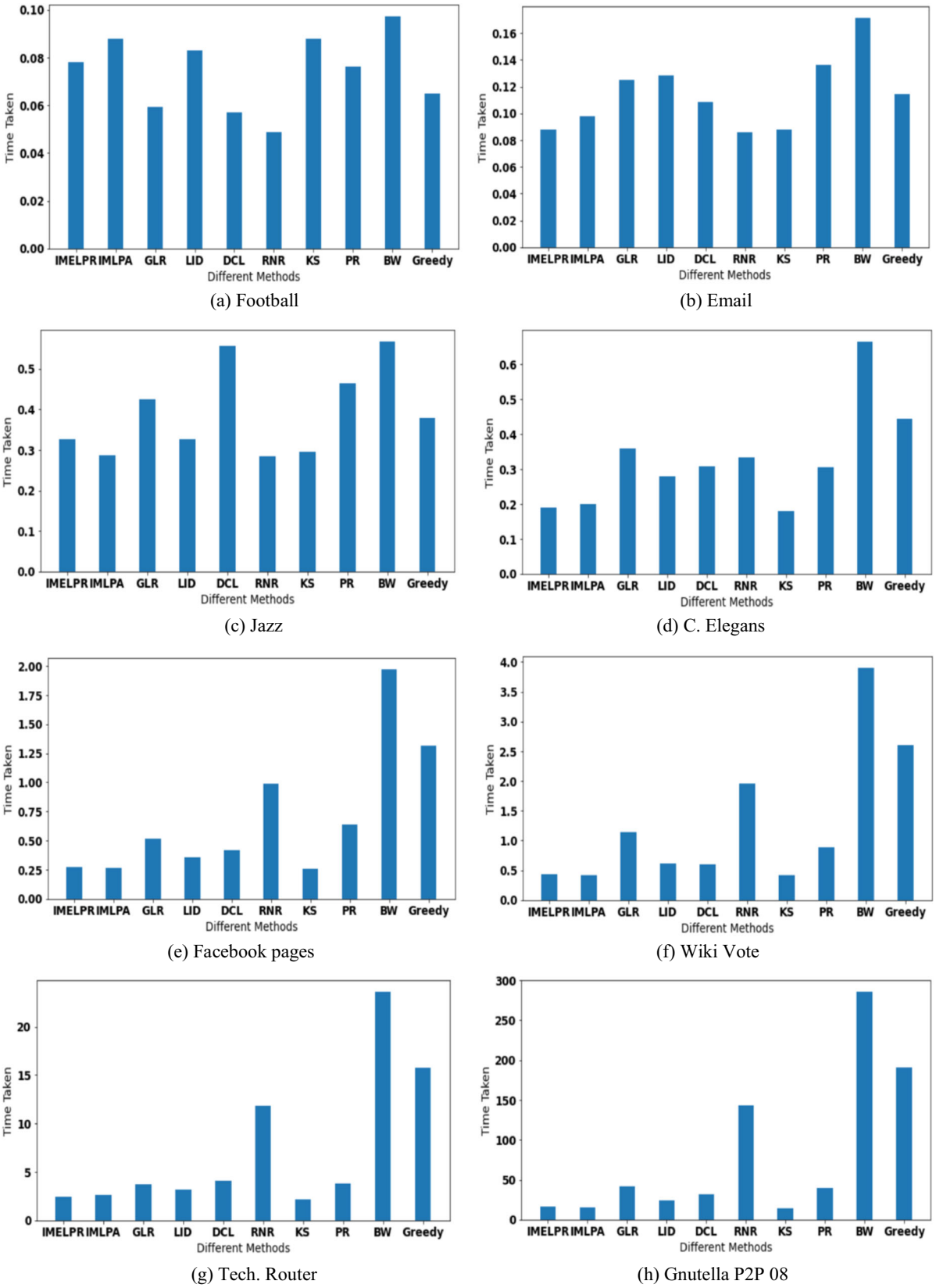


Fig. 7 **a**–**h** Execution time (in second) comparison of various algorithms to rank the spreading capability of nodes for different datasets - Football, Email, Jazz, C. Elegans, Facebook Pages, Wiki Vote, Tech Routers, and Gnutella P2P 08. The result is shown for finding the top 3% of nodes based on their spreading capability

5.4 Average shortest path length (L_s) vs. spreader fraction

After locating the influential spreaders by different algorithms in this study, the average shortest path length (L_s) between selected spreaders is computed using (11). The result is plotted, taking spreader fraction on the x-axis and average shortest path length on the y-axis. The results for L_s vs. spreader fraction are shown in Fig. 6 for all datasets. In all the datasets except football, the value for L_s value attained by IM-ELPR is considerably more than all other approaches, which is the same pattern in case of the final infected scale ($F(t_c)$) Vs. Spreader fraction, as observed in Fig. 3. We can see from Fig. 6, L_s Vs. spreader fraction graph, the influential nodes are chosen by our method is more scattered in the network than the influential nodes of any other method, leading to high influence spread by the proposed method.

5.5 Kendall Tau's correlation

Kendall tau coefficient (τ) is an important metric that can be used to judge the performance of IM algorithms. It is employed to determine the correlation between the rank list generated by each algorithm with the actual rank list obtained from the IC spreading model. In order to get the results, we simulate the IC model independently 50 times. The value of activation probability (p) is considered as 0.075, 0.2, 0.1, 0.1, 0.5, 0.35, 0.35, and 0.165 for datasets Football (FT), Email, Jazz, C. Elegans (CE), Facebook Pages (FB), Wiki vote (WV), Tech. Router (TR), and Gnutella P2P 08 (GP2P), respectively. For each dataset, the activation probability (p) value used here is the same as that used in the metric $F(t_c)$. The obtained results of various algorithms on different datasets are listed in Table 2. In the majority of the datasets, the Kendall tau value produced by various methods is very close. From the results, we can observe that on the Email, Facebook pages, Tech. Routers, the Kendall tau (τ) value attained by the proposed algorithm, is maximum. However, in the case of the Football dataset, RNR performs best followed by IM-ELPR. For the Jazz and C.Elegans datasets, LID produces the best result followed by the RNR method. For the Wiki vote dataset, the greedy method produces the best result followed by IM-ELPR. For Gnutella P2P 08 dataset, IM-ELPR, and LID, both the methods produce the same results and outperform other methods.

5.6 Execution time comparison

We also assess the absolute time required to produce the ranking of nodes in terms of their spreading capability by the proposed model and other methods in comparison. In Fig. 7, we compare the actual running time required to rank the nodes based on the spreading capability using the proposed algorithm, IM-ELPR, and all other algorithms. We performed the examination on all the eight datasets as listed in Table 1. For each dataset, we compute the time taken in seconds by each algorithm to find the top 3% nodes as the chosen seed nodes. We observed that the actual time taken by IM-ELPR for the generation of the ranked list is relatively less as compared to other methods in most of the cases. In the football dataset, our algorithm takes .09 second, which is slightly more than the time taken by GLR, DCL, RNR, and Greedy method. For the Email dataset, the time requires to find the top spreaders by the proposed algorithm is 0.10 second; however, RNR, IM-LPR, RNR, and k-shell each require 0.08 second. For the Jazz dataset, the proposed method takes just 0.28 second to generate the ranking list and outperforms all other techniques. Similarly, in the case of Facebook pages, Wiki Vote, and Tech. Router datasets, our method requires 0.25 seconds, 0.5 seconds, 3 seconds respectively, and beats all other methods. To process 6301 nodes and find the top 3% nodes, i.e., 190 nodes in Gnutella P2P 08 network, the proposed algorithm takes only 21 seconds, which is significantly faster than GLR, DCL, RNR, PageRank, Betweenness, and Greedy method.

According to the results achieved in all the performance metrics namely final infected scale ($F(t_c)$), infected scale ($F(t)$), average shortest path length (L_s), Kendall tau correlation (τ), and actual execution time comparison the proposed algorithm IM-ELPR for influence maximization delivers better results than many well-known classical and recently published methodologies.

6 Conclusion

Finding influential spreaders is a prominent activity to achieve viral marketing and influence maximization in complex networks like social networks. The presence of community structure is quite common in social networks, and information propagation can take place rapidly among the members of the community. In this paper, we introduce the IM-ELPR algorithm as a solution to influence maximization problem using the idea of label propagation and community structure. We obtain the initial seed nodes using extended h-index centrality, and then label propagation is performed where each seed node is assigned a unique label. At the end of label propagation, the detection of communities is accomplished. Further, we merge the related community using a relationship matrix to increase the modularity. The relationship matrix proves to be more effective in handling larger networks as the merging of communities removes the many small communities and causes the maximum spread of information in the network

with minimal effective influential nodes chosen from the merged community. We utilise four performance metrics to judge the performance of our algorithm alongside the performance of other various popular methods of influence maximization. The experimental result obtained on eight real-life networks of different sizes and applications shows that the proposed algorithm outperforms many well-known existing algorithms. Further, the overall time complexity of the IM-ELPR algorithm is linear in terms of the network size, which suggests the effectiveness of the proposed algorithm in large-scale networks.

References

1. Boccaletti S, Latora V, Moreno Y, Chavez M, Hwang DU (2006) Complex networks: Structure and dynamics. Phys Rep 424(4-5):175–308
2. Strogatz SH (2001) Exploring complex networks. Nature 410(6825):268–76
3. Mislove A, Marcon M, Gummadi KP, Druschel P, Bhattacharjee B (2007) Measurement and analysis of online social networks. In: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, pp 29–42
4. Krasnova H, Spiekermann S, Koroleva K, Hildebrand T (2010) Online social networks: Why we disclose. J Inform Technol 25(2):109–125
5. Heidemann J, Klier M, Probst F (2012) Online social networks: a survey of a global phenomenon. Comput Networks 56(18):3866–3878
6. Li Y, Fan J, Wang Y, Tan KL (2018) Influence maximization on social graphs: a survey. IEEE Trans Knowl Data Eng 30(10):1852–72
7. Chen W, Wang C, Wang Y (2010) Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 1029–1038
8. Kempe D, Kleinberg J, Tardos É (2003) Maximizing the spread of influence through a social network. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp 137–146
9. Kumar S, Panda BS (2020) Identifying influential nodes in Social networks: Neighborhood Coreness based voting approach. Physica A: Statistical Mechanics and its Applications, pp 124215
10. Domingos P, Richardson M (2001) Mining the network value of customers. In: Proceedings of the seventh ACM SIGKDD international conference on knowledge discovery and data mining, ACM, pp 57–66
11. Goldenberg J, Libai B, Muller E (2001) Talk of the network: a complex systems look at the underlying process of word-of-mouth. Market lett 12(3):211–23
12. Lü L, Chen D, Ren XL, Zhang QM, Zhang YC, Zhou T. (2016) Vital nodes identification in complex networks. Phys Rep 650:1–63
13. Bródka P, Skibicki K, Kazienko P, Musiał K (2011) A degree centrality in multi-layered social network. In: 2011 international conference on computational aspects of social networks (CASoN), IEEE, pp 237–242
14. Freeman LC (1977) A set of measures of centrality based on betweenness, Sociometry, pp 35–41
15. Bonacich P (2007) Some unique properties of eigenvector centrality. Social Networks 29(4):555–64
16. Page L, Brin S, Motwani R, Winograd T (1999) The pagerank citation ranking: Bringing order to the web. Stanford InfoLab
17. Kitsak M, Gallos LK, Havlin S, Liljeros F, Muchnik L, Stanley HE, Makse HA (2010) Identification of influential spreaders in complex networks. Nat Phys 6(11):888–93
18. Leskovec J, Krause A, Guestrin C, Faloutsos C, VanBriesen J, Glance N (2007) Cost-effective outbreak detection in networks. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 420–429
19. Liu W, Chen X, Jeon B, Chen L, Chen B. (2019) Influence maximization on signed networks under independent cascade model. Appl Intell 49(3):912–28
20. Bozorgi A, Samet S, Kwisthout J, Wareham T (2017) Community-based influence maximization in social networks under a competitive linear threshold model. Knowl Based Syst 134:149–58
21. Newman ME (2004) Detecting community structure in networks. Eur Phys J B 38(2):321–30
22. Ferrara E (2012) A large-scale community structure analysis in Facebook. EPJ Data Sci 1(1):9
23. Fortunato S (2010) Community detection in graphs. Phys Rep 486(3-5):75–174
24. Kumar S, Panda BS, Aggarwal D (2020) Community detection in complex networks using network embedding and gravitational search algorithm. Journal of Intelligent Information Systems, pp 1–22
25. Huang H, Shen H, Meng Z (2020) Community-based influence maximization in attributed networks. Appl Intell 50(2):354–64
26. Chen D, Lü L, Shang M-S, Zhang Y-C, Zhou T (2012) Identifying influential nodes in complex networks. Phys A 391:1777–1787
27. Hirsch JE (2005) An index to quantify an individual's scientific research output. Proc Natl Acad Sci 102(46):16569–72
28. Liu Q, Zhu YX, Jia Y, Deng L, Zhou B, Zhu JX, Zou P. (2018) Leveraging local h-index to identify and rank influential spreaders in networks. Physica A: Statistical Mechanics and its Applications 512:379–91
29. Rui X, Yang X, Fan J, Wang Z (2020) A neighbour scale fixed approach for influence maximization in social networks. Computing, pp 1–23
30. Lü L, Zhou T, Zhang QM, Stanley HE (2016) The H-index of a network node and its relation to degree and coreness. Nat Commun 7:10168
31. Zareie A, Sheikhahmadi A (2019) EHC: Extended H-index Centrality Measure for identification of users' spreading influence in complex networks. Physica A: Statistical Mechanics and Its Applications 514:141–55
32. Raghavan UN, Albert R, Kumara S (2007) Near linear time algorithm to detect community structures in large-scale networks. Phys Rev E 76(3):036106
33. Zhao Y, Li S, Jin F (2016) Identification of influential nodes in social networks with community structure based on label propagation. Neurocomputing 210:34–44
34. Salavati C, Abdollahpouri A, Manbari Z (2019) Ranking nodes in complex networks based on local structure and improving closeness centrality. Neurocomputing 336:36–45
35. Berahmand K, Bouyer A, Samadi N (2019) A new local and multidimensional ranking measure to detect spreaders in social networks. Computing 101(11):1711–33
36. Rui X, Meng F, Wang Z, Yuan G (2019) A reversed node ranking approach for influence maximization in social networks. Appl Intell 49(7):2684–98
37. Wen T, Deng Y (2020) Identification of influencers in complex networks by local information dimensionality. Inf Sci 512:549–62. Feb 1
38. Huang H, Shen H, Meng Z, Chang H, He H (2019) Community-based influence maximization for viral marketing. Appl Intell 49(6):2137–50

39. Goldenberg J, Libai B, Muller E (2001) Talk of the network: a complex systems look at the underlying process of word-of-mouth. Mark lett 12(3):211–23
40. Li X, Zhou S, Liu J, Lian G, Chen G, Lin CW (2019) Communities detection in social network based on local edge centrality. Physica A: Statistical Mechanics and its Applications 531:121552
41. Newman ME (2006) Modularity and community structure in networks. Proc Natl Acad Sci 103(23):8577–82
42. Girvan M, Newman ME (2002) Community structure in social and biological networks. Proc Natl Acad Sci 99(12):7821–6
43. Rossi R, Ahmed N (2015) The network data repository with interactive graph analytics and visualization. In: Twenty-Ninth AAAI Conference on Artificial Intelligence
44. Gleiser PM, Danon L (2003) Community structure in jazz. Advances in Complex Systems 6(04):565–73
45. Duch J, Arenas A (2005) Community detection in complex networks using extremal optimization. Phys Rev E 72(2):027104
46. Rozemberczki B, Davies R, Sarkar R, Sutton C (2019) Gemsec: Graph embedding with self clustering. In: Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp 65–72
47. Leskovec J, Huttenlocher D, Kleinberg J (2010) Signed networks in social media. In: Proceedings of the SIGCHI conference on human factors in computing systems, pp 1361–1370
48. Spring N, Mahajan R, Wetherall D (2002) Measuring ISP topologies with Rocketfuel. ACM SIGCOMM Comput Commun Rev 32(4):133–45
49. Ripeanu M, Foster I, Iamnitchi A (2002) Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system design. arXiv:cs/0209028
50. Xu G, Zhang Y, Li L (2010) Web mining and social networking: techniques and applications. Springer Science & Business Media
51. Kendall MG (1938) A new measure of rank correlation. Biometrika 30(1/2):81–93

Affiliations

Sanjay Kumar[1,2] · Lakshay Singhla[3] · Kshitij Jindal[4] · Khyati Grover[5] · B. S. Panda[1]

Lakshay Singhla
lukkysinghla@gmail.com

Kshitij Jindal
jindal.kshitij@yahoo.com

Khyati Grover
khyati.grover@yahoo.com

B. S. Panda
bspanda@maths.iitd.ac.in

1 Computer Science and Application Group, Department of Mathematics, Indian Institute of Technology Delhi, Hauz Khas, New Delhi, 110016, India

2 Department of Computer Science and Engineering, Delhi Technological University, Main Bawana Road, New Delhi, 110042, India

3 Samast Technologies Pvt Ltd, Sector 29, Gurgaon, 122001, India

4 Nuvogen Limited, Ctra. Moraira- Teulada, 444 C.C. Baraclays, Moraira, 03726, Alicante, Spain

5 Amazon India, Gurgaon, 122004, India