Identifying Influential Nodes for Smart Enterprises using Community structure with Integrated Feature Ranking

Sanjay Kumar, Akshi Kumar *Senior Member, IEEE* and B. S. Panda

Abstract—Finding influential nodes reshuffles the very notion of linear paths in business processes and replaces it with networks of business value within a smart enterprise system. There are many existing algorithms for identifying influential nodes with certain limitations for applying in large-scale networks. In this paper, we propose a community structure with an Integrated Features Ranking (CIFR) algorithm to find influential nodes in the network. Firstly, we use the community detection algorithm to find communities in the system and then we rank the nodes of network based on three factors, namely- local ranking, gateway ranking, and community ranking, collectively termed as integrated features. Our algorithm intends to select influential nodes, which are both globally and locally optimal, leading to overall high information propagation. We perform the experimental results on total eight networks using various evaluation parameters. The obtained results validate superior performance against contemporary algorithms adding value to smart enterprises.

Index Terms—Smart enterprises, Community structure, Business networks, Influential nodes, Node centrality

1 Introduction

A smart enterprise system is a built on the pillars of mobile workforce, flexible on-demand delivery, collaborative communities with trustworthy and secure services. Many enterprises have outspread their business activities to network platforms [1]. A smart enterprise must engage by involving and investing in people who are instrumental in reaching personal, group and organizational goals. Social footprints of key influencers can accelerate returns, assure high level of trust, and a wider reach within a smart enterprise design journey. The influencer pool in collaborative communities can reinvent the customer engagement models and act as the key driver for reshaping smart enterprises.The advantages of various social networks for the promotion of business and viral marketing are universally accepted primarily because of its low cost and fast information diffusion characteristics [2] [3]. Undoubtedly, social networks (SNs) play a crucial role in information propagation and user interactions building smart networks and finding influential nodes in these networks has gained much popularity among researchers because of the business value. The maximum spread of the information in the network can be achieved through the influential nodes. There are many node centrality-based and greedy-based algorithms for identifying influential nodes with certain limitations for applying in large-scale networks.

- *Sanjay Kumar is with the Department of Computer Science and Engineering, Delhi Technological University, New Delhi-110042,India. E-mail: sanjay.kumar@dtu.ac.in, Akshi Kumar is with Department of Computing and Mathematics, Faculty of Science and Engineering, Manchester Metropolitan University, Manchester, United Kingdom. E-mail: akshi.kumar@mmu.ac.uk, B. S. Panda is with Computer Science and Application Group, Department of Mathematics, Indian Institute of Technology Delhi, Hauz Khas, New Delhi-110016, India, E-mail: bspanda@maths.iitd.ac.in*

Manuscript received March 30, 2022; revised Jul 27, 2022

A social network is a complex and dynamic network where users produce and consume a massive amount of content creating economic outcomes. These quickly (re-)configuring links can be modeled as a graph $G = (V, E)$, where V represents a set of people or entities present and E denote edges. An edge connects two nodes if both the nodes have a social connection like friendship, follow-followee, coauthorship, etc. Based on the notion of word-of-mouth strategy in information diffusion and the trust between users, finding influential nodes is one of the most studied problems in the domain of network science in recent years because of its potential business value [4]. Identifying influential nodes is the task of choosing a constant number (k) of seed nodes with a high spreading capability such that if a piece of information originates from them, the information can reach the optimal number of vertices in the system through the diffusion cascade [5], [6]. It is a kind of optimization problem, and Kempe et al. [6] proved that getting an optimal solution is NP-hard under conventional information diffusion models. Mathematically, the problem of finding influential nodes is expressed as follows: "Given a social network $G(V, E)$, information diffusion model M, and a small and constant positive integer k, the aim is to select a subset $S \subset V$ of k users as the seed set to maximize the information spread such that for any other seed subset $S^* \subset V$ of k nodes, the following condition is satisfied:

$$\sigma_{G,M}(S^*) \leq \sigma_{G,M}(S) \tag{1}$$

where $\sigma_{G,M}(S)$ is the expected influence spread or numbers of nodes influenced by the nodes in set S."

Centrality-based methods are the most common way to rank the spreading ability of nodes to uncover influential nodes. These methods assign a rank or provide a score to nodes based on the importance of their topological position under various considerations in the network [7]. Numerous greedy-based algorithms [6], [8] are proposed to

find influential nodes using a greedy strategy and discrete optimization technique. Besides, many community-based, and deep learning based influence maximization methods are proposed with considerably improved performance [9]–[11].

The presence of community structure in business networks where several nodes in a community or module are relatively densely connected and have some similar features allow picking the best capabilities from many business network actors [12]. Community structure can give rise to insight into the system and understand its function. The appearance of such community structures in a network can accelerate the information propagation in the system [13]. In this paper, we propose Community-structure with Integrated Feature Ranking (CIFR) algorithm for finding influential nodes for smart enterprises. We divide the network into various communities using the Leiden algorithm [14], then focus on the influence of nodes by considering their contribution to their local community and the nodes of other communities. After considering communities as building blocks of the network, we also rank all those communities based on their relative importance. The proposed method considers the spreading ability of a node based on the following three parameters: (a) how good a node influences the nodes within the same community, i.e., the local ranking of the node?, (b) how well a node within a community infects the nodes of other communities, i.e., gateway ranking of the node?, and (c) how well is a community based on its number of connections to other communities, i.e., community ranking? Thus, we consider a combined effect of the above-said parameters to compute the final ranking of all the nodes in the network. We term these features of the nodes as integrated features. The proposed work for finding influential nodes for smart enterprises has many augmented intelligence applications as well. One prominent application is viral marketing, where influential nodes can change the decision, opinions, and preferences of many other nodes due to mutual trust. The influential nodes can work as business enterprise leaders, making a product or idea viral and leading many adopters. This can generate high profit for the business and its teams to keep up with the rapid pace of business disruption.
Tab 1 lists the various mathematical terms used in this work.

TABLE 1: Mathematical terms used

Notation	Description
σ	Influence spread
DC	Degree centrailty
KS	K-shell centrality
EV	Eigenvector centrality
HI	h-index centrality
GLR	Gateway local rank
LR	Local Ranking
GR	Global ranking
CR	Community ranking
$F(t)$	Infection scale
τ	Kendall tau correlation
β	Infection rate

The rest of the paper is organized as follows: Section 2 presents the related works and information diffusion models. Section 3 describe the proposed method named CIFR in detail, along with time complexity analysis. Section 4 reports the results and analysis of the proposed method, along with various popular existing methods based on multiple performance parameters. Finally, Section 5 presents the conclusion of the paper with its application in large scale networks.

2 Related Work

In recent years, identifying influential nodes have been studied extensively. Centrality-based algorithms are popular methods for ranking the spreading abilities of nodes. Such methods assign a score to each node by utilizing the topological structure of networks, and nodes having high score values are chosen as the seed nodes. Degree centrality, Betweenness Centrality, and Closeness Centrality, K-shell (KS) centrality [15], [16] are quite common centrality based methods for finding influential nodes. Chen et al. proposed local centrality (LC) [17] as a semi-local measure, which is an extension of degree centrality and considers the 1-hop and 2-hop neighbors into account. The notion of Eigenvector (EV) centrality [18] relies on the fact that a node is vital if it has connectivity with nodes that are considered significant in the system. PageRank [19] is a popular method to rank the web pages used by Google, which is an improved variant of Eigenvector centrality. The h-index (HI) centrality is a popular measure to assess the importance of a node [20]. The value of h-index of a node can be defined as maximum value h such that it should have a minimum of h neighbors with each having degree h or more.

In literature, several algorithms of finding influential nodes are proposed based on the idea of combining local and semi-local features of nodes and produced good results. Berahmand et al. [21] proposed the DCL algorithm for finding influential spreaders by utilizing a combination of local measures such as the number of common neighbors, degree, and inverse clustering coefficient. The reversed node ranking (RNR) method [22] employed the ranking information and the reversed rank as the importance for calculating the influence spread of node. The reversed rank relies upon the idea that the node with a higher rank has more information spreading ability. Wen and Deng [23] introduced a technique named LID for influence maximization using local information dimensionality by considering the local structural properties around the central node. They assumed the quasilocal information and lowered the computational complexity, and further utilized the notion of Shannon entropy.

By utilizing the presence of community structures in networks, the information propagation and influence spread in the system can be enhanced [24], [25]. In recent years, many community-based influence maximization algorithms are proposed. Such algorithms assume independence between communities and perform parallel execution. Cai et al. [10] recommended a heuristic influence maximization technique, which combines community detection and topic awareness into influence diffusion modeling. Salavati et al. [26] proposed the gateway local rank (GLR) method, which improves closeness centrality by community detection and utilizing the local structure of nodes. GLR method first identifies the community, and then in each community, it obtains one best local node and one gateway node using

centrality measures. Generally, the influence maximization method using community structure gives equal importance to each community irrespective of the number of nodes present in them and the number of outgoing edges to other communities.

3 Proposed Work

This section presents our Community structure with Integrated Feature Ranking (CIFR) algorithm for finding influential nodes in smart enterprises. A community in a business network corresponds to a group of nodes that are closely connected among themselves and sparsely connected to the rest of the network. Uncovering community structure in such networks can be utilized to assess the people's behavior in the group and anticipate their future activities. Due to the closed connection, information can rapidly spread in the community. Node with a prime position and sharing many edges with the other members in its community or module may play a vital role in shaping opinions within the group. The vertices present at the boundaries and connecting communities play a crucial role in information exchanges between different modules. Hence, a node with its connections to the nodes of other communities can act as a bridge or gateway for information propagation, and the role of such nodes becomes crucial for influence maximization. We assume each community has different importance for information propagation. Hence, we assign a ranking to each community based on the number of members and outgoing edges to other communities. We define the local importance of a node in its community as local ranking, bridging role between communities of nodes as gateway ranking, and the ranking of community in which a node is present as community ranking. The adopted definitions are as follows.

Defintion 1: **Local ranking**: We define local ranking or local betweenness centrality of a node i in its community (c) is defined as the extent to which the shortest path between a pair of nodes j and k in the same community (c) through node i, i.e.,

$$LR^c(i) = \sum_{i,j,k \in c \cap j \neq k} \frac{\sigma^i_{j,k}}{\sigma_{j,k}} \tag{2}$$

where $\sigma_{j,k}$ is the total number of shortest path between nodes j and k and $\sigma^i_{j,k}$ is the total number of such shortest path that passes through node i in community c. Local ranking can be used to identify crucial nodes for information propagation, as usually, information is flowing through the shortest paths in the network [27]. The high value of the local ranking of a node corresponds to control over a major fraction of the information flow through that node.

Defintion 2: **Gateway ranking**: We define the gateway ranking of a node i lying in community c is defined as the total number of neighbors, which belongs to other communities. Assume input graph is $G(V, E)$ then gateway ranking can be computed using the following steps:

$$G_2 = (V, E \setminus E'), \tag{3}$$

$$GR(i) = DC(i), i \in G_2(V) \tag{4}$$

where G_2 is a graph obtained from input graph G after removing all edges (E') in the same community. Here, the new graph G_2 contains only those edges which connect nodes of different communities or inter-community edges. The node of this graph is a kind of gateway node which is connected to the nodes of other communities as well. Hence, $GR(i)$ is the gateway ranking of node i that is equal to the degree centrality of node i, i.e., $DC(i)$, in the new arrangement. The high value of gateway ranking of a node signifies its importance in influence maximization as it can share the information among nodes of different communities. Hence, local ranking is used to rank the nodes based on the information propagation capability within their community. However, gateway ranking is used to rank the nodes based on the information propagation capability outside the community. Therefore, local ranking is a kind of local measure, and gateway ranking is a type of global measure to assess the importance of a node for information spraed.

Defintion 3: **Community ranking**: we define community ranking that assigns a score to each node based on the significance of the community to which it belongs. To understand the essence of community ranking, we create a multi-graph (G_3) from the input graph $G(V, E)$ as follows:

$$G_3 = (V_c, E_c) \tag{5}$$

where $G_3 = (V_c, E_c)$ is a multi-graph with V_c is the set of communities, and E_c is the edge between these communities. Basically, each node of the graph, G_3, is a community, and each edge connects nodes of two different communities. Hence, there can be a situation where multiple nodes of the same community (c_i) connect to one or more nodes of other communities (c_j), and G_3 is a multi-graph. Now, community ranking of a node i in the community c is defined as the degree of the community c where c is a vertex in the multi-graph G_3 created as per Eq. (7), i.e.,

$$CR(i) = DC(i), i \in c \text{ and } c \in G_3(V) \tag{6}$$

Hence, the community ranking of a node i lying in community c is equal to the number of outgoing connections to other communities from the vertices of community c. Here, all the nodes belonging to a particular community c gets the same value of community ranking. The high value of community ranking of a node implies that the node belongs to a community with more connections to the members of other communities, and the node can play a vital role for influencing other nodes. Since the size of each community may vary in real-life situations, and a core node chosen from a community having greater size can contribute more influence rather than a core node selected from a small community.

We term local ranking, gateway ranking, and community ranking features of nodes as an integrated features. Therefore by considering three factors, namely, the influence and importance of every node in its community, the significance of the node as per inter-community connections or gateway capabilities, and the relative importance and influence of each community, we propose Community structure with Integrated Features Ranking (CIFR) method for finding influential nodes in smart enterprises. The outline of the proposed algorithm is presented in Algorithm 1.

The detailed steps of the proposed algorithms are as follows:

Algorithm 1 : CIFR

Input: $G = (V, E)$ with $\|V\| = n, \|E\| = m$
Output: $\mathcal{R}$: Ranked nodes
1: $C \leftarrow$ Detecting Communities;
2: **for** each $c \in C$ **do**
3: $LR_i^c \leftarrow$ compute local ranking for each node i in c using Eq. (2)
4: $LR_i^c \leftarrow LR_i^c \times A^c$, where A^c is a normalization parameter computed using Eq. (9)
5: **end for**
6: $l_1 \leftarrow$ LR value of each node as obtained in step 4.
7: $l_1 \leftarrow$ sort the l_1 in descending order of LR values and assign rank 1, 2, and so on.
8: **for** each node $v \in V$ **do**
9: compute $GR(v)$, i.e., gateway ranking score using Eq. (4)
10: **end for**
11: $l_2 \leftarrow$ gateway ranking (GR) of all nodes
12: $l_2 \leftarrow$ sort the l_2 in descending order of GR values and assign rank 1, 2, and so on.
13: **for** each $c \in C$ **do**
14: compute $CR(c)$, and assign community ranking score to each node in c using Eq. (6)
15: **end for**
16: $l_3 \leftarrow$ community ranking (CR) of all nodes.
17: $l_3 \leftarrow$ sort the l_3 in descending order ofcommunity ranking (CR) values and assign rank 1, 2, and so on.
18: **for** each node $v \in V$ **do**
19: compute $CIFR(v)$ score using Eq. (8)
20: **end for**
21: $\mathcal{R} = sort(CIFR \text{ score})$
22: **return** $\mathcal{R}$

(i) Step 1: We use an efficient Lieden Algorithm to find the community in large-scale networks, which improves the classical Louvain algorithm and guarantees that communities are well connected. There exist numerous community detection methods. However, the recently developed Lieden method only visits those nodes whose neighborhood has changed [14]. By relying on a fast local move approach, the Leiden algorithm runs in linear time and is a faster algorithm. We can execute community detection process in parallel using Lieden method enabling overall processing faster. the Leiden algorithm can be used to find communities in the network with overlapping communities.

(ii) Step 2: After finding the communities, we consider the communities are separated from each other. Now, within each community, we calculate the local ranking (LR) of each node using Eq. (2), as per line number 3 of the Algorithm 1. This step can be done in parallel in all communities obtained.

(iii) Step 3: Since the size of each community may vary in real-life situations, and a core node chosen from a community having greater size can contribute more influence rather than a core node selected from a small community. Hence, we need a proper normalization parameter to address the difference in the size of communities. We introduce a normalization parameter, A^c, for each community $c \in C$ as follows:

$$A^c = \frac{n_c}{n} \tag{7}$$

where n_c is the number of nodes in community c and n is the total number of nodes in the network. Within each community, the value of local betweenness centrality of each node is multiplied by the parameter A^c, as per line number 4 of the algorithm. This may normalize the effect of the size of the community with the influence of a node in that community.

(iv) Step 4: This step explains line number 6 and 7 of the proposed algorithm. We make a dictionary L_1 that contains the final local ranking score of each node of the network after doing the normalization, as obtained in the previous step 3. Sort the dictionary L_1 based on values in descending order, and assign the value: 1, 2, 3, and so on, where 1 is the new rank value given to the best node, 2 corresponds to a second-best node and so on according to the measure. Till now, we prioritize the nodes of the network based on its local importance for the influence maximization by taking care of the normalization needed to handle the difference in the size of various communities.

(v) Step 5: In this step, which covers line number 8 to 12 of the proposed Algorithm 1, we want to find the global importance of nodes for information propagation based on their bridging capability. A node with its attachments to the vertices of other communities can serve a bridge for information propagation, and the role of such nodes becomes vital for the information spread in the network. We compute the gateway ranking (GR) of each node of the input graph $G(V, E)$ using Eq. (4). Now, we use the dictionary L_2 to contain the nodes and their gateway ranking score. Here, all non-gateway node has a gateway rank value as 0. Similar to step 4, we sort the dictionary L_2 based on values in descending

order and assign the values: 1, 2, 3, and so on, where 1 is the new rank value given to the best node, 2 corresponds to a second-best node and so on. In this arrangement, the nodes which are not gateway nodes, they each get a score as zero, and they get last or maximum rank. Hence, in this step, we prioritize the nodes of the network based on its bridging role or global importance for the information exchange.

(vi) Step 6: In this step, we rank the nodes based on the number of inter-community connections of the community in which they belong. This step discusses the line number 13 to 17 of the proposed Algorithm 1. We compute the community ranking (CR) of each community c using Eq. (6), and assign the same score to all the nodes in that particular community. We utilize the dictionary L_3 to contain the nodes and their community score.
Similar to steps 4 and 5, we sort the dictionary L_3 based on values in descending order, and assign the values: 1, 2, 3, and so on, where 1 is the new rank value given to the best node according to this measure, 2 corresponds to a second-best node, and the last rank is given to the least important nodes. The nodes present in the same community get the same rank in this arrangement. Here, the notion of L_3 ranking signifies, if a node is present in a higher-ranked community, it should get higher priority by this measure.

(vii) Step 7: In the above three measures, namely L_1, L_2, and L_3, we follow the same strategy to handle the case when there are ties, i.e., ranking of nodes having the same score in a particular measure. For example, if there are four nodes, if three of them have the same score, which is higher than the fourth node, e.g., 5,5,4,2, then we rank them as 1,1,3,4. Here, all the nodes having the same score, they each get the same rank, and the next rank is skipped.
Finally, based on three different measures for prioritizing the nodes namely, L_1, L_2 and L_3, we introduce $CIFR$ score of each node (v) as:

$$CIFR(v) = \frac{1}{L_1(v) + L_2(v) + L_3(v)}, v \in V \quad (8)$$

Here, we assign equal weightage to all three measures, namely, local ranking, global ranking, and community ranking of a node, to compute its CIFR score as per Eq.(8). The reason for the same weightage assignment is that all three measures are equally crucial to identifying influential nodes, and there should be no preference for one measure over others. Eventually, we select top-k nodes in this raking as the source influential nodes.

3.1 Time complexity analysis:

We discuss the time complexity of the proposed algorithm based on the pseudo-code of the Algorithm 1. Assume the number of nodes in the network is n, and the number of edges is m. Community detection using the Lieden method in line number 1 can be performed in parallel and takes $O(m + n)$ time. Line number 2 to 5 for computing the local ranking for each node from each community requires $O(m + n)$ time. Line number 6 takes $O(n)$ for maintaining a dictionary containing the LR value of each node, and line 7 uses $O(n \times logn)$ time for sorting activity and prioritizing the nodes. The line numbers from 8 to 10 for computing gateway ranking of all the nodes demands $O(m + n)$ time. The line numbers 11 and 12 for maintaining a dictionary containing the gateway ranking (GR), and then sorting and assigning ranks to all nodes requires $O(n \times logn)$ time. Similarly, line numbers 13 to 15 for computing community ranking requires $O(n)$ times followed by maintaining a dictionary containing the CR, and then sorting and assigning ranks to all nodes requires $O(nlogn)$ time for line numbers 16 and 17. To compute the $CIFR$ score for each node needs $O(n)$ time in the loop of the lines 18 to 20. Finally, the ranking of nodes based on the $CIFR$ score can be done in $O(n \times logn)$ time.

Hence, the overall complexity of the proposed Algorithm is: $O(m + n + n \times logn)$. Since most of the real-life graph is sparse, we can also consider $O(m)$= $O(n)$. Hence, the time complexity of the proposed algorithm is $O(n \times logn)$.

4 Experimental Results and Analysis

To exhibit the effectiveness of the proposed algorithm, we conducted investigations on seven real-life networks, namely PGP, Hep-Th, Cond-Mat, Gr-Qc, Gnutella08, Gnutella09, Facebook social, and one synthetic network, LFR dataset. These datasets are of diverse size, and application, and are listed in Tab. 2 where $< k >$, $< c >$, and d denotes the average degree of the nodes, average clustering coefficients, and diameter, i.e., longest shortest path of the network respectively. We compared the performance of the proposed Community-structure with Integrated Feature Ranking (CIFR) algorithm with the well-known approaches like local centrality (LC) [17], K-shell (KS) centrality [15], Eigenvector (EV) centrality [18], gateway local rank (GLR) [26] and local information dimensionality (LID) [23]. The Lieden method for performing community detection in the proposed algorithm has been exectuted in parallel.

The performance of all the methods are compared using three evaluation criteria, namely the infection scale ($F(t)$) originating from source nodes, the number of influenced nodes vs. the number of source influential nodes, and Kendall tau (τ) correlation. To compute the CIFR score of each node Eq. No. 8 is used, and top k-nodes are chosen as the seed influential nodes. We employed two popular the information diffusion model, namely, Susceptible-infected-recovered (SIR) [33] and independent cascade (IC) model to calculate scale of influence achieved by selected influential nodes using different considered algorithms. SIR model is a well-studied stochastic epidemic-based information propagation model to investigate the performance of the influence maximization algorithm. SIR model splits all nodes into three categories, namely, Susceptible (S), Infected (I), and Recovered (R). This model takes three inputs: initial spreaders, infection rate (β), and recovery rate (γ). The independent cascade (IC) model is a classical information diffusion paradigm where information spreads in the network through a cascade originating from some seed nodes. Every node can be in one of the two states-

TABLE 2: The basic statistical features of used network datasets

Dataset	Type of Network	#nodes	#edges	$< k >$	$< c >$	d
PGP [28]	Trust network	10638	24301	4.57	0.023	10
Hep-Th [29]	Collaboration network	9877	25998	5.2	0.47	17
Cond-Mat [29]	Collaboration network	23133	93439	8.08	0.63	14
Gr-Qc [29]	Collaboration network	5242	14496	5.52	0.529	17
Gnutella08 [30]	File-sharing network	6301	20777	6.59	0.01	9
Gnutella09 [30]	File-sharing network	8114	26013	6.40	0.009	10
Facebook Social [31]	Social network	4039	88234	43.7	0.60	8
LFR-0.2 [32]	Synthetic network	10000	64936	12.987	0.130	16

active or inactive. The active nodes refer to those nodes that got the information, whereas inactive nodes have not yet received the information. The spread of infection scale and the final number of influenced nodes at the end of the spreading process depend on the number of seed nodes and the infection rate (β) in the SIR model and probability P_{uv} in IC model. As the selection of susceptible neighbors for the infection in the SIR and IC model is random, we run the information diffusion model 200 times, and results are averaged over. The experiments have been performed on a personal computer with primary memory 8 GB, and 1.6 GHz Intel Core i5 processor. We performed the simulations of our experiments using Python programming language using various packages like Networkx, Scikit-Learn, matplotlib, panda, etc. The details of all four evaluations are as follows.

4.1 Infection Scale:

The infection scale ($F(t)$) at any time t is the sum of the number of infected nodes ($n_{I(t)}$) at time t, and the number of recovered nodes ($n_{R(t)}$) till time t. The following equation computes the Infection scale, $F(t)$:

$$F(t) = n_{I(t)} + n_{R(t)} \tag{9}$$

To estimate the infection scale ($F(t)$) in terms of the increasing timestamp of the information diffusion process, we run the SIR model using the initial number of influential spreaders, and infection rate (β). We considered different numbers of influential spreaders as the seed nodes for different datasets based on their size. Here, influential spreaders are such nodes that are identified by the various considered algorithms. We considered the top 2%, and 5% nodes as the source spreader nodes depending upon the size of the network. For a relatively large dataset having $10,000$ nodes or more like Cond-Mat, PGP, and LFR top 2% nodes are used as spreader nodes. However, for the remaining datasets, the top 5% nodes are taken as source nodes. The value of the epidemic threshold, i.e., β_{th} are different for different datasets, as mentioned in Tab. 2. To maintain consistency in our evaluation process, we use the same value of β as 0.08 for all the datasets, i.e., when a node is in the infected state, then it can influence 8% of its neighbors randomly. Fig. 1 depicts the infection scale ($F(t)$) with respect to the time for different datasets considered. It is noticeable from the result that in most of the datasets, the proposed algorithm (CIFR), outperforms all approaches, namely, local centrality, K-shell, h-index, Eigenvector, GLR, and LID. LID method almost remains in the second position after CIFR. In the case of Facebook and LFR datasets, which

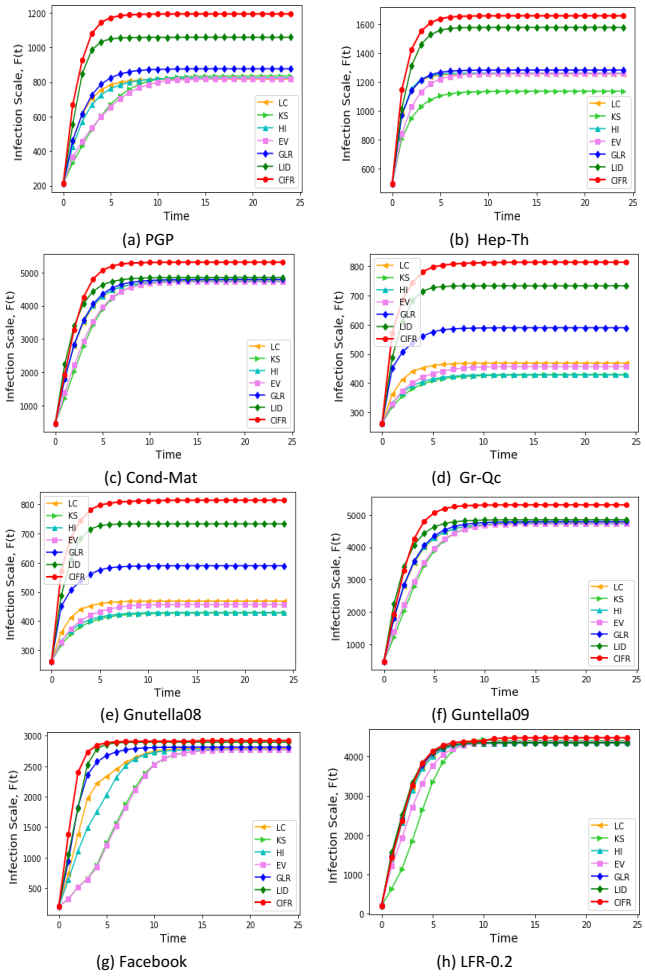

Fig. 1: (a)-(h): Infection scale ($F(t)$) Vs. time using the same rate of infection (β) $= 0.08$ for all networks. The number of initial spreaders is taken as 2% for PGP, Cond-Mat, LFR-0.2 datasets, and 5% for Hep-Th, Gr-Qc, Facebook, Gnutella08, and Guntella09 datasets. The results are obtained over 200 independent simulations of the SIR model

are relatively dense datasets, the performances of all methods very similar. However, CIFR remains at the top. The

obtained result depicts the superiority of the seed selection by our algorithm.

4.2 Influence spread Vs. number of seed influential nodes:

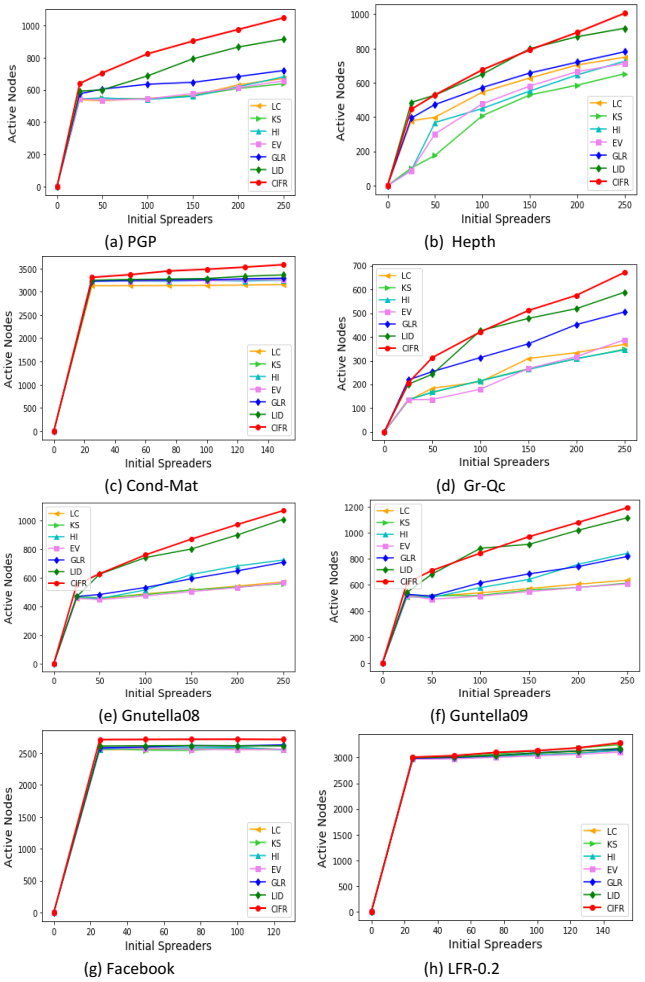

Fig. 2: (a)-(h): Number of influenced nodes at the end of information diffusion process with respect to different number of source influential nodes using the SIR model with infection rate (β) $= 0.08$ for all the datsets. The results are obtained from 200 independent simulations of SIR model

.

The influence spread refers to the extent or fraction of the network to which information originating from the set of seed nodes has finally spread at the end of the information diffusion process. To compute the number of influence spread at the end of the spreading process vs. number of seed influential nodes, we employed both the information spreading model SIR and IC for each dataset. We considered the different number of source nodes for various networks. For the datasets having more than $50,000$ edges like Cond-Mat, Facebook, and LFR-0.2, the number of source nodes were taken in the range of spreaders are taken in the range of 20, 40, 60, 80, 100, and 120. For remaining datasets with less then than $50,000$ edges ,i.e., PGP, Hep-Th, Gr-Qc, Gnutella08,and Gnutella09, we take the number of spreaders as 50, 100, 150,200, and 250. This variation in the considered source spreader nodes to plot influence spread for all the methods is because of the difference in the size of networks and to obtain the results in relatively less time.

Fig. 2 presents the effect of change in the number of initial spreaders, on the number of total influenced nodes using the SIR model with infection rate (β) as 0.08 for all the datasets. The received results are averaged over 100 simulations of the SIR model. For all datasets, we keep the same value of β in the SIR model, to maintain consistency in the evaluation. From the results, it is clear that the proposed algorithm performs better than all other methods. From the results, it is evident that the proposed algorithm performs better than all other methods in the case of PGP, Hep-Th, Gr-Qc, Cond-Mat, Gnutella08, and Gnutella09 datasets. However, in the case of relatively dense datasets like Facebook and LFR-0.2 datasets, the performances of various methods are very close to each other, but CIFR manages to remains at the top. Similarly, Fig. 3 displays the effect of change in the number of seed influential nodes, on overall influence spread using the IC model with $p_{uv} = 0.12$. The received results are averaged over 100 simulations IC model. For all datasets, we keep the same value of p_{uv} in the IC model to maintain consistency in the evaluation. The results show that the proposed algorithm performs better than all other methods on the datasets PGP, Hep-Th, and Gr-Qc. However, in the case Cond-Mat, Gnutella08, Gnutella09, and LFR-0.2 datasets, the performances of CIFR and LID methods are very close to each other with CIFR is performing slightly better then LID and beating remaining all other techniques. From both the Fig. 2 and 3, it is visible that the proposed algorithm performed consistently well in all datasets. Hence, we can infer that the selection of influential nodes using the proposed CIFR algorithm maintains its performance with an increase in the number of source influential nodes.

4.3 Kendall Tau (τ) Matrix:

We used Kendall's tau coefficient (τ) [34] to assess the correctness of the ranking methods. This evaluation metric examines the correlation between the two rank lists, namely X and Y. Here, X is the value of the centrality score of nodes under various methods, and Y is the actual value of influence spread for all nodes. If node v_i appears before node v_j in list X and Y both, then such pair are called concordant pair (n_c), otherwise discordant pairs (n_d). When $n_c > n_d$, the coefficient has a positive value, indicating similarity. Kendall's tau coefficient value ranges between $(-1, 1)$. The following formula represents Kendall's tau correlation values.

$$\tau = \frac{n_c - n_d}{\frac{n(n-1)}{2}} \tag{10}$$

where n is the total number of vertices in each list. We computed Kendall's tau (τ) correlation of the ranked nodes iden-

TABLE 3: Kendall's Tau (τ) corelation values of different real-life network using SIR information diffusion model and infection rate (β) $= 0.1$

Dataset	$\tau(LC,\sigma)$	$\tau(KS,\sigma)$	$\tau(HI,\sigma)$	$\tau(EV,\sigma)$	$\tau(LID,\sigma)$	$\tau(GLR,\sigma)$	$\tau(CIFR,\sigma)$
PGP	0.059	0.014	0.035	0.050	0.031	0.014	**0.138**
Hep-Th	0.318	0.142	0.029	0.045	0.167	0.232	**0.376**
CondMat	0.136	0.149	0.185	0.258	0.444	0.553	**0.581**
Gr-Qc	0.294	342	0.348	0.520	**0.591**	0.197	0.455
Gnutella08	0.120	0.334	0.502	0.500	0.363	0.305	**0.511**
Gnutella09	0.492	0.241	0.40	0.552	0.363	0.541	**0.569**
Facebook	**0.127**	0.052	0.038	0.021	0.009	0.023	0.033
LFR-0.2	0.160	0.203	0.020	0.285	0.340	**0.514**	0.453

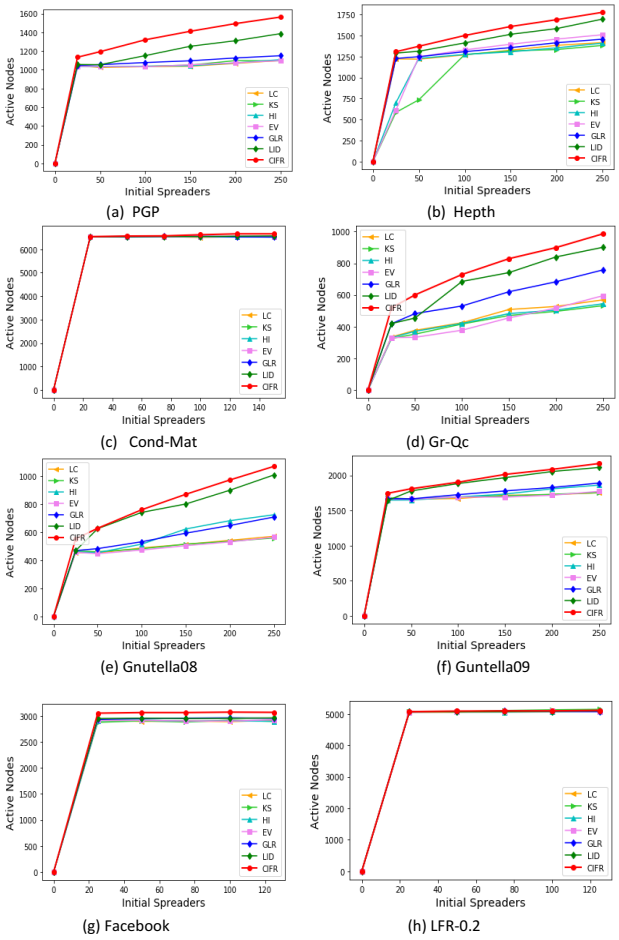

Fig. 3: (a)-(h): Number of influenced nodes at the end of information diffusion process with respect to different number of seed influential nodes using the IC model with $P_{uv} = 0.12$ for all the datasets. The results are obtained from 200 independent simulations of IC model

tified by different methods, namely local centrality (LC), K-shell, h-index, Eigenvector, gateway local rank (GLR), local information dimensionality (LID) and proposed algorithm (CIFR). For this calculation, we applied the SIR spreading model. To maintain consistency in the various performance measures, the value of the infection rate (β) is taken as 0.01 for all the networks. Tab. 3 presents the values of τ obtained using all the algorithms. On the PGP, Hep-Th, Cond-Mat, Gnutella08, and Gnutella09 datasets, proposed algorithm produces the best τ values among all the approaches. However, local centrality produced the best result on Facebook, LID attains the best reslut on Gr-Qc and GLR obtains the best value on the LFR-0.2 dataset. Hence, on a majority of the networks, the proposed CIFR method produced best result.

5 Conclusion

Influential nodes are the nodes having high information spreading ability, and influence maximization can be achieved through these nodes in business networks of smart enterprises. In this paper, we proposed the Community structure with the Integrated Features Ranking (CIFR) algorithm for detecting influential nodes by utilizing the feature of community spread in the network. Initially, we divided the network into various communities for each community. We ranked all the nodes according to their local and gateway influences. Now, considering those communities as building blocks of the whole network, we rank all those communities. Hence, the importance of a node is weighed based on its influence in its community and community-specific global indicators like inter-connection, bridging roles, and relative importance of communities. Our algorithm utilizes the community spread selects top spreaders, which are both globally and locally optimal, leading to a high information propagation in their community and other connected communities as well. Our method, CIFR, is applicable in large-scale business networks that is for large as well as small-scale enterprises to locate the most desirable seed nodes with the high spreading capability, and it has more economical time complexity. We performed the experimental results on eight networks using various evaluation parameters. The proposed algorithm performed better against contemporary algorithms based on evaluation criteria like infection scale, influence spread, and Kendall tau correlation. The obtained results suggest the efficacy of the work and hence add value to smart enterprises. The proposed work may give rise to relatively inadequate performance if the underlying business network has unclear community structure. The future extension of the current

work can accommodate more features independent of the network and community structure to identify influential nodes in the multi-layer business network.

REFERENCES

[1] Y. C. H. Z. Zhang, Yongping and F. Tao, "Long/short-term preference based dynamic pricing and manufacturing service collaboration optimization," *IEEE Transactions on Industrial Informatics*, 2022.

[2] A. Kumar and A. Jaiswal, "A deep swarm-optimized model for leveraging industrial data analytics in cognitive manufacturing," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2938–2946, 2020.

[3] J. Tang, X. Tang, and J. Yuan, "Profit maximization for viral marketing in online social networks: Algorithms and analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 6, pp. 1095–1108, 2017.

[4] Y. Li, J. Fan, Y. Wang, and K.-L. Tan, "Influence maximization on social graphs: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 10, pp. 1852–1872, 2018.

[5] W. Chen, C. Wang, and Y. Wang, "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*, 2010, pp. 1029–1038.

[6] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003, pp. 137–146.

[7] P. Basaras, G. Iosifidis, D. Katsaros, and L. Tassiulas, "Identifying influential spreaders in complex multilayer networks: A centrality perspective," *IEEE Transactions on Network Science and Engineering*, vol. 6, no. 1, pp. 31–45, 2017.

[8] G. Song, Y. Li, X. Chen, X. He, and J. Tang, "Influential node tracking on dynamic social network: An interchange greedy approach," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 2, pp. 359–372, 2016.

[9] Y.-C. Chen, W.-Y. Zhu, W.-C. Peng, W.-C. Lee, and S.-Y. Lee, "Cim: Community-based influence maximization in social networks," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 5, no. 2, pp. 1–31, 2014.

[10] F. Cai, L. Qiu, X. Kuai, and H. Zhao, "Cbim-rsrw: An community-based method for influence maximization in social network," *IEEE Access*, vol. 7, pp. 152 115–152 125, 2019.

[11] S. Kumar, A. Mallik, A. Khetarpal, and B. Panda, "Influence maximization in social networks using graph embedding and graph neural network," *Information Sciences*, 2022.

[12] W. Luo, D. Zhang, L. Ni, and N. Lu, "Multiscale local community detection in social networks," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[13] Z. Liu and B. Hu, "Epidemic spreading in community networks," *EPL (Europhysics Letters)*, vol. 72, no. 2, p. 315, 2005.

[14] V. A. Traag, L. Waltman, and N. J. Van Eck, "From louvain to leiden: guaranteeing well-connected communities," *Scientific reports*, vol. 9, no. 1, pp. 1–12, 2019.

[15] M. Kitsak, L. K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. E. Stanley, and H. A. Makse, "Identification of influential spreaders in complex networks," *Nature physics*, vol. 6, no. 11, pp. 888–893, 2010.

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

[17] D. Chen, L. Lü, M.-S. Shang, Y.-C. Zhang, and T. Zhou, "Identifying influential nodes in complex networks," *Physica a: Statistical mechanics and its applications*, vol. 391, no. 4, pp. 1777–1787, 2012.

[18] A. N. Langville and C. D. Meyer, "A survey of eigenvector methods for web information retrieval," *SIAM review*, vol. 47, no. 1, pp. 135–161, 2005.

[19] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Tech. Rep., 1999.

[20] J. E. Hirsch, "An index to quantify an individual's scientific research output," *Proceedings of the National academy of Sciences*, vol. 102, no. 46, pp. 16 569–16 572, 2005.

[21] K. Berahmand, A. Bouyer, and N. Samadi, "A new local and multidimensional ranking measure to detect spreaders in social networks," *Computing*, vol. 101, no. 11, pp. 1711–1733, 2019.

[22] X. Rui, F. Meng, Z. Wang, and G. Yuan, "A reversed node ranking approach for influence maximization in social networks," *Applied Intelligence*, vol. 49, no. 7, pp. 2684–2698, 2019.

[23] T. Wen and Y. Deng, "Identification of influencers in complex networks by local information dimensionality," *Information Sciences*, vol. 512, pp. 549–562, 2020.

[24] H. Huang, H. Shen, Z. Meng, H. Chang, and H. He, "Community-based influence maximization for viral marketing," *Applied Intelligence*, vol. 49, no. 6, pp. 2137–2150, 2019.

[25] S. Kumar, A. Gupta, and I. Khatri, "Csr: A community based spreaders ranking algorithm for influence maximization in social networks," *World Wide Web*, pp. 1–20, 2022.

[26] C. Salavati, A. Abdollahpouri, and Z. Manbari, "Ranking nodes in complex networks based on local structure and improving closeness centrality," *Neurocomputing*, vol. 336, pp. 36–45, 2019.

[27] J. S. Mitchell, "Shortest paths and networks," in *Handbook of discrete and computational geometry*. Chapman and Hall/CRC, 2017, pp. 811–848.

[28] M. A. Serrano, M. Boguná, R. Pastor-Satorras, and A. Vespignani, "Correlations in complex networks," *Large scale structure and dynamics of complex networks: From information technology to finance and natural sciences*, pp. 35–66, 2007.

[29] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, pp. 2–es, 2007.

[30] M. Ripeanu and I. Foster, "Mapping the gnutella network: Macroscopic properties of large-scale peer-to-peer systems," in *international workshop on peer-to-peer systems*. Springer, 2002, pp. 85–93.

[31] J. Leskovec and J. J. Mcauley, "Learning to discover social circles in ego networks," in *Advances in neural information processing systems*, 2012, pp. 539–547.

[32] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical review E*, vol. 78, no. 4, p. 046110, 2008.

[33] C. Xia, Z. Liu, Z. Chen, S. Sun, and Z. Yuan, "Epidemic spreading behavior in local-world evolving networks," *Progress in Natural Science*, vol. 18, no. 6, pp. 763–768, 2008.

[34] M. G. Kendall, "A new measure of rank correlation," *Biometrika*, vol. 30, no. 1/2, pp. 81–93, 1938.

Sanjay Kumar is currently an Assistant Professor in the Department of Computer Science and Engineering, Delhi Technological University, New Delhi, India. He did Ph.D. degree in Computer Applications, Indian Institute of Technology (IIT) Delhi. He has completed M.Tech in Computer Application from Indian Institute of Technology (IIT) Delhi, India. Previously, he has worked with National Informatics Centre, Govt. of India as Scientist-B. His research interests include AI, Social Network Analysis, Graph Algorithm, Machine Learning, and Design and Analysis of Algorithms.

Akshi Kumar, SMIEEE, is a Post-doc from Federal Institute of Education, Science and Technology of Ceará, Fortaleza, Brazil and PhD from Faculty of Technology, University of Delhi, New Delhi, India. She is currently a Senior Lecturer-Data Science in the Department of Computing & Mathematics, Faculty of Science & Engineering, Manchester Metropolitan University, United Kingdom. She has worked as an Associate Professor at the Department of Information Technology, Netaji Subhas University of Technology (NSUT), New Delhi, India and as an Assistant Professor at the Delhi Technological University (DTU), formerly, Delhi College of Engineering, New Delhi, India.

B. S. Panda is currently a Professor in Computer Science and Application group of the Department of Mathematics, Indian Institute of Technology (IIT) Delhi, India. He did his Ph.D. in the Department of Mathematics, IIT Kanpur in 1994 in Algorithmic Graph Theory. He has served as an Assistant Professor in the Department of Computer and Information Sciences, University of Hyderabad. He was also in the Department of Computer Science and Engineering, University of Texas at Arlington, USA as a visiting researcher. His research interests are Graph Algorithms, Social Network Analysis, and Design and Analysis of Algorithms.