Applied Intelligence (2022) 52:5173–5184
https://doi.org/10.1007/s10489-021-02656-0

CIM: clique-based heuristic for finding influential nodes in multilayer networks

Venkatakrishna Rao. K[1] · Mahender Katukuri[1] · Maheswari Jagarapu[2]

Accepted: 14 June 2021 / Published online: 5 August 2021

Abstract

Identifying Influential nodes (Influence maximization) in complex networks is an essential factor for spreading and controlling the information spreading dynamics in social networks. The majority of the influence maximization problems are in monolayer networks. After advancements and increased social network usage, the need to perform influence maximization in multilayer networks has increased. The critical issue is to identify the influential nodes that can effectively spread the information across the networks. Detecting such influential nodes with high precision in multilayer networks is a challenging and yet unexplored task. Based on our experiments, it is observed that a potential node may have strong connections in both the interlayers and intralayers. A comparative study of various influence maximization algorithms in multilayer networks is carried out with this observation. We propose a novel algorithm, clique-based influence maximization (CIM) in a multilayer network. We also propose ignoring noted nodes in the network to increase the efficiency of influence maximization and remove the information redundancy. CIM is generating better results for influence spread in multilayer networks compared to the other algorithms. The simulation studies have shown that CIM can detect influential nodes on both real and synthetic networks under various environments.

Keywords Social networks · Multilayer networks · Influenced nodes · Cliques · Degree

1 Introduction

With the tremendous popularity of the internet, many social networks like Facebook, Twitter, and Linked-in have emerged. They have become standard communication tools for billions of users across the world. With the rapidly increasing trend of social networks and the participation of many users, they have become good and economical marketing platforms. In this context, finding the set of nodes, which maximizes the information spread across the networks, is an interesting research problem and is termed as influence maximization (IM) [2].

Pedro and Mat [13] have introduced the influence maximization problem in a graph; the core issue here is to spread the information from one node to another. Authors modeled the problem using Markov Random Fields and proposed heuristic solutions. D. Kempe et al. [2] studies influence maximization as an optimization problem and reported it as an NP-hard problem. They have proposed a greedy algorithm for influence maximization under classical diffusion models such as Independent Cascade (IC) and Linear Threshold (LT) models. In order to improve the efficiency of the greedy algorithm, Leskovec et al. [14] have proposed CELF algorithm, which utilizes the submodular property of diffusion function to reduce the number of Monte Carlo simulations. CELF is 700 times faster than the greedy algorithm. Inspired by the idea of CELF, Goyal et al. [15] have introduced an algorithm CELF++ which is 55% faster than CELF. Chen et al. [4] have introduced an algorithm based on degree discount. The work is based on the observation that once a node is selected as a seed node, that node is no longer available for seed selection. Therefore, a degree discount step is required for all the direct neighbors of that node. Algorithms to address the influence maximization problem can be classified as greedy-based [30, 31], heuristic-based [16, 17, 25, 27], meta-heuristics based [23, 24, 26], and sampling-based algorithms [28, 29].

✉ Venkatakrishna Rao. K
vrkatakamsetty.rs.cse18@itbhu.ac.in

1 Department of Computer Science and Engineering, Indian Institute of Technology (BHU), Varanasi, India

2 Department of Computer Science and Engineering, Jawaharlal Nehru Technological University, Kakinada, India

The majority of the existing studies focus on influence maximization in single-layer networks and ignore some critical factors such as user engagement across the networks, the network of networks, etc. Therefore, we need to consider multilayer networks, as many users in the networks are actively engaged. Due to this, they propagate information, ideas, and innovations across the networks simultaneously. Henceforth, we need to account for the individual's influence across the networks instead of a single network to estimate an individual's overall influence accurately. To incorporate multiple networks in influence estimation, we considered a multilayer network framework. In this framework, each of the networks is considered as a layer. The information spreading dynamics in multilayer network is complex compared to a single layer network. Finding out influence maximization in such multilayer networks is a relatively less explored research problem.

The influence maximization problem is not fully explored in multilayer networks. There are some efforts [9, 18–20] done in this direction. Basaras et al. [18] used the power community index (PCI) for finding the most influential nodes in multilayer networks. Degree heuristics considers only the degree of the node, but PCI intends to find the node with a dense neighborhood. Wang et al. [19] have proposed multilayer collective influence (MCI) to identify influential nodes, which utilizes topological and dynamic properties instead of the local degree of the node. Wang et al. [20] have proposed an algorithm, essential nodes determining based on CP tensor decomposition (EDCPTD), centrality to find the most influential seed nodes. They applied CANDECOMP/PARAFAC(CP) tensor decomposition to get some significant factors such as principle singular vectors and produced quadruplets vector to obtain hub and authority scores of each node across the layers. The seed selection in the greedy algorithm is not feasible for large networks due to time-consuming MC simulations. Therefore, Kuhnle et al. [9] have proposed a two-phase approximation algorithm called knapsack seeding of networks(KSN). The first phase runs parallelly in each layer and stores the list of activated nodes per seed node. In the second phase selects seed nodes based on the multiple-choice knapsack (MCK) problem.

This study addresses the influence maximization problem in multiple social networks using a multilayer framework by utilizing clique structure. Also, it proposes a seed selection algorithm, clique-based influence maximization (CIM). The contributions of the proposed work are:

- We adapted the maximal clique selection algorithm [6] and extended it to a multilayer framework.
- We propose a seed selection algorithm CIM by considering maximal clique structure for percolated and non percolated cliques.
- We propose to ignore the *noted nodes* for influence maximization to reduce the redundancy and reducing the spreading time without losing influence maximization.
- We compare the proposed algorithm's performance against the state-of-the-art algorithms under the IC diffusion model on real-world social networks.

The rest of the paper is organized as follows. In Section 2, discusses the multilayer framework with the influence maximization problem. In Section 3, we present the proposed algorithm along with the complexity analysis. In Section 4, we discuss the experimental setup and evaluation metrics. Results are discussed in Section 5. Finally, Section 6 concludes our work.

2 Preliminaries

2.1 Basic model of multilayer networks

A multilayer network model is intra and inter connected graph network $M = (V^M, E^M, V, L)$, where $V^M \subseteq V \times L^1 \times L^2 \times .. \times L^m$ is the set of node layer combinations, $E^M \subseteq V^M \times V^M$ is the edge set in multilayer network containing the possible pair of combinations and elementary layers, V is set of nodes in multilayer network and $L = \{L^\alpha\}_{\alpha=1}^m, \alpha \geq 2$. Nodes need not be homogeneous and a node u can exist in many layers [8]. A node layer tuple u^l indicates a node u in layer l, $V^i \in V^M = [v_1^i, v_2^i, ...v_n^i]$ and $V = [V^1, V^2...V^m]$, where $\mid V \mid= m * n$. Different notations have been used in this paper, Table 1 presents the notations used in this paper.

The information propagation dynamics of a multilayer network are different from a single-layer network. To demonstrate the information spreading in a multilayer network, let us consider a simple multilayer network in

Table 1 Notations used in this paper

Notation	Description
V, E	Nodes, edges in a graph
K	Number of seed nodes
p	Propagation probability in IC model
$\sigma(S)$	Influence Spread achieved by seed set S
d_v^l	Degree of v in l^{th} layer
dd_v^l	Degree discount of v in l^{th} layer
$nbrs(v^l)$	Neighbours of node v in l^{th} layer
$actnbrs(v^l)$	Active neighbours of node v in l^{th} layer
t_v^l	Number of neighbours of node v in layer l that are already selected as seed node
u^l	Node u in layer l

Fig. 1. The network consists of two layers, $L1$, and $L2$; each layer consists of six nodes, A, B, C, D, E, and F. Assume that initially, node C in $L1$ and E in $L2$ are the most influential nodes, so select them as seed nodes. The information spread in the network can be analyzed iteratively. In the first iteration, the seed nodes activate their neighbors using one of the two propagation models, i.e., Independent Cascade model and Linear Threshold model. In the subsequent iterations, the activated nodes will activate their neighbors, and the process continues until no node is left to participate. After applying propagation model to Fig. 1A at time step 1, three extra nodes are activated i.e., A in $L1$ and B, D in $L2$. Heuristics to find the seed nodes in the influence maximization of multilayer networks are discussed in the subsequent section.

2.2 Influence maximization

Multilayer influence maximization refers to the process of finding the set of most influential nodes from all the layers, called seed nodes. The expected influence spread according to any one of the propagation models is maximized in multilayer networks by activating them. The propagation model plays a crucial role in the analysis of influence maximization. Two classical propagation models generally used are independent cascade (IC) and linear threshold (LT) models. In both these models, a node is assumed to be either in an activated state or an inactive state [9].

In the independent cascade model, when a node becomes active at time step t, it will get one chance to activate its inactive neighbors at time step $t + 1$ with propagation probability p. In the Linear threshold model, every node is associated with an activation threshold between 0 and 1. At any time step t, if the sum of the incoming edge weights is greater than the activation threshold of the node, then the node enters into an activated state. The sum of all the incoming edge weights of any node is assumed to be at most 1.

A node v^l influenced by each neighbour w^l according to b_{w^l,v^l} such that

$$\sum_{w^l \in nbrs(v^l)} b_{w^l,v^l} \leq 1$$

A node v^l becomes active when at least (weighted) θ_{v^l} fraction of its neighbors are active, i.e.,

$$\sum_{w^l \in actnbrs(v^l)} b_{w^l,v^l} \geq \theta_{v^l}$$

In both the models, influence propagation happens until no new user is turned active.

Let χ be a collection of coin flips on edges and $R(v^l, \chi)$ be the set of all nodes which can be reached from node v^l on a path consists of totally live edges, we can estimate approximate influence spread $\sigma(S)$ of seed set S [7].

$$\sigma(S) = \cup_{v^l \in S} R(v^l, \chi)$$

where, $\sigma(S)$ is the expected number of nodes influenced by seed set S after applying one of the propagation models.

Definition 1 (Monotone) f is monotone, if $\sigma(A) \leq \sigma(B)$ whenever $A \subseteq B$. i.e. $\sigma(S \cup \{v^l\}) \geq \sigma(S)$. Influence spread is more in super set when compare to subset,

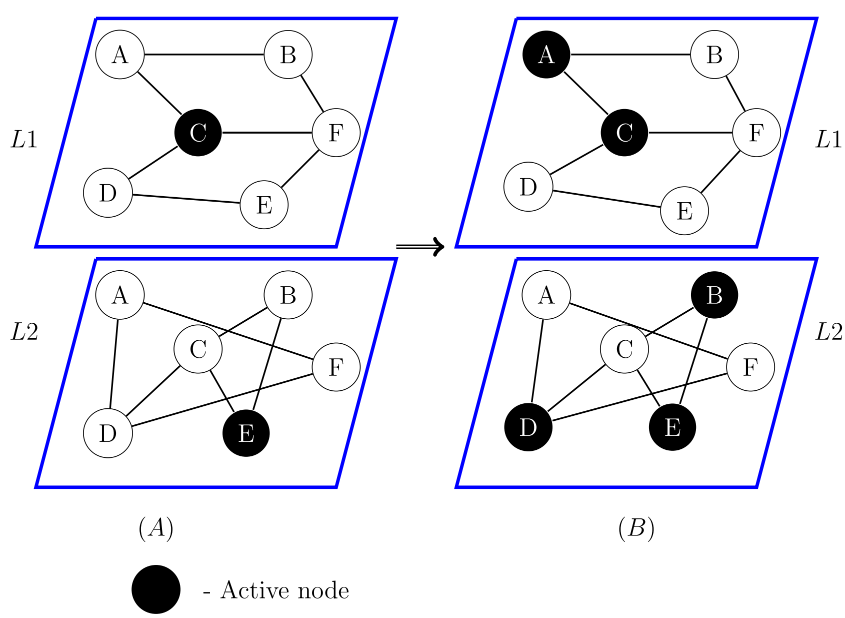


Fig. 1 (A) Dark nodes are active nodes in Layer 1 and 2 before influence maximization (B) Dark nodes are active nodes in layer 1 and 2 after influence maximization

Definition 2 (Submodularity) The function $\sigma : 2^V \rightarrow \Re$ is submodular if for all $X, Y \subset V$

$$\sigma(X) + \sigma(Y) \geq \sigma(X \cup Y) + \sigma(X \cap Y) \quad (1)$$

If all the elements are submodular functions, then we say that a collection σ of functions from $2^V \, to \, \Re$ is submodular [12].

Monotonicity is where the effect of a larger set L on node u is stronger than the effect of a smaller set K. Submodularity is the condition where the effect (diminishing returns) of adding a node u to a smaller set K is more than the effect of adding the same node u to a larger set $L, (K \subset L)$ [3].

Lemma 1 A function $\sigma : 2^V - > \Re$ is submodular Iff

$$\sigma(A \cup \{v^l\}) - \sigma(A) \geq \sigma(B \cup \{v^l\}) - \sigma(B) \quad (2)$$

for all $A \subset B \subset B \cup \{v^l\} \subset V$, and $v^l \in V \setminus B$.

Proof From definition 2, assume $X = A \cup \{v^l\}$, $Y = B$ and substitute in (1)

$\sigma(A \cup \{v^l\}) + \sigma(B) \geq \sigma(A \cup \{v^l\} \cup B) + \sigma(A \cup \{v^l\} \cap B)$
$\sigma(A \cup \{v^l\}) + \sigma(B) \geq \sigma(B \cup \{v^l\}) + \sigma(A)$
$\sigma(A \cup \{v^l\}) - \sigma(A) \geq \sigma(B \cup \{v^l\}) - \sigma(B)$

Hence, the proof that (2) satisfies submodularity property. For the IC and LT models, σ is monotone and submodular [11]. □

3 Proposed model

In this section, we discuss the proposed algorithm. We explain the procedure of identifying all maximal cliques along with the time complexity analysis. We present the seed selection process using the clique-based topological feature. Finally, we present the proposed CIM algorithm.

3.1 Identification of all maximal cliques

Moser et al. [5] proved that a graph with n vertices could have at most $3^{n/3}$ maximal cliques. Hence, a multilayer network with m layers and in each layer with n nodes can have at most $3^{(m \times n)/3}$ maximal cliques. Any algorithm which operates by finding all the maximal cliques of a multilayer network can explore $3^{(m \times n)/3}$ maximal cliques out of the network. Kerbosch et al. [10] proposed an algorithm to find all the existing maximal cliques in a single layer network. The algorithm works recursively for all the cliques for the generation of maximal cliques in the graph. The algorithm performs recursion for all the cliques, whether it is a maximal clique or not; due to that, the time complexity of the algorithm is slightly higher. Tomita et al. [6] have introduced a variant for the above algorithm, which finds all the existing maximal cliques. The algorithm backtracks in the graph's branch search if it has no maximal clique so that time is reduced. Both of the algorithms generate all the existing maximal cliques. Algorithms use depth-first search (DFS) to generate all the maximal cliques in the given graph.

To identify all the maximal cliques, we adapt the algorithm proposed by [6] for a single layer network to multilayer networks. This algorithm is used to generate all the maximal cliques of a given multilayer network graph. The algorithm begins with an empty set and expands the set in stages recursively till all the maximal cliques are identified.

Algorithm 1 CIM: clique-based influence maximization algorithm.

Initialize $Min, Upto$
/* $NOMC$- number of generated maximal cliques; K- number of Required Seed nodes, SOC- Size of Clique*/
$Min = K/NOMC$; $Upto$= $K \; mod \; NOMC$
Case 1: If $NOMC \geq K$ **then**:
  Sort all the cliques in descending order based on size
  Select highest degree node from each clique, Starting from largest size of the clique.
  This set of K nodes selected from first K cliques are seed set.
Case 2: If $NOMC \leq K$ **then**:
  Step 1: Select $Min(= K/NOMC)$ number of Highest Degree nodes from each clique
  Step 2: Select $Min + 1$ from the first $K \; mod \; NOMC$ cliques.
Case 3: If *any clique size* $\leq Min$ **then**:
  Select $Size\,of\,Clique\,(SOC)/2$ highest degree nodes from the clique
  Remaining seed nodes will be selected based on satisfaction of case 1 or case 2.
Case 4: In percolated cliques,
  If the same node with highest degree is in different cliques then select that node from largest clique.
  Remaining seed nodes will be selected based on satisfaction of case 1 or case 2.

3.2 Identification of seed users

This section presents how to identify seed nodes after finding all the existing maximal cliques in a multilayer network. We select seed nodes from the cliques based on the number of generated maximal cliques (NOMC). Since a node in a clique is connected to all the other nodes, its influence is more. The following cases explain the selection of seed nodes from the generated maximal cliques in a multilayer network. Algorithm 1 presents the procedure of seed nodes selection.

$$Min = K/NOMC$$

$$Upto = K \mod NOMC$$

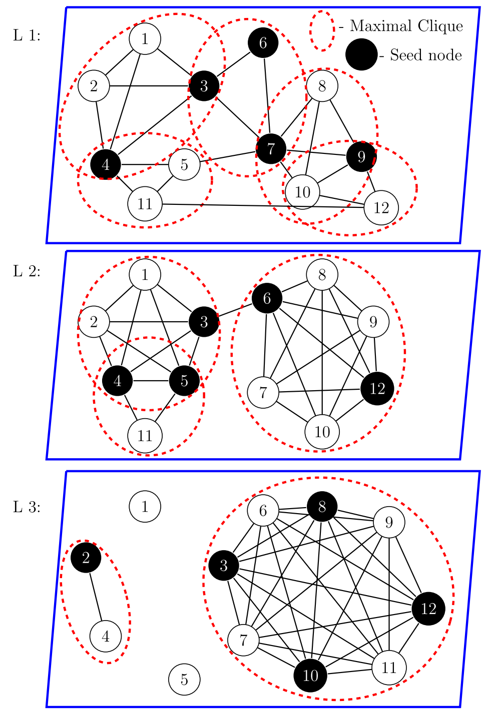
Case 1: If generated $NOMC$ are more than or equal to the required number of seed nodes, sort the cliques in descending order based on their size. Select the highest degree node (yet to be selected) from each clique, starting from the largest clique. This set of K nodes selected from the first k cliques will be the seed set.

Case 2: If $NOMC$ are less than the required number of seed nodes K, then seed nodes will be selected in two steps. In the first step, select the Min (=$K/NOMC$) number of highest degree nodes (yet to be selected) from each clique. In the second step, select the nodes (i.e., Min+1 node, yet to be selected) from the first K mod $NOMC$ cliques.

Case 3: While selecting the seed nodes from each clique, If the size of any clique is less than or equal to Min, then select $SOC/2$ highest degree nodes (yet to be selected) from the clique, and remaining seed nodes will be selected based on satisfaction of case 1 or case 2.

Case 4: While finding all the maximal cliques, there may be percolated cliques. In percolated cliques, the same node with the highest degree may be present in different cliques. Select the highest degree node from the larger clique, and remaining seed nodes will be selected based on the satisfaction of case 1 or case 2.

Figure 2 Illustrates the selection of seed nodes in a multilayer network. We took the number of seed nodes as five. The network consists of three layers; layer one generated C1, C2, C3, C4, and C5 cliques, satisfying case 1. Therefore, node 3 from C1, node 7 from C2, node 6 from C3, node 4 from C4, and node 9 from C5 are selected as seed nodes. Layer two generated C6, C7, and C8 cliques, and it satisfies case 2, so the seed node selection is made in two steps; in the first step, select the highest degree node (i.e., Min=1) from each clique, i.e., 6 from C6, 4 from C7, and 5 from C8, In the second step select the next highest degree node starting from the largest clique, i.e., 12 from C6 and 3 from C7. Layer three generated C9 and C10, size of C10 is two and Min=2, so, select $SOC/2$ highest degree nodes from C10, i.e., 2 or 4 from C10, and the remaining four seed nodes are selected from C9, i.e., 3, 8, 10, and 12.



$L1 = \{C1 = \{1,2,3,4\}, C2 = \{7,8,9,10\}, C3 = \{3,6,7\}, C4 = \{4,5,11\}, C5 = \{9,10,12\}$
$L2 = \{C6 = \{6,7,8,9,10,12\}, C7 = \{1,2,3,4,5\}, C8 = \{4,5,11\}$
$L3 = \{C9 = \{3,6,7,8,9,10,11,12\}, C10 = \{2,4\}$

Fig. 2 Illustration of identifying five seed nodes from each layer after finding maximal cliques

3.3 Time Complexity

Let the complexity of identifying all the cliques in a graph of n vertices be $O(Y(n))$. The complexity of finding the size of each clique and sorting the cliques is $O(rNOMC + NOMClogNOMC)$, where r is the size of the largest clique. The complexity of selecting K seed nodes will be $O(K)$ (assuming that the degrees of nodes are available and sorted within clique).

3.4 Ignoring noted nodes for influence maximization

If a node is a seed node in one layer, then that node will be ignored for influence maximization in all the other layers where it is a non-seed node. We propose this to reduce the spreading time over the network and remove redundancy without losing influence maximization. Nodes that are ignored in other layers are termed as *noted nodes*. The idea is, if a node is activated in one layer, then it need not be reactivated in other layers since it is already influenced and information is available to that node. From this assumption, only non-seed nodes of a layer will be ignored for influence maximization, so ignoring these nodes will not affect much of the influence maximization. The number of newly influenced nodes after ignoring the nodes remains unchanged compared to the number of nodes influenced before ignoring them. The time complexity of influence maximization under the IC model after ignoring *noted nodes* is less comparatively. In Fig. 3, nodes 2, 3, and 5 are seed nodes in layer 1 and nodes 1, 5, and 8 are seed nodes in layer 2. Nodes 2, 3 and their links are ignored in the second layer since they are already seed nodes in layer one and non-seed nodes in layer 2. Similarly, nodes 1, 8 and their links are ignored for influence maximization in the first layer because they are seed nodes in the second and non-seed nodes in the first layer.

4 Experimental setup

To evaluate the performance, **pymnet** library [1] is used for generating synthetic multilayer network. All the experiments were executed on a system with the following configuration, Intel Xeon(R) workstation with 2.20Ghz CPU, 32GB RAM, and Ubuntu 16.04.

4.1 Dataset description

To compare the proposed algorithm's performance with the established algorithms, we use several real and synthetic datasets. Three real-time networks are used (i) Celegans multiplex GPI network [21], where multiplex consists of layers corresponding to different synaptic junctions. (ii) Drosophila multiplex GPI network [22], it consists of different types of genetic interaction for organisms in the general biological repository. (iii) Homo multiplex GPI network [32], it disseminates genetic and protein interaction data from humans and model organisms. In addition to the real datasets, a synthetic four-layer network $G = \{L^1, L^2, L^3, L^4\}$ is generated for the evaluation using the PYMNET library. In the synthetic network, each layer is generated with a node count of 150, i.e., $|V^i|$=150. All the four layers together constitute 600 nodes, i.e.$|V|$=600. Edges are placed randomly between the nodes, irrespective of the layer. The edge generation probability parameter is fixed as 0.5 so that half of the possible edges will be present in the network (Table 2).

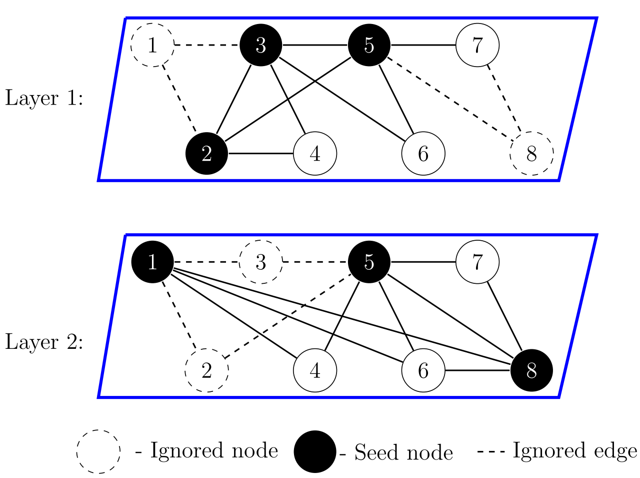


Fig. 3 Illustration of ignoring noted nodes for influence maximization

4.2 Methods to compare

- **ShaPley value-based Influential Nodes (SPINs)** [33] : In SPIN, the shapely value concept in the cooperative game theory problem is used for finding the top k nodes. The SPIN approach consists of two steps, computing a ranked list of nodes based on shapely value generation and choosing the top k nodes from the ranked list.
- **KSN** [9]: Knapsack seeding of networks (KSN) is an approximation algorithm, which parallelizes the problem in terms of the components of the layer. The difficulty lies in combining the solutions to the influence maximization problem on the separate layers to find the multiplex influence maximization (MIM). KSN achieves this by approximating the solution to the multiple-choice knapsack problem.
- **DISCO** [34]: Deep Learning-Based Influence Maximization(DISCO) is a learning-based framework. It extracts the nodes' features and trains the data using Deep Q Network (DQN) to identify seed nodes. It predicts the marginal influence maximization and selects the seed nodes.
- **DD**: The running time for the greedy heuristic is too large, so it may not be suitable for large and complex networks. So, other heuristics need to be applied. Degree-based heuristics are giving the best results. The basic idea behind the degree discount heuristic is that the nodes with a maximum degree as seed nodes give more influence than the other nodes. However, if the two highest-degree nodes are neighbors, they will not result in influence maximization. In such a case, the degree discount heuristic discounts/reduces the degree of a node whenever it has a connection with a node,

Table 2 Datasets used

Dataset	# of Nodes	# of Edges	# of Layers
Celegans multiplex GPI network	3879	8181	6
Drosophila multiplex GPI network	8215	43366	7
Homo multiplex GPI network	18222	170899	7
Pymnet synthetic network	600	179700	4

which falls under another seed node's influence. After discounting all the nodes' degrees, the seed set will be computed based on the higher degree nodes.

5 Results

In this section, we compare and analyze the results. As the proposed algorithm is clique based influence maximization, it is more effective than other algorithms. Any node in a clique is connected to all other nodes of the clique and due to this property, the influence maximization of our algorithm is better than other algorithms. In the first subsection, we compared the results of running time and algorithm performance on synthetic networks. In the second subsection, we compare the results of running time and algorithms performance on real networks. In each subsection, we have discussed two parameters for analysis and comparison of the performance of the heuristics. The first parameter is to find the number of influenced nodes on actual datasets and after ignoring *noted nodes* in actual datasets. The second parameter is the execution time to find the seed nodes, to find influenced nodes on actual datasets under the IC model, and to find influenced nodes after ignoring the *noted nodes* in actual datasets.

5.1 Synthetic networks

We considered synthesized multiplexes generated from PYMNET Model [1] with 600 nodes, 179700 edges i.e., half of the maximum possible number of edges in 4 layers. We assigned synthetic multiplex network to IC diffusion model with edge weights and thresholds chosen randomly in [0,1].

5.1.1 Algorithm performance on synthesized networks

In Fig. 4, we compared the performance of all five algorithms in two scenarios. The first scenario is in the generated synthetic network from the PYMNET model, and the second scenario is in the generated synthetic network after ignoring the *noted nodes* for influence maximization. We have selected 50 seed nodes from the network and assigned them to the IC model for influence maximization. In synthetic networks, results are different. In the first scenario, CIM performed better than other algorithms. After CIM, DD has given better results. DISCO has performed better than the KSN and SPIN. SPIN influence count is less than all other heuristics. In the second scenario, CIM influence count is more than other heuristics in the synthetic dataset after ignoring *noted nodes*, but DISCO, SPIN, and DD influence count is almost the same.

5.1.2 Running time on synthesized networks

Figure 5 presents the three different scenarios on the synthetic network dataset. In the first scenario, we compare the running time of identifying seed nodes under different heuristics in the generated synthetic network. In this, SPIN takes more time, and DD takes less time among all the

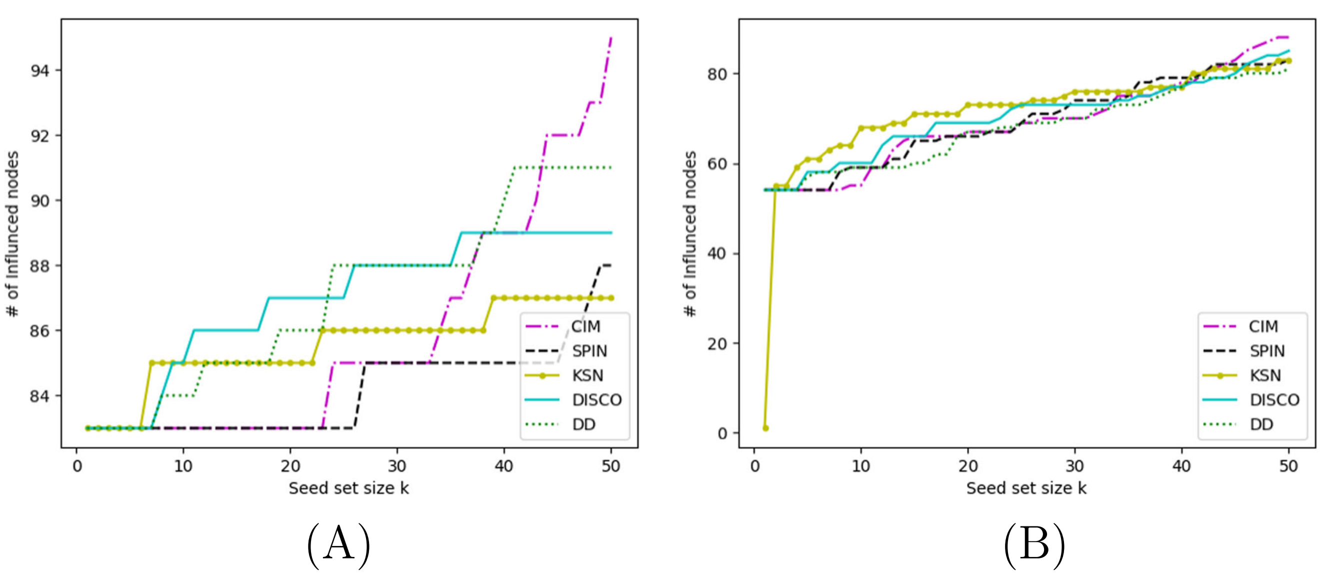


Fig. 4 **A** Influence spread of 50 seed nodes for different heuristics under IC model on Synthetic Network before ignoring *noted nodes*. **B** Influence spread of 50 seed nodes for different heuristics under IC model on Synthetic Network after ignoring the *noted nodes*

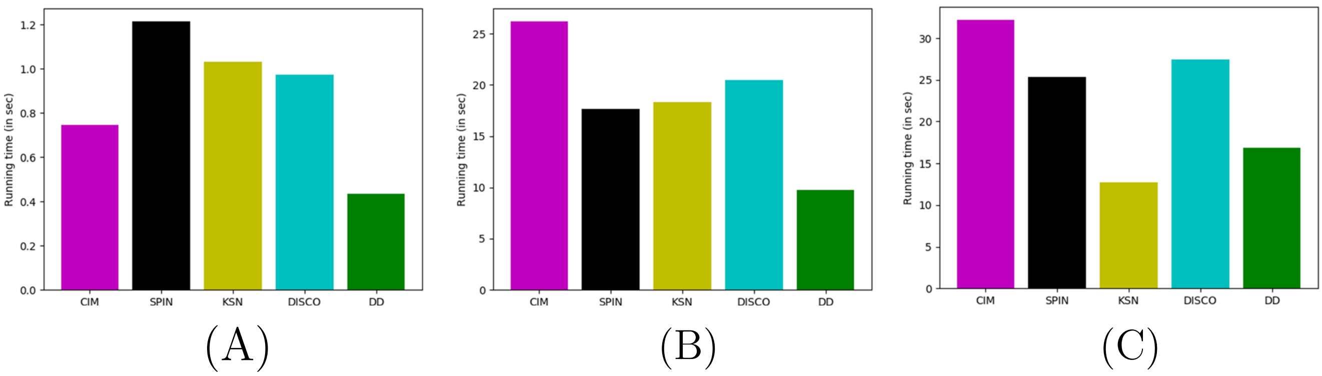


Fig. 5 **A** Running time of different heuristics for finding 50 seed nodes in Synthetic Network (**B**) Running time of IC Model for influence maximization on Synthetic Network. **C** Running time of IC Model on Synthetic Network after ignoring *noted nodes*

heuristics. CIM takes less time than DISCO, KSN, and SPIN, and takes more time than DD.

In the second scenario, we compare the running time of influence maximization under the IC model on the actual dataset. CIM takes more time for influence maximization than any other heuristic since it gives more influence count, and KSN takes less time than all other heuristics.

In third scenario, we compare the running time of influence maximization under the IC model after ignoring *noted nodes* in the actual dataset. In this DD takes less time than any other algorithm. DISCO takes more time than DD, SPIN, and KSN. Time complexity of all heuristics for influence maximization after ignoring *noted nodes* in the actual dataset is less than the before ignoring *noted nodes* in the actual dataset.

5.2 Real networks

5.2.1 Algorithm performance on real networks

Here, we present influence maximization on three real-time datasets, and after ignoring *noted nodes* in real datasets for influence maximization. Under each real-time dataset, we compare the number of influenced nodes under different heuristics using the IC model of information spread. Figure 6 shows influence maximization on Celegans multiplex GPI network dataset and after ignoring *noted nodes* in Celegans multiplex GPI network dataset. CIM produced the best results among all the heuristics under different test case conditions. DISCO has influenced more number of nodes than KSN, SPIN, and DD. Some seed

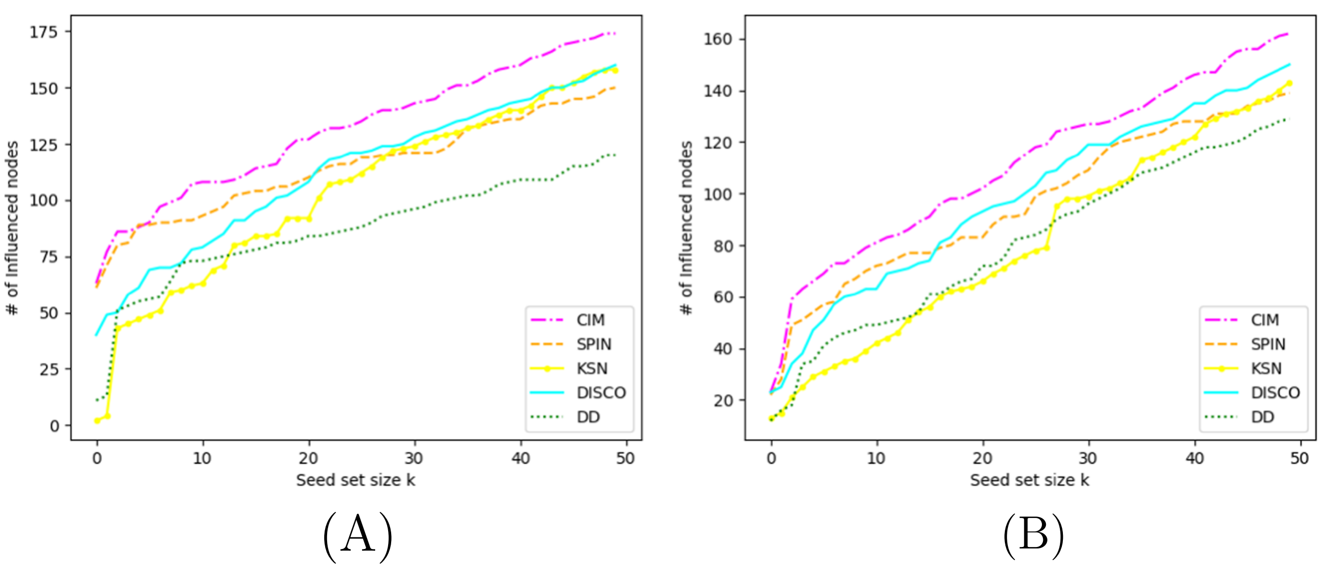
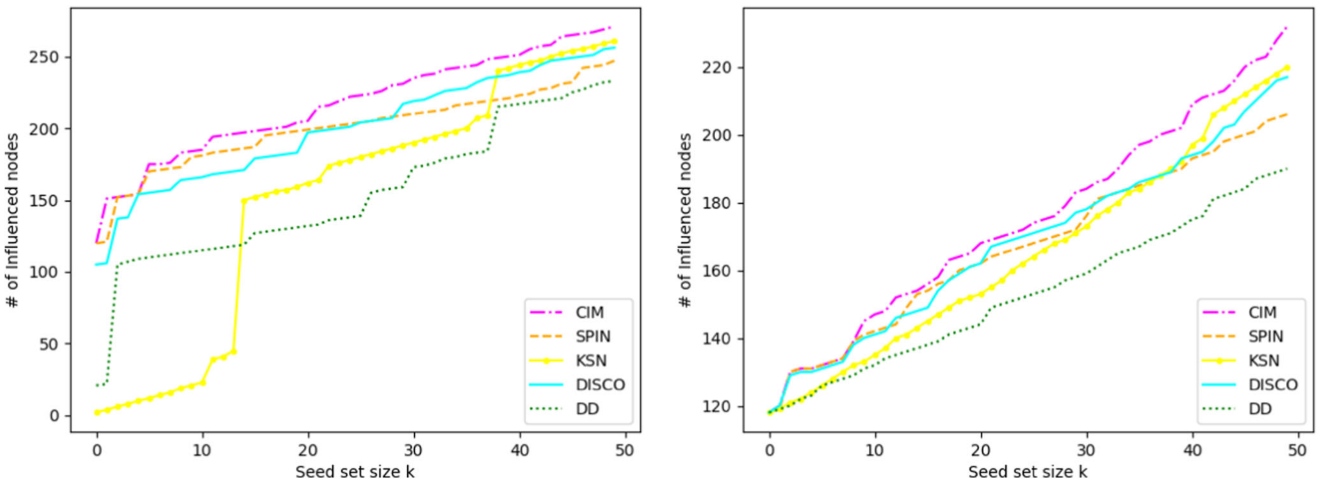


Fig. 6 **A** Influence spread of 50 seed nodes for different heuristics under IC model on Celegans multiplex GPI network. **B** Influence spread of 50 seed nodes for different heuristics under IC model on Celegans multiplex GPI network after ignoring *noted nodes*



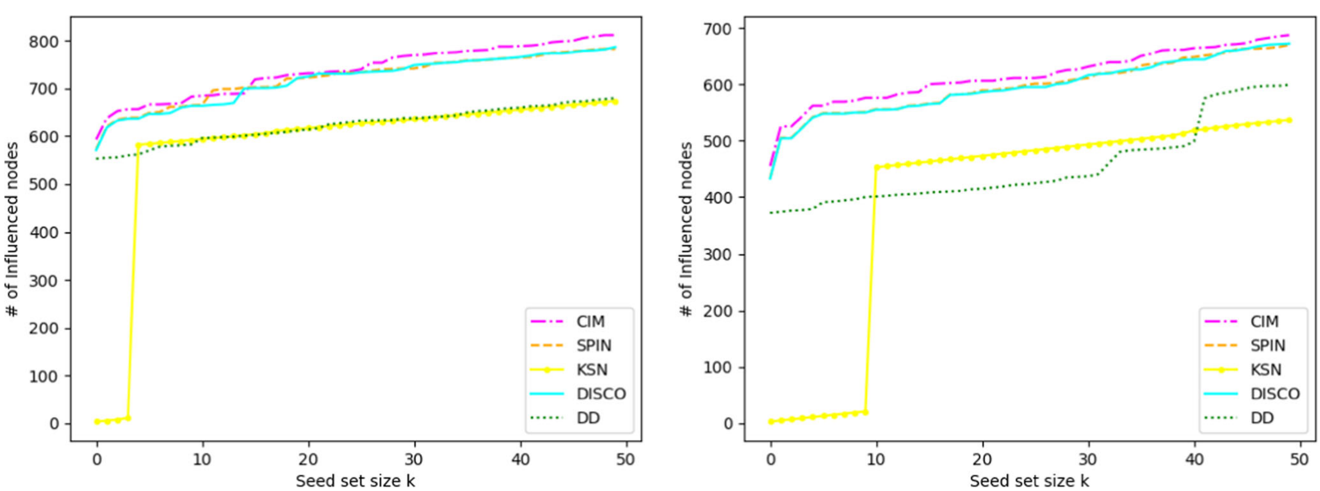
(A) (B)

Fig. 7 **A** Influence spread of 50 seed nodes for different heuristics under IC model on Drosophila multiplex GPI network for influence maximization. **B** Influence spread of 50 seed nodes for different heuristics under IC model on Drosophila multiplex GPI network after ignoring *noted nodes* for influence maximization

nodes can influence exponential number of nodes based on the network. Due to this, there is a sudden rise of some curves in the results.

Figure 7 presents influence maximization on Drosophila Multiplex GPI Network dataset and after ignoring *noted nodes* in Drosophila Multiplex GPI Network dataset. CIM performed better than other heuristics in both cases. KSN and DISCO have influenced the same number of nodes.

Figure 8 shows influence maximization on Homo multiplex GPI network and after ignoring the *noted nodes* in Homo multiplex GPI network. DISCO performance is marginally more than KSN and DD, DISCO, and SPIN influenced same number of nodes. In some cases, influenced nodes in before ignoring *noted nodes* (actual dataset) is marginally more than the dataset after ignoring the *noted nodes* because *noted nodes* does not come under influence



(A) (B)

Fig. 8 **A** Influence spread of 50 seed nodes for different heuristics under IC model on Homo multiplex GPI network for influence maximization. (B) Influence spread of 50 seed nodes for different heuristics under IC model on Homo multiplex GPI network after ignoring *noted nodes* for influence maximization

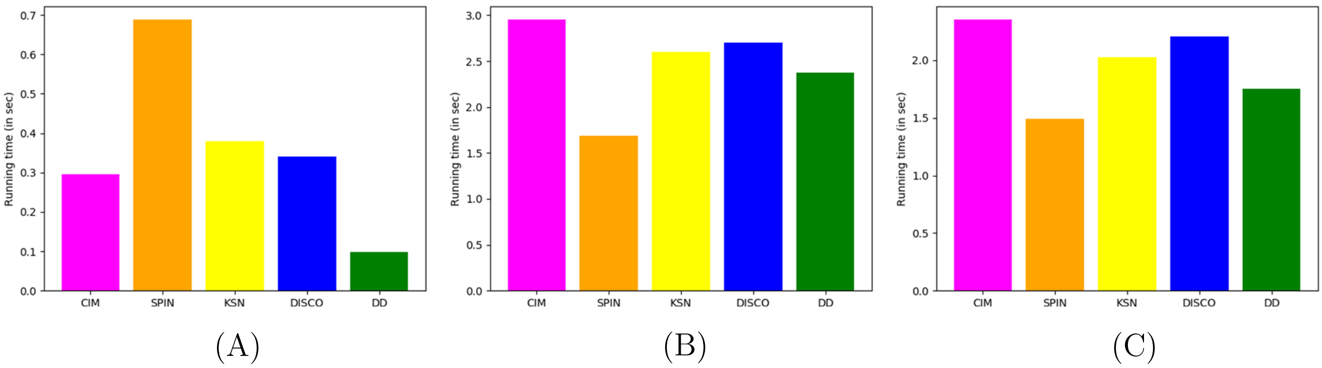


Fig. 9 **A** Running time of different heuristics for finding 50 seed nodes in Celegans multiplex GPI network (**B**) Running time of IC Model on Celegans multiplex GPI network for influence maximization. **C** Running time of IC Model on Celegans multiplex GPI network after ignoring *noted nodes* for influence maximization

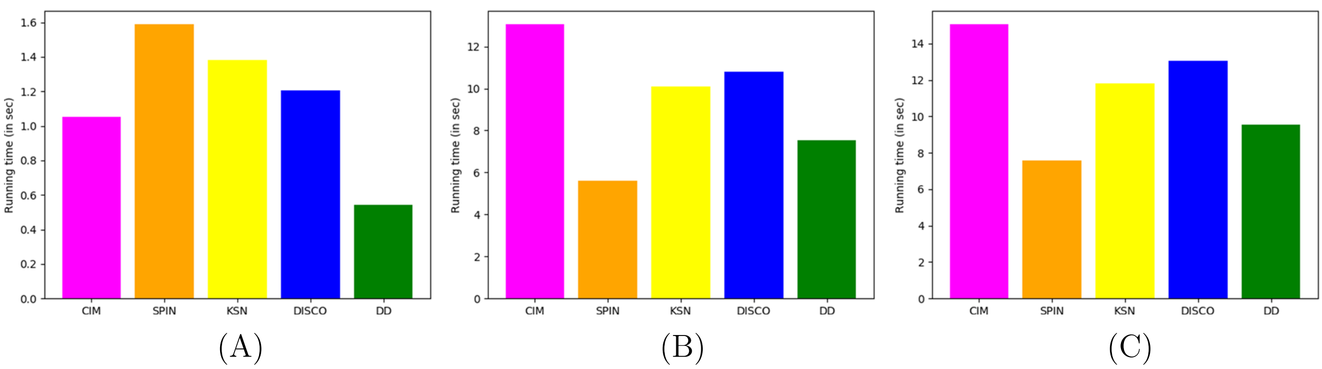


Fig. 10 **A** Running time of different heuristics for finding 50 seed nodes in Drosophila multiplex GPI network (**B**) Running time of IC Model on Drosophila multiplex GPI network for influence maximization. **C** Running time of IC Model on Drosophila multiplex GPI network after ignoring *noted nodes* for influence maximization

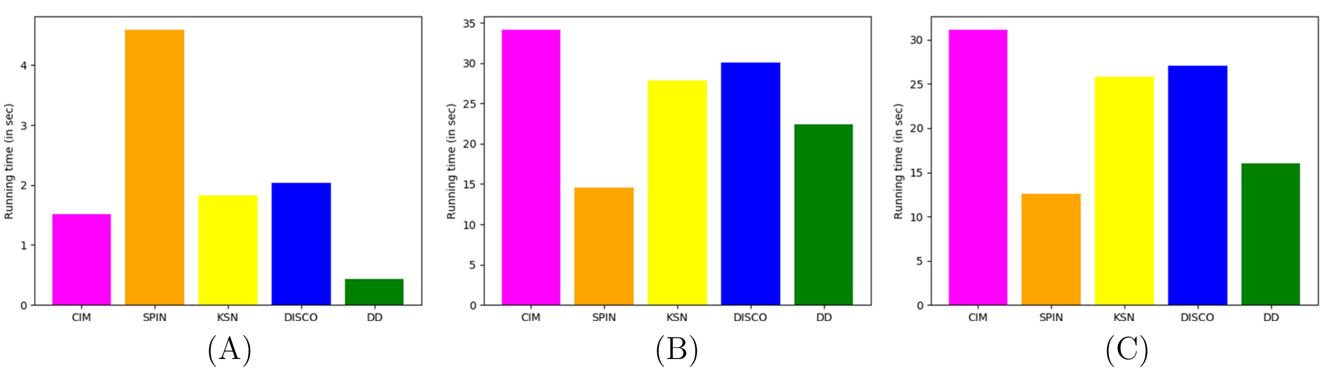


Fig. 11 **A** Running time of different heuristics for finding 50 seed nodes in Homo multiplex GPI network (**B**) Running time of IC Model on Homo multiplex GPI network for influence maximization. **C** Running time of IC Model on Homo multiplex GPI network after ignoring *noted nodes* for influence maximization

count. In three real-time datasets the results are similar, and in synthetic datasets results are different.

5.2.2 Running time on real networks

Here, we compare the three different scenarios in mentioned datasets; first scenario is the time performance of finding seed nodes under different heuristics. Second scenario is the time performance of influence maximization under the IC model for different real datasets. The third scenario is the time performance of influence maximization after ignoring *noted nodes*. Figure 9 shows the three-time scenarios on Celegans multiplex GPI network. Figure 10 shows the three-time scenarios on the Drosophila multiplex GPI network dataset, and Fig. 11 presents the three-time scenarios on the Homo multiplex GPI network dataset. Comparatively, the results on all three datasets are similar in each time scenario. In the first time scenario SPIN takes more time, and DD takes less time among all the heuristics. CIM takes less time than DISCO, KSN, and SPIN, and takes more time than DD. In the second time scenario, among all the heuristics in real-time datasets SPIN takes less time, and CIM takes more time for influence maximization. KSN takes more time than SPIN, DD and less time than DISCO. Though CIM takes more time for synthetic datasets, KSN takes less time. DISCO takes more time than SPIN, DD, and KSN. Similarly, in the third time scenario, among all the heuristics in real-time datasets SPIN takes less time, and CIM takes more time. DISCO takes more time than SPIN, DD, and KSN.

6 Conclusion and future work

This paper has introduced an efficient CIM algorithm for influence maximization in multilayer networks under the independent cascade diffusion model. CIM performs better than other algorithms. Our CIM algorithm considers connectivity of the graph since a node in a clique connects to all other nodes. Therefore, the probability of influencing large number of nodes is more. We came to this conclusion after conducting extensive experiments on different datasets and synthetic networks.

Complex networks are a continually focused concept in the realm of network science. Influence maximization emerged as an essential topic in complex networks. So far, researchers have put efforts into influence maximization in single-layer networks. The progress in detecting seed nodes in multilayer networks is limited. Influence maximization in multilayer networks has not been explored intensively. As part of this work, few heuristics are proposed and applied in the influence maximization of multilayer networks. A performance evaluation testbed is developed and used to analyze and compare the results of the heuristics. This work can be extended by the usage of other innovative heuristics in the multilayer influence maximization. The evaluation platform described in the paper can be used to analyze and compare the new heuristics results.

References

1. Kivelä M, Arenas A, Barthelemy M, Gleeson JP, Moreno Y, Mason A (2014) Porter† Multilayer networks. J Compl Netw 2:203–271
2. Kempe D, Kleinberg JM, Tardos E (2003) Maximizing the spread of influence through a social network. In: KDD
3. Nemhauser G, Wolsey L, Fisher M (1978) An analysis of the approximations for maximizing submodular set functions. Math Programm 14:265–294
4. Chen W, Wang Y, Yang S (2009) Efficient influence maximization in social networks. In: KDD
5. Moon JW, Moser L (1965) On cliques in graphs. Israel J Math 23_28
6. Tomita E, Tanaka A, Takahashi H (2006) The worst-case time complexity for generating all maximal cliques and computational experiments. Theor Comput Sci 363:28–42
7. Mohite M, Narahari Y Incentive Compatible Influence Maximization in Social Networks and Application to Viral Marketing, arXiv:1102.0918
8. Tomasini M An Introduction to Multilayer Networks, https://doi.org/10.13140/RG.2.2.16830.18243
9. Kuhnle A, Md AA, Li X, Zhang H, My T (2018) Thai multiplex influence maximization in online social networks with heterogeneous diffusion models. IEEE Trans Comput Soc Syst Page(s):418–429
10. Bron C, Kerbosch J (1973) Algorithm 457: Finding all cliques of an undirected graph. Commun ACM 16:575–577
11. Li Q, Chen W, Sun X, Zhang J Influence Maximization with Almost Submodular Threshold Functions. NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing
12. Mossel E, Roch S (2009) Submodularity of Infuence in Social Networks: From Local to Global. SIAM Journal on Computing
13. Domingos P, Richardson M (2001) Mining the network value of customers. In: KDD, pp 57–66
14. Leskovec J, Krause A, Guestrin C, Faloutsos C, Van-Briesen J, Glance N (2007) Cost-effective outbreak detectionn in networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, pp 420–429
15. Goyal A, Lu W, Lakshmanan L (2010) Celf++: Optimizing the greedy algorithm for influence maximization in social networks. In: Proceedings of the 20th International Conference Companion on World Wide Web, Hyderabad, pp 47–48
16. 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, Washington, pp 1029–1038
17. Chen W, Yuan Y, Zhang L (2010) Scalable influence maximization in social networks under the linear threshold model. In: Proceedings of the 10th IEEE International Conference on Data Mining, pp 88–97
18. Basaras P, Iosifidis G, Katsaros D, Tassiulas L (2019) Identifying Influential Spreaders in Complex Multilayer Networks: A Centrality Perspective. IEEE Trans Netw Sci Eng 2, vol 6

19. Wang J, Zhang R, Wei W, Pei P, Zheng Z (2009) On the stability of multilayer Boolean networks under targeted immunization. KDD
20. Wang D, Wang H, Zou X (2017) Identifying key nodes in multilayer networks based on tensor decomposition. Chaos: An Interdisciplinary Journal of Nonlinear Science
21. Stark C, Breitkreutz B-J, Reguly T, Boucher L, Breitkreutz A, Tyers M (2006) Biogrid: a general repository for interaction datasets. Nucleic Acids Res 34(1):D535–D539
22. De Domenico M, Nicosia V, Arenas A, Latora A (2015) Structural reducibility of multilayer networks. Nat Commun 6:6864
23. Singh SS, Kumar A, Singh K, Biswas B (2019) LAPSO-IM: A Learning-based influence maximization approach for social networks. Applied Soft Computing, Volume 82, 105554, ISSN, 1568–4946. https://doi.org/10.1016/j.asoc.2019.105554
24. Singh SS, Kumar A, Singh K, Biswas B (2020) IM-SSO: Maximizing influence in social networks using social spider optimization. Concurrency Computat Pract Exper 32:e5421. https://doi.org/10.1002/cpe.5421
25. Singh SS, Kumar A, Singh K, Biswas B (2019) C2IM: Community based Context-aware Influence Maximization in Social Networks. In: Physica a: Statistical mechanics and its applications, vol 514. Elsevier, pp 796–818, ISSN: 0378-4371
26. Singh SS, Singh K, Kumar A, Biswas B (2019) ACO-IM: Maximizing influence in social networks using ant colony optimization. In: Soft computing, springer, ISSN: 1433–7479
27. Singh SS, Singh K, Kumar A, Biswas B (2019) MIM2: Multiple Influence maximization across multiple social networks. In: Physica a: Statistical mechanics and its applications, Elsevier, ISSN: 0378-4371
28. Cheng S et al (2013) StaticGreedy. Proceedings of the 22nd ACM international conference on Conference on information and knowledge management - CIKM '13: n. pag. Crossref. Web
29. Ohsaka N, Akiba T, Yoshida Y, Kawarabayashi K-I (2014) Fast and accurate influence maximization on large networks with pruned Monte-Carlo simulations. Inproceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI'14). AAAI Press, pp 138–144
30. Chen Y-C, Zhu W-Y, Peng W-C, Lee W-C, Lee S-Y (2014) CIM: Community-Based Influence Maximization in Social Networks. ACM Trans Intell Syst Technol 5, 2, Article 25:31. https://doi.org/10.1145/2532549
31. Singh SS, Singh K, Kumar A, Biswas B (2018) CoIM: Community-based Influence Maximization in Social Networks. In: Second international conference on advanced informatics for computing research (ICAICR-2018), Shimla
32. De Domenico M, Porter MA, Arenas A (2015) MuxViz: A Tool for Multilayer Analysis and Visualization of Networks . J Complex Netw 3(2):159–176
33. Narayanam R, Narahari Y (2011) A shapley Value-Based approach to discover influential nodes in social networks. IEEE Trans Autom Sci Eng 8(1):130–147
34. Li H, Xu M, Bhowmick SS, Sun C, Jiang Z, Cui J (2019) DISCO Influence Maximization Meets Network Embedding and Deep Learning. arXiv:1906.07378

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Venkatakrishna Rao. K is working toward his PhD in the department of Computer science and engineering, Indian Institute of Technology (BHU), Varanasi and he received his M.Tech degree from Indian Institute of Technology, Madras. His research interests are data analytics and data mining in social network analysis.

Mahender Katukuri is working toward his PhD in the department of Computer science and engineering, Indian Institute of Technology (BHU), Varanasi and he received his M.Tech degree from Indian Institute of Technology, Madras. His research interests are data analytics and data mining in social network analysis.

Maheswari Jagarapu is pursuing her M.Tech at department of computer science and engineering, Jawaharlal Nehru Technological University, Kakinada and she received her B.Tech degree from Jawaharlal Nehru Technological University, Kakinada. Her research interests are data analytics and data mining in social network analysis.