Neurocomputing 453 (2021) 151–163

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Link prediction-based influence maximization in online social networks

Ashwini Kumar Singh *, Lakshmanan Kailasam *

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

A R T I C L E I N F O

Article history:
Received 16 September 2020
Revised 30 March 2021
Accepted 22 April 2021
Available online 30 April 2021
Communicated by Zidong Wang

Keywords:
Online influence maximization
Dynamic link prediction
Evolutionary social networks

A B S T R A C T

Influence Maximization (IM) is the problem of finding a small set of highly influential users in the social networks. The influence spreads according to an explicit influence propagation model. IM is an essential component in many applications such as Network Monitoring and Viral Marketing. Most of the present IM solutions neglect the highly dynamic behavior of social networks. It can result in either deprived seed qualities or a prolonged processing time. In this paper, we study the IM problem in a social network that evolves with time and proposes a new Link Prediction based Influential Node Tracking (LPINT) framework. In the proposed model, we apply the conditional temporal Restricted Boltzmann Machine (ctRBM) to predict the upcoming snapshot of the graph by predicting the links that may appear in the network by considering the evolutionary network's temporal and structural pattern. And then, we apply an efficient IM technique for finding the seed nodes in the predicted snapshot of the network. Finally, we evaluate the spread of influence in the latest snapshot of the graph using predicted seed nodes. Extensive experimentation on four real large-scale datasets confirms that our LPINT model attains better performance in terms of influence coverage and influence spread time for considered networks compared to the baseline techniques.

1. Introduction

In recent decades Internet technology has grown significantly, digital world affected human activities and lifestyle in various aspects. Communication, the spread of news, trade, propagation of information have taken new forms [1]. Different forms of social network platforms have become popular in society. Users communicate, share data, or exchange views by joining these networks. A large number of users and rapid exchange of information among them has made social networks a powerful means for information spreading. Various companies use these networks for the marketing [2] of their products. However, it is impractical for them to connect with each user individually. We can also observe that the users often decide to purchase a product on the advice of their friends [3]. Thus, the more practical solution would be to find the most influential users as seed nodes set and target them for advertisement. The problem of finding the top k most influential users is known as influence maximization in literature which is an *NP-Hard problem* [4].

Studies have modeled the process of influence maximization in social networks for application domains such as social media [5], epidemiology [6], viral marketing [7–10], political campaigning [11], fake news containment [12,13] and many more. Several models have been proposed to formulate the diffusion process. Independent Cascade (IC) model and Linear Threshold (LT) model are the two elementary diffusion models proposed by Kemp et al. [4]. In the *IC model*, a node has a probability of convincing each of its neighbors. And in the *LT model*, a node accepts a new idea if the influence from all its neighbors has crossed a threshold. Based on these two, many other models [14–16] have been built. Researchers have extensively studied and proposed various algorithms [4,17–19,2,20–22] to influence maximization in a static network. However, it is natural and significant to understand that social networks have a continuous change in structure [23–25]. These changes in structure often occur in real applications; for instance, connections appear and disappear when users friend/unfriend others on Facebook [26]; alternatively, they follow/unfollow others on Twitter [27]. With the change in the structure of social networks, the amount of influence a person can have on others also keeps changing. People also get more affected by neighbors she/he communicates more often with. A neighbor's impact decreases if he/she does not communicate with the person for a long time. Using static seed nodes for different snapshots of the graph may not give good performance for influence maximization at different timestamps. And we have observed in our experiments that using dynamic seed nodes depending on the timestamp turns out to be crucial for the success of influence maximization in dynamic social

* Corresponding authors.
E-mail addresses: ashwinikrsingh.rs.cse17@iitbhu.ac.in (A.K. Singh), lakshmanank.cse@iitbhu.ac.in (L. Kailasam).

https://doi.org/10.1016/j.neucom.2021.04.084
0925-2312/

networks. Some papers have considered IM problems in dynamic networks and proposed algorithms like [28–30]. However, they all fail to ensure an efficient seed set that maximizes the influence as the network evolves.

In dynamic networks, the number of snapshots is the graph instances taken after a fixed interval of time (timestamps) for an evolutionary graph in which the edges are being added/removed with the increase of time. Here, we need to select suitable seed nodes at different timestamps for achieving the maximum influence spread. It is because the changing structure affects the spreading capacity of seed nodes. To demonstrate the idea of link prediction-based IM in dynamic social networks, consider an example shown in Fig. 1. This example shows the links (edges) between users (nodes) at different timestamps. Each edge indicates that a user can influence another user. Figs. 1-a, 1-b, 1-c, 1-d shows the snapshots G_0, G_1, G_2, G_3 of an evolving graph at the time-stamps $t = 0, 1, 2, 3$, respectively. The most influential node at timestamp $t = 0$ in snapshot G_0 is b, as it seems to influence the maximum number of nodes. Similarly, at time $t = 1,\ t = 2$ and $t = 3$, nodes $e,\ e$ and d are the most influential nodes, respectively. Even in this simple example, we can see that the most influential node can be different for different snapshots of the graph.

For dynamic social networks, the seed nodes selected for a particular snapshot may not be influential at other snapshots of the network. So, we need to compute the seed nodes for each snapshot of the network. However, the seed selection process itself takes a

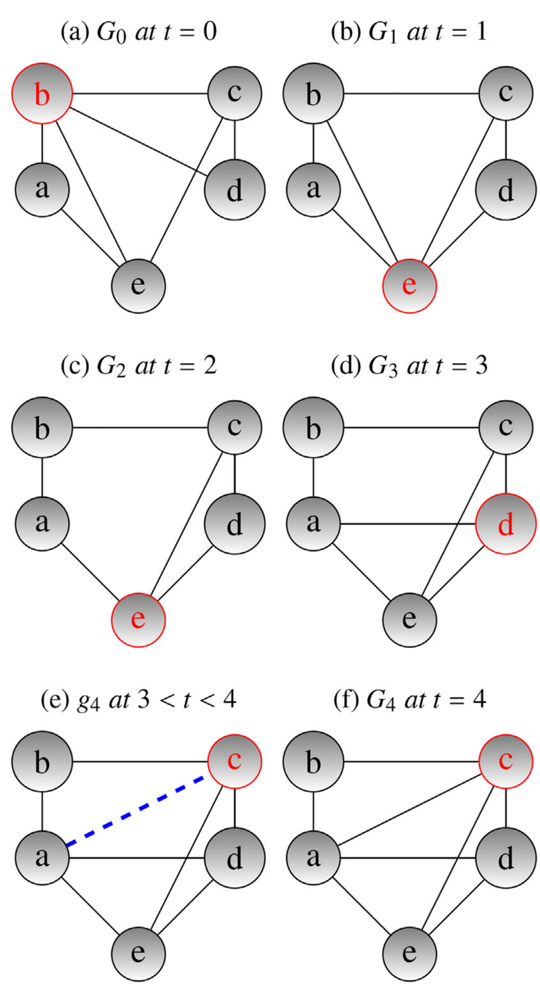

Fig. 1. (a), (b), (c), (d), (f) are the snapshots of the graph $G = \{G_0, G_1, G_2, G_3, G_4\}$ respectively, and (e) is the predicted $g_{t+1} = g_4$, here in g_4, link $a - c$ is expected to appear in snapshot G_4.

significant amount of time for large networks, and if the network is highly dynamic in nature, then the computed seed nodes may not be the best ones due to continuous change in network structure.To deal with this problem, we propose a new influence maximization algorithm, LPINT. In our proposed algorithm, the crucial idea is to predict the next snapshot of the graph by considering the evolving graph's temporal and structural behavior. For example, in Fig. 1 (e), g_4 is the predicted snapshot at $3 < t < 4$, and node c is chosen as the seed node for the predicted graph g_4. We use this predicted seed node for information diffusion in actual upcoming snapshot G_4 at time $t = 4$ for efficient and faster information diffusion. In our proposed method, we predict the next snapshot of the graph using an efficient link prediction method. We then find the seed set for the predicted snapshot using an efficient IM technique. This seed set is used for information diffusion in the next snapshot of the graph.

For link prediction, we use the ctRBM technique, which combines the temporal as well as structural behavior of the nodes in evolving graphs to predict the upcoming links. Next, to select the efficient seed set, we tried various state-of-the-art algorithms of influence maximization and found that the Upper Bound based Lazy Forward (UBLF) [31] approach works better in our problem. We use an improved *UBLF* algorithm named *EXCHANGE* algorithm in our proposed LPINT framework. To make the *EXCHANGE* algorithm run faster, we find the most influential nodes in the current snapshot starting from the set of seed nodes found from the previous snapshot rather than from an empty set.

In this work, we assume that the active nodes are the ones that have communicated in past N snapshots. We find seed nodes only among these active nodes. For example, let us consider the friendship/following relationship of the social network. Many nodes in the network do not communicate frequently, or some of them are less active or inactive. Considering these inactive nodes for the IM problem would be a waste of time and resources. These inactive nodes are rarely helpful in IM applications like viral marketing or fake news containment. In the proposed work, we consider only the active nodes in the seed selection process. Implementing this assumption is novel and makes our goal of influence maximization more efficient and effective.

Briefly, our contributions can be stated as follows:

- We define a novel Influential Node Tracking problem to maximize the influence spread in an online social network.
- We propose an LPINT framework for efficient and effective influence maximization in dynamic social networks. The proposed model uses the link prediction technique to predict the upcoming snapshot of the graph and then computes the seed set for influence maximization.
- Experimentally, we show that the proposed framework performs better in terms of influence spread in comparison to the considered baseline techniques on considered datasets.

Paper Organization. We review the relevant literature in Section 2. In Section 3, we formally frame our link prediction based influence maximization problem after introducing the link prediction, diffusion, and IM problem. We then described our proposed methodology and LPINT algorithm in Section 4. We have shown our experimental outcomes on four real-world, large-scale online social networks in Section 5, and finally we conclude our work along with possible future research directions in Section 6.

2. Related work

The study of influential nodes in viral marketing was first proposed in [8] by Domingos and Richardson. Further, Kemp *et al.* in

[4] formulated the influence maximization problem as a combinatorial optimization problem and suggested a greedy algorithm using IC and LT models with an approximation guarantee of $(1 - 1/e)$. However, for large networks, the proposed solution does not scale due to its requirement of a large number of Monte-Carlo simulations for the estimation of influence spread. Several techniques [32–36] have been proposed to handle this issue for influence maximization in static networks.

Broadly, the solutions proposed for influence maximization can be divided into two categories: the algorithms of the first category aims to improve the performance of the greedy algorithm and give an approximation guarantee of $(1 - 1/e)$ [31,37]. Alternatively, the second category of algorithms put on several heuristics but lack verifiable approximation guarantee [17–19,2]. However, all these approaches consider only static networks.

Some of the IM methods in the online social network consider the snapshots of the graph and then apply static IM algorithms. However, these approaches do not handle the real dynamics of the network. Some methods consider the graph streams, but they do not provide a theoretical guarantee of their seed quality and may return arbitrarily bad solutions. For instance, Aggarwal et al. [38], Zhuang et al. [39], and Song et al. [40] focuses on $t + \delta$ given the changing aspects of the progress of the network throughout the interval $[t, t + \delta]$, where δ denotes the small change in time. They apply diffusion maximization independently in each static graph G_t and using S_{t-1} as seed node set for influence spread. However, the previous seed set might become inefficient at a later stage because of the graph's dynamic nature. So, these solutions are not effective for dynamic social networks.

Recently, there are many studies about IM in online social networks. A few significant contributions are discussed in this section. Wang et al. in [41] proposed a "Pairwise Factor Graph (PFG) model" to formalize the problem of IM in social network using a probabilistic model and further extended it by incorporating the time information, which results in the "Dynamic Factor Graph (DFG) model" for IM in the dynamic social network. Aggarwal et al. in [38] use the communications of given social network entities, which can frequently be predicted based on past behavior of the evolving network, and these represented future interactions, which were used to model the spread of information. Rodriguez et al. in [42] developed a method INFLUMAX for IM that considers time-based dynamics underlying the diffusion processes. This method allows for variable transmission (influence) rates between nodes of a network.

Zhuang et al. in [39] proposed an algorithm to determine a subset of seed nodes in the network so that the particular information diffusion method in the network can be best projected with the probing nodes. That is, it decreases the likely error among the evaluated network and the real network. Gayraud et al. in [29] introduced a persistent and transient variation of IC and LT model for justifying network evolution. Li et al. in [43] proposed a novel conformity-aware greedy algorithm called CINEMA for a conformity-aware cascade model that integrates the interplay among conformity and influence. Han et al. in [44] proposed a dynamic probing context that accepts the community structure as a unit and updates network topology to investigate the genuine changes of network and employs community-based influence maximization. Wang et al. in [45] proposed an IM query named Stream Influence Maximization (SIM) on social streams, it embraces the sliding window model and keeps up a set of k seeds with the most significant influence value over the latest social activities. Tong et al. in [30] demonstrated the dynamic IC model and presented the idea of an adaptive seeding technique with a provable performance guarantee. However, all these approaches require high computation costs, and there is still a lot of scopes to do better for dynamic social networks.

In this paper, our goal is to ensure the maximum information spread in a minimum time span in online social networks. For achieving this goal, we predict the upcoming snapshot of the graph as G_{t+1} by considering the temporal and structural behavior of the evolving graph, and then we detect the seed set that maximizes the information spread in the upcoming snapshot of the graph.

For predicting the upcoming snapshot of the graph, we have used the link prediction method for the prediction of upcoming links in the dynamic networks. In literature, most of the link prediction methods are based on heuristics, such as the sum of mutual neighbors. Many heuristic methods [46–48] do not consider the dynamic and evolutionary behaviors of the networks and consider one static snapshot of the graph for link prediction. However, graph data of social networks generally have significant evolution information, such as the pattern of change in the structure of the graph [49]. Some probability-based models of link prediction [50–53] have problems related to the model capacity and computation. In this paper, we have used a "Conditional Temporal Restricted Boltzmann Machine (ctRBM) model" [54] for the prediction of graph links based on temporal and structural patterns of the dynamic social network. The proposed method considers the temporal and structural behavior of the dynamic social network over a period to predict the upcoming snapshot of the graph by using the ctRBM method. The most influential set of seed nodes for the network is determined by the proposed algorithm in advance and used for information spread on the actual snapshot of the graph.

3. Preliminaries and problem statement

In this section, first, we explain the concept of *Link Prediction* and then introduce the *Independent Cascade (IC) Model* as the information diffusion model. Next, we define the *Influence Maximization* problem for social networks. Further, we formally defined our pro-

Table 1
Some of the notations used.

Notations	Descriptions
$G(\mathbb{V}, E)$	a simple graph
$\mathbb{V}$	the vertex set
E	the edge set
$\rho = \{G_t\}_1^T$	set of snapshots of an online social network
$G_t = (\mathbb{V}, E_t)$	a snapshot of ρ at time t
E_t	the edge multiset of G_t
S	the seed set
S_t	the seed set at timestamp t
k	the size of seed set
$\sigma(S)$	expected number of nodes influenced by S
$p_{u,v}^G$	the propagation probability between node u and v in snapshot G
N	window size
W_A	weight matrix of ctRBM at time $t-1$
θ_m	parameter of ctRBM model R_m
R_m	ctRBM for node m
$\mathbb{R}$	collection of ctRBMs for all nodes
η_m^t	neighbors impact on node m at time t
V	visible layer in RBM
$\hat{p}$	total number of nodes in the visible layer
H	hidden layer in RBM
$\tilde{V}$	reconstructed data (model's estimation)
$N_{\overline{V}}$	number of visible variables in ctRBM
$N_{\overline{H}}$	number of hidden variables in ctRBM
β	hyperparameter balancing temporal variations
γ	hyperparameter controlling node neighbors impact
x	bias for visible layer $\overline{V}$ in RBM
$\bar{x}^t$	adaptive bias of V at t for ctRBM model
y	bias for hidden layer $\overline{H}$ in RBM
$\Delta_{e,e_s}(S_t)$	the replacing gain of changing from e_s to e

posed *LPINT* problem. The symbol notations used are listed in Table 1.

3.1. Link prediction

In the link prediction problem, we need to predict the edges that are expected to be added to the network at a future time t' given the snapshot of the network at time t [47].

Link predictor: Consider a graph $G(\mathbb{V}, E)$, where $\mathbb{V}$ is the set of nodes, and E is the set of connections. Multiple connections and self-association are not permitted. Denoted by U, the universal set containing all $\frac{|\mathbb{V}|.(|\mathbb{V}|-1)}{2}$ potential connections, where $|\mathbb{V}|$ signifies the number of elements in set $\mathbb{V}$. The set of nonexistent connections is $U - E$. We expect some missing connections (or the connections that will show up in the future) from the set $U - E$ and to discover this connection is the task of link predictor.

Traditional Link Prediction method: All strategies give an association weight score $s(x, y)$ to pair of nodes (x, y) in the input graph G. And then produce a positioned list in decreasing order of the score $s(x, y)$. In this way, they can be seen as computing a vicinity or similarity between nodes x and y concerning the network topology.

Time series based Link prediction: In time series based link prediction, the essential idea is to make a time-based ordering of each non-connected pair of nodes of the network utilizing similarity scores are given by a topological metric [55]. This value between the pair of nodes is calculated from unsupervised or supervised learning-based link prediction methods.

In this work, we have used time-series-based link prediction using the ctRBM method, which is described in Section 4.1. A general RBM is introduced here.

Restricted Boltzmann Machine (RBM) [54]: is a particular case of Markov Random Field, which has two layers of variables, Visible layer variables are denoted as set V and Hidden layer variables are denoted as set H. A typical RBM is represented in Fig. 2. Here, set V and set H to form a fully-connected bipartite graph with undirected edges. RBM defines a distribution over $(V, H) \in \{0,1\}^{N_V} \times \{0,1\}^{N_H}$, where N_V and N_H is the dimension of V and H layers. The joint probability distribution for RBM is defined as:

$$P(V, H) = \exp\left(V'WH + x'V + y'H\right)/Z \tag{1}$$

Here $Z = \sum_{V,H} \exp\left(V'WH + x'V + y'H\right), W \in \mathbb{R}^{N_V \times N_H}$ is the weight between layers V and H, x, y is the biases for V and H, respectively. V', x', and y' are the transposes of V, x, and y, respectively.

Due to the bipartite nature of the RBM, there is no interaction between nodes in individual layer, so the conditional probability distributions are fully factorial and represented by:

$$\begin{aligned} P(H_j = 1|V) &= \omega\left(y_j + W'_{:,j}V\right), \\ P\left(\widetilde{V}_i = 1|H\right) &= \omega(x_i + W_{i,:}H), \end{aligned} \tag{2}$$

Here, ω is the logistic function given as $\omega(a) = (1 + \exp(-a))^{-1}, \widetilde{V}$ is the reconstructed data showing the model's evaluation, i and j are row and column index. The aim of learning is to minimize the gap between V and $\widetilde{V}$.

3.2. Diffusion model

In this work, we study the influence maximization in social networks using the Independent Cascade model. In the IC model, the social network is modeled as a directed graph $G = (P, E)$, where P relates to the people, while E represents the set of social connections between the people. Additionally, each edge $(u, v) \in E$ is given a propagation probability $p^G_{u,v}$ indicating the strength of influence of the individual u has on v.

The IC model [4] describes a straightforward and intuitive diffusion process. Beginning from a seed set S, which is active, the diffusion process happens in discrete-time steps as follows. When a node u becomes active in step t, it attempts to activate all of its inactive neighbors in step $t + 1$. For each neighbor v, it succeeds with the known probability p_{uv}. If it succeeds, v ends up active; else, v stays inactive. When u have made all these attempts, it does not get the chance to make further activation attempts at later occasions.

3.3. Influence maximization

Given the seed set S, we describe the influence spread of S as the expected number of activated nodes when the diffusion procedure stops, represented by the influence function $\sigma(S)$. The Influence Maximization (IM) problem is to find a seed set $S \subseteq V$ of maximum

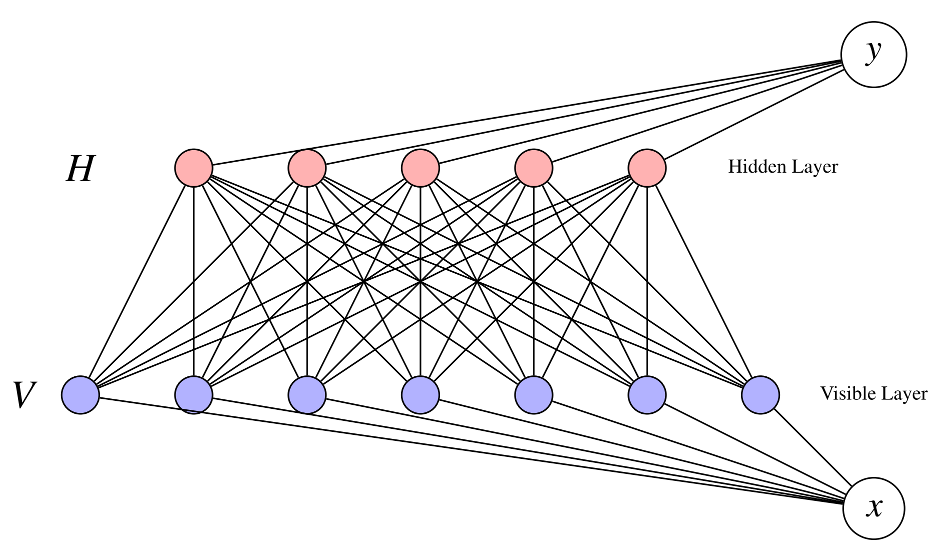

Fig. 2. Restricted Boltzmann Machine.

size k to maximize the influence function $\sigma(S)$. Formally, IM is defined as the following optimization problem:

$$S^* = \underset{|S| \leqslant k}{\arg\max}\, \sigma(S). \quad (3)$$

It has been shown by Kempe et al. in [4] that the IM problem under the IC model is NP-hard. It can be shown that the influence function $\sigma(S)$ under the IC model is monotone and submodular. A set function f is monotone if $f(S+e) \geqslant f(S)$ for an element e; and f is submodular if it has diminishing returns as $f(S+e) - f(S) \geqslant f(T+e) - f(T)$ for an element e whenever $S \subseteq T$. These properties of the IC model allow for an approximation algorithm with a guarantee.

In particular, there is an elementary Greedy Algorithm 1 proposed by Nemhauser et al. in [56] for maximizing monotone submodular functions. The greedy algorithm repeatedly picks the node with maximum marginal gain and adds it to the present seed set until the budget k is reached. It can be shown that this algorithm approximates the optimal solution with a factor of the $(1 - \frac{1}{e})$ for the IM problem.

Algorithm 1. *GREEDY*$(G = (\mathbb{V}, E), k)$

1: Initialize: $S = \varnothing$
2: **for** $i = 1$ to k
3: $\quad e^* = \underset{e \in \mathbb{V}-S}{\arg\max}\{\sigma(S \cup \{e\}) - \sigma(S)\}$
4: $\quad S \leftarrow S + \{e^*\}$
5: **end for**
6: **return** S.

Though accurate calculation of marginal gain is NP-hard, an approximate estimate can be obtained by multiple Monte-Carlo simulations. However, this is inefficient for large networks. Various strategies have been proposed to handle the inefficiency of the greedy algorithm. In our experiments, we also observed that the greedy algorithm takes a significant amount of time to run for large networks.

3.4. Influential nodes in dynamic networks

To find the most influential seed nodes in online social networks, we need to track the dynamic behavior of the networks. Here, we consider the sequence of the snapshot graphs $G_0, G_1, \ldots, G_T$ at time-stamp $t = 0, 1, 2, \ldots, T$, respectively. In this paper, we consider the dynamics of the graph in terms of edge change with time, i.e., an edge gets added in the graph if there occurs communication between a pair of nodes in the networks. Each snapshot graph shows an undirected graph termed as a **growing graph** and denoted as $G_t = (\mathbb{V}, E_t)$, where $\mathbb{V}$ is the set of nodes and E_t is the set of edges showing the nodes communication between time-stamp $t-1$ and t. A propagation probability $P^t_{u,v}$ is associated with each edge of every snapshot graph G_t.

As the selection of seed set S_t itself takes significant time for large graphs, it is possible that by the time S_t is computed, the graph might evolve from G_t to $G_{t+\delta}$, where we consider $computation\ time\ for\ S_t \leqslant \delta < snapshot\ interval$. So S_t may become less effective for influence maximization in the actual snapshot $\widehat{G}_{t+\delta}$. To handle this problem, we propose a novel approach in which we use a time series dependent link prediction method to predict G_{t+1} by considering the evolution pattern of the graph then we find the probable seed set S_{t+1} based on predicted G_{t+1}. And we use the predicted seed set S_{t+1} for influence maximization in the actual snapshot $\widehat{G}_{t+1}$ at time $t+1$.

Our goal is to predict graph G_{t+1} from graph snapshots up to G_t and find seed set S_{t+1} based on predicted G_{t+1}, which maximizes the influence function $\sigma_{t+1}(\cdot)$ at every snapshot $\widehat{G}_{t+1}$ at time $t+1$. We now formally define this Link Prediction based Influential Node Tracking (LPINT) problem in online social networks.

3.5. Problem definition

Link Prediction based Influential Nodes Tracking. Let $\rho = \{G_t\}_0^T$ be an online social network. The LPINT problem is to find a sequence of seed sets $S_1, \ldots, S_T$ whose size is maximum k, such that $S_t = \underset{S_t \in V, |S_t| \leqslant k}{\arg\max}\, \sigma_t(S_t)$ for all snapshot graphs $G_t, t = 1, \ldots, T$.

We propose a straightforward method to solve the LPINT problem. This method uses an efficient link prediction technique on the sliding window (a set snapshots) of snapshots of the dynamic graph to predict the next snapshot of the graph and then applies an effective Influence Maximization algorithm to find the predicted set of seed nodes in the predicted snapshot of the graph. This predicted seed set is used for influence maximization in the actual upcoming snapshot of the graph.

4. Proposed method

In an online social network, sudden changes in graph structure or topology within a short duration of time are likely. This change in the structure or topology of the graph could also lead to a change in seed sets. If the considered graph is highly dynamic, then the problem of finding a suitable seed set becomes more difficult. In many existing IM solutions, researchers have considered a snapshot of the graph at a fixed interval of time and finding the seed set on each snapshot. However, such techniques do not work well when the graph is rapidly changing.

In this paper, we proposed a novel LPINT algorithm for the IM problem in dynamic social networks. The proposed LPINT framework is explained by a block diagram in Fig. 3. Here, we consider the dynamic behavior of the graph in the past and depending on the time-series pattern of graph snapshots, we are predicting the next snapshot of a graph using the Link Prediction method. We

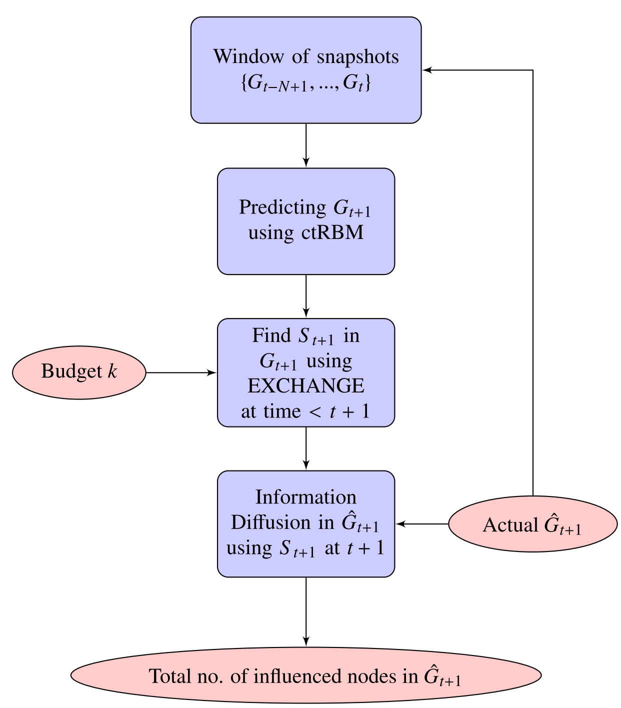

Fig. 3. Block diagram showing LPINT steps.

then apply the proposed IM algorithm to find the probable seed set for the upcoming graph snapshot. More detailed descriptions about how our method works on the snapshot graphs and dynamic networks are presented in the next two subsections.

4.1. Predicting G_{t+1} using link prediction

We predict the appearance of new links by exploring the evolution pattern of the graph. We have used ctRBM [54], which adopts temporal variations (temporal connections) and neighbor opinions (neighbor connections) during the training phase and performs prediction dependent on the existing time window of snapshots and the local neighbor's predictions of each pair of nodes. The model trains a ctRBM denoted as R_m for each node m, and then collects a set of ctRBMs denoted by $\mathbb{R}$. The prediction is therefore G_{t+1} done by aggregating the results from each R_m Fig. 4.

4.1.1. Temporal connections

Let N be the size of the window, which is a tunable parameter indicating the number of time steps (graph snapshots) we have to look back. In modeling a highly dynamic graph, N depends on the snapshots interval so as to better manage evolutionary networks. We assume that the graph snapshots at $t-N,\cdot,t-1$ is integrated into a vector $V^{<t}$ of dimension $N \cdot N_V$. The $N \cdot (N_V \times N_H)$ size weight matrix W_A provides the weights for temporal connections (see Fig. 4). Since the model has transitory information, the conditional probability at time t is represented as:

$$P(H^t|V^t, V^{<t};\theta) = \beta \cdot \omega\left(y + W_A' V^{<t}\right) + (1-\beta)\cdot\omega\left(y + W' V^t\right), \tag{4}$$

$$P\left(\widetilde{V}^t|H^t\right) = \omega(x + WH^t), \tag{5}$$

Here, weight matrix W_A is a parameter to model the temporal variations and β is a hyperparameter used to balance the static and dynamic characteristics of the graph Fig. 5.

4.1.2. Neighbor connections

We can explain the common assumption that an individual's conduct is influenced by his/her neighbor's circle by considering this. To formulate this, we define the neighbor impact as the desirability of its prediction (see Fig. 5). If the total number of nodes in $\mathbb{V}$ is $\hat{p}$, the neighbor impact can be written as:

$$\eta^t_m = \frac{1}{U^t_m}\sum_{n=1}^{\hat{p}} l(x^t_m, x^t_n) \times P\left(\widetilde{V}^t_n|H^t_n\right) \times P\left(H^t_n|V^t_n, V^{<t}_n;\theta_n\right), \tag{6}$$

Here, $U^t_m = \sum_{n=1}^{\hat{p}} l(x^t_m, x^t_n)$, and the indicator function l is 1 if node n is connecting to node m at time t, and 0 otherwise. And θ_n is the parameter of ctRBM model R_n for neighbor node n.

Neighbors make their predictions for node m based on their past. It can be seen from Eq. (6) that the opinion η^t_m of a node m is an average of its neighbor's opinions. Since models are already trained using last $t-1$ snapshots, $P\left(\widetilde{V}^t_n|H^t_n\right) \cdot P\left(H^t_n|V^t_n, V^{<t}_n;\theta_n\right)$ can be effectively computed by Eqs. (4) & (5) by substituting V^t in Eq. (4) by V^t_m.

4.1.3. Training and inferences on ctRBM

Following the standard for RBMs [57], the state of the hidden units is controlled by the contribution from individual perception V^t and $V^{<t}$ and the input η^t from neighbors. Given V^t and $V^{<t}$, the hidden units at time t are restrictively independent. The impact of the neighbor influence can be seen as adaptive bias:

$$\bar{x}^t = \gamma \cdot x + (1-\gamma)\cdot \eta^t \tag{7}$$

which includes the static bias, x for the current observation, and the contribution from the neighbors. Here γ is a hyperparameter which says how much an individual comply with his/her neighbors. In our experiments, we set it to be 0.5. Hence in Eq. (5) x is replaced with $\bar{x}^t$ to obtain:

$$P\left(\widetilde{V}^t|H^t\right) = \omega(\bar{x}^t + WH^t) \tag{8}$$

The detailed algorithm for inference used by ctRBM is shown in Algorithm 2.

Algorithm 2. PREDICT($\{G_t\}^t_{t-N+1}, \mathbb{R}$) [54]

Require: A trained $\mathbb{R}$ for all nodes, in which $R_m \in \mathbb{R}$ have parameters $\theta_m : \{W_{Am}, W_m, x_{Am}, x_m, y_m\}$, Snapshots $\{G_{t-N+1}, \ldots, G_t\}$, here N is the size of window
l**Output:** Predicted graph G_{t+1}
Initialize: $m \leftarrow 1, G_{t+1} \leftarrow zero(size(\mathbb{V}), size(\mathbb{V}))$
for $m < size(\mathbb{V}) + 1$ **do**
  $V^{<t+1}_m \leftarrow \{G^m_{t-N+1}, \ldots, G^m_t\}$
  $V^{t+1}_m \leftarrow one(1, size(\mathbb{V})).5$
  `Take neighbor indicator:`$Jdx \leftarrow find(G^m_t == 1)$
  `Take neighbor models detail:`$\mathbb{R}_{nbr} \leftarrow \mathbb{R}(Jdx)$
  `Determine` η^{t+1}_m `by Eq. (6) provided` $\mathbb{R}_{nbr}$
  $x^{t+1}_m \leftarrow x_m + \eta^{t+1}_m$
  `Determine` $\widetilde{V}^{t+1}_m$ `by Eq. (4) & (8) replacing` V^t `and` $V^{<t}$ `by` V^{t+1}_m `and` $V^{<t+1}_m$
    $G^m_{t+1} \leftarrow \widetilde{V}^{t+1}_m$
end for
return G_{t+1}.

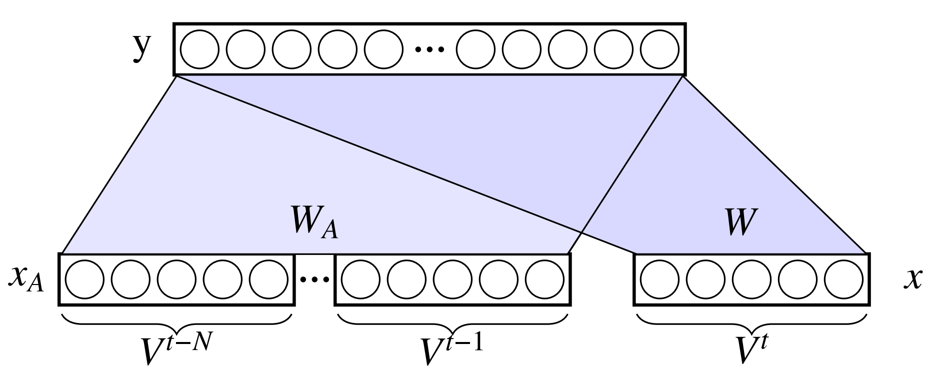

Fig. 4. Restricted Boltzmann Machine with temporal information, here the window size is N.

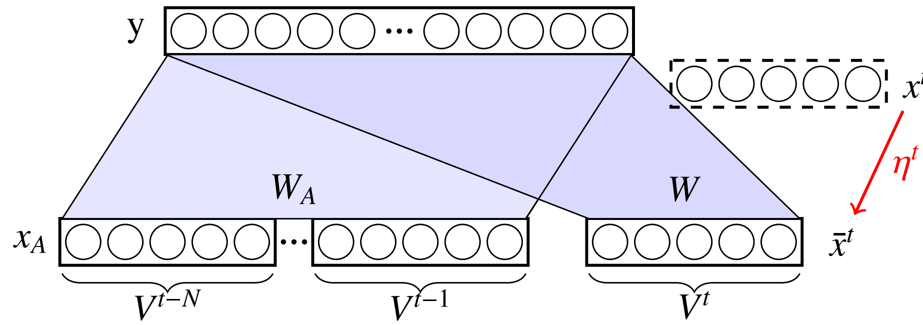

Fig. 5. A Conditional Restricted Boltzmann Machine with summarized neighbor influence η^t integrated into an adaptive bias into the energy function.

4.2. Finding seed nodes for influence maximization

The Interchange Heuristic proposed in [56] is utilized to replace the nodes in the seed set S_t. Beginning from a self-assertive set $S_t \subseteq \mathbb{V}$, Interchange Heuristic discovers a set $S_t' \subseteq \mathbb{V}$ that contains all nodes present in S_t except one node and the number of nodes remains the same. According to Nemhauser et al., when we apply the Interchange Heuristic on any submodular monotone function till maximum improvement gives a solution having an approximation guarantee of 1/2 [56].

Algorithm 3 describes how to get the set S_{t+1} using the Interchange Heuristic. We select S_{t+1} so that the gain accomplished by means of substitution of any fixed $e_s \in S_t$ to $e \in \mathbb{V} - S_t$ is maximized. We evaluate $e^* = \arg\max_{e \in \mathbb{V} - S_t} \Delta_{e,e_{s_t}}(S_t)$, by choosing $S_{t+1} = S_t - e_s + e^*$, here $\Delta_{e,e_{s_t}}(S_t)$ is replacing gain from node e_{s_t} to e. The upper bound [40] on the replacement gain is denoted by $\bar{\Delta}_{e,e_{s_t}}(S_t)$. We stop to find another e_{S_t} for interchange if the largest replacing gain $\Delta_{e,e_{s_t}}$ is less than a given threshold $\chi \geqslant 0$. We are doing this to increase the speed of the process of interchange and reduce the computations for the case of minor improvements.

We determine the gain by substituting e_s with any node in $e \in \mathbb{V} - S_t$, which requires $|\mathbb{V} - S_t|$ influence estimations. Monte-Carlo simulation-based calculation for the above task is excessively costly even for a network with a moderate size. To decrease the number of iterations for influence estimation, we use the upper bound on replacing gain as proposed in [40]. The subroutine in Algorithm 4 performs the Interchange Heuristic for any fixed $e_s \in S_t$.

Algorithm 3. EXCHANGE($G_{t+1}, S_t, e_s, \bar{\Delta}_{\cdot,e_s}(S_t)$)

Set $\Delta_{e,e_{s_t}} \leftarrow \bar{\Delta}_{e,e_{s_t}}(S_t)$, $e \in V - S_t$
Set $curr_e \leftarrow false$, $e \in V - S_t$
while *TRUE* **do**
  $e^* \leftarrow \arg\max_{e \in \mathbb{V} \setminus S_t} \left\{ \Delta_{e,e_{s_t}} \right\}$
  if $\Delta_{e^*,e_{s_t}} \leqslant \chi\sigma(S_t)$ **then**
  break
  end if
  if $curr_{e^*}$ **then**
  $S_t \leftarrow S_t - e_{s_t} + e^*$
   break
   else
   $\Delta_{e^*,e_{s_t}} \leftarrow \sigma(S_t - e_{s_t} + e^*) - \sigma(S_t)$
   $curr_{e^*} \leftarrow TRUE$
   end if
end while
$S_{t+1} = S_t$
return S_{t+1}.

4.3. Link prediction based influential node tracking

By using the strategy of interchange from Algorithm 3, we describe our Link Prediction based Influential Node Tracking, in short, LPINT as Algorithm 4. In Algorithm 4, when we start the process; firstly, we find the seed set in given snapshot G_1 by applying Algorithm 1, then we predict the upcoming graph snapshot using Algorithm 2, and then we apply Algorithm 3 to do at most k rounds of a replacement instead of doing it until no further improvement is possible, and finally, we got the fresh seed set S_{t+1} for snapshot G_{t+1}. Hence we ignore the insignificant performance improvement for reducing the computations and time of processing and hence increasing the efficiency of the overall process of seed set selection.

Algorithm 4. LPINT($\{G_t\}_{t-N+1}^{t}, k$)

Require: Snapshots of the graph $\{G_t\}_{t-N+1}^{t}$, size of seed set k
Output: Seed set S_{t+1} with k nodes for snapshot G_{t+1}
$S_1 = $ **GREEDY**$(G_1(\mathbb{V}, E_1), k)$
`Predicting` G_{t+1} `using Algorithm 2`
`For snapshot` G_{t+1} `compute` $\bar{\Delta}_{e,e_s}(S_t)$ `for` $e \in \mathbb{V} - S_t, e_s \in S_t$
for $i = 1$ `to` k **do**
  $e_s^* \leftarrow \arg\max_{e_s \in S_t} \{\bar{\Delta}_{\cdot,e_s}(S_t)\}$
  $S_t \leftarrow$ **EXCHANGE**$(G_{t+1}, S_t, e_s^*, \bar{\Delta}_{\cdot,e_s}(S_t))$
  `Update` $\bar{\Delta}_{e,e_s}(S_t)$ `for any` $e \in \mathbb{V} - S_t, e_s \in S_t$ `according to the` **EXCHANGE** `result`
end for
$S_{t+1} = S_t$
return S_{t+1}.

4.4. Theoretical results

We present some theoretical results on influence maximization for dynamic networks.

Theorem 1. *For growing graph $G(\mathbb{V}, E_t)$, snapshots G_t and G_{t+1} at timestamp t and $t+1 > t$, if $G_t \subseteq G_{t+1}$ and k size seed set $S_t, S_{t+1} \subseteq \mathbb{V}$ for graph G_t and G_{t+1} respectively, then the equation for the influence spread is related as:*

$$\arg\max_{|S_{t+1}|=k} \sigma(G_{t+1}, S_{t+1}) - \arg\max_{|S_t|=k} \sigma(G_t, S_t) \geqslant 0 \tag{9}$$

Proof. To establish this result, we need to consider Claim 2.3 proposed in [4]. According to the claim, a node x ends up being active if and only if there is a path from a node in S to node x composed only of live edges. We also have the following relation

$G_t \subseteq G_{t+1}$ implies$E_t \subseteq E_{t+1}$.

The active node estimator function $E[\text{Activenodes}]$ can be written as

$$E\left[\sum_{x\in\mathbb{V}} I_x\right]$$

where

$$I_x = \begin{cases} 1 & \text{if } node\ x \text{is active} \\ 0 & \text{otherwise.} \end{cases}$$

Thus,

$$E\left[\sum_{x\in\mathbb{V}} I_x\right] = \sum P_x.$$

We define $P_x^{G_T}$ as the probability of a path containing only live edges from x to S in G_T. Thus,

$$P_x^{G_t} \leqslant P_x^{G_{t+1}}.$$

In words, the probability of x being active in $G_t \leqslant$ probability of x being active in G_{t+1}.

Moreover, if G_t has m such live paths, then G_{t+1} also has at least m live paths because all the edges of G_t are in G_{t+1} but not in converse. Hence, we have:

$$\arg\max_{|S_t|=k} \sigma(G_t, S_t) \leqslant \arg\max_{|S_{t+1}|=k} \sigma(G_{t+1}, S_{t+1})$$

which concludes the proof. □

Theorem 2. *If the accuracy of link prediction is high, the predicted seed set S_{t+1} is nearer to the expected seed set $\widehat{S}_{t+1}$ and thus, the influence spread is high.*

Proof. Let the propagation probability for G_t and G_{t+1} is P_{uv}. Let the accuracy of the prediction of edges be ξ, then:

$$\frac{|E_{t+1} \cap \widehat{E}_{t+1}|}{|E_{t+1} \cup \widehat{E}_{t+1}|} \geqslant \xi. \tag{10}$$

Here $\widehat{E}_{t+1}$ is the actual number of edges at time $t+1$ and E_{t+1} is predicted edges at time $t+1$. We have for any graph G_{t+1},

$$|\arg\max_{\sigma}(G_{t+1}, S_{t+1})| \leqslant |\arg\max_{\sigma}\left(G_{t+1}, \widehat{S}_{t+1}\right)| \leqslant \lambda, \tag{11}$$

for some $\lambda > 0$. Now assume $P_{uv} = p$, by Eq. (10) we have:

$$\begin{aligned}
&|\arg\max_{\sigma}\left(\widehat{G}_{t+1}, \widehat{S}_{t+1}\right) - \arg\max_{\sigma}(G_{t+1}, S_{t+1})| \\
&\leqslant |\arg\max_{\sigma}\left(\left(\widehat{G}_{t+1} \cup G_{t+1}\right), \widehat{S}_{t+1}\right) \\
&\quad - \arg\max_{\sigma}\left(\left(\widehat{G}_{t+1} \cap G_{t+1}\right), S_{t+1}\right)| \\
&\leqslant |\arg\max_{\sigma}\left(\left(\widehat{G}_{t+1} \cup G_{t+1}\right), \widehat{S}_{t+1}\right) \\
&\quad - \arg\max_{\sigma}\left(\left(\widehat{G}_{t+1} \cup G_{t+1}\right), S_{t+1}\right) \cdot \xi|.
\end{aligned} \tag{12}$$

Combining Eqs. (11) and (12), we get

$$|\arg\max_{\sigma}\left(\widehat{G}_{t+1}, \widehat{S}_{t+1}\right)| - |\arg\max_{\sigma}(G_{t+1}, S_{t+1})| \leqslant \lambda(1-\xi).$$

Thus we can conclude if the prediction accuracy ξ is high the difference between influence spread in the predicted and actual graph is small. □

5. Experimental details

In this section, we give the details of the experiments conducted.

5.1. Datasets

We performed our experiments on four real-world dynamic networks: College, Mathoverflow, Askubuntu, and Wikitalk datasets. The datasets are available on the web at https://snap.stanford.edu/data/.

- The **College** [58] is a transient network dataset comprises of private messages sent on an online social network at the University of California, Irvine. Here, clients search the network for different clients, and after that, start a discussion based on profile data. An edge (p, q, t) signifies that client p sent a message to client q at time t.
- The **Mathoverflow** [59] is a temporal network of interactions on the stack-exchange web site mathoverflow. The data consists of a directed edge (p, q, t), where user p interacts with user q at time t.
- The **Ask-ubunbtu** dataset used in [59] is an online social network of interactions on the stack-exchange website askubuntu. This network data also consists of a directed edge (p, q, t), where user p interacts with user q at time t.
- The **Wiki-talk** dataset used in [60] is an online social network representing Wikipedia users editing each other's talk page. A directed edge (p, q, t) shows that user p edited user q's talk page at time t.

We generate the snapshot $G^t = (\mathbb{V}, E^t), V = \bigcup \mathbb{V}^t$ at timestamp t by considering all the edges appearing during the time $[t-\delta t, t]$, where δt is the time duration between two snapshots. The basic statistics of the dataset networks are given in Table 2. The dynamic behavior of the dataset is represented in the graphs shown in Fig. 6. It shows the graph plot of timestamp versus the number of communications for different datasets.

5.2. Baseline methods

Basic approaches used for comparison of influence maximization in evolving networks with and without link prediction are:

- **DegreeDiscount** [17]: A degree discount heuristic algorithm developed for the Independent Cascade model with uniform propagation probability.
- **LPDegreeDisount**: A degree discount heuristic algorithm implemented on the link prediction based predicted snapshot for seed selection.
- **PageRank** [61]: A link analysis algorithm that positions the priority of pages in a Web graph.
- **LPPageRank**: The PageRank algorithm for IM implemented on the link prediction based predicted snapshot for seed selection.
- **UBLF**: The Upper Bound based Lazy Forward (UBLF) algorithm [31] for IM derive an upper bound to reduce the number of spread estimations in the initialization step of influence maximization.

Table 2
Statistics of datasets.

Dataset	Nodes	Time span (days)
college	1899	193
mathoverflow	21,688	2350
ask-ubuntu	137,517	2613
wiki-talk	1,140,149	2320

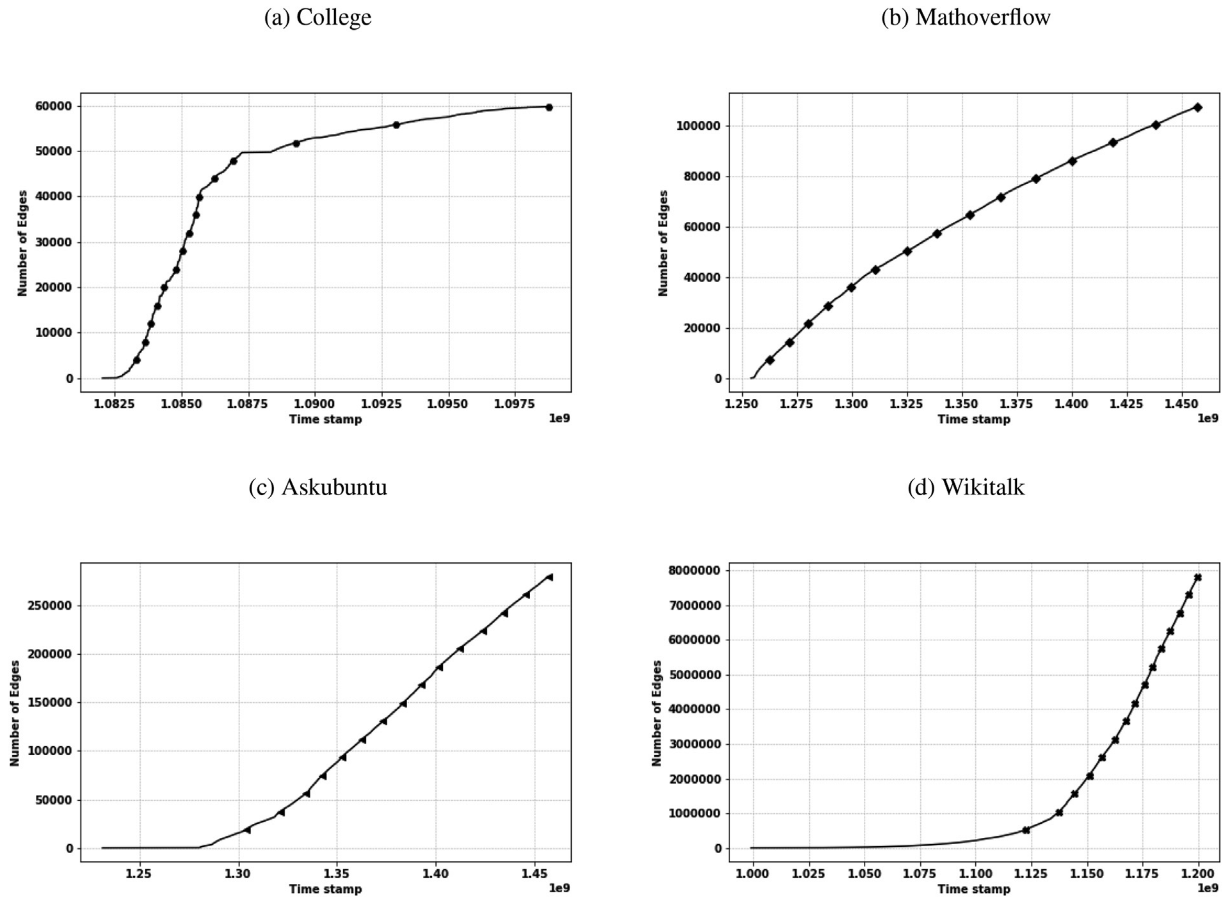

Fig. 6. Number of interactions versus time-stamp graph for datasets.

- **LUBLF**: The UBLF algorithm implemented on the link prediction based predicted snapshot for seed selection.l
- **EXCHANGE**: Our proposed upper bound based algorithm which computes the upper bound of marginal gain while evaluating the upper bound of node replacement gain.

Approaches used for comparison of IM with the proposed method are:

- **CIM** [62]: CIM (Continuous Influence Maximization) adopted the IC model as an influence model and pp=0.01 as propagation probability.
- **OIM** [63]: OIM (Online Influence Maximization) uses explore-exploit strategies for IM problems in dynamic networks. Here the propagation probability is taken as 0.01, and we have evaluated the result for varying seed sets using the greedy approach.
- **INT** [40]: INT (Influential node tracking) algorithm using for influence estimation with propagation probability 0.01. The initial seed set S_0 is set up by the Greedy algorithm. Then by using the UBI algorithm, we calculate an upper bound of marginal gain and an upper bound of node replacement gain.
- **LPINT**: Our LPINT algorithm using ctRBM for link prediction to predict the upcoming snapshot of the graph G^{t+1} and then uses the above IM algorithms for predicting the probable seed set S^{t+1} for efficient Influence Maximization in the dynamic social network.

5.3. Quality metric for influence spread

To demonstrate the effectiveness of our proposed solution, we find the seed sets produced by all the strategies described above for every window shift. When a seed set S_{t+1} is returned by an algorithm, we evaluate the influence spread under the IC model by 10,000 rounds of Monte-Carlo simulation on the current snapshot of the graph, i.e., $\widehat{G}_{t+1}$. Note that we assume that the graph is highly dynamic, so it evolves quickly after the selection of seed sets. Finally, we take the average influence spread over all windows as the quality metric to compare different approaches.

5.4. Experimental settings

All experiments are conducted on a server machine running CentOS-7 with a Quad-Core 2.1 GHz Intel Xeon Silver 4110 processor and 64 GB memory. All the algorithms were implemented in python.

For the information diffusion process, we use Independent Cascade model. In this model, most of the literature [17] have done experiments using a small propagation probability of $p = 0.01$. Larger p values such as $p = 0.1$ are not considered due to insensitivity to different algorithms. In this paper, we use $p = 0.01$ as the value of propagation probability. In the first step, the dataset is divided into different snapshots. We have divided the dataset according to a fixed timeframe, which is different for the different datasets. For all datasets, we have divided them into $T = 25$ snapshots. Now, we have $[G_1, G_2, \ldots c, G_{25}]$ snapshots of the graph. We use $N = 10$ as the size of the window; it means there are 10 snapshots in each window. Here, we use the first 10 snapshots for training the ctRBM using Algorithm 2 for link prediction task and then predict the newly arrived edges in newly added snapshot after window shift of 1 snapshot and compare it with original edges at latest snapshot for testing and repeat the same process for each window shift. For influence maximization, we find the seed node set by applying Algorithm 3 on the predicted snapshot graph and use this

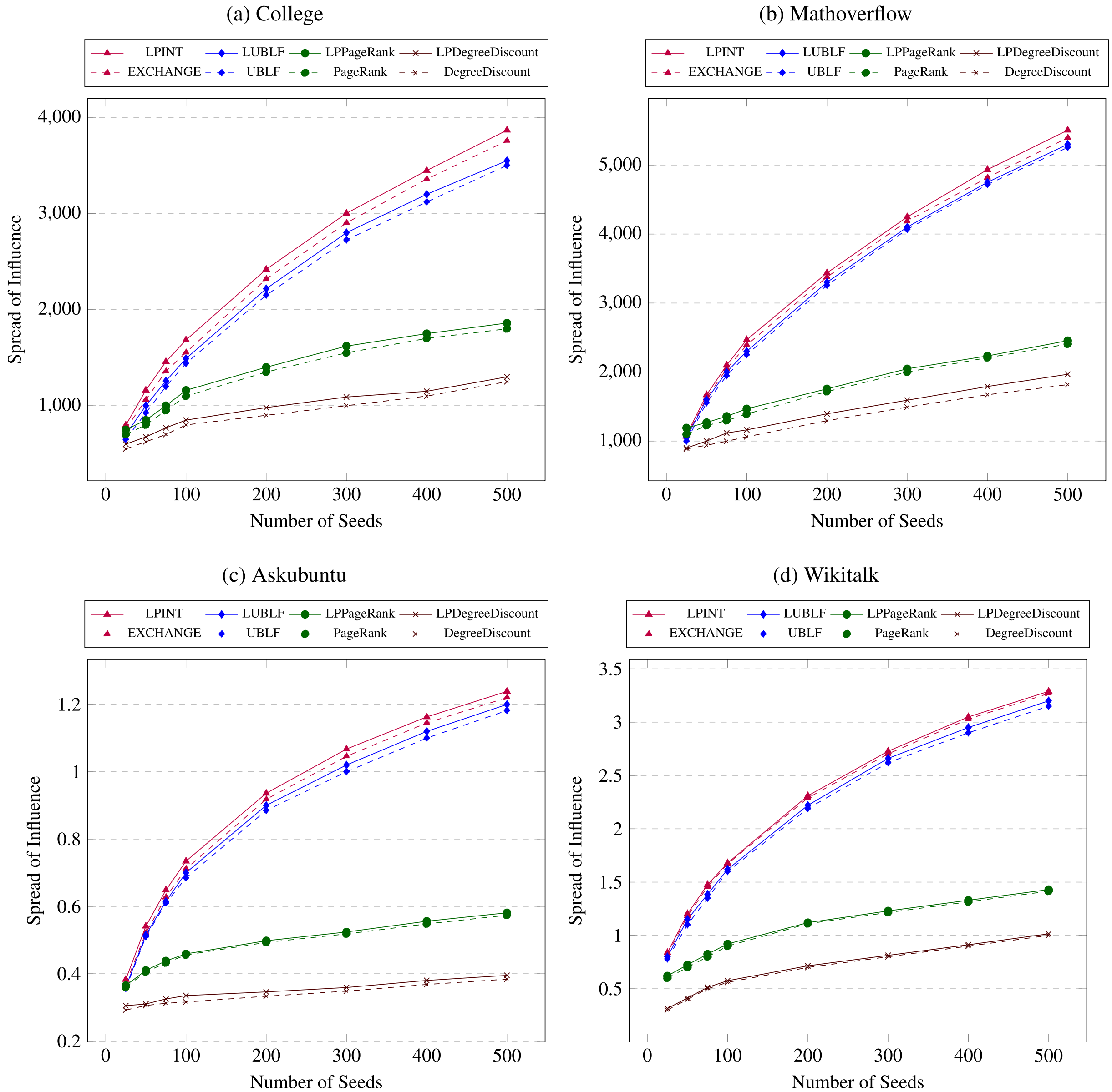

Fig. 7. Seed vs Influence spread for different static IM technique with and without link prediction on the snapshot of different dynamic networks.

seed set for evaluating the influence spread on the newly added snapshot graph. In our experiments, we are varying the size of seed set k represented on the x-axis, and the y-axis represents the information spread in the target snapshot of the graph. In our experiments, we predict 5 percent links, which may appear in the upcoming snapshot of the graph. We use the average of 10,000 rounds of Monte-Carlo simulations to estimate the actual influence spread and thereby assess the seed set found by the algorithms.

6. Results and analysis

6.1. Comparing different approaches with and without using link prediction

The results in Fig. 7 shows the influence spread for different datasets against varying seed nodes with and without link prediction technique for the dynamic network. The seed set size value varies as $\{25, 50, 75, 100, 200, 300, 400, 500\}$. Here, we compared different static IM techniques applied with and without link prediction technique for IM in dynamic social networks. Here we use the link prediction technique for predicting the next snapshot. Then, we find the seed nodes in predicted snapshot using different static IM techniques. Further, we find the influence spread using those seed nodes in the actual snapshot. In this way, we get better results in terms of influence spread for all the considered algorithms when link prediction technique used. We see that the EXCHANGE algorithm gives an improvement in influence spread as compared with other baseline IM algorithms, and it becomes even better when used with link prediction as LPINT.

6.2. Comparing dynamic approaches with our proposed LPINT algorithm

We compare our proposed LPINT model for influence maximization with existing techniques for influence maximization in dynamic networks. The results are presented in graphs shown in Fig. 8, here we have shown the influence spread by varying the seed set size k on the target snapshot of the graph. We can see that the proposed LPINT algorithm outperforms other considered algorithms in terms of influence spread.

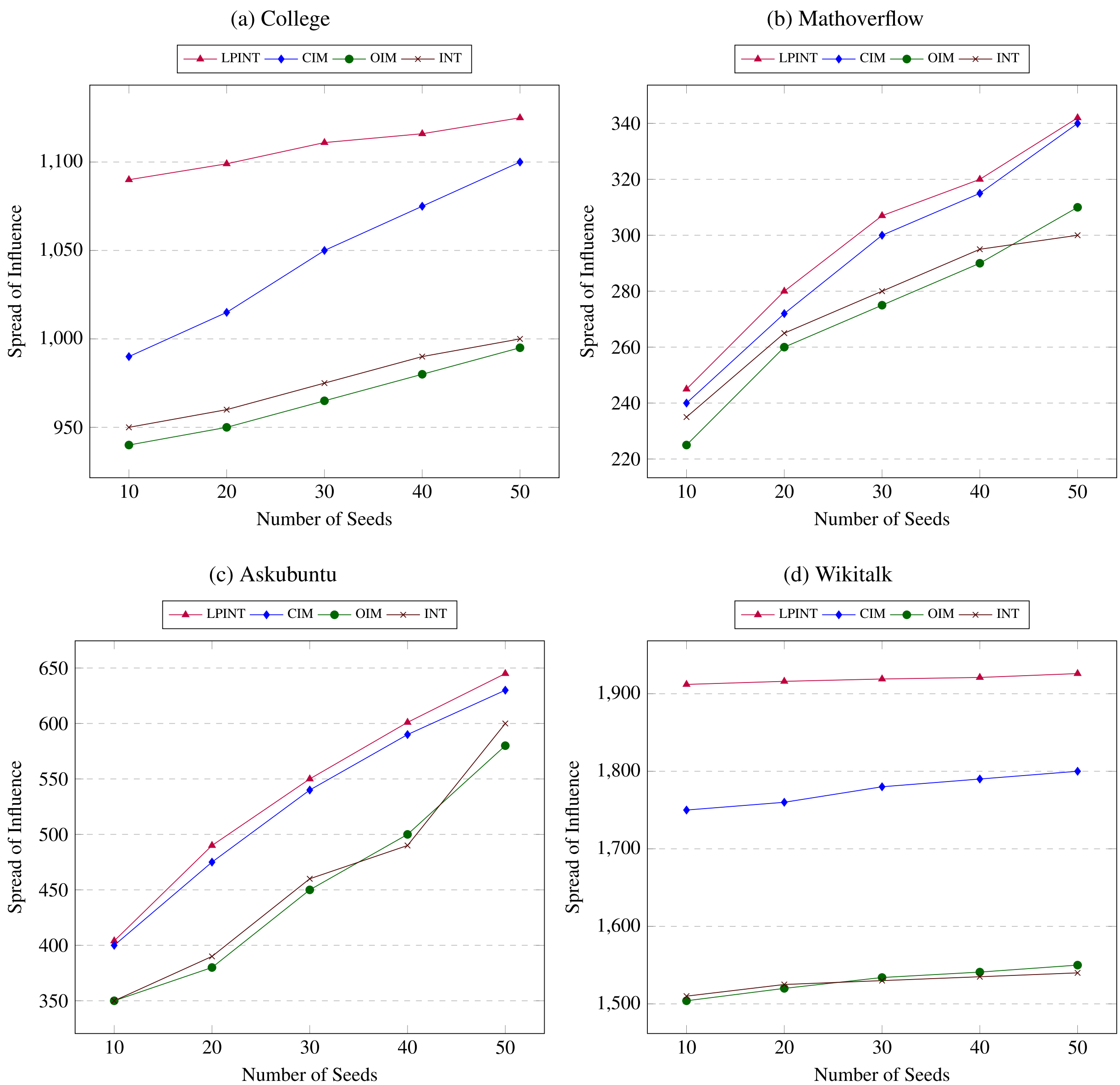

Fig. 8. Comparison of different online IM techniques with LPINT on the snapshot of different dynamic networks by showing seed vs influence spread.

Table 3
Average running time for influence spread.

Datasets	Methods			
	INT	OIM	CIM	LPINT
college	46 ms	40 ms	32 ms	21 ms
mathoverflow	38 ms	33 ms	28 ms	24 ms
ask-ubuntu	1.1 s	59 ms	53 ms	40 ms
wiki-talk	1.6 s	1.2 s	1.0 s	59 ms

6.3. Comparison of average running time for influence spread

In Table 3, we can see the average running time of influence spread for a snapshot graph using benchmark algorithms and LPINT method. Notice that LPINT performs significantly better in terms of time required for influence spread. The reason for this faster influence spread is the selection of better seed nodes. If we choose more effective seed nodes, it takes less time for influence spread as it requires fewer iterations to complete the influence spread process using the Independent Cascade model. This lower time for influence spread in LPINT again confirms its efficiency over benchmark methods.

6.4. Insightful discussion

In our proposed influence maximization algorithm for the dynamic social network, we show the improvement in results in terms of influence spread experimentally and theoretically. In our proposed work, once the behavior of nodes for making the new links are learned, the prediction of the upcoming snapshot becomes efficient and effective. At each snapshot, there is no need to explore all the nodes to find suitable seed nodes. Efficient seed nodes reduce the number of iteration in the IC model for influence spread and hence take less time for information spread as compared to other considered baseline algorithms.

The limitation of our proposed method includes the overhead of prediction of the upcoming snapshot; however, with the increase of time system learns for efficient prediction. Here, we have not considered the situation where any node behaves randomly, although it is also not considered by the baseline algorithms. Our proposed model can also be implemented with other diffusion models, which is not explored here.

7. Conclusion and future works

In this paper, we proposed a link prediction based influential node tracking method to find seed nodes for information spread in the dynamic social network. We use the ctRBM based deep learning technique for link prediction to predict the next snapshot of the graph. We then find the seed set in the predicted snapshot using the EXCHANGE algorithm. This seed set is used for actual influence spread in the real snapshot of the graph. This method improves the influence spread in terms of the number of influenced nodes in highly dynamic social networks. Extensive experiments on four real social networks demonstrate that our method outperforms the baselines in terms of influence coverage and influence spread time. In the future, we plan to generalize our LPINT algorithm for other diffusion models under dynamic networks. Making this process online instead of executing it on snapshots of the graph is another possible future work.

CRediT authorship contribution statement

Ashwini Kumar Singh: Conceptualization, Data curation, Investigation, Methodology, Resources, Software, Validation, Visualization, Writing - original draft, Writing - review & editing. **Lakshmanan Kailasam:** Conceptualization, Resources, Supervision, Validation, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] K. Klemm, M.Á. Serrano, V.M. Eguíluz, M. San Miguel, A measure of individual role in collective dynamics, Sci. Rep. 2 (2012) 1–8.
[2] W. Chen, C. Wang, 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, KDD '10, New York, NY, USA, 2010, pp. 1029–1038.
[3] J.A. Chevalier, D. Mayzlin, The effect of word of mouth on sales: Online book reviews, J. Market. Res. 43 (2006) 345–354.
[4] D. Kempe, J. Kleinberg, 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, ACM, 2003, pp. 137–146.
[5] J. Leskovec, Social media analytics: tracking, modeling and predicting the flow of information through networks, in: Proceedings of the 20th international conference companion on World wide web, ACM, 2011, pp. 277–278.
[6] R. Lemonnier, K. Scaman, N. Vayatis, Tight bounds for influence in diffusion networks and application to bond percolation and epidemiology, in: Advances in Neural, Information Processing Systems (2014) 846–854.
[7] V. Mahajan, E. Muller, F.M. Bass, New product diffusion models in marketing: A review and directions for research, J. Market. 54 (1990) 1–26.
[8] P. Domingos, M. Richardson, Mining the network value of customers, in: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2001, pp. 57–66.
[9] J. Goldenberg, B. Libai, E. Muller, Using complex systems analysis to advance marketing theory development: Modeling heterogeneity effects on new product growth through stochastic cellular automata, Acad. Market. Sci. Rev. 9 (2001) 1–18.
[10] J. Tang, X. Tang, J. Yuan, Profit maximization for viral marketing in online social networks: Algorithms and analysis, IEEE Trans. Knowl. Data Eng. 30 (2018) 1095–1108.
[11] R.M. Bond, C.J. Fariss, J.J. Jones, A.D. Kramer, C. Marlow, J.E. Settle, J.H. Fowler, A 61-million-person experiment in social influence and political mobilization, Nature 489 (2012) 295.
[12] L. Fan, Z. Lu, W. Wu, B. Thuraisingham, H. Ma, Y. Bi, Least cost rumor blocking in social networks, in: 2013 IEEE 33rd International Conference on Distributed Computing Systems, IEEE, pp. 540–549.
[13] N.P. Nguyen, G. Yan, M.T. Thai, S. Eidenbenz, Containment of misinformation spread in online social networks, in: Proceedings of the 4th Annual ACM Web Science Conference, ACM, 2012, pp. 213–222.
[14] K. Saito, R. Nakano, M. Kimura, Prediction of information diffusion probabilities for independent cascade model, in: International conference on knowledge-based and intelligent information and engineering systems, Springer, 2008, pp. 67–75.
[15] S. Galhotra, A. Arora, S. Virinchi, S. Roy, Asim: A scalable algorithm for influence maximization under the independent cascade model, World Wide Web, ACM, 2015, pp. 35–36.
[16] N. Barbieri, F. Bonchi, G. Manco, Topic-aware social influence propagation models, Knowled. Inf. Syst. 37 (2013) 555–584.
[17] W. Chen, Y. Wang, S. Yang, Efficient influence maximization in social networks, in: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2009, pp. 199–208.
[18] M. Kimura, K. Saito, Tractable models for information diffusion in social networks, in: European conference on principles of data mining and knowledge discovery, Springer, 2006, pp. 259–271.
[19] Y. Wang, G. Cong, G. Song, K. Xie, Community-based greedy algorithm for mining top-k influential nodes in mobile social networks, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '10, New York, NY, USA, 2010, pp. 1039–1048.
[20] W. Chen, W. Lu, N. Zhang, Time-critical influence maximization in social networks with time-delayed diffusion process, in: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, AAAI Press, 2012, pp. 592–598.
[21] W. Chen, Y. Yuan, L. Zhang, Scalable influence maximization in social networks under the linear threshold model, in: Proceedings of the 2010 IEEE International Conference on Data Mining, IEEE Computer Society, Washington, DC, USA, 2010, pp. 88–97.
[22] N. Du, L. Song, M. Gomez-Rodriguez, H. Zha, Scalable influence estimation in continuous-time diffusion networks, in: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13, Curran Associates Inc., USA, 2013, pp. 3147–3155.
[23] J. Leskovec, J. Kleinberg, C. Faloutsos, Graphs over time: Densification laws, shrinking diameters and possible explanations, in: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, KDD '05, ACM, New York, NY, USA, 2005, pp. 177–187.
[24] J. Leskovec, J. Kleinberg, C. Faloutsos, Graph evolution: Densification and shrinking diameters, ACM Trans. Knowl. Discov. Data 1 (2007).
[25] J. Leskovec, L. Backstrom, R. Kumar, A. Tomkins, Microscopic evolution of social networks, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '08, New York, NY, USA, 2008, pp. 462–470.
[26] S.G. Mazman, Y.K. Usluel, Modeling educational usage of facebook, Comput. Educ. 55 (2010) 444–453.
[27] D. Murthy, Twitter, Polity Press Cambridge, UK, 2018.
[28] N. Ohsaka, T. Akiba, Y. Yoshida, K.-I. Kawarabayashi, Dynamic influence analysis in evolving networks, Proc. VLDB Endowment 9 (2016) 1077–1088.
[29] N.T. Gayraud, E. Pitoura, P. Tsaparas, Diffusion maximization in evolving social networks, in: Proceedings of the 2015 acm on conference on online social networks, ACM, pp. 125–135.
[30] G. Tong, W. Wu, S. Tang, D.-Z. Du, Adaptive influence maximization in dynamic social networks, IEEE/ACM Trans. Netw. (TON) 25 (2017) 112–125.
[31] C. Zhou, P. Zhang, J. Guo, X. Zhu, L. Guo, Ublf: An upper bound based approach to discover influential nodes in social networks, in: 2013 IEEE 13th International Conference on Data Mining, IEEE, pp. 907–916.
[32] Y. Li, J. Fan, Y. Wang, K.-L. Tan, Influence maximization on social graphs: A survey, IEEE Trans. Knowl. Data Eng. 30 (2018) 1852–1872.
[33] M. Li, X. Wang, K. Gao, S. Zhang, A survey on information diffusion in online social networks: Models and methods, Information 8 (2017) 118.
[34] R. Rabade, N. Mishra, S. Sharma, Survey of influential user identification techniques in online social networks, in: Recent advances in intelligent informatics, Springer, 2014, pp. 359–370.
[35] A. Guille, H. Hacid, C. Favre, D.A. Zighed, Information diffusion in online social networks: A survey, ACM Sigmod Record 42 (2013) 17–28.
[36] J. Sun, J. Tang, A survey of models and algorithms for social influence analysis, in: Social network data analytics, Springer, 2011, pp. 177–214.
[37] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, N. Glance, Cost-effective outbreak detection in networks, in: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2007, pp. 420–429.
[38] C.C. Aggarwal, S. Lin, P.S. Yu, On influential node discovery in dynamic social networks, in: Proceedings of the 2012 SIAM International Conference on Data Mining, SIAM, pp. 636–647.

[39] H. Zhuang, Y. Sun, J. Tang, J. Zhang, X. Sun, Influence maximization in dynamic social networks, in: 2013 IEEE 13th International Conference on Data Mining, IEEE, pp. 1313–1318.
[40] G. Song, Y. Li, X. Chen, X. He, J. Tang, Influential node tracking on dynamic social network: an interchange greedy approach, IEEE Trans. Knowl. Data Eng. 29 (2017) 359–372.
[41] C. Wang, J. Tang, J. Sun, J. Han, Dynamic social influence analysis through time-dependent factor graphs, in: 2011 International Conference on Advances in Social Networks Analysis and Mining, IEEE, pp. 239–246.
[42] M.G. Rodriguez, B. Schölkopf, Influence maximization in continuous time diffusion networks, arXiv preprint arXiv:1205.1682 (2012).
[43] H. Li, S.S. Bhowmick, A. Sun, J. Cui, Conformity-aware influence maximization in online social networks, The VLDB Journal-The International Journal on Very Large Data, Bases 24 (2015) 117–141.
[44] M. Han, M. Yan, Z. Cai, Y. Li, X. Cai, J. Yu, Influence maximization by probing partial communities in dynamic online social networks, Transactions on Emerging Telecommunications Technologies 28 (2017) e3054.
[45] Y. Wang, Q. Fan, Y. Li, K.-L. Tan, Real-time influence maximization on dynamic social streams, Proc. VLDB Endowment 10 (2017) 805–816.
[46] L.A. Adamic, E. Adar, Friends and neighbors on the web, Soc. Netw. 25 (2003) 211–230.
[47] D. Liben-Nowell, J. Kleinberg, The link-prediction problem for social networks, J. Am. Soc. Inf. Sci. Technol. 58 (2007) 1019–1031.
[48] T. Tylenda, R. Angelova, S. Bedathur, Towards time-aware link prediction in evolving social networks, in, in: Proceedings of the 3rd workshop on social network mining and analysis, ACM, 2009, pp. 1–10.
[49] D.Q. Vu, A.U. Asuncion, D. Hunter, P. Smyth, Continuous-time regression models for longitudinal networks, Advances in neural information processing systems 24 (2011).
[50] L.E. Baum, T. Petrie, Statistical inference for probabilistic functions of finite state markov chains, Ann. Math. Stat. 37 (1966) 1554–1563.
[51] Z. Ghahramani, G.E. Hinton, Variational learning for switching state-space models, Neu. Comput. 12 (2000) 831–864.
[52] P. Sarkar, D. Chakrabarti, M. Jordan, Nonparametric link prediction in dynamic networks, arXiv preprint arXiv:1206.6394 (2012).
[53] U. Sharan, J. Neville, Temporal-relational classifiers for prediction in evolving domains, in: 2008 Eighth IEEE International Conference on Data Mining, IEEE, pp. 540–549.
[54] X. Li, N. Du, H. Li, K. Li, J. Gao, A. Zhang, A deep learning approach to link prediction in dynamic networks, in: Proceedings of the 2014 SIAM International Conference on Data Mining, SIAM, pp. 289–297.
[55] Z. Huang, D.K. Lin, The time-series link prediction problem with applications in communication surveillance, INFORMS J. Comput. 21 (2009) 286–303.
[56] G.L. Nemhauser, L.A. Wolsey, M.L. Fisher, An analysis of approximations for maximizing submodular set functions–i, Math. Program. 14 (1978) 265–294.
[57] G.E. Hinton, Training products of experts by minimizing contrastive divergence, Neu. Comput. 14 (2002) 1771–1800.
[58] P. Panzarasa, T. Opsahl, K.M. Carley, Patterns and dynamics of users' behavior and interaction: Network analysis of an online community, J. Am. Soc. Inform. Sci. Technol. 60 (2009) 911–932.
[59] A. Paranjape, A.R. Benson, J. Leskovec, Motifs in temporal networks, in: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, ACM, 2017, pp. 601–610.
[60] J. Leskovec, D. Huttenlocher, J. Kleinberg, Governance in social media: A case study of the wikipedia promotion process, in: Fourth International AAAI Conference on Weblogs and Social Media, 2010.
[61] S. Brin, L. Page, The anatomy of a large-scale hypertextual web search engine, Comp. Netw. ISDN Syst. 30 (1998) 107–117.
[62] Y. Yang, X. Mao, J. Pei, X. He, Continuous influence maximization: What discounts should we offer to social network users? in: Proceedings of the 2016 international conference on management of data, ACM, pp. 727–741.
[63] S. Lei, S. Maniu, L. Mo, R. Cheng, P. Senellart, Online influence maximization, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 645–654.

Ashwini Kumar Singh is working toward the Ph.D. degree in the department of Computer Science and Engineering, Indian Institute of Technology (BHU), Varanasi. His research interests lie in Network Science and Applications of machine learning and data mining in Social Networks.

Lakshmanan Kailasam received his Ph.D. degree from the Indian Institute of Science, Bangalore in 2013. He has held positions of postdoctoral fellow at IIT Bombay, University of Leoben, Austria and National University of Singapore. He is an assistant professor in the department of Computer Science and Engineering, Indian Institute of Technology (BHU), Varanasi. His research interests include Reinforcement Learning, Simulation Optimization and Network Science.