

Article

Influence Maximization Based on Snapshot Prediction in Dynamic Online Social Networks

Lin Zhang and Kan Li *

School of Computer Science, Beijing Institute of Technology, Beijing 100081, China; bitszhl@bit.edu.cn

* Correspondence: likan@bit.edu.cn

Abstract: With the vigorous development of the mobile Internet, online social networks have greatly changed the way of life of human beings. As an important branch of online social network research, influence maximization refers to finding K nodes in the network to form the most influential seed set, which is an abstract model of viral marketing. Most of the current research is based on static network structures, ignoring the important feature of network structures changing with time, which discounts the effect of seed nodes in dynamic online social networks. To address this problem in dynamic online social networks, we propose a novel framework called Influence Maximization based on Prediction and Replacement (IMPR). This framework first uses historical network snapshot information to predict the upcoming network snapshot and then mines seed nodes suitable for the dynamic network based on the predicted result. To improve the computational efficiency, the framework also adopts a fast replacement algorithm to solve the seed nodes between different snapshots. The scheme we adopted exhibits four advantages. First, we extended the classic influence maximization problem to dynamic online social networks and give a formal definition of the problem. Second, a new framework was proposed for this problem and a proof of the solution is given in theory. Third, other classical algorithms for influence maximization can be embedded into our framework to improve accuracy. More importantly, to reveal the performance of the scheme, a series of experiments based on different settings on real dynamic online social network datasets were carried out, and the experimental results are very promising.

Keywords: dynamic replacement; dynamic online social networks; influence maximization

MSC: 05C90



Citation: Zhang, L.; Li, K. Influence Maximization Based on Snapshot Prediction in Dynamic Online Social Networks. *Mathematics* **2022**, *10*, 1341. <https://doi.org/10.3390/math10081341>

Academic Editor: Sergio Gómez

Received: 18 March 2022

Accepted: 14 April 2022

Published: 18 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the popularization of the mobile Internet and the vigorous development of new media, online social networks have changed many aspects of human daily life. People can carry out a series of activities in online social networks, such as sharing ideas, communicating, receiving news, establishing friendships, and so on. Mass users and real-time information spreading make online social networks a new carrier of information diffusion. More and more companies are beginning to use online social networks to market their products. This trend has attracted the interest of researchers in many different areas. Understanding the information diffusion process in social networks is beneficial to reveal the structure of human society and influence the strategies for marketing products.

Viral marketing based on the word-of-mouth effect is an important application of online social networks. This marketing pattern can be abstractly described as an influence maximization problem, which is an indispensable branch of social network analysis [1]. The problem of influence maximization is to select a small group of seed nodes in an online social network to maximize their influence on other nodes in the network. It is proved that influence maximization is an NP-hard problem under the linear threshold and independent cascade model.

There has been a lot of research around influence maximization. In these studies, an online social network is usually regarded as a graph in which nodes represent users and edges represent the relationships between users. The researchers analyze the process of influence diffusion based on the graphs and then use the greedy algorithm or heuristic algorithm to find the most influential seed set. However, we found that the online social network structure generally remained unchanged in their study. In reality, the structure of online social networks is constantly changing over time, which is an important characteristic of online social networks. For example, in Twitter, a user follows a singer, and after a while, he may no longer like the singer, unfollows the singer, and then follows another singer. Once the network structure changes, the influence of users also changes, and individuals are more inclined to be influenced by people who are closely related to them. Therefore, using static social networks to study influence maximization in dynamic social networks eventually leads to finding suboptimal seeds. Some studies have considered the dynamic characteristics of networks, but none of them have perfectly solved the problems of performance and efficiency.

In dynamic networks, snapshots can be used to record the topology of the network at different times. To select the most influential seed node in the whole process, we need to find the optimal solution in different snapshots. This is because as the network structure changes, the influence of seed nodes also change. To facilitate understanding of the problem in dynamic social networks, we illustrate this concept with an example. Figure 1 shows snapshots of an online social network at different time stamps. G_i represents the snapshot at time $t = i$. This network contains 4 users, the connections between users are represented by edges, and the two connected users can influence each other. It is easy to find that the network structure has changed over time. At $t = 0$, the most influential user is v_1 , with the dynamic change of the network structure, the most influential user becomes v_4 at $t = 2$, and becomes v_2 at $t = 3$. This shows the importance of dynamic changes to the network. Therefore, to select the most influential seed set in a dynamic social network, we need to mine seed nodes from each snapshot. Ignoring changes between network snapshots may lead to poor results. In response to this problem, we propose a new framework called Influence Maximization based on Prediction and Replacement (IMPR). First, predict the upcoming network topology based on the previous network snapshots, and then use the prediction result to mine the seed nodes. For example, in Figure 1, G_0 and G_1 are used to predict G_2 , and the prediction result is G' , and the seed node at time $t = 2$ is calculated according to G' . In addition, to improve the computational efficiency, we adopted a fast replacement algorithm to mine the seed set under the new snapshot.

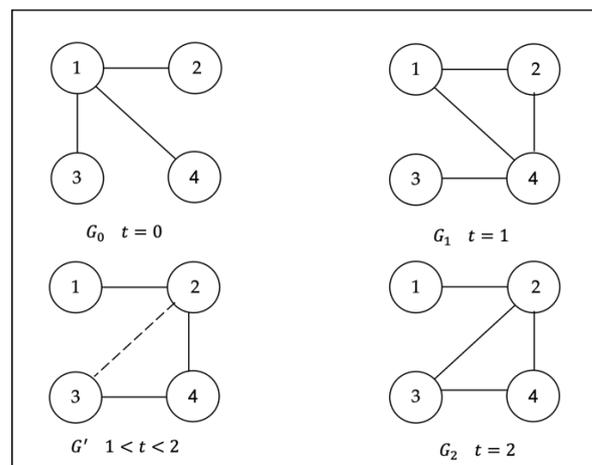


Figure 1. The snapshots of a dynamic online social network at different time stamps.

In short, the contributions of this paper are fourfold. First, we extended the classic influence maximization problem to dynamic online social networks and give a formal

definition of the problem. Second, a new framework was proposed for this problem and a proof of the solution is given theoretically. Third, the accuracy of traditional methods can be improved based on our proposed framework. Finally, a series of experiments with different specifications and settings were conducted on real dynamic online social network datasets to examine the advantages of the framework, which prove to be very promising.

The organization of this paper is as follows. We summarize the literature related to influence maximization in dynamic online social networks in Section 2. In Section 3, we give a formal definition of the problem and introduce the proposed framework in detail. In Section 4, we conduct a series of experiments based on real online social network data sets to reveal the performance of our framework. Finally, conclusions and future work are presented in Section 5.

2. Related Work

The study of influence maximization was first proposed in 2001 by Domingos and Richardson [2]. Based on this research, Kempe et al. [3] defined the problem as a discrete optimization problem, which was a milestone for influence maximization research. They defined the problem as mining K seed nodes that maximize the spread of influence in an online social network based on a given diffusion model. In addition to this, they also proved that the influence maximization problem is an NP-hard problem when the given information propagation model is an Independent Cascade or a Linear Threshold model. For the solution of the problem, they proposed a greedy algorithm that can guarantee the approximate optimality of $1 - 1/e - \epsilon$ [4]. Sviridenko [5] extended this greedy framework with a non-uniform cost function. Since the greedy algorithm involves a large number of Monte Carlo simulations, to reduce the computational complexity, many researchers have improved it. Leskovec et al. [6] and Goyal et al. [7] proposed Cost Effective Lazy Forward schema (CELF) and CELF++, respectively, using the sub-mode attribute of the influence function to reduce the number of Monte Carlo simulations for each seed node selection. Estevez et al. [8] discarded the overlapping part with the neighbors of the seed node when selecting the seed node, and this method is called the Set Covering Greedy algorithm (SCG). Chen et al. [9] removed those edges that could not successfully propagate information in the iterative process and proposed a new algorithm called New Greedy-IC. Following these, Zhou et al. [10] found the upper limit of the marginal benefit of node influence diffusion in the influence function and proposed an Upper Bound based Lazy Forward algorithm (UBLF). UBLF shortens the computation time and achieves similar accuracy to the greedy algorithm. In addition, the improved methods based on greedy algorithm include cascade discount algorithm (CD), influence maximization based on learning automata algorithm (IMLA), hybrid potential-influence greedy algorithm (HPG), and so on [11–13].

Although the greedy algorithm is very accurate, computational complexity remains a huge challenge when the network scales up. This is because Monte Carlo simulation is a time-consuming operation. Therefore, some researchers start to use heuristic algorithms to solve this problem. Chen et al. [9] studied the relationship between node influence and node degree and proposed the Degree-Discount algorithm. This algorithm greatly reduces the computational time complexity but sacrifices some accuracy. Khomanmi et al. [14] considered the influence of community structure on propagation and proposed a fast and scalable algorithm, called Community Finding Influential Node (CFIN). Kundu et al. [15] proposed the diffusion degree of a node, which is used to represent the influence of a node on other nodes. They use this centrality measure to select seed nodes. Kim et al. [16] proposed the Independent Path Algorithm (IPA) by using the independent influence paths to evaluate the influence of nodes. Apart from this, there are other heuristics based on influence paths, such as Influence Maximization Shortest Path (IMSP) [17], SIMPATH [18], and LDAG [19]. To support computation on large-scale networks, Tang et al. [20] proposed the TIM algorithm. TIM can significantly increase computation speed without compromising performance. Furthermore, there are many other heuristics CGA [21], ACO-IM [22], IRIE [23], and others [24,25].

Classical influence maximization algorithms are mainly divided into two categories: greedy algorithms and heuristic algorithms. The greedy algorithm has high precision but is computationally time-consuming, while the heuristic algorithm is efficient but sacrifices some precision. Most importantly, these studies are based on static networks.

Recently, some researchers have begun to devote themselves to the study of the influence maximization in dynamic networks. Currently these studies can be divided into two categories. The first one mainly considers the dynamics in the process of information dissemination, such as dynamic activation probability, dynamic threshold, and dynamic perception. The second is to consider the dynamic of the network topology, where edges are added or removed over time. Hao et al. [26] considered the dynamic changes in the propagation process, and proposed two models to solve the influence maximize in dynamic networks. The activation probability between two individuals in the first model depends on previous activation trials. The second is the dynamic variable threshold model, which argues that an individual's activation threshold can change according to an individual's attitude toward information. Considering user preferences and social influence, Teng et al. [27] used the knowledge graph to capture the dynamic perception of users, proposed a new problem of maximizing influence based on dynamic personal perception, and gave an approximate solution. Ge et al. [28] considered the dynamic changes of user interests in online social networks. Additionally, Li et al. [29] explored the dynamics of propagation and the influence of local aggregation factors on influence diffusion, and proposed a dynamic influence maximization algorithm based on cohesive entropy. This type of research [26–31] focuses on the dynamics of propagation. The influence between individuals in the propagation process is dynamically variable, but the network topology remains fixed.

This paper focuses on the influence maximization problem when the network topology changes dynamically. In response to this situation, to quantify the influence between two nodes in a dynamic network, Wang et al. [32] proposed a dynamic factor graph model (DFG) to calculate the dynamic influence of nodes. Agarwal et al. [33] studied the interaction patterns of users in dynamic social networks and proposed a globally optimized forward trace approach to mine key nodes in the propagation. Considering the situation that the influence between users changes with time and the network topology remains unchanged, Rodriguez et al. [34] proposed the continuous-time influence maximization problem and gave an approximate solution method. Moreover, Peng et al. [35] studied the influence maximization problem when social networks expand over time, and they proposed an adaptive sampling method to transform the influence maximization problem into a $MAX - K$ coverage problem.

In addition, there are some other studies based on dynamic networks, among which are similar to ours including Meng et al. [36] studied the diffusion mode of information in multiple networks, and proposed the influence maximization problem of dynamic multi-social networks based on common friends. They combined multiple social networks into a dynamic network to study the influence maximization problem. Song et al. [37] studied the problem of tracking the most influential node sets in dynamic social networks and proposed an Upper Bound Interchange Greedy algorithm (UBIG). UBIG updates the seed set under different snapshots by calculating the difference between network snapshots with different timestamps. Wang et al. [38] defined a stream influence maximization (SIM) problem and proposed a sliding window model to maintain a set of k seeds that have the largest influence over the most recent social behaviors. Jia et al. [39] proposed a community-based influence maximization (CIM) algorithm to solve the problem in dynamic networks. CIM first divides the network into communities, then calculates the candidate seed nodes in each community after updating the network structure, and finally selects the k most influential nodes from the candidate seed nodes. However, these studies ignore that the network topology is updated in real-time in dynamic online social networks. Using snapshots of the network or existing update operations to mine seeds, the resulting seed set may not be

optimal under the current network, and there is a lag between the seed set and the current network changes. Therefore, there is still a lot of research space for this issue.

In this paper, since the dynamic evolution of online social networks is continuous, we used historical network snapshots to predict the network topology at the next moment and then mined the seed nodes on the prediction result. Our goal was to maximize the influence of the seed set on the current network and weaken the impact of network changes on the results. To predict the structural changes in online social networks, we employed the link prediction technique in this paper. Methods for link prediction can be divided into three categories, including learning-based, probabilistic models, and similarity-based models [40]. There are three types of measures commonly used in similarity-based methods, including local, global, and quasi-local similarity measures. The local similarity index mainly utilizes local neighborhood information. The global similarity index is calculated based on the topology information of the entire network. Global similarity indices contain more information about the entire structure, but they are more complex to compute than local similarity indices. The quasi-local similarity index combines these two similarity measures and aims to find a balance between local and global. In this paper, we fused three similarity indexes to construct the feature vectors of edges in the network and then used a neural network to construct a prediction model.

3. Methodology

This section is mainly divided into three parts. First, we give a formal definition of the influence maximization problem in dynamic online social networks. Next, we introduce the computational framework proposed in this paper. Finally, we give a theoretical proof of the solution.

3.1. Preliminaries

An online social network is usually represented by a graph $G = (V, E)$, where the node set V represents the user set, $|V| = N$ indicates that there are N users, and the edge set E represents the relationship between different users. Information propagates along the edges in the network.

The classic influence maximization can be defined as an optimization problem in which the network topology is static. That is, given an online social network $G = (V, E)$ and an information diffusion model M that simulates how information spreads in the network, this optimization problem can be defined as selecting K nodes from V as seed nodes such that the number of affected nodes is maximized after the end of the propagation process based on the diffusion model M in G . Assuming that S represents the set of seed nodes, the number of nodes affected by the seed nodes is denoted by $R(S)$. Formally, the classical influence maximization problem can be defined as follows

$$S^* = \arg \max_{S \subset V, |S|=K} R(S). \quad (1)$$

In a dynamic social network, as the network topology is constantly changing, network snapshots can be used to record the updates. In this study, we only consider the changes of edges over time, the nodes remain unchanged, so we denote the network snapshot at time t by $G_t(V, E_t)$, where V is the set of nodes and E_t is the set of edges in the network at timestamp t . Since the network topology is constantly changing, the seed set S_t will also change constantly, where S_t indicates the seed set at time t . Referring to the classical definition of influence maximization, influence maximization in dynamic online social networks can be defined as follows.

Definition 1. *The influence maximization of a dynamic online social network is to find a seed set sequence $\{S_t\}_0^T$ containing K nodes, so that under a given dynamic online social network $\{G_t\}_0^T$ and an information diffusion model M , the sum of the number of affected nodes at all times is the*

largest. Let $R_t(S_t)$ is the number of nodes affected by the seed nodes in the network based on M at time t . The formal expression is as follows:

$$S_t = \arg \max_{S_t \in V, |S_t|=K} R_t(S_t), t = 1, 2, \dots T. \tag{2}$$

In this paper, the information diffusion model adopted the Independent Cascade model. In the Independent Cascade model, each edge in the network is assigned an independent probability p , which represents the strength of the influence between adjacent nodes. If a node is activated, it has only one chance to activate its inactive neighbor nodes. Additionally, once a node is activated, it remains activated throughout the process.

3.2. Proposed Method

Analyzing the evolution process of the dynamic online social network, it can be found that if the most influential seed set S_t is mined based on G_t , then S_t may become less effective in practice. This is because the network is constantly updated and it takes time to calculate the S_t . when the computation of S_t is done, the network may have evolved to $G_{t+\sigma}$, where we assume that the computation time of S_t is $f(S_t)$, the time interval between adjacent snapshots is d and $f(S_t) \leq \sigma < d$. To avoid this problem, we propose a novel framework—Influence Maximization based Prediction and Replacement (IMPR), which first predicts the upcoming network snapshot based on historical snapshots, and then mines seed nodes on the predicted results. The obtained seed nodes are applied to the latest network as the most influential nodes. This is a near real-time scheme that improves the matching between seed nodes and the dynamic network.

3.2.1. Predict Upcoming Network Snapshot

Predicting the upcoming network topology becomes a link prediction problem when only considering the dynamic changes of links in dynamic online social networks. We can solve this problem with machine learning methods. IMPR uses a neural network algorithm (NN) for link prediction. The structure of the neural network is shown in Figure 2. This algorithm uses non-linear activation functions and multiple hidden layers to model complex patterns of edges in dynamic online social networks.

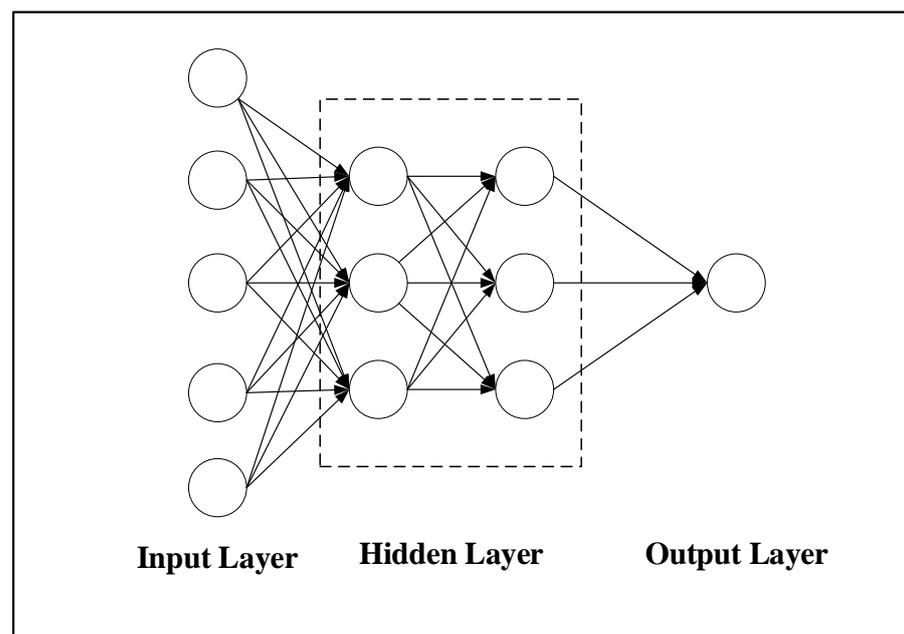


Figure 2. The structure of neural network.

The IMPR framework uses a feature fusion algorithm that fuses different similarity measures together to generate a feature vector, which is then passed to the input layer of the neural network.

The local similarity indices used in the feature vector generation process include Adamic–Adar index (AA), Common Neighbors (CN), Preferential Attachment (PA), and Jaccard Coefficient (JC). The AA index S_{AA} is to measure the similarity between two entities based on the shared features of the two entities. Let $N(\alpha)$ and $N(\beta)$ denote the neighbor node sets of nodes α and β , respectively, and d_γ represents the degree of node γ . The Adamic–Adar index can be expressed as:

$$S_{AA}(\alpha, \beta) = \sum_{\gamma \in N(\alpha) \cap N(\beta)} \frac{1}{\log d_\gamma} \tag{3}$$

The CN index S_{CN} between two nodes represents the size of the intersection of the neighbors of the two nodes, which is defined as follows.

$$S_{CN}(\alpha, \beta) = |N(\alpha) \cap N(\beta)| \tag{4}$$

The JC index S_{JC} is similar to common neighbors. It normalizes the number of common neighbors and can be defined as:

$$S_{JC}(\alpha, \beta) = \frac{|N(\alpha) \cap N(\beta)|}{|N(\alpha) \cup N(\beta)|} \tag{5}$$

The preferential attachment property was first used in network generation models. The PA index S_{PA} between node α and β is defined as:

$$S_{PA}(\alpha, \beta) = d(\alpha) \cdot d(\beta) \tag{6}$$

The global similarity indices usually contain more complete topological information about the network. The global similarity indices adopted in IMPR include cosine based on L^+ (Cos^+), Shortest Path (SP), Average Commute Time (ACT), and Matrix Forest index (MF).

Let L denote the Laplacian matrix of the network, which is widely used in graph theory as an alternative representation for graphs. L^+ denotes the pseudo-inverse of the L matrix computed by Moore–Penrose. Each entry of L^+ can be used to represent the similarity score between two corresponding nodes. Therefore, the Cos^+ index S_{COS^+} between nodes α and node β can be expressed as follows:

$$S_{COS^+}(\alpha, \beta) = \frac{L_{\alpha, \beta}^+}{\sqrt{L_{\alpha, \alpha}^+ L_{\beta, \beta}^+}} \tag{7}$$

The SP index S_{SP} represents the shortest distance from a node to another node in the network. The shortest path between node α and node β is defined as:

$$S_{SP}(\alpha, \beta) = -|D(\alpha, \beta)| \tag{8}$$

where $D(\alpha, \beta)$ represents the shortest distance between nodes α and β calculated using the Dijkstra algorithm [41].

The ACT index is based on the concept of random walk. The ACT similarity index S_{ACT} between node α and node β is defined as the average number of steps required by a random walker to go from start node α to target node β and back to start node α . If $s(\alpha, \beta)$ is the average number of steps required to get from α to β , the following formula captures this concept.

$$S_{ACT}(\alpha, \beta) = s(\alpha, \beta) + s(\beta, \alpha) \tag{9}$$

The MF index S_{MF} is based on the concept of spanning trees. The similarity between nodes α and β can be calculated with the following formula. $(I + L)_{(\alpha,\beta)}$ represents the number of spanning trees rooted at node α and containing both nodes α and β .

$$S_{MF}(\alpha, \beta) = (I + L)_{(\alpha,\beta)}^{-1} \tag{10}$$

The quasi-local indices are a trade-off between global and local metrics. These metrics are computationally more efficient than global metrics. The quasi-local matrices used by IMPR are Path of Length 3 (L3) and Local Path Index (LP).

The L3 index was first used in protein–protein interaction networks. The L3 similarity index $S_{L3}(\alpha, \beta)$ between node α and node β is defined as:

$$S_{L3}(\alpha, \beta) = \sum_{\vartheta, \mu} \frac{a_{\alpha, \mu} \cdot a_{\mu, \vartheta} \cdot a_{\vartheta, \beta}}{\sqrt{d_{\mu} \cdot d_{\vartheta}}} \tag{11}$$

where $a_{\alpha, \mu}$ represents the interaction strength between node α and node μ , and d_{μ} is the degree of node μ .

The LP index S_{LP} is a local path-based metric that trades off accuracy and computational complexity. This metric can be expressed as follows, where A represents the adjacency matrix of the network and ρ represents a free parameter.

$$S_{LP} = A^2 + \rho A^3 \tag{12}$$

The local similarity index has high computational efficiency, the global index has more comprehensive information, and the quasi-local index ignores the information with lower correlation. To extract more comprehensive feature information and improve the performance of prediction, we employ a feature fusion scheme. The edge feature vector of dynamic online social networks is generated by the fusion of local similarity indices, global similarity indices, and quasi-local similarity indices, as in Algorithm 1. To obtain the best-performing feature vector, we fused these similarities in different combinations. The optimal feature vector is eventually used as input to the neural network algorithm to predict the structure of the upcoming network.

3.2.2. Mining Seed Nodes for Influence Maximization

In a dynamic online social network, the network topology changes over time, but is unlikely to change drastically in a short period of time. Therefore, the network structure in two adjacent snapshots is similar, which also leads to the possibility that the most influential seed nodes may be similar. To solve the influence maximization problem in dynamic networks, based on this idea, the IMPR framework adopts a fast replacement algorithm. In this algorithm, if the seed set S_t in the network snapshot G_t at time t has been obtained, then when calculating the seed set S_{t+1} at time $t + 1$, S_{t+1} can be obtained by directly replacing and updating the nodes in S_t . This avoids building from scratch and greatly saves computing time.

We adopt the Interchange Heuristic proposed by Fisher et al. [42] as our strategy for replacing nodes in S_t . The Interchange Heuristic changes only one element of the set at a time, and they have proved that when the objective function is a monotonic submodular function, it is possible to quickly find the set that can no longer be improved. The influence function is a monotone submodular function that satisfies the applicable conditions.

The purpose of updating S_t to S_{t+1} according to the Interchange Heuristic strategy is to obtain the maximum gain. Let $\delta_{v, v_s}(S_t)$ denote the gain brought by replacing node $v_s \in S_t$ with node $v \in V - S_t$, then the replacement rule can be expressed as: $v^* = \arg \max_v \delta_{v, v_s}(S_t)$, $S_{t+1} = S_t - v_s + v^*$, where V represents the set of nodes in the network.

Algorithm 1 Generate the input feature vector

Input: Snapshots of a dynamic online social network $\{G_t\}_0^t$
Output: Feature set for edges $Edge_fs$

- 1: **for** $snapshot$ in $G_0, G_1, G_2, G_3, \dots, G_t$ **do**
- 2: **for** each $edge_curr$ in $snapshot$ **do**
- 3: $node1, node2 \leftarrow edge_curr$
- 4: $cn, aa, jc, pa \leftarrow calculate\ the\ local\ similarity\ index(node1, node2, snapshot)$
- 5: $mf, act \leftarrow calculate\ the\ global\ similarity\ index(node1, node2, snapshot)$
- 6: $cos+, sp \leftarrow calculate\ the\ global\ similarity\ index\ (node1, node2, snapshot)$
- 6: $lp, l3 \leftarrow calculate\ the\ quasi - local\ similarity\ index(node1, node2, snapshot)$
- 7: **If** $Edge_fs(edge_curr)$ not empty **then**
- 8: $temp = Edge_fs[edge_curr]$
- 9: **else**
- 10: $temp = []$
- 11: **end if**
- 12: $Edge_fs[edge_curr] \leftarrow temp + [cn, aa, jc, pa, mf, act, cos, sp, lp, l3]$
- 13: **end for**
- 14: **end for**

We can find that this strategy involves a lot of Monte Carlo simulation processes, which is a time-consuming operation. To improve efficiency, we use an upper bound on the gain to reduce a large number of computational processes. Algorithm 2 describes the process of selecting a node to replace a fixed node v_s $v_s \in S_t$ in the seed set. If the maximum replacement gain δ_{v, v_s} is less than a given threshold λ , the search is abandoned and we can then reselect a node from the seed set for replacement. This loses some improvements but speeds up the update process. Additionally, the improvement below the threshold is negligible and wastes computation time. In our framework, in order to calculate the seed set S_{t+1} at time $t + 1$, we only need to select the node with the greatest possible replacement gain from the seed set S_t at time t , and use the above algorithm to exchange it.

Algorithm 2 Select a candidate seed node

Input: Snapshot $G_t(V, E)$, Seed set S_t at time $t, v_s \in S_t$, The upper bound on replacing gain $\bar{\delta}_{v, v_s}(S_t)$
Output: A candidate seed node v^*

- 1: **Set** $\delta_{v, v_s} \leftarrow \bar{\delta}_{v, v_s}(S_t), v \in V - S_t$
- 2: **Set** $cur_v \leftarrow false, v \in V - S_t$
- 3: **while true do**
- 4: $v^* = arg\ max_{v \in V - S_t} \{ \delta_{v, v_s} \}$
- 5: **if** $\delta_{v^*, v_s} \leq \lambda R(S_t)$ **then**
- 6: $v^* = NULL$
- 7: **break**
- 8: **else if**
- 9: **if** cur_{v^*} **then**
- 10: **break**
- 11: **else**
- 12: $\delta_{v^*, v_s} \leftarrow R(S_t - v_s + v^*) - R(S_t)$
- 13: $cur_{v^*} \leftarrow true$
- 14: **end if**
- 15: **end while**

With the above two important parts, the problem of influence maximization in a dynamic social network can be solved easily by IMPR. At the beginning of the algorithm, we use the greedy algorithm to obtain the seed set S_1 on the initial snapshot G_1 . The next process of the whole framework is shown in Figure 3. We first use the historical snapshots $\{G_t\}_1^t$ to predict the upcoming network snapshot G_{t+1} , then use the fast replacement algorithm to update the seed set on the predicted network snapshot, and finally get the fresh seed set S_{t+1} for the network at time $t + 1$. The complete prediction and fast

replacement process are described in Algorithm 3. This seed set has the highest matching degree with the dynamic network and has the largest influence on the network at time $t + 1$.

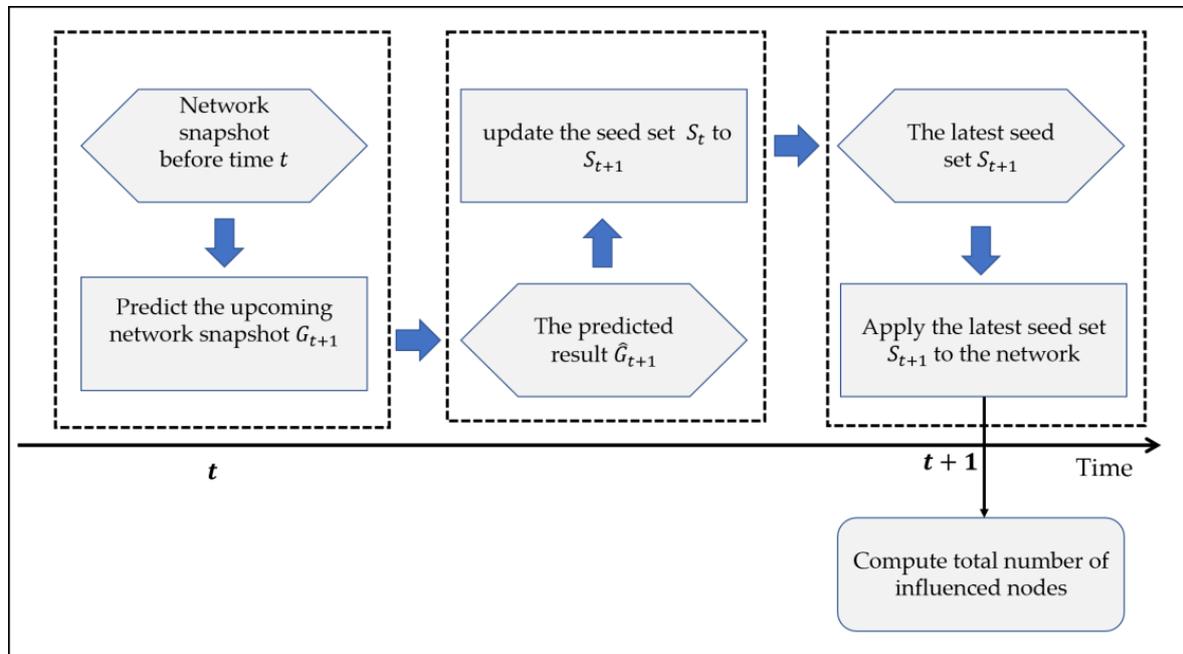


Figure 3. The flow chart of the IMPR framework.

Algorithm 3 Influence maximization based on prediction and fast replacement

Input: Snapshot $\{G_t\}_1^t$, The size of seed nodes K

Output: Seed node set S_{t+1}

- 1: $S_1 = greedy(G_1, K)$
- 2: $\hat{G}_{t+1} \leftarrow$ predict the upcoming network snapshot G_{t+1}
- 3: compute $\bar{\delta}_{v,v_s}(S_t)$ based on $\hat{G}_{t+1}, v_s \in S_t$
- 4: **for** $i = 1$ to K **do**
- 5: $v_s^* = arg \max_{v_s \in S_t} \{ \bar{\delta}_{v,v_s}(S_t) \}$
- 6: $S_{t+1} = S_t - v_s^* +$ Select a candidate seed node $(G_t, S_t, v_s^*, \bar{\delta}_{v,v_s}(S_t))$
- 7: update $\bar{\delta}_{v,v_s}(S_t)$ for any $v \in V - S_t$
- 8: **end for**
- 9: $S_{t+1} = S_t$

3.3. Theory Proof

In this section, we give a theoretical proof of the scheme proposed in this paper.

Theorem 1. The higher the accuracy of the prediction result, the closer the seed set S_{t+1} obtained according to the prediction result is to the expected seed set \bar{S}_{t+1} , and the greater the influence.

Proof of Theorem 1. Suppose the set of edges in the network at time $t + 1$ is E_{t+1} , the prediction result is \hat{E}_{t+1} , the probability of information spreading in the network is P_{uv} , and the accuracy of structure prediction is η , then:

$$\frac{|E_{t+1} \cap \hat{E}_{t+1}|}{|E_{t+1} \cup \hat{E}_{t+1}|} \geq \eta \tag{13}$$

Assuming that the influence function is denoted by $R(S_t)$, the following inequality is satisfied for any dynamic online social network, where $\varepsilon > 0$

$$\left| \operatorname{argmax}_R(G_{t+1}, S_{t+1}) \right| \leq \left| \operatorname{argmax}_R(G_{t+1}, \bar{S}_{t+1}) \right| \leq \varepsilon \tag{14}$$

Combining these two formulas, we can obtain

$$\begin{aligned} & \left| \operatorname{argmax}_R(\hat{G}_{t+1}, \bar{S}_{t+1}) - \operatorname{argmax}_R(G_{t+1}, S_{t+1}) \right| \\ & \leq \left| \operatorname{argmax}_R((G_{t+1} \cup \hat{G}_{t+1}), \bar{S}_{t+1}) - \operatorname{argmax}_R((G_{t+1} \cap \hat{G}_{t+1}), S_{t+1}) \right| \\ & \leq \left| \operatorname{argmax}_R((G_{t+1} \cup \hat{G}_{t+1}), \bar{S}_{t+1}) - \operatorname{argmax}_R((G_{t+1} \cup \hat{G}_{t+1}), S_{t+1}) \right| \cdot \eta \end{aligned} \tag{15}$$

Combining the above equations, we can obtain

$$\left| \operatorname{argmax}_R(\hat{G}_{t+1}, \bar{S}_{t+1}) \right| - \left| \operatorname{argmax}_R(G_{t+1}, S_{t+1}) \right| \leq \varepsilon(1 - \eta) \tag{16}$$

So far, it can be proved that if the prediction accuracy η is more accurate, the seed set obtained based on the prediction will be more closely matched with the expected result. \square

4. Experiments and Discussion

In this section, the performance and efficiency of the proposed scheme are verified through a series of experiments. The experiments are mainly divided into two parts. The first part verifies the accuracy of our prediction module, and the second part compares the classical methods and other similar algorithms with our framework.

4.1. Datasets

To evaluate the performance of the proposed framework, we conduct experiments on four different dynamic network datasets, all of which are real dynamic online social networks. Table 1 shows the information of the datasets. The second column of the table specifies the name of the dataset, the third column indicates the total number of temporal edges included in each dataset, and the last column shows the time span. As can be seen from the table, in order to make the experiments more convincing, we use datasets of different scales.

Table 1. Dataset information.

No.	Name	Temporal Edges	Time Span (Days)
1	CollegeMsg	59,835	193
2	Mathoverflow	107,581	2350
3	Superuser	430,033	2773
4	Stackoverflow	17,823,525	2774

4.2. Evaluate the Prediction Module

To evaluate the algorithm, we adopt a widely used metric in link prediction--AUC. This metric can be interpreted as the probability that the score of an edge in the test set is higher than the score of a randomly selected edge that does not exist. The larger the AUC value, the higher the accuracy of the algorithm prediction. The following formula explains the AUC calculation process:

$$AUC = \frac{n' + 0.5 * n''}{n} \tag{17}$$

where n is the number of comparisons, n' is the number of times the edge has a larger score in the test set, and n'' is the number of cases where two scores are the same.

In the experiment, TensorFlow was used to build our prediction model. In the model, the hidden layer of the neural network is two layers, and each hidden layer has 1024 neurons. The activation functions used in the model are the ReLU function and the sigmoid function. The learning rate used during training was 0.001 and the batch size was 32 for training purposes with epoch 5. The model utilized an Adam optimizer to minimize cross-entropy. All datasets are divided into 20 equally spaced snapshots by time interval, and the first 19 snapshots are used to train the model. After the model is trained, it is used to predict the edges in the last snapshot.

During the experiment, we tested the effect of different feature fusion methods to construct the model input vector. The AUC values of four prediction methods are shown in Table 2, where NNLG (neural network based on local and global similarity indices) represents the fusion of local similarity indices and global similarity indices to generate feature vectors, NNLQ means fusing local similarity indices and quasi-local similarity indices. Similarly, NNGQ and NNLGQ represent different fusion methods of the three similarity measures, respectively.

Table 2. The AUC value of four feature fusion algorithms on different datasets.

Dataset	NNLG	NNLQ	NNGQ	NNLGQ
CollegeMsg	0.57	0.68	0.53	0.59
Mathoverflow	0.76	0.82	0.71	0.77
Superuser	0.68	0.73	0.67	0.68
Stackoverflow	0.74	0.82	0.72	0.74

Analyzing the experimental results, it can be found that the NNLQ that fuses local features and quasi-local features exhibits the best performance. Although the input vector of the NNLGQ algorithm contains local features, global features, and quasi-local features, the effect is not as good as that of NNLQ. In-depth analysis of the reason behind this phenomenon revealed that LQ contains local information and quasi-local information, but not global information, and this combination captures the most accurate features of link prediction, while the redundant information in NNLGQ may interfere with prediction results. Therefore, in our IMPR framework, the local similarity indexes and quasi-local similarity indexes were fused to construct the feature vector.

4.3. Evaluation of the Proposed Framework

In order to reveal the performance of our framework, we first embedded classical influence maximization algorithms into our framework for experiments. Moreover, we also compared the proposed framework with some existing algorithms on dynamic networks.

4.3.1. Baseline Algorithms

In order to demonstrate the superiority of our framework, we compared the classical influence maximization algorithm embedded in the framework and not embedded in the framework. The algorithms used for comparison in the experiments are summarized as follows.

Upper Bound based Lazy Forward (UBLF) [10]: This is a typical representative of a greedy-based influence maximization algorithm, which uses an upper bound on the gain of the influence function to speed up the computational process. Compared with other greedy algorithms, the UBLF algorithm was more efficient.

Prediction-based Upper Bound based Lazy Forward (PUBLF): This was to embed UBLF into our IMPR framework and add the prediction part to the original.

Degree-Discount (DD) [9]: This was the most typical algorithm based on heuristic information, which selects the seed node according to the degree of the node.

Prediction-based Degree-Discount (PDD): This was to embed the Degree-Discount algorithm into our IMPR framework and add the prediction part to the original.

Community Finding Influential Node (CFIN) [14]: This was a recently proposed algorithm based on community structure. First, the network is divided into communities, and then the seed nodes are found in the community according to the dynamic programming algorithm.

Prediction-based Community Finding Influential Node (PCFIN): This was to embed the CFIN algorithm into our IMPR framework.

Furthermore, a series of experiments are conducted to compare our framework with some algorithms in dynamic networks to demonstrate the advantages of our framework. A brief description of these algorithms is given below.

Upper Bound Interchange Greedy algorithm (UBIG) [31]: This algorithm was used to track the influence nodes in the dynamic network, and the result set was continuously updated by comparing the changes of the network structure.

Community-based influence maximization (CIM) [33]: This algorithm mainly uses the community structure to mine the seed nodes in the community, and then decides whether to update the seed set according to the changes of the community structure.

Influence Maximization based Common Users (IMCU) [30]: This algorithm is based on common users and studies the influence maximization problem in dynamic networks from the perspective of users.

Influence Maximization based Prediction and fast Replacement (IMPR): This is a new computational framework proposed in this paper. First, we predicted changes in network structure based on historical snapshots, and then dynamically updated the seed set based on the differences between snapshots.

4.3.2. Evaluation Metric

According to the existing analysis, the purpose of maximizing the influence of dynamic online social networks is to find the K nodes with the greatest influence at each moment in the network as the seed set. To evaluate the performance of our proposed framework, we first assumed that the network was continuously changing dynamically, and then obtained the seed set for each time window in the network according to different algorithms. When the calculation of the seed set was completed, based on the network structure at the current moment and a given information diffusion model, the seed node was used as the information source to simulate the information diffusion process. The number of affected nodes when the propagation ends was used as the influence spread of the seed set. It is important to note that to avoid randomness of the results, each propagation process goes through 100 iterations.

To compare different models, we took the average influence spread of all snapshots as the evaluation metric for different models.

4.3.3. Result and Discussion

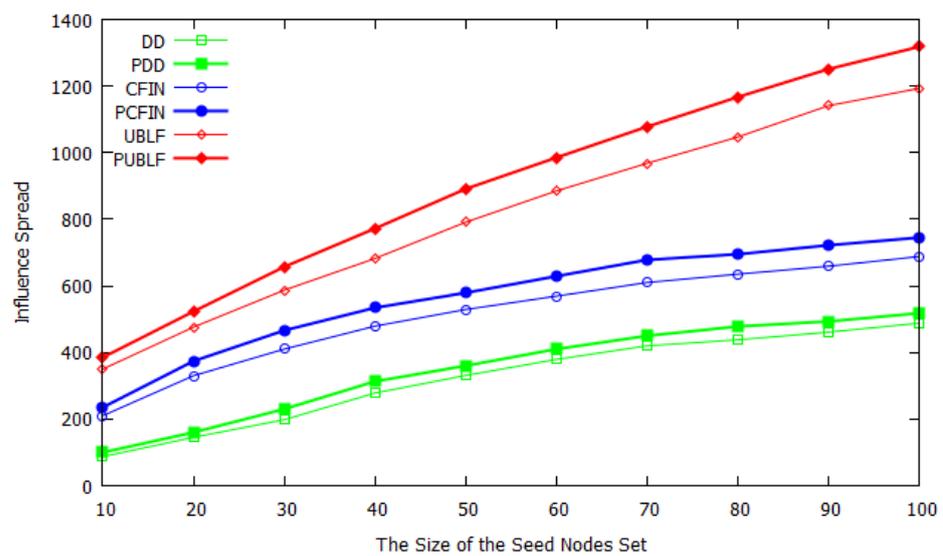
In the experimental process, in order to facilitate comparison with other methods, we adopted the most widely used independent cascade model for the information diffusion model, and the probability of information propagation between adjacent nodes is set to $p = 0.06$. For datasets, we split each dataset into 20 snapshots in an equally spaced manner. We trained our predictive model with the first 10 snapshots. After the model training was completed, we calculated the seed nodes according to different algorithms and used the influence spread as a metric to evaluate the seed nodes.

To demonstrate the importance of the prediction module in our framework, we first compared the classical influence maximization algorithm embedded in our framework with the case without embedding. The results on different datasets are shown in Figure 4, where the abscissa k represents the size of the seed set. During the experiment, the value of K ranges from 10 to 100, with 10 as the interval.

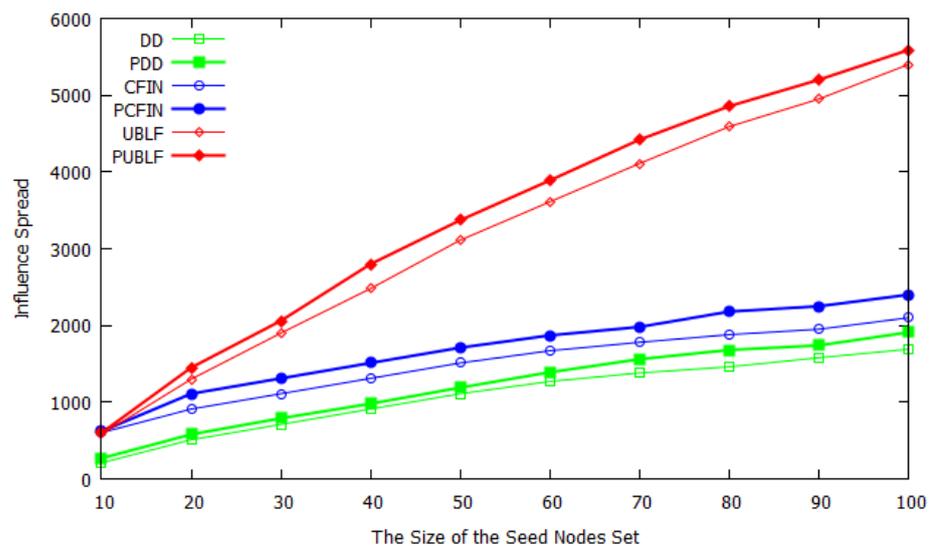
After careful analysis of these figures, it is easy to observe that as the size of the seed set increases, the influence spread of the seed nodes in all datasets gradually becomes larger. The greedy algorithm UBLF exhibited the best performance, and the Degree–Discount

algorithm exhibited the worst effect. This was because the Degree–Discount algorithm only considers the information of the node degree, which sacrifices accuracy in exchange for efficiency improvement. Most importantly, we found that prediction techniques help each algorithm improve accuracy and achieve better results.

Next, we compared our algorithm with some existing influence maximization algorithms in dynamic networks on different datasets. Figure 5 shows the experimental results. Comparing these figures, it can be found that our proposed scheme outperformed other algorithms. This is because our framework could better predict the upcoming network snapshot compared to other algorithms. Mining seed nodes on the prediction network can maximize the fit between the seed nodes and the dynamic network. While other algorithms used outdated network snapshots, when the seed node was calculated, the network structure had changed.

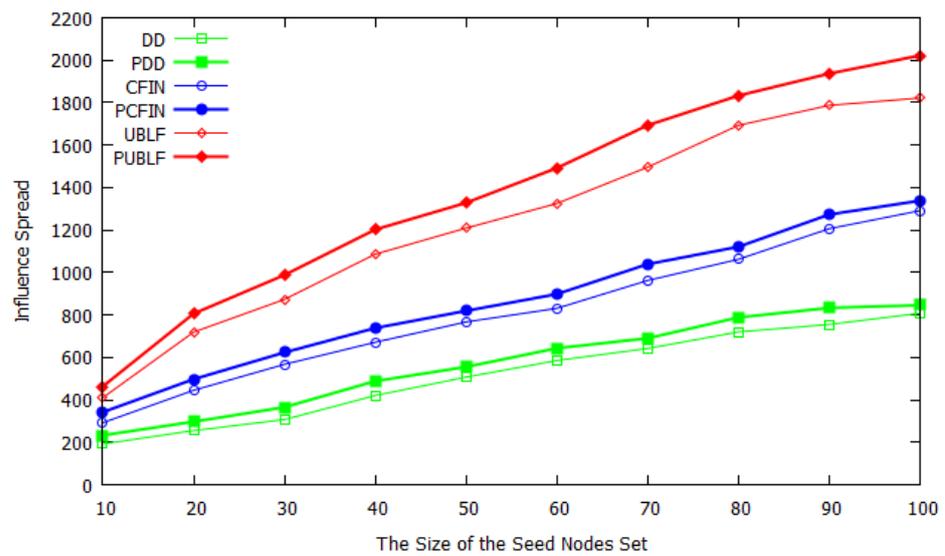


(a)

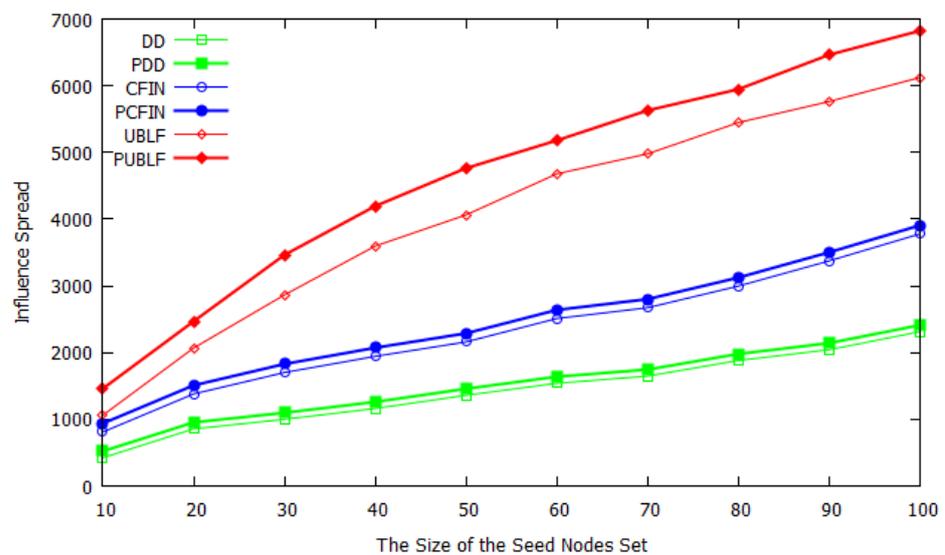


(b)

Figure 4. Cont.



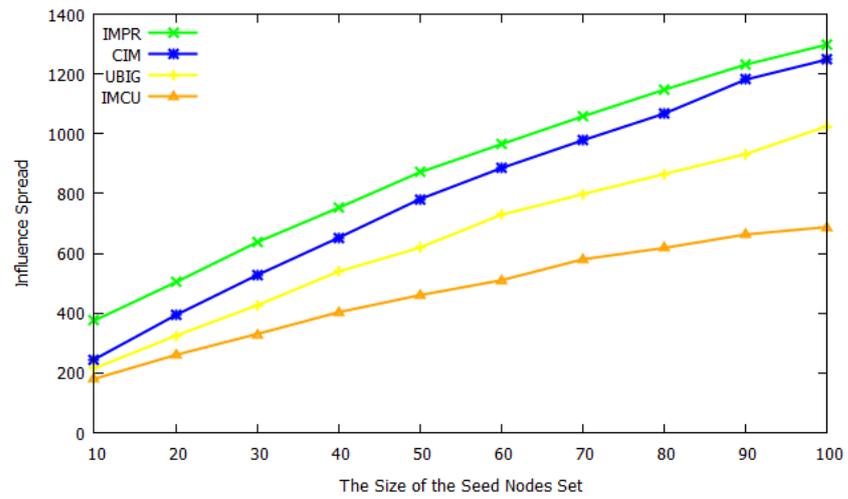
(c)



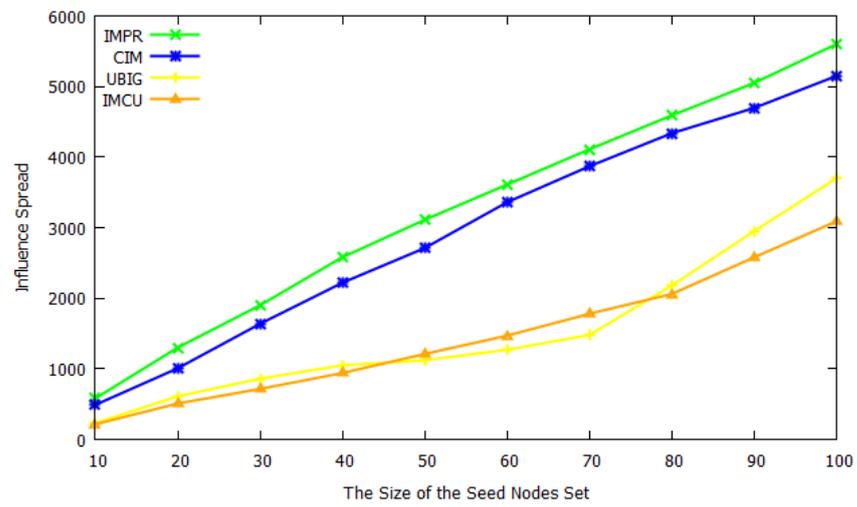
(d)

Figure 4. Comparison of the influence spread of classical influence maximization algorithm embedded and not embedded in IMPR framework. (a) Comparison of the performance of different algorithms on the CollegeMsg dataset. (b) Comparison of the performance of different algorithms on the Mathoverflow dataset. (c) Comparison of the performance of different algorithms on the Superuser dataset. (d) Comparison of the performance of different algorithms on the Stackoverflow dataset.

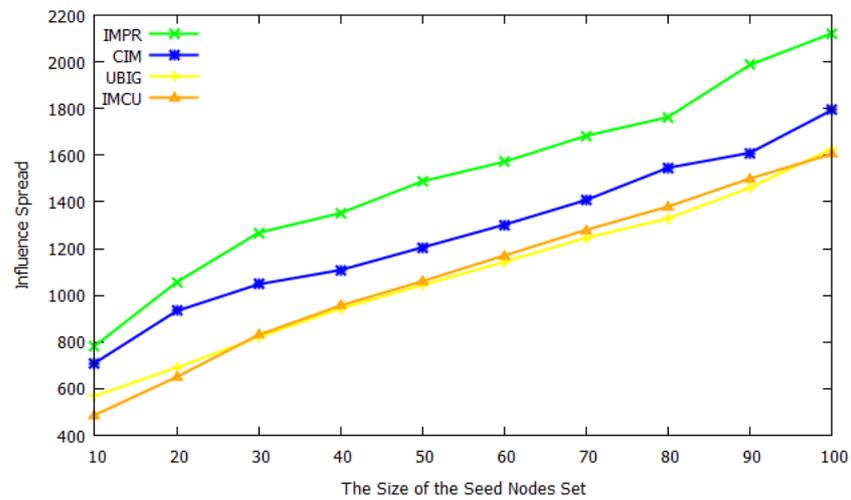
Finally, we compared the running time of different algorithms on four datasets, where we fixed the size of the seed set to 50. The experimental results are shown in Figure 6. It can be seen intuitively from the figure that the UBIG algorithm has the shortest running time, followed by the algorithm proposed in this paper. This is because the UBIG algorithm only calculates the seed node based on the existing historical snapshot every time, and there is no network update operation. Other algorithms include the operation of updating the network structure.



(a)



(b)



(c)

Figure 5. Cont.

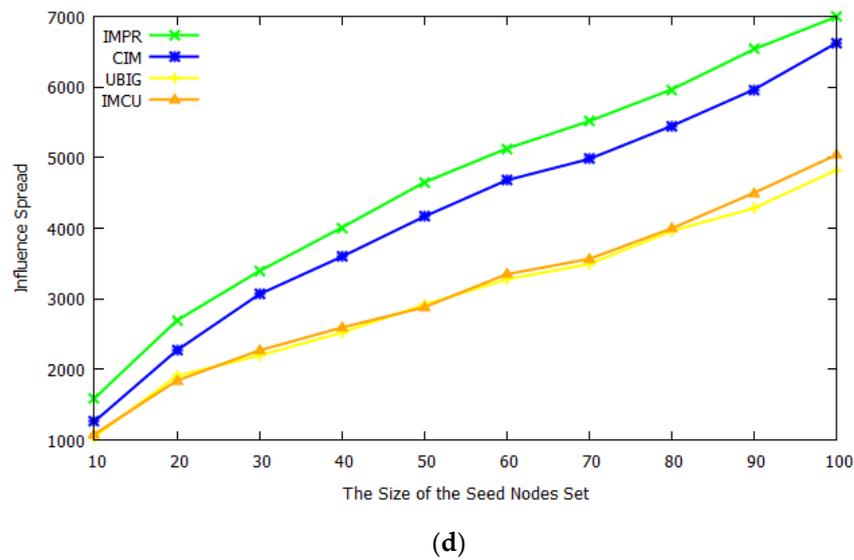


Figure 5. Influence spread of seed nodes selected by IMPR and three other algorithms on different datasets. (a) Comparison of IMPR and three other algorithms on the CollegeMsg dataset. (b) Comparison of IMPR and three other algorithms on the Mathoverflow dataset. (c) Comparison of IMPR and three other algorithms on the Superuser dataset. (d) Comparison of IMPR and three other algorithms on the Stackoverflow dataset.

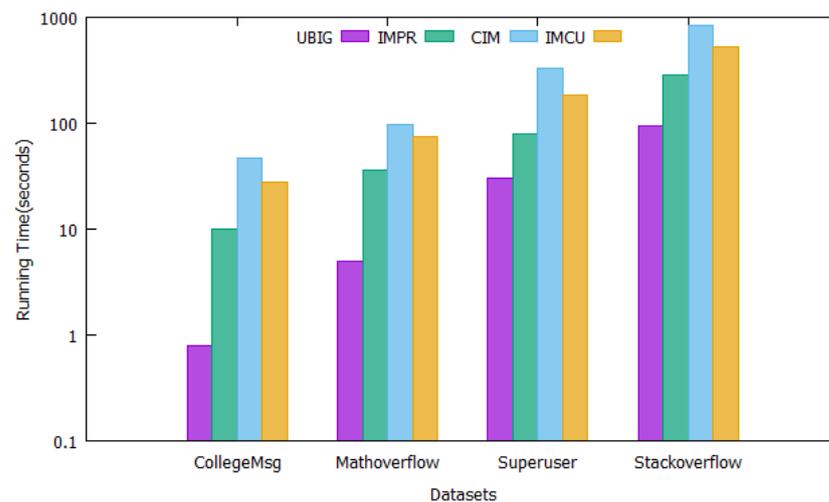


Figure 6. Comparison of IMPR and three other algorithms in running time.

Combining the experimental results, we can conclude that our proposed computational framework is more suitable for solving the influence maximization problem in dynamic networks, especially for those that change continuously. The limitation of our scheme is that it requires a training process; however, training can improve the accuracy of the results.

5. Conclusions

With the continuous development of the mobile Internet, online social networks have changed many aspects of our lives. Many researchers are devoted to the study of online social networks. Influence maximization is one of the important issues of research in this field. Most of the existing research is based on static network structure, but in fact the network structure changes dynamically with time. To this end, we delved into the problem of influence maximization in dynamic online social networks.

In this paper, we propose a novel computational framework for solving the influence maximization problem in dynamic online social networks. Our framework first predicts upcoming network snapshots based on historical network snapshots, and then mines the most influential seed nodes on the predicted results. We theoretically demonstrate the proposed scheme. Moreover, a series of experiments on four real dynamic online social network datasets were conducted to reveal the advantages of our scheme, and the experimental results show that our algorithm can improve the accuracy of the results and the computational efficiency.

In the future, we will continue to study issues related to online social networks. There are two potential research directions, one is to study the influence maximization when the network topology is unavailable, and the other is to study the information diffusion on multilayer networks and extend our model to multilayer networks

Author Contributions: Formal analysis, K.L.; supervision, K.L.; project administration, K.L.; resources, K.L.; writing—review and editing, K.L. and L.Z.; conceptualization, L.Z.; methodology, L.Z.; investigation, L.Z.; writing—original draft, L.Z.; software, L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Beijing Natural Science Foundation, China (No. 4222037, L181010) and the National Key R & D Program of China (No. 2016YFB0801100).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All datasets used in this paper can be downloaded at <http://snap.stanford.edu/data/>, accessed on 10 December 2021.

Conflicts of Interest: The authors declare no conflict of interest. Funders did not interfere in the research process.

References

1. Li, K.; Zhang, L.; Huang, H. Social influence analysis: Models, methods, and evaluation. *Engineering* **2018**, *4*, 40–46. [[CrossRef](#)]
2. Domingos, P.; Richardson, M. Mining the network value of customers. In Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 26–29 August 2001; ACM: New York, NY, USA, 2001; pp. 57–66.
3. Kempe, D.; Kleinberg, J.; Tardos, E. Maximizing the spread of influence through a social network. In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 24–27 August 2003; pp. 137–146.
4. Kempe, D.; Kleinberg, J.; Tardos, E. Influential nodes in a diffusion model for social networks. In Proceedings of the 32nd International Conference on Automata, Languages and Programming, Lisbon, Portugal, 11–15 July 2005; pp. 1127–1138.
5. Sviridenko, M. A note on maximizing a submodular set function subject to a knapsack constraint. *Oper. Res. Lett.* **2004**, *32*, 41–43. [[CrossRef](#)]
6. Leskovec, J.; Krause, A.; Guestrin, C.; Faloutsos, C.; VanBriesen, J.; Glance, N. Cost-effective outbreak detection in networks. In Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD2007, San Jose, CA, USA, 12–15 August 2007; Association for Computing Machinery: New York, NY, USA, 2007; pp. 420–429.
7. Goyal, A.; Lu, W.; Lakshmanan, L.V.S. CELF++: Optimizing the greedy algorithm for influence maximization in social networks. In Proceedings of the 20th International Conference Companion on World Wide Web 2011, Hyderabad, India, 28 March–1 April 2011; ACM: New York, NY, USA, 2011; pp. 47–48.
8. Estevez, P.; Vera, P.; Saito, K. Selecting the Most Influential Nodes in Social Networks. In Proceedings of the International Joint Conference on Neural Networks, Orlando, FL, USA, 12–17 August 2007; IEEE: Piscataway, NJ, USA, 2007; pp. 2397–2402.
9. Chen, W.; Wang, Y.; Yang, S. Efficient influence maximization in social networks. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, 28 June–1 July 2009; ACM: New York, NY, USA, 2009; pp. 199–208.
10. Zhou, C.; Zhang, P.; Zang, W.; Guo, L. On the upper bounds of spread for greedy algorithms in social network influence maximization. *IEEE Trans. Knowl. Data Eng.* **2015**, *27*, 2770–2783. [[CrossRef](#)]
11. Lu, F.; Zhang, W.; Shao, L.; Jiang, X.; Xu, P.; Jin, H. Scalable influence maximization under independent Cascade model. *J. Netw. Comput. Appl.* **2017**, *86*, 15–23. [[CrossRef](#)]

12. Ge, H.; Huang, J.; Di, C.; Li, J.; Li, S. Learning automata-based approach for influence maximization problem on social networks. In Proceedings of the 2017 IEEE Second International Conference on Data Science in Cyberspace (DSC), Shenzhen, China, 26–29 June 2017; pp. 108–117.
13. Zhang, L.; Li, K. Influence maximization based on backward reasoning in online social networks. *Mathematics* **2021**, *9*, 3189. [[CrossRef](#)]
14. Khomami, M.M.D.; Rezvanian, A.; Meybodi, M.R.; Bagheri, A. CFIN: A community-based algorithm for finding influential nodes in complex social networks. *J. Supercomput.* **2020**, *77*, 2207–2236. [[CrossRef](#)]
15. Kundu, S.; Murthy, C.; Pal, S. A new centrality measure for influence maximization in social networks. In Proceedings of the Pattern Recognition & Machine Intelligence-International Conference, Moscow, Russia, 27 June–1 July 2011; pp. 242–247.
16. Kim, J.; Kim, S.; Yu, H. Scalable and parallelizable processing of influence maximization for large-scale social networks. In Proceedings of the Twenty-Ninth International Conference on Data Engineering, Brisbane, Australia, 8–12 April 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 266–277.
17. Kimura, M.; Saito, K. *Tractable Models for Information Diffusion in Social Networks*; PKDD 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 259–271.
18. Goyal, A.; Lu, W.; Lakshmanan, L. Simpath: An efficient algorithm for influence maximization under the linear threshold model. In Proceedings of the 2011 IEEE 11th International Conference on Data Mining, ICDM, Vancouver, BC, Canada, 11–14 December 2011; pp. 211–220.
19. Chen, W.; Yuan, Y.; Zhang, L. Scalable influence maximization in social networks under the linear threshold model. In Proceedings of the 2010 IEEE 10th International Conference on Data Mining, ICDM, Sydney, Australia, 13–17 December 2010; pp. 88–97.
20. Tang, Y.; Xiao, X.; Shi, Y. Influence maximization: Near-optimal time complexity meets practical efficiency. In Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, Snowbird, UT, USA, 22–27 June 2014; pp. 75–86.
21. Wang, Y.; Cong, G.; Song, G.; Xie, K. 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, Washington, DC, USA, 25–28 July 2010; ACM: New York, NY, USA, 2010; pp. 1039–1048.
22. Singh, S.; Singh, K.; Kumar, A.; Biswas, B. ACO-IM: Maximizing influence in social networks using ant Colony optimization. *Soft Comput.* **2020**, *24*, 10181–10203. [[CrossRef](#)]
23. Jung, K.; Heo, W.; Chen, W. IRIE: Scalable and robust influence maximization in social networks. In Proceedings of the 2012 IEEE 12th International Conference on Data Mining, ICDM, Brussels, Belgium, 10–13 December 2012; pp. 918–923.
24. Zhang, L.; Li, K.; Liu, J. An Information Diffusion Model Based on Explosion Shock Wave Theory on Online Social Networks. *Appl. Sci.* **2021**, *11*, 9996. [[CrossRef](#)]
25. Tian, J.; Wang, Y.; Feng, X. A new hybrid algorithm for influence maximization in social networks. *Chin. J. Comp.* **2011**, *34*, 1956–1965. [[CrossRef](#)]
26. Hao, F.; Zhu, C.; Chen, M.; Yang, L.; Pei, Z. Influence strength aware diffusion models for dynamic influence maximization in social networks. In Proceedings of the 2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing, Washington, DC, USA, 19–22 October 2011; pp. 317–322.
27. Teng, Y.; Shi, Y.; Tai, C.; Yang, D.; Lee, W.; Chen, M. Influence maximization based on dynamic personal perception in knowledge graph. In Proceedings of the 2021 IEEE 37th International Conference on Data Engineering (ICDE), Chania, Greece, 19–22 April 2021; pp. 1488–1499.
28. Ge, J.; Shi, L.; Wu, Y.; Liu, J. Human-driven dynamic community influence maximization in social media data streams. *IEEE Access* **2020**, *8*, 162238–162251. [[CrossRef](#)]
29. Li, W.; Zhong, K.; Wang, J.; Chen, D. A dynamic algorithm based on cohesive entropy for influence maximization in social networks. *Expert Syst. Appl.* **2021**, *169*, 114207. [[CrossRef](#)]
30. Cai, Z.; Brede, M.; Gerding, E. Influence maximization for dynamic allocation in voter dynamics. In *Complex Networks & Their Applications IX*; Springer: Cham, Switzerland, 2021; Volume 943, pp. 382–394.
31. Min, H.; Cao, J.; Yuan, T.; Liu, B. Topic based time-sensitive influence maximization in online social networks. *World Wide Web* **2020**, *23*, 1831–1859. [[CrossRef](#)]
32. Wang, C.; Tang, J.; Sun, J.; Han, J. Dynamic social influence analysis through time-dependent factor graphs. In Proceedings of the 2011 International Conference on Advances in Social Networks Analysis and Mining, IEEE, Kaohsiung, Taiwan, 25–27 July 2011; pp. 239–246.
33. Aggarwal, C.; Lin, S.; Yu, P. On influential node discovery in dynamic social networks. In Proceedings of the 2012 SIAM International Conference on Data Mining, SIAM, Anaheim, CA, USA, 26–28 April 2012; pp. 636–647.
34. Rodriguez, M.; Schölkopf, B. Influence maximization in continuous time diffusion networks. In Proceedings of the 29th International Conference on Machine Learning, Edinburgh, Scotland, UK, 26 June–1 July 2012.
35. Peng, B. Dynamic influence maximization. *Adv. Neural Inf. Processing Syst.* **2021**, *34*, 10718–10731.
36. Meng, Y.; Chen, N.; Yi, Y.; Wang, S.; Pei, C. Research on the dynamic multisocial networks influence maximization problem based on common users. *IEEE Access* **2021**, *9*, 127407–127419. [[CrossRef](#)]
37. Song, G.; Li, Y.; Chen, X.; He, X.; Tang, J. Influential node tracking on dynamic social network: An interchange greedy approach. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 359–372. [[CrossRef](#)]

38. Wang, Y.; Fan, Q.; Li, Y.; Tan, L. Real-time influence maximization on dynamic social streams. *Proc. VLDB Endow.* **2017**, *10*, 805–816. [[CrossRef](#)]
39. Jia, W.; Cui, Z.; Qiu, L.; Niu, W. A community-based algorithm for influence maximization on dynamic social networks. *Intell. Data Anal.* **2020**, *24*, 959–971.
40. Kumar, A.; Singh, S.; Singh, K.; Biswas, B. Link prediction techniques, applications, and performance: A survey. *Phys. A Stat. Mech. Its Appl.* **2020**, *553*, 124289. [[CrossRef](#)]
41. Dijkstra, E. A note on two problems in connexion with graphs. *Numer. Math.* **1959**, *1*, 269–271. [[CrossRef](#)]
42. Fisher, M.; Nemhauser, G.; Wolsey, L. An analysis of approximations for maximizing submodular set functions–1. *Math. Program.* **1978**, *14*, 265–294.