Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/mbe

MBE, 20(8): 13660–13680.
DOI: 10.3934/mbe.2023609
Received: 02 May 2023
Revised: 01 June 2023
Accepted: 01 June 2023
Published: 15 June 2023

Research article

Influenced node discovery in a temporal contact network based on common nodes

Jinjing Huang[1] **and Xi Wang**[1,2,*]

[1] School of Software and Services Outsourcing, Suzhou Vocational Institute of Industrial Technology, Suzhou 215004, China

[2] Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China

* **Correspondence:** Email: wangxi@siit.edu.cn; Tel: 15962120827.

Abstract: Verification is the only way to make sure if a node is influenced or not because of the uncertainty of information diffusion in the temporal contact network. In the previous methods, only N influenced nodes could be found for a given number of verifications N. The target of discovering influenced nodes is to find more influenced nodes with the limited number of verifications. To tackle this difficult task, the common nodes on the temporal diffusion paths is proposed in this paper. We prove that if a node v is confirmed as the influenced node and there exist common nodes on the temporal diffusion paths from the initial node to the node v, these common nodes can be regarded as the influenced nodes without verification. It means that it is possible to find more than N influenced nodes given N verifications. The common nodes idea is applied to search influenced nodes in the temporal contact network, and three algorithms are designed based on the idea in this paper. The experiments show that our algorithms can find more influenced nodes in the existence of common nodes.

Keywords: temporal contact network; information diffusion; temporal diffusion path; influenced nodes; infection probability

1. Introduction

The study of information diffusion in social networks has drawn the attention of academia and industry. Finding influenced nodes quickly in the process of information diffusion is practical and valuable for advertising recommendation, virus prevention, network public opinion control and so on. For example, it is necessary for advertisers to know who will be influenced by their advertisements to plan the marketing strategy. As we know, information is diffused via contacts under the closed-world assumption [1] and contacts are varying over time [2]. An information diffusion network can

be modeled as a temporal contact network, which records when and between whom a contact happens without knowing what is diffused [3]. This paper studies how to find as many influenced nodes as possible in a temporal contact network.

However, the information diffusion is uncertain, so which node will be influenced is unpredictable in a temporal contact network. In fact, there exist time intervals and diffusion probability during the information diffusion; it is necessary to track and verify the result of information diffusion in the process of discovering influenced nodes. It is intuitive that only reachable nodes can be influenced nodes under the closed world assumption. To tell whether a node is influenced accurately, we need to confirm with the node. For example, we do not know if Tom has an infectious virus until he is tested. However, node confirmation requires a large amount of computing resources and the number of confirmations is usually limited [4]. Given the number of verifications N, it is assumed that every predicted node is influenced exactly; only N nodes can be found in the best case and the upper limit N cannot be exceeded. In [3], the heuristic diffusion simulation (HDS) algorithm was able to find the influenced nodes in the h-hops range of the initial influenced nodes; however, it could not find more than N influenced nodes within the limitation of verification times N. How to find more influenced nodes with the limited number of verifications is a challenge.

To address the above challenge and find more influenced nodes, we try to find these nodes which can be viewed as influenced nodes directly without verification. If more than one influenced node can be found in one verification, the number of influenced nodes will likely increase. In this paper, we find that if there exist some common nodes on the temporal diffusion paths, these nodes are likely to be influenced. We propose the idea of common nodes and design three algorithms based on common nodes. We prove that if there exist common nodes on the temporal diffusion paths, there are more influenced nodes that can be found with the limited number of verifications.

The main contributions of this paper are as follows:

1) We prove that the common node on the temporal diffusion paths do not need to be verified, if the node after the common node is identified as an influenced node.
2) We present three algorithms based on the idea of common nodes. When a node is verified as an influenced node, the common nodes on the temporal diffusion paths are influenced nodes that do not need to be verified in our algorithms, which may find more than N influenced nodes before check time t under N verification times.
3) We do some comparison experiments using our three algorithms and measure the performance of our algorithms against the HDS and HDS based on adjacent nodes (AHDS) algorithms by using precision and recall. The experiments show that more influenced nodes can be found with the limited verification times because of the common nodes.

The rest of this paper is organized as follows. Section 2 gives some related work. We formalize some relative definitions and the influenced node discovery problem in temporal contact networks based on the independent cascade model [5] in Section 3, and we propose the idea of common nodes on the temporal diffusion paths. Section 4 introduces our methods based on common nodes, which can find more than N influenced nodes after N times of verification in the best case. In Section 5, experiments are conducted to evaluate influenced node queries on different datasets, and the performance of our algorithms are compared with the HDS algorithm and AHDS algorithm. Finally, Section 6 concludes the paper.

2. Related works

The information diffusion problem is a topic that has been receiving much attention in several contexts in recent years, such as in information diffusion modeling [6–9], diffusion probability precision [10–13], link prediction [14, 15], privacy in diffusion networks [16–18], influence maximization in social networks [19–22] and so on. Discovering influenced nodes is similar to influence maximization (IM) and information diffusion prediction in that they all predict the result of information diffusion.

IM is a well-known problem in the information diffusion research field [23], which has been studied extensively. For example, Tong et al. [24] proposed an analysis of the time-constrained adaptive IM problem, which includes the hardness result in the computation of the optimal policy, a lower bound of the adaptive gap and a series of seeding policies. Wang et al. [25] studied the problem of IM in multi-relational social networks and proposed a novel framework consisting of a truncated meta-seed generator and a structural seed extender. These authors aimed to find a seed set in a social network to maximize the expected number of influenced nodes, while discovering influenced nodes in our paper entails trying to find as many influenced nodes as possible.

Researchers have made efforts to explore information diffusion prediction. Independent cascade and the linear threshold [26] are two seminal diffusion models in information diffusion prediction. There were some researchers to predict the diffusion range by using deep learning techniques [27–32] in recent years; for example, Feng et al. [33] developed a new model Inf2vec which combines both the local influence neighborhood and global user similarity to learn the representations. However, these models cannot model the temporal dynamics of a diffusion cascade. Liu et al. [34] proposed a temporal evolving interest-driven cascade prediction framework called TEIC to predict the incremental diffusion scale. Wu et al. [35] proposed a personalized graph neural network to model the diffusion process. Yang et al. [36] proposed a novel full-scale diffusion prediction model based on reinforcement learning and employed an effective structural context extraction strategy to utilize the underlying social graph information. These methods focus on predicting the ultimate range of information diffusion and require a certain amount of training data; meanwhile, it is hard to predict the nodes and when will they become influenced because of the uncertainty of information diffusion. Verifying the result of information diffusion is the only way to make sure if a node is influenced or not. The authors of [3] and [4] described the method of computing the infection probability based on an independent cascade model and proposed some algorithms such as the HDS and AHDS algorithms, to discover the influenced nodes. However, given the number of verifications N, no more than N influenced nodes can be found in the best case. In this paper, we propose the idea of common nodes on temporal diffusion paths in a temporal contact network, which can find more influenced nodes with the limited number of verifications.

3. Related definitions

We now present the necessary background and define the problem formally, as well as propose the definition of common nodes on the temporal diffusion paths.

3.1. Preliminaries and problem formulation

The influenced node discovery problem is related to a lot of prerequisites. In this section, some notations and definitions are introduced successively.

For ease of description, some notations are listed first. Table 1 defines the notations used in this paper.

Table 1. Definitions of notations.

Notation	Description
$\mathbb{G}(\mathbb{V}, \mathbb{E})$	Temporal contact network with node set $\mathbb{V}$ and edge set $\mathbb{E}$
$\mathbb{L}$	Set of initial influenced nodes
h	Hop limit
$\mathbf{Pr}(e)$	Diffusion probability of edge e
$c_{v,s,i}$	Contact from node v_s to v_e

Information diffusion in social networks relies on the contacts between nodes which are varying over time, as in [3] and [4], and the information diffusion network is modeled as a temporal contact network.

Definition 1 ***Contact:*** Let v_s and v_e represent two nodes, respectively; a contact from v_s and v_e is a 5-tuple $c_{v,s,i} = (v_s, v_e, t_i, d_i, p_i)$, in which t_i means the start time of information diffusion, d_i means the delay of information diffusion and p_i represents the diffusion probability.

Definition 2 ***Temporal Contact Network:*** A temporal contact network can be represented as $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ with node set $\mathbb{V}$ and edge set $\mathbb{E}$, in which each edge represents a contact between the two adjacent nodes, and $\mathbb{E} = \{c_{a,b,i}\}$ where $a \in \mathbb{V}$, $b \in \mathbb{V}$, $i = \{1, 2, 3...\}$.

Definition 3 ***Temporal Diffusion Path:*** Given a temporal contact network $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, a temporal diffusion path is a sequence of m contacts, denoted $P = \{c_{0,1,1}, c_{1,2,2}, ..., c_{m,n,n}\}$ where the start diffusion time of the latter contact $c_{i,i+1,i+1}$ and the previous contact $c_{i-1,i,i}$ should satisfy the condition $t_i + d_i \leq t_{i+1}$.

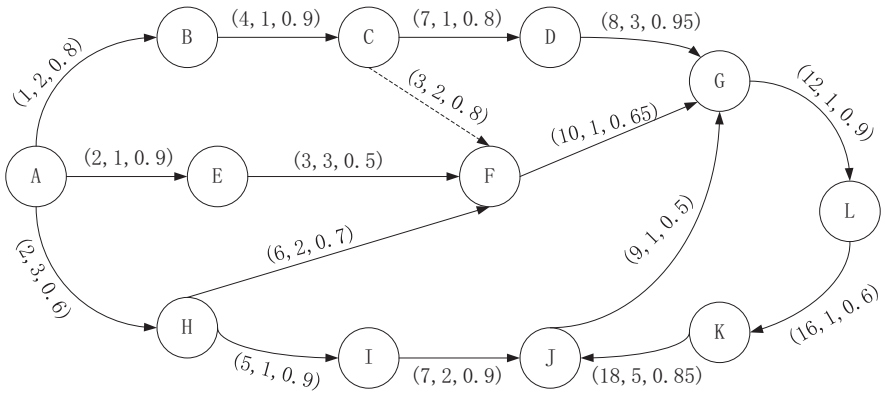


Figure 1. Illustration of temporal contact network and temporal diffusion path.

Figure 1 shows a temporal contact network which is composed of nodes linked by edges with the start time of influence, delay of information diffusion and influence probability. In Figure 1, $(B, C, 4, 1, 0.9)$ represents the contact between node B and node C. $P = \{(A, B, 1, 2, 0.8), (B, C, 4, 1,$

$0.9), (C, D, 7, 1, 0.8)\}$ is a temporal diffusion path, the start time of the previous one of the two adjacent contacts $(A, B, 1, 2, 0.8)$ and $(B, C, 4, 1, 0.9)$ is 1, the delay of information diffusion is 2 and the start time of the latter one is 4; it is obvious that $1 + 2 < 4$, which satisfies the condition $t_1 + d_1 \leq t_2$. Similarly, the two adjacent contacts $(B, C, 4, 1, 0.9)$ and $(C, F, 3, 2, 0.8)$ do not meet the time constraint of a temporal diffusion path, so $P = \{(A, B, 1, 2, 0.8), (B, C, 4, 1, 0.9), (C, F, 3, 2, 0.8)\}$ is not a temporal diffusion path.

Definition 4 ***Time Diffusion Path Length:*** Given a temporal diffusion path, the number of contacts that make up a temporal diffusion path is called the time diffusion path length.

For example, $P = \{(A, B, 1, 2, 0.8), (B, C, 4, 1, 0.9), (C, D, 7, 1, 0.8)\}$ in Figure 1, whose time diffusion path length is 3.

Definition 5 ***Influenced Node:*** Given a known influenced node v_s, v_e is a node of the temporal contact network; if there is a temporal diffusion path from node v_s to v_e, v_e is regarded as an influenced node only when v_e is verified and actually influenced.

For example, node A is an initial influenced node and there is a temporal diffusion path from node A to node C; whether node C is an influenced node needs to be confirmed.

Definition 6 ***Nodes Within h-hops of Influenced Node:*** Given a known influenced node v_0; if v_i is the node within h-hops of v_0 means that there is a temporal diffusion path from v_0 to v_i, at least, and the time diffusion path length of this temporal diffusion path l must be $l \leq h$.

For example, node A is an initial influenced node, and $\{B, C, E, F, H, I\}$ are nodes within 2-hops of node A in Figure 1.

Definition 7 ***Influenced Time:*** Suppose that v_s is a known influenced node, v_e is an adjacent node of v_s and there is a contact $c_{s,e,i}$ from node v_s to v_e. If node v_e is affected by node v_s, the influenced time of v_e is $t = t_i + d_i$.

Definition 8 ***Infection Probability:*** There is an influence probability for each contact in the temporal diffusion path, so each temporal diffusion path has infection probability $\mathbf{Pr}(e)$. The method of computing nodes' infection probability on the temporal diffusion path is described in [3] and [4] and it is based on the inclusion-exclusion principle. However, the value is approximate probability.

[Problem Formulation] Given a temporal contact network $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, an initial influenced node set $\mathbb{L} \in \mathbb{V}$, the number of verifications N, time limit T and hop limit h, the target is to find influenced nodes as possible out of $\mathbb{V} - \mathbb{L}$ with N times of verification. These undetermined nodes are within h-hops of known influenced nodes during the dynamic querying process and their influenced time t needs to be less than T.

3.2. Common nodes on temporal diffusion paths

As we know, it is impossible to test all nodes in the time contact network one by one to discover more influenced nodes within the limited verifications. The probabilities of nodes within h-hops of an influenced node are computed in [3] and [4], and these nodes with a high probability are selected to be verified. It is clear that, no matter which node is selected, only one node can be verified for each verification. Suppose that each selected node is an influenced node; N influenced nodes can be obtained after N times of verification in the best case. However, if more than one influenced node can be found during one verification, the number of influenced nodes can increase.

If some temporal diffusion paths have a common node, it means that these temporal diffusion paths must pass through the node. In this section, we will discuss the common nodes on temporal diffusion

paths.

Definition 9 ***Common Nodes on Temporal Diffusion Paths:*** If there are several temporal diffusion paths from the known influenced nodes to node v and all these temporal diffusion paths pass through node v_x, then the common node among these temporal diffusion paths is v_x.

Given a temporal contact network $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, $\mathbb{L}$ is an initial influenced node set and node v_m is confirmed to be an influenced node at time t. There are k temporal diffusion paths $\{P_1, P_2, ...P_k\}$ from nodes in $\mathbb{L}$ to node v_m, and v is a common node on these temporal diffusion paths; then, node v must be an influenced node under the closed-world assumption.

For convenience, we only record the nodes of the temporal diffusion path; for example in Figure 1, this temporal diffusion path, $P = \{(A, B, 1, 2, 0.8), (B, C, 4, 1, 0.9)\}$, is marked as $P = \{A \to B \to C\}$, so the k temporal diffusion paths can be expressed as follows:

$$\begin{cases} P_1 :< v_0 \to a_1 \to a_2 \to v \to a_n \to v_m > \\ P_2 :< v_1 \to b_1 \to v \to ... \to b_n \to v_m > \\ ..... \\ P_k :< v_0 \to k_1 \to ... \to v \to k_n \to v_m > \end{cases} \tag{3.1}$$

where a_i, b_i and k_i are nodes of the temporal contact network with $i = \{1, 2, 3...\}$. Because node v is the common node among these k temporal diffusion paths, any P_i of k temporal diffusion paths would pass through node v. The information must pass through node v when it travels along P_i from the initial node to v_m, which means that information arrives to v_m from the initial node by way of node v. Node v_m is a known influenced node, so node v must have been influenced before v_m became the influenced node.

For example, in Figure 2, node A is the initial influenced node. If node F is confirmed to be an influenced node, nodes B, D, E are all influenced nodes that do not need to be verified. Node F is the influenced node, and information can only be diffused to node F from node E under the closed-world assumption; thus, it is not necessary to verify node E, which must be an influenced node. Likewise, nodes D, B are also influenced nodes. However, node C cannot be regarded as an influenced node; because there is a temporal diffusion path from node B to node D, information may be diffused from node B to node D without going through node C.

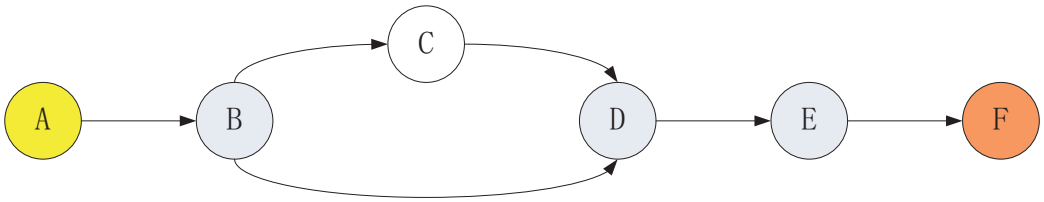


Figure 2. Illustration of common nodes in temporal diffusion paths.

However, not all common nodes need to be obtained by path calculation. If v is a new influenced node and the indegree of v is one, the previous node v_f which points to v must be an influenced node. An indegree of v equal to one means that there is only one path from node v_f to node v; if v is an influenced node, information can only travel from node v_f to node v. For example, the indegree of

node F is one in Figure 2 and node F is an influenced node; the previous node E must be an influenced node. Similarly, the indegree of node E is one; the previous node D is an influenced node.

It is clear that, if a node v is verified as an influenced node and there exist common nodes on the temporal diffusion paths from the known influenced nodes to node v, these common nodes can be viewed as influenced nodes directly.

4. Effective methods for influenced nodes discovery based on common nodes

It is possible that more influenced nodes can be found if there exist common nodes on the temporal diffusion paths, as described in Section 3.2. In this section, the idea of common nodes is applied to methods for influenced nodes discovery. We design three algorithms based on common nodes and discuss each algorithm separately.

4.1. CHDS algorithm

At first, we discuss a HDS algorithm based on common nodes (CHDS), which adds the common nodes idea to the HDS algorithm directly. The infection probability of nodes within h-hops of influenced nodes are computed in the HDS algorithm, and the nodes with maximum infection probability will be selected to verify after computation. Then, the HDS algorithm updates the infection probability of nodes in time and carries out a new round of discovering influenced nodes. If a new influenced node is found, we try to find the common nodes on the temporal diffusion paths to the new influenced node. These common nodes can be regarded as influenced nodes without verification according to the common nodes idea we have discussed in Section 3.2.

The implementation of the CHDS algorithm is as follows:

[Step 1] Generate the candidate set.

The CDHS algorithm computes the infection probabilities of nodes within h-hops of each influenced nodes in $\mathbb{L}$ and puts them into the candidate set. In the process, it makes a note of which nodes affect these nodes.

[Step 2] Verify nodes and search for common nodes on the temporal diffusion paths.

Node v_k, which has maximum infection probability, is selected to be verified in the candidate set. If v_k is influenced, the common nodes on the temporal diffusion paths from those nodes which affect v_k to node v_k are found. Those nodes which affect node v_k are marked in the candidate set.

[Step 3] Update infection probability of the candidate node set.

When finding the new influenced nodes, the infection probability of nodes in the candidate node set will change. It is necessary to update the infection probability of the nodes in the candidate node set. Meanwhile, new candidate nodes may be generated that are within the h-hops range of the new influenced node. These new candidate nodes should be appended to the candidate node set.

In contrast to the HDS algorithm, the CHDS algorithm searches for the common nodes after verification. Algorithm 1 is the implementation of searching for the common nodes from the initial node set V to node v_k. If the indegree of v_k is one, the previous node of v_k can be added to the result set $\mathbb{C}$. Then, all nodes with 1 indegree before v_k will be added to the result set $\mathbb{C}$ via a while loop, as shown in Line 2. Line 7 serves to find the temporal diffusion paths within h-hops from the initial node set V to node v, whose indegree is more than 1. Lines 9–13 serve to find the common nodes on these temporal diffusion paths.

Algorithm 1: Finding Common Nodes: FindCN($\mathbb{V}$,v_k,h)

Input: $\mathbb{V}$: the set of nodes that affect v_k, h: hop limit
Output: $\mathbb{C}$: the set of common nodes on temporal diffusion paths

$\mathbb{C} \leftarrow \phi$, $v \leftarrow v_k$;
while *the indegree v is 1* **do**
  $\mathbb{C} \leftarrow \mathbb{C} \cup v$;
  $v \leftarrow$ the previous of v;
end
for $v_i \in \mathbb{V}$ **do**
  $P \leftarrow getPath(v_i, v, h)$;
end
for $v_i \in P_0$ **do**
  if *(v_i on the paths* $\{P_1, P_2, ...P_n\}$ **then**
    $\mathbb{C} \leftarrow \mathbb{C} \cup v_i$;
  end
end
return $\mathbb{C}$;

4.2. ICHDS algorithm

It is necessary to compute the infection probability of all nodes within h-hops of the known influenced nodes, which consumes a lot of computational resources. In this section, we talk about the ICHDS algorithm, which is an HDS algorithm based on indegree and common nodes. The ICHDS algorithm selects nodes within h-hops of influenced nodes for calculation by indegree. As far as the node indegree is concerned, the more indegree of a node the more likely it is to receive messages from other nodes. The efficiency of the algorithm is improved by reducing the number of computed nodes.

The ICHDS algorithm also generates a candidate set first. The difference between the ICHDS and CHDS algorithms is that the ICHDS algorithm does not aim to compute the probability of all nodes within h-hops of the influenced nodes, but to sort the indegree of nodes within h-hops of the initial influenced nodes and compute the infection probability of the node with the maximum indegree. If there are some nodes with the same indegree, they are put into the candidate set in order of their probabilities.

Algorithm 2: Get Candidate Set: getCandidateSet($\mathbb{G}$,$\mathbb{L}$,h,t)

Input: $\mathbb{G}$: the temporal contact network, $\mathbb{L}$: the set of initial influenced nodes , h: hop limit, t: time limit
Output: $\mathbb{C}$: the candidate set
$\mathbb{C} \leftarrow \phi$;
$\mathbb{V} \leftarrow getMaxIndegreeV(\mathbb{L}, h)$;
$\mathbb{C} \leftarrow ComputePro(\mathbb{G}, \mathbb{L}, \mathbb{V}, t)$;
$SortV(\mathbb{C})$;
return $\mathbb{C}$;

Algorithm 2 is the process of getting the candidate set. The second line obtains the node set with the maximum indegree within h-hops of the initial influenced nodes. The third line computes the infection probability of the above nodes with the maximum indegree and puts them into the candidate set. The fourth line sorts the nodes in the candidate set according to their probabilities.

After getting the candidate set, we should select a node to verify and search for the common nodes on temporal diffusion paths. Specifically, the node marked v with maximum infection probability in the candidate set is selected to be verified. If node v is influenced, the common nodes on these temporal diffusion paths from the initial influenced nodes to node v should be found. These common nodes are put into the result set if the common nodes exist. If v is not an influenced node, the next node is selected from the candidate set for verification. When the candidate set is empty, it needs to be regenerated, that is, the nodes whose indegree is 1 less than the last time are selected for infection probability computation, and these nodes are put into the candidate set for verification.

For example, A is the initial influenced node in Figure 2; suppose that the hop limit is 3 and the indegree of node D is 2, which is the maximum indegree. Node D is selected to be verified based on the above method. There are two temporal diffusion paths from node A to node D, that is, $\{A \rightarrow B \rightarrow C \rightarrow D\}$ and $\{A \rightarrow B \rightarrow D\}$. Node B is the common node of the two temporal diffusion paths; if node D is influenced, node B must be an influenced node.

When the new influenced nodes are found, the nodes in the candidate set need to be updated in time, because nodes within h-hops change with the new influenced node. These nodes with maximum indegree within h-hops of the new influenced node should be found first. If their indegree is greater than or equal to the indegree of the nodes in the current candidate set, the infection probability of these nodes are computed, and these nodes are put into the candidate set in order of their infection probabilities. Otherwise, the candidate set will not be updated.

Algorithm 3: Update Candidate Set: updateCandidateSet($\mathbb{G}$,$\mathbb{C}$,h,v)

Input: $\mathbb{G}$: the temporal contact network, $\mathbb{C}$: the candidate set , h: hop limit, v: the new influenced node

Output: $\mathbb{C}$: the candidate set

1 $\mathbb{V} \leftarrow getMaxIndegreeV(v, h)$;
2 $n \leftarrow getIndegree(\mathbb{V})$;
3 **if** $(n > getIndegree(\mathbb{C})$ **then**
4 $\quad \mathbb{C} \cup ComputePro(\mathbb{G}, \mathbb{L}, v, t)$;
5 $\quad SortV(\mathbb{C})$;
6 **end**
7 **return** $\mathbb{C}$;

Algorithm 3 shows the update of the candidate set. Lines 1 and 2 obtain the nodes with the maximum indegree within h-hops of the new influenced node and the indegree is marked as n. Lines 3–6 update the candidate set according to the size of n and the indegree of the original candidate set.

4.3. PCHDS algorithm

Although infection probability is not computed for all nodes within h-hops of the influenced nodes in the ICHDS algorithm, it also takes time to find the nodes with the maximum indegree and the common nodes on the temporal diffusion paths. In addition, the infection probability is usually an approximate probability, which is inaccurate and requires a lot of computation. In this section, an HDS algorithm based on path and common nodes (PCHDS) is proposed, which gives some effective measures for improvement.

First, we will discuss the improvements of the PCHDS algorithm, which has two primary new improvements.

[Improvement 1] The exception of common nodes.

Suppose that v_0 is the initial influenced node and v_m is an influenced node by verification. If there is only one temporal diffusion path from v_0 to v_m, it is obvious that all nodes between v_0 and v_m are influenced nodes. The computation of common nodes is easier when we try to find as many such temporal diffusion path as possible.

[Improvement 2] The computation of infection probability.

The process of calculating the infection probability of nodes in the temporal contact network is not simple, a previously published paper [4] introduces the calculation methods of different structures. The simplest and most accurate method of infection probability is the linear structure, which has only one temporal diffusion path from the known influenced node to the node that is to be calculated. In this structure, the infection probability of the node to be computed is the product of the probability of each contact on the temporal diffusion path. In such a temporal diffusion path, the indegree of the rest of the nodes in the temporal contact network is one, except for the head node. For the convenience of expression, we call this temporal diffusion path as a temporal diffusion path with 1 indegree.

Then, we will describe the implementation of the PCHDS algorithm. The main idea of the PCHDS algorithm is to find the temporal diffusion paths with 1 indegree as far as possible from the known influence nodes and select the node that is far away from the known influenced nodes on the temporal diffusion path to confirm. The flow chart of the PCHDS algorithm is shown in Figure 3.

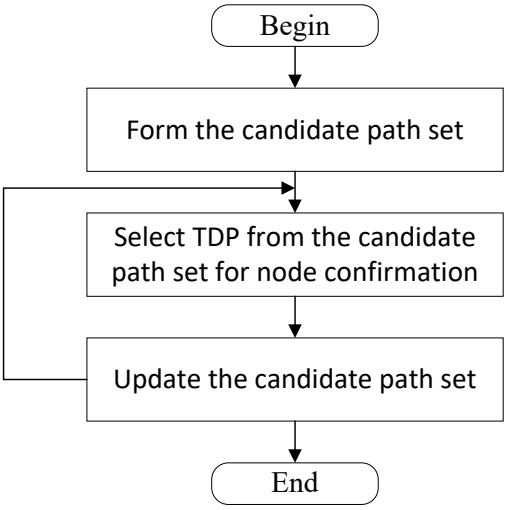


Figure 3. Flow chart of PCHDS algorithm.

The specific method is as follows:

[Step 1] The temporal diffusion path within h-hops of the initial influenced nodes are obtained to form the candidate path set.

We can know that, if the node at the end of the temporal diffusion path is verified as an influenced node and the temporal diffusion path is composed of nodes with 1 indegree, all nodes on the whole temporal diffusion path can be regarded as the influenced nodes without confirmation, according to the previous description. Because each contact has a certain probability, it is obvious that, if the temporal diffusion path length is longer, the infection probability of the last node on the path is lower. We set a threshold p_{min} for the minimum infection probability in the PCHDS algorithm in order to ensure the validity of verification. The infection probability of the tail node on the temporal diffusion path is required to be greater than or equal to p_{min}. These temporal diffusion paths are put into the candidate path set in accordance with the length of the temporal diffusion paths. If the temporal diffusion paths have the same length, we can put these temporal diffusion paths into the candidate path set in accordance with their infection probabilities. If there is no temporal diffusion path with 1 indegree or the infection probabilities of all temporal diffusion paths with 1 indegree are less than p_{min}, the node adjacent to the known influenced node and with the maximum infection probability within h-hops is selected to form a candidate path with the known influenced node.

Algorithm 4: Get Candidate Path Set: getCandidatePath($\mathbb{G}$,$\mathbb{L}$,h,t,p)

Input: $\mathbb{G}$: the temporal contact network, $\mathbb{L}$: the set of initial influenced nodes , h: hop limit, t: time limit, p: the minimum infection probability threshold

Output: $\mathbb{C}$: the candidate path set

```
1 ℂ ← ϕ;
2 for v_i ∈ 𝕃 do
3 |   r ← getPath(𝔾, v_i, h, t, p);
4 |   SortV(ℂ ∪ r);
5 end
6 return ℂ;
```

Algorithm 4 gets these temporal diffusion paths with 1 indegree within h-hops of the known influenced nodes to form the candidate path set. The function getPath of line 3 obtains the temporal diffusion paths with 1 indegree and an infection probability that is greater than or equal to p_{min} within h-hops of node v_i, and it records the infection probability of each node on the temporal diffusion path. Line 4 adds the new temporal diffusion path to the candidate path set according to the length and infection probability of the new temporal diffusion path.

[Step 2] The temporal diffusion path is selected from the candidate path set for node confirmation.

The farthest node of the first temporal diffusion path of the candidate path set is selected to be verified. If the node is influenced, all nodes on the temporal diffusion path can be put into the result set. Otherwise, the farthest node needs to be removed from the temporal diffusion path and the temporal diffusion path length is reduced by 1. This new temporal diffusion path is put into the candidate set according to the length and infection probability; then, the first temporal diffusion path of this current candidate path set is selected to be verified as described earlier.

[Step 3] The candidate path set should be updated.

The temporal diffusion paths with 1 indegree within h-hops of the new influenced node should be obtained in this step. If two adjacent nodes (called A and B) are all influenced nodes, node A can be removed from this temporal diffusion path $\{A \rightarrow B \rightarrow ...\}$.

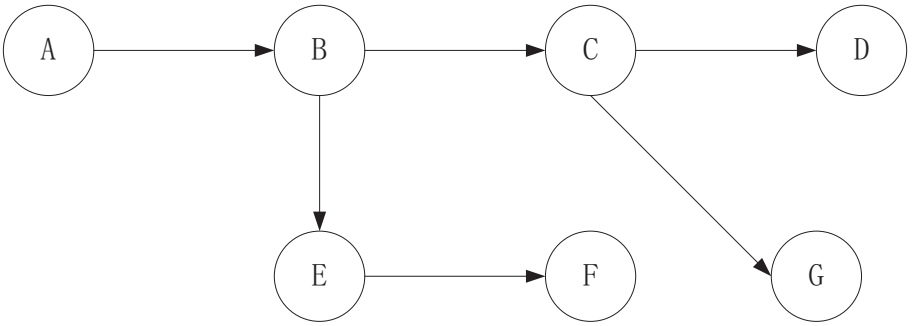


Figure 4. Illustration of updating process for the candidate path set.

As is shown in Figure 4, suppose that node A is the initial influenced node and node D is the verified influenced node; nodes B and C can be regarded as influenced nodes without verification. These temporal diffusion paths with 1 indegree from node B, node C and node D need to be found while updating the candidate path set. As nodes C and E are adjacent nodes of node B and node C is an influenced node, it is unnecessary to find the temporal diffusion path along node C from node B. Finally, two new temporal diffusion paths are $\{B \rightarrow E \rightarrow F\}$ and $\{C \rightarrow G\}$. Algorithm 5 shows the process of updating the candidate path set.

Algorithm 5: Update Candidate Path Set: updateCandidatePath($\mathbb{G}$,$\mathbb{C}$,h,$\mathbb{V}$,$\mathbb{R}$,p)

Input: $\mathbb{G}$: the temporal contact network, $\mathbb{C}$: the candidate path set, h: hop limit, $\mathbb{V}$: the influenced node set on the new temporal diffusion path, $\mathbb{R}$: the result set, p: the minimum infection probability threshold

Output: $\mathbb{C}$: the candidate path set

```
for v_i ∈ 𝕍 do
    A ← getAdjV(v_i);
    for a_i ∈ A do
        if (a_i ∉ ℝ) then
            r ← v_i ∪ getPath(𝔾, a_i, h − 1, t, p);
            Sort(ℂ ∪ r);
        end
    end
end
return ℂ;
```

5. Experiments

We performed some relative experiments to analyze the performance of our algorithms in this section.

5.1. Experimental setup

[Experimental setting] All of these experiments were conducted on a computer equipped with an Intel Core i5 processor and 16GB memory. The operating system version was Windows 10. These experiments were programmed by using Java language.

[Experimental Datasets] We used the Kronecker graph model [37] to generate datasets, which is often used to generate synthetic datasets for information diffusion. Kronecker graphs can model realistic networks very well [38]. However, the Kronecker graph is static, so we added time and diffusion probability to the graph to form a temporal diffusion network. In this dataset, the temporal diffusion path with 1 indegree, whose length is greater than or equal to 2, accounts for approximately 5% of the total number of edges. The dataset was marked as $s-5$. In order to verify the impact of common nodes on the experimental result, we added some temporal diffusion paths with 1 indegree whose length is 2 to $s-5$ to form two datesets donated as $s-10$ and $s-15$. Meanwhile, we deleted all the temporal diffusion paths with 1 indegree, whose length are greater than or equal to 2, from $s-5$ to form a new dateset called $s-0$. The precision and efficiency of our algorithms were tested on the four datasets.

[Calculation method of precision and recall] The precision was evaluated using the method described in [3] and [4], where C is the node set selected for verification and $St(u_i|t)$ is the influenced state of u_i at time t. If the number of verifications is N, $|C| = N$. If u_i is influenced, $St(u_i|t) = 1$; otherwise, $St(u_i|t) = 0$. The recall test method was extended from the formula proposed in [3]. Because of the common nodes, new nodes can be added to the result set without verification. Therefore, a new parameter n was added to the formula of recall, which is the number of influenced nodes without verification. The parameter $\mathbb{V}$ in Eq (3) is the node set of the temporal contact network. It is obvious that a higher precision and recall means more effective verification, and more influenced nodes can be found in the time contact network with a limited number of verification times.

$$Precision = \frac{1}{|\mathbb{C}|}\sum_{u_i\in\mathbb{C}}\mathbf{St}(u_i|t) \tag{5.1}$$

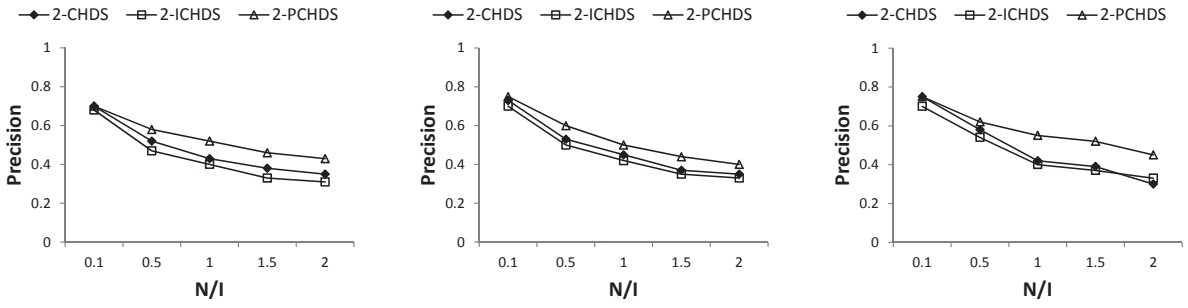
$$Recall = (\sum_{u_i\in\mathbb{C}}\mathbf{St}(u_i|t) + n)/\sum_{u_i\in\mathbb{V}}\mathbf{St}(u_i|t) \tag{5.2}$$

5.2. Contrast experiment

1) Comparison of our algorithms CHDS, ICHDS and PCHDS

We propose three algorithms in this paper; some comparison experiments for the three algorithms are presented in this section. Figures 5 and 6 show the comparison of precision and recall within 2-hops of the influenced nodes of the three algorithms on different datasets.

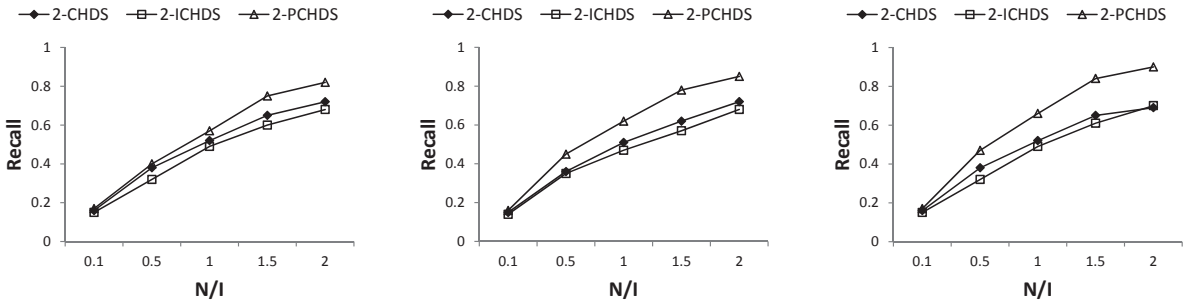
Figure 5 is the comparison of precision on different datasets, where N is the verification time and I is the predetermined influenced node set. On the whole, the PCHDS algorithm had the high precision,



(a) $s-5$ (b) $s-10$ (c) $s-15$

Figure 5. Precision on different datasets.

and the ICHDS algorithm precision was lower than that of the other two algorithms. The PCHDS algorithm is limited by the minimum infection probability threshold, as it will select the adjacent node with the maximum probability within h-hops for verification. The ICHDS algorithm takes the indegree of nodes within h-hops as a condition for node selection; however, some nodes have a larger indegree but there are less temporal diffusion paths to these nodes, which causes the lower precision.



(a) $s-5$ (b) $s-10$ (c) $s-15$

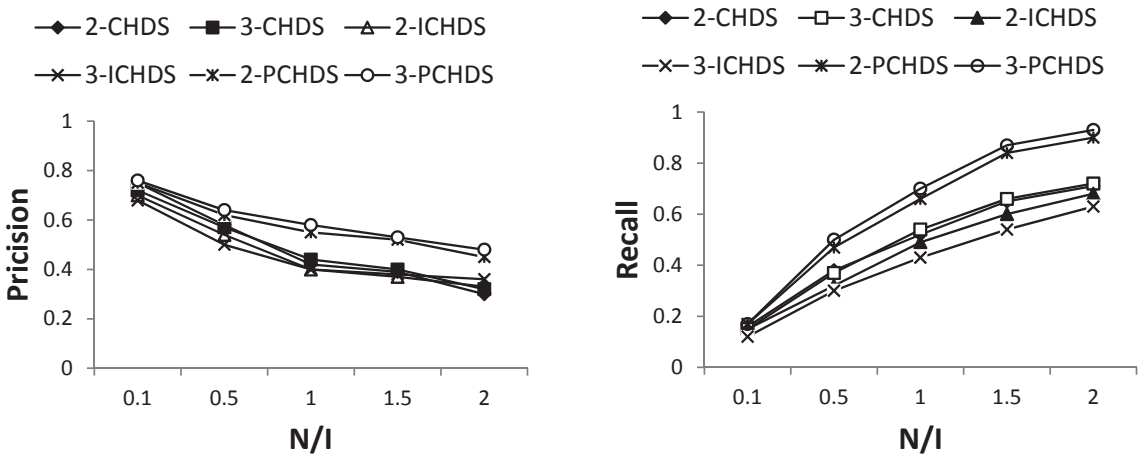
Figure 6. Recall on different datasets.

Figure 6 is the comparison on recall of different datasets. On the whole, the recall of the PCHDS algorithm is higher than that of the other two algorithms because of the common nodes. If a node is verified as an influenced node, these common nodes of temporal diffusion paths can be regarded as influenced nodes without verification. However, it is difficult to find common nodes when the hop limit is 2 in the CHDS and ICHDS algorithms. As can be seen from Figure 6, the number of temporal diffusion paths with 1 indegree is different on the three datasets; the $s-15$ dataset had the most temporal diffusion paths with 1 indegree, so the PCHDS algorithm had the best recall. Through the experimental comparisons, with the increase of common nodes in the dataset, the recall of the PCHDS algorithm is improved.

The ICHDS algorithm had lower precision and recall than the CHDS algorithm on different datasets according to the experiments. However, instead of computing the infection probability of all nodes

within h-hops of the influenced nodes, the ICHDS algorithm selects nodes to compute the infection probability according to the indegree. The ICHDS algorithm does not consume a lot of computing power to calculate the infection probability of nodes.

In addition, the number of hops has an effect on the efficiency. It is clear that, when h is larger, more candidate nodes need to be calculated and more time is spent. As there is no common node when h is 1, the precision and recall of the three algorithms in the 2-hops and 3-hops range on the $s-15$ dataset were compared. As shown in Figure 7, the precision and recall of the three algorithms did not vary significantly with the increase of hops. This is because of the lower infection probability of the node which is far away from the initial influenced nodes with the increase of hops. It is less likely to select the distant node when selecting nodes to be verified. Therefore, there is little difference between the 2-hops and 3-hops in precision and recall. However, these algorithms need to spend more time when the hop is three.



(a) Precision (b) Recall

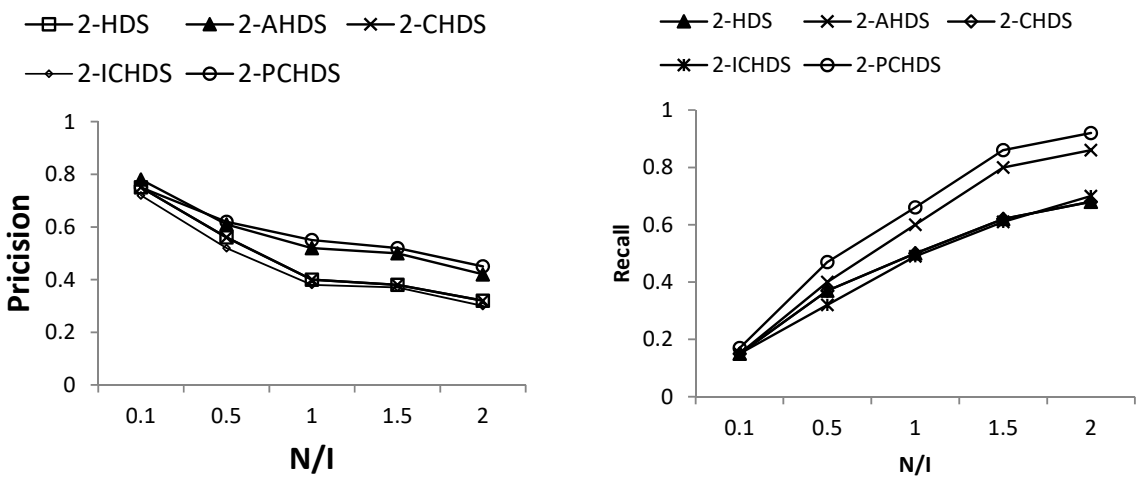
Figure 7. Precision-recall of different hops.

Then, we analyzed the time complexity of the algorithms. For the CHDS and ICHDS algorithms, the first step is to compute the infection probability of the nodes and select one node to verify; then, the query of the common nodes is carried out if the verification is successful. The CHDS algorithm adds the common node query to the HDS algorithm directly, and the HDS algorithm computes the infection probability of all nodes within the h-hops range of the known influenced nodes. The time complexity of computing the infection probability of each node is $O(d^h)$, where d is the out-degree of a node and h is the hot limit, so the time complexity of computing the infection probability of all nodes is $O(d^h * n)$, where n is the number of nodes within the h-hops range of the known influenced nodes. The ICHDS algorithm does not need to compute the infection probability of all nodes within the h-hops range. It selects the node with the largest indegree to compute the infection probability, which can be obtained quickly by setting the index, so the time complexity is $O(d^h)$. Both the CHDS algorithm and ICHDS algorithm need to search the common nodes on temporal diffusion paths; the time complexity of the whole query process is $O(l_1 * l_2)$ at best, where l_1 and l_2 mean the number of nodes on the first and

second temporal diffusion paths, respectively. However, it would go through every node on all temporal diffusion paths at worst. The main time consumption of the PCHDS algorithm involves finding these temporal diffusion paths with 1 indegree in the range of h-hops, whose time complexity is $O(d^h)$. It takes the least time because there is no common node query process in the PCHDS algorithm.

2) Comparison of our algorithms, HDS and AHDS

There are no obvious effects on the result between 2-hops and 3-hops, but 3-hops increases the complexity of the calculation, so we compared our algorithms with the HDS and AHDS algorithms within 2-hops on the $s-15$ dataset. Figure 8 is the experiment result. From the comparison, both the precision and recall of algorithms HDS, CHDS and ICHDS were lower than those of the other two algorithms, AHDS and PCHDS. The CHDS algorithm is based on the HDS algorithm, which add a common node query to the HDS algorithm, so the precision of the CHDS and HDS are similar. The ICHDS algorithm selects a node for probability calculation according to the indegree; however, a node with a large indegree does not mean it has more temporal diffusion paths and a larger infection probability, so the precision of ICHDS was lower than that of the HDS algorithm. The recall of algorithms 2-HDS, 2-CHDS and 2-ICHDS were similar because it is difficult to find common nodes within 2-hops. The precision and recall of ICHDS are more dependent on the network topology as the number of hops grows. The following section is the analysis of the AHDS and PCHDS algorithms. The results show that PCHDS and AHDS have similar precision because of the minimum infection threshold in the PCHDS algorithm. When the infection probability of a candidate node is low, the adjacent nodes of the influenced nodes are selected to verified, but the idea of the AHDS algorithm is to verify these nodes which are adjacent to the influenced nodes. The recall of the PCHDS algorithm is higher than that of the AHDS algorithm because of common nodes, which means that effective searching can bring more than one influenced node and increase the total influenced nodes in the result set. Obviously, if the number of common nodes is larger, the recall of the PCHDS algorithm is better.



(a) Precision (b) Recall

Figure 8. Precision-recall of different algorithms.

5.3. Ablation experiment

We can see from the previous comparison experiments that the PCHDS algorithm is the best of our three algorithms and the AHDS algorithm is better than the HDS algorithm. In this section, we present some ablation experiments about the PCHDS and AHDS algorithms on two factors: one is common nodes, and the other is verification times.

First, we ran the two algorithms on the $s-0$ dataset. It is clear that the PCHDS algorithm cannot find common nodes on the $s-0$ dataset, as no more than one influenced node could be obtained in each verification. The two algorithms have similar precision and recall because the PCHDS algorithm selects adjacent nodes, which has maximum infection probability when the length of temporal diffusion paths with 1 indegree is one or zero. However, the PCHDS algorithm spends more time in this case because it searches temporal diffusion paths with 1 indegree at first, and then it computes the adjacent nodes infection probability.

Then, we consider the second factor: the predetermined influenced node set. The comparison experiment was carried out on the three datasets $s-0$, $s-5$ and $s-15$. The $s-0$ dataset had no common node, the $s-5$ dataset did not change the topology of the Kronecker graph and the $s-15$ dataset included more common nodes. The experiment validates the effect of verification times on the query of influenced nodes on the same dataset. All nodes on the three datasets were preset as influenced nodes, so each verification was successful. We compared the two algorithms 2-AHDS and 2-PCHDS in this case. Table 2 was the experiment result, which shows that the number of influenced nodes found in the AHDS algorithm cannot exceed the limit of verification times, while the PCHDS algorithm can breakthrough the limitation. For example, when the number of verifications is 50, the 2-PCHDS algorithm can find 53 and 58 influenced nodes on the $s-5$ and $s-15$ datasets, respectively, while the AHDS algorithm can only get 50 influenced nodes in two datasets. However, if there is no common node on the temporal diffusion paths, the PCHDS algorithm cannot find more influenced nodes, just like the $s-0$ dateset.

Table 2. Comparison of the number of verifications and influenced nodes.

Algorithms	Verification times	$s-0$	$s-5$	$s-15$
2-AHDS	50	50	50	50
	150	150	150	150
	300	300	300	300
2-PCHDS	50	50	53	58
	150	150	153	163
	300	300	311	329

The experimental results show that, if there are more common nodes in the temporal contact network and each node selected to be verified is an influenced node, more nodes than verification times can be found with the limited number of verifications. In an ideal world, more than N influenced nodes can be found given N verification times.

6. Conclusions

The goal of the influenced nodes discovery in the temporal contact network is to find as many influenced nodes as possible before the check time t under k times of verification. In order to find more influenced nodes, we consider the common nodes on temporal diffusion paths in this paper. We designed three algorithms, i.e., CHDS, ICHDS and PCHDS, which are based on common nodes. The CHDS algorithm adds a common node searching step to the HDS algorithm, and the ICHDS algorithm considers node indegree in infection probability calculation. The PCHDS algorithm takes account of the exception of common nodes and the simplicity of infection probability calculation. We did some comparison experiments and ablation experiments, which show that our methods can find more than N influenced nodes given the number of verifications N in the best case.

Use of AI tools declaration

We declare that we have not used artificial intelligence tools in the creation of this article.

Acknowledgments

This work was supported by the Qing Lan Project of Jiangsu Province and Planning Subject for the 14th Five-Year Plan of Jiangsu Education Science (C-b/2021/03/06), Future Network Scientific Research Fund Project (FNSRFP-2021-YB-60), Natural Science Fund for Colleges and Universities in Jiangsu Province (21KJB520026), the Fundamental Research Funds for the Central Universities JL (93K172020K25), Innovative Research Team Project of Suzhou Institute of Industrial Technology (2021KYTD003).

Conflict of interest

We declare that we have no conflict of interest regarding this work.

References

1. A. Guille, H. Hacid, C. Favre, D. A. Zighed, Information diffusion in online social networks: A survey, *ACM SIGMOD Record*, **42** (2013), 17–28. https://doi.org/10.1145/2503792.2503797
2. J. Byun, S. Woo, D. Kim, Chronograph: Enabling temporal graph traversals for efficient information diffusion analysis over time, *IEEE Trans. Knowl. Data Eng.*, **32** (2020), 424–437. https://doi.org/10.1109/TKDE.2019.2891565
3. J. J. Huang, T. Q. Lin, A. Liu, Z. Li, H. Yin, L. Zhao, Influenced nodes discovery in temporal contact network, in *Proceedings of the 18th International Conference on Web Information Systems Engineering*, (2017), 472–487. https://doi.org/10.1007/978-3-319-68783-4_32
4. T. Q. Lin, *Diffusion range prediction in temporal diffusion networks*, Master's thesis, Soochow University, 2017.

5. Y. X. Wang, H. Z. Wang, J. Z. Li, H. Gao, Efficient influence maximization in weighted independent cascade model, in *Proceedings of the 21st International Conference on Database Systems for Advanced Applications*, (2016), 49–64. https://doi.org/10.1007/978-3-319-32049-6_4

6. T. Iwata, A. Shah, Z. Ghahramani, Discovering latent influence in online social activities via shared cascade poisson processes, in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, (2013), 266–274. https://doi.org/10.1145/2487575.2487624

7. M. Gomez-Rodriguez, L. Song, Diffusion in social and information networks: Research problems, probabilistic models and machine learning methods, in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (2015), 2315–2316. https://doi.org/10.1145/2783258.2789991

8. M. Nickel, M. Le, Modeling sparse information diffusion at scale via lazy multivariate hawkes processes, in *Proceedings of the 21th International Conference of World Wide Web*, (2021), 706–717. https://doi.org/10.1145/3442381.3450094

9. L. Sun, Y. Rao, X. B. Zhang, Y. Lan, S. Yu, MS-HGAT: memory-enhanced sequential hypergraph attention network for information diffusion prediction, in *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, (2022), 4156–4164. https://ojs.aaai.org/index.php/AAAI/article/view/20334

10. M. Gomez-Rodriguez, J. Leskovec, A. Krause, Inferring networks of diffusion and influence, in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, (2010), 1019–1028. https://doi.org/10.1145/1835804.1835933

11. K. Kutzkov, A. Bifet, F. Bonchi, A. Gionis, STRIP: stream learning of influence probabilities, in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (2013), 275–283. https://doi.org/10.1145/2487575.2487657

12. A. Sepehr, H. Beigy, Viral cascade probability estimation and maximization in diffusion networks, *IEEE Trans. Knowl. Data Eng.*, **31** (2019), 589–600. https://doi.org/10.1109/TKDE.2018.2840998

13. H. Wang, C. Yang, C. Shi, Neural information diffusion prediction with topic-aware attention network, in *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, (2021), 1899–1908. https://doi.org/10.1145/3459637.3482374

14. D. A. Vega-Oliveros, L. Zhao, A. Rocha, L. Berton, Link prediction based on stochastic information diffusion, *IEEE Trans. Neural Networks Learn. Syst.*, **33** (2022), 3522–3532. https://doi.org/10.1109/TNNLS.2021.3053263

15. J. Y. Chen, J. Zhang, Z. Chen, M. Du, Q. Xuan, Time-aware gradient attack on dynamic network link prediction, *IEEE Trans. Knowl. Data Eng.*, **35** (2023), 2091–2102. https://doi.org/10.1109/TKDE.2021.3110580

16. F. Granese, D. Gorla, C. Palamidessi, Enhanced models for privacy and utility in continuous-time diffusion networks, *Int. J. Inf. Sec.*, **20** (2021), 763–782. https://doi.org/10.1007/s10207-020-00530-7

17. D. Gorla, F. Granese, C. Palamidessi, Enhanced models for privacy and utility in continuous-time diffusion networks, in *Proceedings of the 16th International Conference on Theoretical Aspects of Computing*, (2019), 313–331. https://doi.org/10.1007/978-3-030-32505-3_18

18. X. D. Wu, L. Y. Fu, H. Long, D. Yang, Y. Lu, X. Wang, Adaptive diffusion of sensitive information in online social networks, *IEEE Trans. Knowl. Data Eng.*, **33** (2021), 3020–3034. https://doi.org/10.1109/TKDE.2020.2964242

19. W. Wang, H. L. Yang, Y. F. Lu, Y. Zou, X. Zhang, S. Guo, et al., Influence maximization in multi-relational social networks, in *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, (2021), 4193–4202. https://doi.org/10.1145/3459637.3481928

20. T. T. Cai, J. X. Li, A. Mian, R. H. Li, T. Sellis, J. X. Yu, Target-aware holistic influence maximization in spatial social networks, *IEEE Trans. Knowl. Data Eng.*, **34** (2022), 1993–2007. https://doi.org/10.1109/TKDE.2020.3003047

21. S. X. Huang, W. Q. Lin, Z. F. Bao, J. C. Sun, Influence maximization in real-world closed social networks, *Proc. VLDB Endow.*, **16** (2022), 180–192.

22. C. Feng, L. Y. Fu, B. Jiang, H. Zhang, X. Wang, F. Tang, et al., Neighborhood matters: Influence maximization in social networks with limited access, *IEEE Trans. Knowl. Data Eng.*, **34** (2022), 2844–2859. https://doi.org/10.1109/TKDE.2020.3015387

23. G. D'Angelo, L. Severini, Y. Velaj, Influence maximization in the independent cascade model, in *Proceedings of the 17th Italian Conference on Theoretical Computer Science*, (2016), 269–274.

24. G. M. Tong, R. Q. Wang, Z. Dong, X. Li, Time-constrained adaptive influence maximization, *IEEE Trans. Comput. Soc. Syst.*, **8** (2021), 33–44. https://doi.org/10.1109/TCSS.2020.3032616

25. C. Wang, Y. M. Liu, X. F. Gao, G. H. Chen, A reinforcement learning model for influence maximization in social networks, in *Proceedings of the 26th International Conference on Database Systems for Advanced Applications*, (2021), 701–709. https://doi.org/10.1007/978-3-030-73197-7_48

26. R. Michalski, Linear threshold model in temporal networks: Seed selection for social influence, in *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, (2015), 922–923. https://doi.org/10.1145/2808797.2809346

27. C. W. Tian, M. H. Zheng, W. M. Zuo, B. Zhang, Y. Zhang, D. Zhang, Multi-stage image denoising with the wavelet transform, *Pattern Recognit.*, **134** (2023), 109050. https://doi.org/10.1016/j.patcog.2022.109050

28. C. W. Tian, Y. N. Zhang, W. M. Zuo, C. W. Lin, D. Zhang, Y. Yuan, A heterogeneous group CNN for image super-resolution, *IEEE Trans. Neural Networks Learn. Syst.*, https://doi.org/10.48550/arXiv.2209.12406

29. Q. Zhang, J. Xiao, C. W. Tian, J. C. Lin, S. Zhang, A robust deformed convolutional neural network (CNN) for image denoising, *CAAI Trans. Intell. Technol.*, 2022. https://doi.org/10.1049/cit2.12110

30. C. W. Tian, X. Y. Zhang, J. C. Lin, W. Zuo, Y. Zhang, C. Lin, Generative adversarial networks for image super-resolution: A survey, preprint, arXiv: 2204.13620.

31. C. W. Tian, Y. Xu, W. M. Zuo, B. Zhang, L. Fei, Coarse-to-fine CNN for image super-resolution, *IEEE Trans. Multim.*, **23** (2021), 1489–1502. https://doi.org/10.1109/TMM.2020.2999182

32. C. W. Tian, Y. Xu, W. M. Zuo, C. W. Lin, D. Zhang, Asymmetric CNN for image superresolution, *IEEE Trans. Syst. Man Cybern. Syst.*, **52** (2022), 3718–3730. https://doi.org/10.1109/TSMC.2021.3069265

33. S. S. Feng, G. Cong, A. Khan, X. Li, Y. Liu, Y. M. Chee, Inf2vec: Latent representation model for social influence embedding, in *Proceedings of the 34th IEEE International Conference on Data Engineering*, (2018), 941–952, https://doi.org/10.1109/ICDE.2018.00089

34. Y. Y. Liu, J. R. Gao, Z. F. Zhao, H. Wu, Z. Yue, J. Li, Evolving interest for information diffusion prediction on social network, in *Proceedings of the 25th International Conference on Advanced Communication Technology*, IEEE, (2023), 130–136. https://doi.org/10.23919/ICACT56868.2023.10079436

35. Y. Wu, H. Huang, H. Jin, Information diffusion prediction with personalized graph neural networks, in *Proceedings of the 13th International Conference on Knowledge Science*, (2020), 376–387. https://doi.org/10.1007/978-3-030-55393-7_34

36. C. Yang, H. Wang, J. Tang, C. Shi, M. Sun, G. Cui, et al., Full-scale information diffusion prediction with reinforced recurrent networks, *IEEE Trans. Neural Networks Learn. Syst.*, **34** (2023), 2271–2283. https://doi.org/10.1109/TNNLS.2021.3106156

37. N. Du, Y. Y. Liang, M. Balcan, L. Song, Influence function learning in information diffusion networks, in *Proceedings of the 31st International Conference on Machine Learning*, (2014), 2016–2024.

38. J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, Z. Ghahramani, Kronecker graphs: An approach to modeling networks, *J. Mach. Learn. Res.*, **3** (2010), 985–1042.

AIMS Press