IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 31, 2020, accepted August 30, 2020, date of publication September 10, 2020, date of current version September 24, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.3023185*

# Minimize Social Network Rumors Based on Rumor Path Tree

**SONGTAO YE[1,2], JUNJIE WANG[1,2], AND HONGJIE FAN[3]**
[1]School of Cyberspace Security, Xiangtan University, Xiangtan 411105, China
[2]School of Computer Science, Xiangtan University, Xiangtan 411105, China
[3]Department of Science and Technology, China University of Political Science and Law, Beijing 100871, China

Corresponding author: Hongjie Fan (hjfan@cupl.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61802327, and in part by the Natural Science Foundation of Hunan Province under Grant 2018JJ3511.

**ABSTRACT** Social networks have become a powerful information spreading platform. How to limit rumor spread on social networks is a challenging problem. In this article, we combine information spreading mechanisms to simulate real-world social network user behavior. Based on this, we estimate the risk degree of each node during the hazard period and analyze the hazard level that other nodes are potentially affected by when a node is infected by a rumor. We use the Rumor Path Tree (*RPT*) to analyze the rumor spreading path. By comparing the rumors and truths propagation to a certain node, the steps taken by the rumor node to propagation are estimated. In order to identify the truth node, we construct a fractional function to calculate the effective influence nodes, and select the node with the highest score from the generated *RPT* pool. Based on the truth node we effectively block the spread of rumors. Finally, experimental results and comparisons on the real datasets prove that our method is effective and efficient.

**INDEX TERMS** Social network, rumor spread, risk level, rumor path tree.

## I. INTRODUCTION

Social networks provide users with a new way to spread messages. Users can share recent updates, recommended music and videos via social networks. Due to the high openness and spread of message transmission, the network is full of false and even harmful rumors. Therefore, limiting the spread of rumors and minimizing their influence have become the challenging problem.

As shown in Fig. 1, node $U_1$ is the rumor initiator. Through rumor propagation, nodes $U_2$, $U_3$, $U_4$ are becoming recipients. In the rumor propagation, after accepting the rumor by $U_1$, node $U_2$ is not only the receiver, but also the initiator. Consequently that, $U_2$ passes the rumor to $U_3$ and $U_4$ again. Through the above propagation, node $U_3$, $U_4$ receive rumor twice.

In response to this problem, how to estimate the risk degree of each node at any time during the hazard period and choose the influencing node (*truth node*) to effectively block the rumor propagation is a challenging problem.

The associate editor coordinating the review of this manuscript and approving it for publication was Chunsheng Zhu.

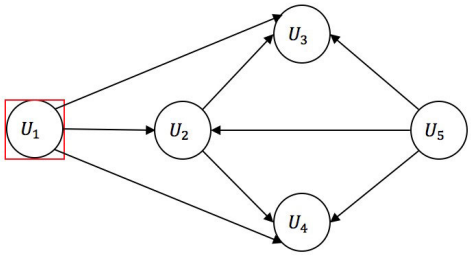



**FIGURE 1.** During social network with rumor propagation, node $U_3$ and $U_4$ receive rumor twice by $U_1$ and $U_2$.

There are two typical methods can be implemented to address this problem. The first one is to define some nodes/edges that make rumors unreachable, that is, immune nodes [1]–[4]. For example, in the above example, setting the $U_2$ node as a immune node can block on the $< U_1, U_2 >$ path. But if you want to completely block the spread of rumors, $U_2$, $U_3$, $U_4$ should also be set as immune nodes. The second strategy is to define some key nodes as the truth initiators in the social network. When rumors spread in social networks, the truth initiators also propagate the truth [5]–[7]. This strategy assumes that when the user is aware of the existence of

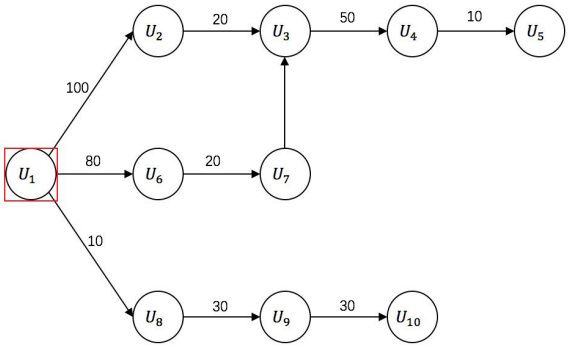



**FIGURE 2.** An Example of Social Network with Entropy.

the truth, then the user will be immune to the rumor and will not be attacked by rumors. For example, if $U_5$ is set as the truth initiator, in the same step, $U_5$ and $U_1$ can simultaneously propagate the truth/rumor to $U_2$, $U_3$, $U_4$. Then $U_2$, $U_3$, $U_4$ are protected by the truth node $U_5$. Obviously the second strategy is more efficient.

However, entropy values have an impact on rumor propagation and affect truth node selection. As shown in Fig.2.

(1) $U_1$ is the rumor initiator, and the weight on each path represents the entropy value consumed by the rumor propagation.

(2) Suppose that without considering the entropy value, if we only set one truth node in the network, it is obvious that node $U_2$ will be the best choice. Node $U_6$ always arrives at node $U_3$, $U_4$, $U_5$ before $U_1$, and can successfully protect node $U_3$, $U_4$, $U_5$ and $U_7$ from rumors.

(3) Assuming that the initial entropy value of the rumor is 100, the entropy value will vary according to the path during propagation. In this case, according to the rumor propagation minimization theory, compared to the node $U_2$, selecting and setting the node $U_8$ as truth node is the best choice.

Therefore, considering the entropy as the driving force for rumor diffusion, the process of identifying immune nodes and truth initiators is totally different from the existing works. In light of this, we have designed a solution for a rumor propagation network that the diffusion of rumors are driven by entropy values. First we estimate the risk degree of each node during the hazard period. It is to confirm the number of nodes that are potentially affected when a neighbor node is infected by a rumor, that is, the potential hazard level of any node. Subsequently, we use the Rumor Path Tree (*RPT*) to determine the infect probability between two nodes. By constructing a *RPT*, we can analyze the propagation path of rumors to determine the order in which rumors and truths propagate to a certain node. Finally, we construct a fractional function to calculate the effective influence nodes of each node in the rumor hazard period, and select the node with the highest score from the generated *RPT* pool as the truth node, which effectively block the rumor propagation. Besides, in order to validate the effectiveness of the proposed method, we have conducted a number of experimental comparisons on real data sets.

### A. PROBLEM DEFINITION

#### 1) SOCIAL NETWORK

A social network can be formally defined as $G = \{V, E\}$, where $V$ is the set of users, and $E$ represents the relationship among users. $(u, v) \in E$ indicates that there is a direct relationship $(u, v \in E)$ between user $u$ and user $v$. $\alpha_{uv} \in \{0, 1\}$ represents the correlation coefficient, and if $\alpha_{uv} = 1$, it means that there is an association relationship $(u, v)$, otherwise it does not exist. We use $p(u, v)$ to indicate the probability that user $u$ will delivery information to user $v$ and user $v$ will accept it.

#### 2) INFORMATION DIFFUSION MODEL

In social networks, the transmission mechanism of rumors is similar to the spread of infectious diseases [8]. According to the Susceptible-Infected-Recovered model (*SIRmodel*) [9]–[12], each user will always be in the following status: Susceptible, Infected, and Recovered. An susceptible status indicates that the user has not been infected by a rumor, but is rumored to be infected at any time. The infected status indicates that the user has been infected by the rumor and spread the rumor. The recovered status indicates that the user is aware of the existence of rumors and is immune to rumors.

In social networks, there are two typical diffusion models, the Linear Threshold (*LT*) model [5], [13], [14] and the Independ Cascade (*IC*) model [15]–[18].

In the *LT* model, all nodes are divided into active status and inactive status. For each node, there is a threshold $\gamma_v \in [0, 1]$. When the threshold $\gamma_v \geqslant \sum \gamma_{u \in C}$ of the node $v$ ($C$ is the set of active nodes in the pioneer node of $v$) indicates that the node $v$ transitions from the inactive status to the active node.

The *IC* model simulates two simultaneous activities, denoted as $C$ (Campaign) and $L$ (Limiting Campaign). The model represents $A_C$ as the initial set of active nodes in $C$, and $A_L$ represents the set of initial active nodes in $L$. These two events can also be considered as a kind of "good" (truth) and a kind of "bad" (rumor). Both events are simultaneously propagated in the social network. When the nodes in the two propagate to the same node at the same time, the node chooses to believe the "good" event. In this article, we choose the *IC* model as the message propagation model, and our goal is to maximize the spread of "good" activities throughout the network and minimize the spread of "bad" activities.

#### 3) WEIGHT MODEL

In a social network, each user's social status is different. We assign each user a fixed weight based on certain attributes of the user in the social network. A user with a strong weight indicates that he/she has a higher status, which means he/she has more follows. Users with significant weights pass information to users with small weights, and users who accept information are more willing to choose users who believe in weight. By assigning each user a weight value, we can more clearly define the information transfer probability $p(u, v)$ between users.

#### 4) ENTROPY MODEL

The uncertainty of the rumor can be measured by entropy. In this article, we assume that the amount of the entropy of a rumor at the beginning stage is $H$. Note that the beginning stage means that the rumor is known to no user in the social network. $H^+$ and $H^-$are used to represent the entropy of the truth and the rumor, respectively. $H = H^+ + H^-$. Due to the immobility of the entropy value, the larger $H^+$, the smaller $H^-$, and vice versa. When the rumor is initiated, $H^+ < H^-$ with the spread of rumors, $H^-$ is decreasing. We assume a constant $\varepsilon$ that is much smaller than the initial entropy value H as the critical value of the enthalpy entropy $H^-$. When $H^- \leqslant \varepsilon$, the rumor no longer propagates. In the process of rumor entropy $H^- \rightarrow \varepsilon$, the relationship between the decrease $\Delta H^-$ of each propagation entropy and the number of propagation $\zeta$ is presented as:

$$\Delta H^- = H^- e^{(-\lambda\zeta)} \tag{1}$$

where $\lambda$ is a constant. After each rumor spread is complete, we update the entropy of the rumor:

$$H^- = H^- - \Delta H^- \tag{2}$$

According to the entropy model, we can consider the process of reducing $H^-$ to $\varepsilon$ as a process in which a rumor tends to be stable and no longer propagates. It is worth noting we choose the IC model as the message propagation model, so one propagation is performed in one time step, that is, the $\zeta$ propagation of the hierarchy can be regarded as a $\zeta$ times step.

#### 5) RUMOR PROPAGATION MODEL

After clarifying the model, we formalize the rumor propagation model. $R$ is the set of rumor initiators in the social network, $Z$ is the set of truth initiators, and $|Z|$ is the number of truth initiators that exist in the network. We define $\phi(R, Z, H)$ as the rumor initiator, the truth initiator, and the set of nodes affected by the initial entropy value $H$. Our goal is to select the appropriate node in the network as the truth initiator, We denote the choice of the truth node as:

$$D^* = argmax_z(|\phi(R, \Phi, H)| - |\phi(R, Z, H)|) \tag{3}$$

where $D^*$ represents the difference of nodes. $\phi(R, \Phi, H)$ means that there is no truth and only affected by rumor $R$.

### B. OUR SOLUTION

We propose a solution to determine the $top - k$ node as the truth initiator and effectively block the rumor propagation. The solution consists of two phases:

(1) In phase one, we define the social network $G$ and the nodes that the rumor node can harm when the entrench entropy value $H^- \geqslant \varepsilon$ is specified. We estimate the potential hazard of an infected node by calculating the risk level of each node.

(2) In phase two, we generate a $RPT$ to determine the rumor propagation path and the time step of estimating the propagation of a rumor node to other nodes. Finally, we select the appropriate $top - k$ nodes from the $RPT$ pool as the truth initiator, which effectively block the rumor spreading.

### C. OUR CONTRIBUTIONS

Our main contributions in this article:

(1) We combine information spreading mechanisms to simulate real-world social network user behavior. Based on this, we estimate the risk degree of each node during the hazard period and analyze the hazard level that other nodes are potentially affected by when a node is infected by a rumor.

(2) We use the Rumor Path Tree ($RPT$) to analyze the rumor spreading path. By comparing the rumors and truths propagation to a certain node, the steps taken by the rumor node to propagation are estimated.

(3) We construct a fractional function to calculate the effective influence nodes, and select the node with the highest score from the generated RPT pool.

(4) Experimental results and comparisons on the real datasets prove that our method is effective and efficient.

We organize the paper as follows:

Section 1 is introduction. We introduce the research status of rumor communication and typical communication model. Section 2 is related work. In section 3, we propose a rumor path tree structure that creates a beta equation by analyzing the nodes in the structure. We choose the truth node by the level of the node score in section 4. We demonstrate the effectiveness and efficiency of our method by comparing the experimental results with the existing methods in section 5. Finally, section 6 gives conclusions and future work.

## II. RELATED WORKS

As early in 1940s and 1990s, a group of outstanding scholars emerged to deeply analyze the reasons for the spread of personal and group rumors [15], [16], [19], [20]. [15] demonstrated that rumors will mutate during the process of communication and construct corresponding rumor formulas. [16] studied the causes and results of rumors, and proved that rumors caused not only negative consequences. The theoretical analysis is used to extract the propagation path of rumors, and the initiators and communicators of rumors are distinguished [19].

In recent years, research on preventing the spread of false news has emerged in social networks. References [1]–[4] takes the node/edge setting level to filter the false information. References [5]–[7] controlled the spread of malicious information through the definition of anti-rumbling activities in the network. References [1]–[3] selected nodes to immunize the attacks of the rumor nodes, and sets nodes that can maximize propagation into the immune nodes in the tree structure. Reference [3] considered the user experience when blocking occurs in a social network, while using a time window to simulate the social experience when the user is blocked.

A growing body of research has shown that it is more effective to initiate a campaign to counter the spread of rumors than to set up a rumor immune checkpoint node.

Reference [7] defined a multi-objective activity independent cascade model to describe the *EIL* problem, and selects the nodes with the greatest impact through a large number of simulations. General greedy similarity algorithm to estimate the local structure of each node against the attack of false information [5], [13]. Reference [21] drawed on Sina Weibo's social network platform to analyze the relationship between its users. Reference [6] proposed the Local Shortest-Paths For Multiple Influencers(LSMI) algorithm to measure the performance of selected nodes. Reference [22] proposed a distributed expression model of users combined with emotional factors to solve the problem of serious imbalances in positive and negative cases. References [15], [19], [20] constructed the Independent Cascade Model with Login Event (IC-L) model to simulate the delay propagation process. Reference [19] proposed a regression equation to explain the relationship between the distance between nodes in a social network and the probability of being infected.

## III. SINGLE NODE ATTRIBUTE CALCULATION

### A. NODE WEIGHT AND PROPAGATION PROBABILITY

We calculate the weight $weight(v)$ of each node $v \in V$, which denoted as:

$$weight(v) = \frac{2\vartheta}{\pi} \arctan followers(v) \tag{4}$$

where $\vartheta$ is the scale factor and $followers(v)$ represents the number of node $v$. Since the transfer probability $p(u, v)$ of $u$, $v$ is based on the weight of the two nodes, we use the inverse tangent function to limit the value range $weight(v) \in [0, 1]$.

When $weight(u) \geqslant weight(v)$, the probability/transplication probability $p(u, v)$ of the truth/proverb from $u \rightarrow v$ is denoted as:

$$p(u, v) = \frac{|weight(v) - weight(u)|}{\theta} \tag{5}$$

When $weight(u) < weight(v)$, $p(u, v)$ is represented as:

$$p(u, v) = \theta \, |weight(v) - weight(u)| \tag{6}$$

where $\theta$ is to ensure that the probability of a user who is moving from a powerful user to a user with a small weight. Which is a constant and $\theta \in [0, 1]$.

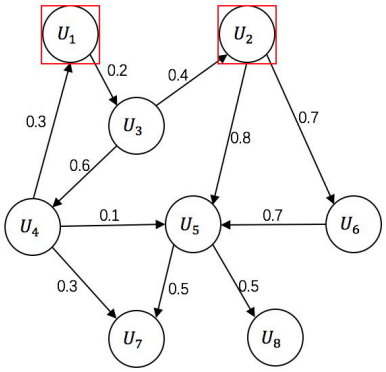
### B. NODE INFLUENCE

We define the influence of a node $u$ on the successor nodes, denoted as $L_{uv}$. It represents the probability that $v$ node is only affected by the pioneer node $u$ and not by other pioneer nodes.

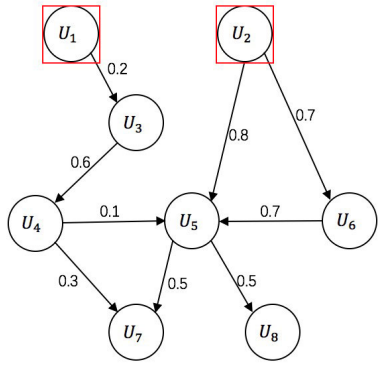
Assuming that $Q$ is the set of pioneer nodes of node $u$, then the influence of $u$ on successor node $v$ is:

$$L_{uv} = p(u, v) \prod_{w \in Q \setminus \{u\}} (1 - p(w, v)) \tag{7}$$

The influence of a node can indicate the hazard level of a node to its successor node when the entropy is greater than the particular value. It is needed to be clear that nodes with larger entrench entropy have higher risk levels. When the entropy of the $u$ is greater than $\varepsilon$, $u$ poses a threat to the successor node. When $H^-$ is larger, the number of times




(a) G and Propagation Probability $p(u, v)$




(b) DAG Corresponding to G

**FIGURE 3.** DAG Representation of Social Networks.

$u$ is propagated will also increase. And after being propagated to the successor nodes, the successor nodes will have more rumor entropy values and carry out the next round of propagation.

We perform a Depth-First Search (DFS) algorithm by $\zeta$ steps based on the period of rumor propagation, and form the nodes involved in the algorithm into a set $S$. Based on $S$, we perform an Acyclic algorithm [23] on it to find the Directed Acyclic Graph (DAG). It is also for us to build the *RPT* structure in the next step.

As shown in Fig.3(a), a social network graph $R = \{U_1, U_2\}$, the number on the relationship represents the propagation probability $p(u, v)$. Fig.3(b) is the *DAG* processed by the Acyclic algorithm, where $\{U_3, U_6, U_4, U_5, U_7, U_8\}$ is a topological sorting structure in the *DAG*.

### C. NODE RISK LEVEL

We define $risk(u, t)$ of node $u$ at time $t$, which represents the expected number of influences. If $t = 0$ and $risk(u, 0) = 1$, then it indicates that the number of nodes affected by the $u$ node is 1. $risk(u, t)$ denoted as:

$$risk(u, t) = \sum_{v \in C} (L_{uv} \sum_{s=1}^{t} (weight(v) risk(v, t - s))) \tag{8}$$

where $C$ is the set of successor nodes of $u$. The higher the degree of risk, the greater the hazard of the node at this time. Similarly, the risk degree is also an important parameter in

our final score function. We present algorithm 1 as node risk degree computation.

**Algorithm 1** Risk Degree Computation

**Input:**

1. Graph $G = (V, E)$.
2. Initialized Rumor's Entropy $H^-$ and Critical Value $\varepsilon$.
3. Rumor starter $R$.

**Output:**

Nodes in Set $S$ reachable by $R$ and Their Risk Degree.

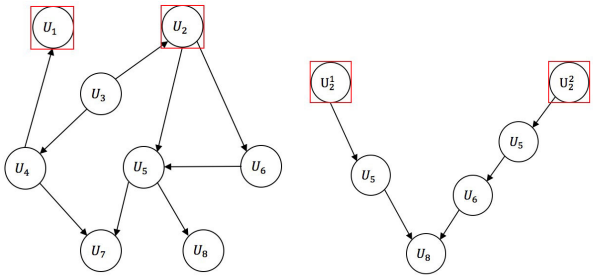
1: Compute propagate times $\zeta$ with $H^-$, $\varepsilon$;
2: Initialize $S = \varnothing$;
3: **for** each $u \in R$ **do**
4: Perform $DFS$ from $u$ and insert visited nodes with $\zeta$ hops into $S$;
5: **end for**
6: Apply Acyclic on $S$ to generate a $DAG$ and a topological ordering;
7: **for** each node $u$ in the topological ordering **do**
8: **for** each successor $v$ of $u$ **do**
9: compute $L_{uv}$ with Eq.(7);
10: **end for**
11: $risk(u, 0) = 1$;
12: **end for**
13: **for** $t = 1, \ldots, \zeta$ **do**
14: compute $risk(u, t)$ with Eq.(8);
15: **end for**
16: Return $S$ and $risk(u, t)$ for all $u \in S, t = 0, 1, \ldots, \zeta$;

Algorithm 1 gives the process of calculating the risk level of any node. Line 1 indicates the critical value $\varepsilon$ and the number of propagation times $\zeta$ based on the initial value $H^-$ of the rumor that it no longer propagates. Lines 2 to 4 represent a $\zeta - step\ DFS$ algorithm for the rumor initiator $R$ to derive the range of nodes that the rumor can affect. Line 5 represents the use of the Acyclic algorithm to obtain the $DAG$ map and topological ordering of $S$. Lines 7-8 indicate that for each node $u$, the influence $L_{uv}$ on its successor node $v$ is calculated using Eq.(7). Lines 10-11 indicate that the risk degree $risk(u, t)$ of the $u$ node at time $t = 1, \ldots, \zeta$ is calculated using Eq.(8). Finally, in the 16th line, we return $S$ and the risk degree for all $u \in S, t = 0, 1, \ldots, \zeta$.

## IV. TRUTH NODE SELECTION

### A. RPT GENERATION

We operate on a network graph $G$ containing the rumor initiator $R$. According to the $RIS$ algorithm [24], we calculate the propagation probability $p(u, v)$ by weight, and remove the edges in the network with the probability of $1 - p(u, v)$ to obtain the simple graph $g$. In the simple graph $g$, we perform a reverse Breadth First Search (BFS) algorithm on the node $r \in S$ to generate an $RPT$ structure rooted at $r$. Whenever a node $u$ is reached, we create the corresponding node and add the node and the edge it is connected to into the $RPT$ structure. If the node $v$ has already been accessed, then copy the $v$ node again.




(a) Sample Graph g (b) RPT $T_{u_8}$

**FIGURE 4.** Generate a RPT structure.

If the created $RPT$ structure does not contain the rumor initiator $R$, then it is not considered and removed until the iteration is terminated. By generating an $RPT$ structure, we can clarify all the rumor propagation paths in the network. We analyze the degree of danger based on the distance between the node and the rumor initiator in the tree structure path, and also the cost of evaluating the rumor and the truth to a particular node in the tree. It is of significance for us to choose the truth node.

Fig.4(a) is a sample graph $g$ generated after the processing of Fig.3(a). Fig.4(b) is a $RPT$ structure diagram $T_{u_8}$ of node $U_8$. Since there are two paths from the rumor node $U_2$ to the node $U_8$, we create two $U_2$ nodes in the $RPT$ tree and the $U_5$ nodes passing by. In order to distinguish the two paths, the rumors are respectively denoted as $U_2^1$ and $U_2^2$.

We use $T_r$ to represent the $RPT$ structure of a $(\zeta + 1)$ layer of noder. Each path $p \in T_r$ from $v$ to its descendant noder also corresponds to the path from $v$ to $r$ in the simple network $g$. Each node $v$ in $T_r$ is combined into one $(\zeta+1)$ layer vector $B_v$, and the probability that the node $v$ of the $j - th$ layer reaches the root node in step $j$ is denoted as $B_v[j]$. The vector of the root node $r$ is represented as $B_r = [1, 0, \ldots, 0]$.

We assume that $v$ is the $d - th$ layer in the $RPT$ structure, then it needs at least $d$ steps to reach the root node $r$. In other words, there are at least $d$ nodes on the path from $v$ to $r$. When $i < d$, the probability that $v$ can reach $r$ is zero. When $i \geqslant d$, let $w$ be the $v$ node to reach the current successor node on the path of the root node $r$. Then the probability that $v$ can reach $r$ in step $i$ multiplied by the probability that $w$ reaches $i$ in step $i - j$, is denoted as

$$B_v[i] = \begin{cases} 0, & i < d \\ \sum_{j=1}^{i} weight(w) B_w[i-j], & Otherwise \end{cases} \tag{9}$$

After modeling the rumor path into a vector structure, we determine the probability that the node $u$ reaches the root node $r$ at time $t$ and before the rumor in the $RPT$ tree structure, and such a probability is denoted as $\beta(u, T_r, t)$. It helps us to determine the order in which rumors and truths arrive at a particular node as they propagate, thereby prioritizing rumors and truth.

Let $R' \subset R$ denote the rumor initiator in $T_r$. We have the following definitions:

(1) If $u$ is the pioneer node of a rumor initiator $w \in R'$, $\beta(u, T_r, t) = 0$, sincew always arrives at $r$ before $u$.

(2) If all the nodes in $R'$ are the pioneer nodes of $u$, then $u$ will always arrive at $r$ before any rumors. $\beta(u, T_r, t) = B_u[t]$. That is, the probability of $\beta(u, T_r, t)$ is the probability that $u$ reaches the root node at time $t$.

(3) If $\exists R_u \subseteq R'$ and node in $R_u$ is neither the pioneer node of $u$ nor the successor node of $u$, and $R_u \neq \varnothing$. Obviously, there is a rumor that there is a rumor that $w \in R_u$ reaches $r$ from 0 to time $t-1$ as $\prod_{s=0}^{t-1}(1 - B_w[s])$. However, there is no rumor from 0 to time $t-1$. The probability that $w \in R_u$ reaches $r$ is $\prod_{w \in R_u} \prod_{s=0}^{t-1}(1 - B_w[s])$.

In summary, we can get:

$$\beta(u, T_r, t) = B_u[t] \times \prod_{w \in R_u}(\prod_{j=0}^{t-1}(1 - B_w[j])) \quad (10)$$

If $u$ reaches the root node $r$ in the $RPT$ tree at time $t$ than any of the rumors of $R'$, then all nodes potentially affected by $r$ in the entire network $G$ can be prevented. If the $u$ node is used as the origin initiation point, it can propagate to the root node $r$ before other nodes in the existing $RPT$ tree structure, then the $r$ node will not be affected by any rumors. Correspondingly, the nodes whose $r$ nodes are potentially affected in the entire network will not be attacked by rumors.

Reviewing the previously calculated risk level $risk(r, t)$ can give the expected number of influences at time $t$. Then, from $t$ to $\zeta$, the sum of nodes $r$ can affect other nodes is denoted as $\sum_{s=0}^{\zeta - t} risk(r, s)$.

We construct the fractional function of node $u$ based on the work done before. The score obtained by this fractional function indicates the sum of the number of nodes that can be effectively affected by node $u$ in the order $\zeta$ propagation, which is denoted as:

$$score(u, T_r, \zeta) = \sum_{t=1}^{\zeta}(\beta(u, T_r, t) \cdot \sum_{j=0}^{\zeta - t} risk(u, j)) \quad (11)$$

The fractional function consists of two important parameters. The risk level indicates the sum of the nodes that the node can affect over a period of time. The $\beta$ function represents the probability that the node will reach the root node before the rumor in the rumor path tree structure. Through these two parameters, we can summarize its influence at the macro level (the whole social network) and the micro (each $RPT$) level, so such a score can best prepare for the influence of a node as a truth node. We generates the $RPT$ structure and computes the node scorein Algorithm 2.

We first randomly extract a node $r$ as the root node of the $RPT$ in $S$, and implement a $BFS$ algorithm in reverse along the path of the node $r$ pioneer pointing to itself. For each node $v \in F$, calculate their $B_v$ vector in line 6. We determine if the $v$-node is a rumor initiator. If not, add a copy of node $v$ to queue $F$. If $v$ is a rumor initiator, then we end the traversal of the current branch and move horizontally, while removing node $v$ from $R_u$. After building the $RPT$ structure $T_r$, we check in line 15 whether there are rumors in the tree. $R_u = \varnothing$ indicates that node $u$ is a successor of all rumors, define $\beta(u, T_r, t) = B_u[t]$. Otherwise calculate $\beta(u, T_r, t)$ according to Eq.(10). Finally, we calculate the score of node $u$ and return $RPT$ through Eq.(11).

**Algorithm 2** Generate a Rumor Paths Tree

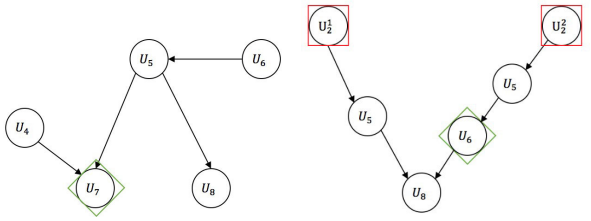
**Input:**
1. $GraphG = (V, E)$, Hops $\zeta$, Rumor starters $R$.
2. Set S of nodes reachable from $R$.
3. Risk Degree $risk(u, t)$ for each $u \in S$.

**Output:**
$A$ $RPT$.

1: Initialize processing queue $F = \varnothing$;
2: Randomly choose a node $r \in S$;
3: $F.enqueue(r)$;
4: **while** $F \neq \varnothing$ **do**
5: $v = F.dequeue()$;
6: Compute $B_v$ with Eq.(9);
7: **if** $v \notin R$ **then**
8: **if** $BFS$ is within $\zeta$ levels from $r$ **then**
9: Create a copy of $u$;
10: $F.enquenue(u)$;
11: Initialize $R_u = R$;
12: **end if**
13: **else**
14: **for** each node $u$ on the path from $v$ to $r$ **do**
15: $R_u = R_u\{v\}$;
16: **end for**
17: **end if**
18: **end while**
19: **if** $T_r$ contains any node in $R$ **then**
20: **for** each node $u$ in the tree **do**
21: **if** $R_u = \varnothing$ **then**
22: **for** $i = 0, 1, \ldots, \zeta$ **do**
23: $\beta(u, T_r, t) = B_u[t]$;
24: **end for**
25: **else**
26: **for** $i = 0, 1, \ldots, \zeta$ **do**
27: compute $\beta(u, T_r, t)$ with Eq.(10);
28: **end for**
29: **end if**
30: Compute $score(u, T_r, \zeta)$ with Eq.(11);
31: **end for**
32: **return** the generated $RPT$;
33: **else**
34: **return** **void**;
35: **end if**

### B. NODE UPDATE AND NODE SELECTION

After determining the score for each node, we need to select nodes with high scores as the truth nodes among the huge social networks. In order to simplify the problem and consider some cases of node conflicts, we perform a modular operation on all nodes in the network. We use the Dynamic: Stop-and-Stare ($D-SSA$) algorithm [25] to generate a random $RPT$ pool, and all nodes in the social network form a number of $RPT$ structures by random sampling. The $D-SSA$ algorithm




(a) Update DAG (b) Update RPT

**FIGURE 5.** Update Node Score.

can be seen as a process of generating an independent *RR* set in two stages, where the first stage is to find the largest subset, and the second stage is to evaluate the influence of the subset. The above $score(u, T_r, \zeta)$ represents the sum of the potential protection nodes of the node $u$ in the *RPT* structure with $r$ as the root node. As shown in Fig.5(a), if the $U_7$ node is a truth initiator, the risk level of $U_7'$s pioneer nodes $U_4$, $U_5$ will be reduced because they cannot affect $U_7$ as a truth node. If $U_4$ or $U_5$ is also the root node of an *RPT*, their risk level will be updated to:

$$risk(r,t) = risk(r,t) - L_{ru} \cdot \left(\sum_{s=1}^{t} weight(u) \cdot risk(u, t-s)\right) \quad (12)$$

After updating the risk level of such a node, we recalculate the scores of the nodes in its corresponding *RP* tree.

In addition, we cannot ignore the fact that the selected truth node $r$ also exists in another *RPT* structure. As shown in Fig.5(b), the node scores in the *RPT* may change at this time. It is because when node $u_6$ is selected as the truth node, the node in the *RPT* not only needs to reach the root node before the rumor, but also needs to arrive before the $U_6$ node. Otherwise, the $r$ node has been affected by the $U_6$ truth node, and it is meaningless to select a node $v$ as the truth node. When this happens, we define:

$$\Delta\beta(u, T_r, t) = \beta(Z \cup \{v\}, T_r, t) - \beta(Z, T_r, t) \quad (13)$$

For each $T_r$ where node $u$ exists, we replace $\Delta\beta(u, T_r, t)$ with $\beta(u, T_r, t)$ and then recalculate the fraction of the nodes.

Based on the last updated score for each node, we can clearly see the number of potential protection nodes per node as the truth initiator. We can choose the top-k node as the truth point in the social network, thus minimizing the impact of rumors on the network and blocking the spread of rumors.

Algorithm 3 gives the process of picking $k$ nodes from the *RPT* samples as the truth node. Line 1 first uses the $D-SSA$ algorithm to generate a series of *RPT* pools. Then we select the node $u$ with the highest score among all *RPTs*. Since some nodes are selected as the truth node, it may cause the pioneer node's risk degree to change, so we update its risk level and recalculate the score of the node in the *RPT* with $u'$s pioneer node as the root node. Lines 12-17 are the scores of other nodes in the *RPT* that have the truth node present. After all the scores have been updated, we select the node with the highest score from all the nodes as our truth initiating node. Repeat the previous steps until the node selection is complete.

**Algorithm 3** Node Selection

**Input:**
1. Number of nodes to select $k$.
2. Hops $S$.
3. Nodes reachable from rumor starters $S$.
4. Risk Degree $risk(u, t)$ for $u \in S$.

**Output:**
Set $S$ of nodes reachable by $R$ and their risk degree.

1: call $D-SSA$ to generate a pool of *RP* tree $\psi$;
2: Initialize $Z = \varnothing$;
3: **repeat**
4: Let $u$ be the node with the highest $score(u, T_r, \zeta)$;
5: $Z = Z \cup u$;
6: **for** each pioneer $v$ of $u$ **do**
7: **for** $t = 0, \ldots, \zeta$ **do**
8: Update $risk(u, t)$ with Eq.(12);
9: **end for**
10: **for** each *RP* tree $T_v$ **do**
11: **for** each $w$ in $T_v$ **do**
12: Update $score(u, T_r, \zeta)$ with Eq.(11);
13: **end for**
14: **end for**
15: **end for**
16: **for** each $T_r \in \psi$ involving $u$ **do**
17: set $score(u, T_r, \zeta)$=0;
18: **for** each node $w$ in $T_r$ **do**
19: **for** $t = 1 \ldots, \zeta$ **do**
20: Compute $\Delta\beta(u, T_r, t)$ with Eq.(13);
21: **end for**
22: Update $score(u, T_r, \zeta)$ with Eq.(11);
23: **end for**
24: **end for**
25: **until** $|Z| = k$;
26: **return** $Z$;

## V. EXPERIMENTAL RESULTS AND ANALYSIS

### A. EXPERIMENTAL SETTING

We acquired more than 500,000 user nodes and their associations on the *Zhihu website*[1] using the Scrapy framework-based *crawler*[2]. We use the *Neo4j*[3] graph database storing all these nodes and relationships. We set up the experimental configuration parameters and created different training sets so that our comparison experiments can be performed in different environments.

All experiments were run on 2.2 GHz Intel Core i7 CPUs,16 GB 1600 MHz DDR3 RAM, and Mac 10.13.3 operating systems.

We extracted 100, 1000, 10,000, and 100,000 user training sets in the database and tested them as $T_1$, $T_2$, $T_3$, and $T_4$. The user and relationships in each training set are shown in Table 1.

[1]https://www.zhihu.com
[2]https://scrapy.org
[3]https://neo4j.com

**TABLE 1. Statistics of User and Relationships among Training Sets.**

| Training Set | Number of Users | Number of Relationships |
|---|---|---|
| $T_1$ | 100 | 612 |
| $T_2$ | 1000 | 40628 |
| $T_3$ | 10000 | 534738 |
| $T_4$ | 100000 | 2188229 |

**TABLE 2. Statistics of Rumors and Truth Initiators among Training Sets.**

| Training Set | Number of Rumors | Number of Truth Initiators |
|---|---|---|
| $T_1$ | 5 | $1 \sim 5$ |
| $T_2$ | 10 | $2 \sim 10$ |
| $T_3$ | 20 | $3 \sim 15$ |
| $T_4$ | 50 | $5 \sim 25$ |

We measure the effectiveness of each method by Salvation Ratio (*SR*) [1], [20], [26]. It gives the protected nodes proportion by setting truth node, which denoted as:

$$SR(Z) = \frac{\phi(R, \varnothing, H) - \phi(R, Z, H)}{\phi(R, \varnothing, H)} \tag{14}$$

where $\phi(R, Z, H)$ represents the user set under the influence of the initial entropy value $H$, the spoof initiator $R$ and the truth initiator $Z$.

As shown in Table 2, in order to facilitate the experimental argumentation, we first compare a set of cases that do not consider the entropy value of the rumor, that is, $H^- = \infty$. The number of rumors spread on the network is not limited until the entire propagation is completed. Then we set the appropriate initial enthalpy value $H^-$ and the critical value $\varepsilon$ so that the number of rumors spread $\zeta = 10$. At the same time, we set the appropriate number of rumor initiators and truth nodes in the four training sets, which makes the experimental results more representative.

### B. EXPERIMENTAL EVALUATION AND ANALYSIS

We evaluate the effectiveness and efficiency of our approach by comparing it with other methods:

- PageRank [27]: This method selects nodes by page rank (PageRank) score.
- LSMI [6]: This method evaluates the influence of each node through the shortest path, and selects the node with high influence as the truth node.
- LargeInf [7]: This method estimates the score of the node on the reachable path of the rumor node through the simulation method, and also selects the truth node based on such a score.

We set the appropriate rumor entropy and the critical value. We set the number of propagation is $\zeta = 10$. In other words, all we have to do is choose the appropriate truth node within the ten spreads of the rumor. The performance of the four methods under such settings is shown in Fig.6-Fig.9. We observe that when $k = 3$, our method exceeds the rescue rate by 55% over the second-ranked LargeInf method and by

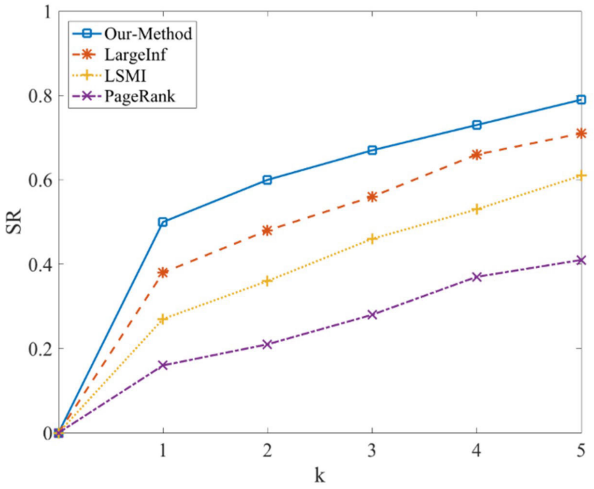



**FIGURE 6.** SR on $T_1$ training results when $\zeta = 10$.

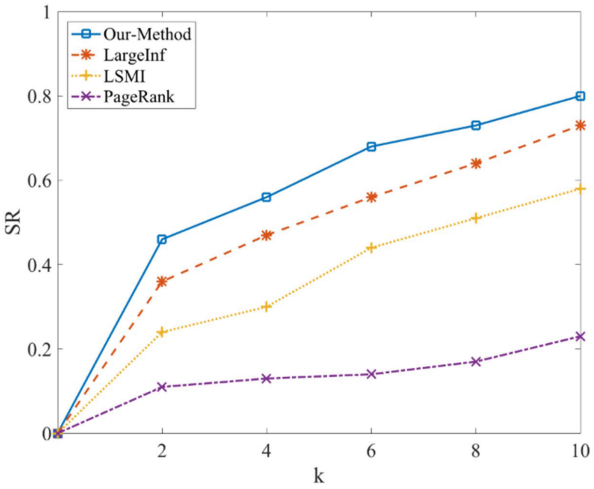



**FIGURE 7.** SR on $T_2$ training results when $\zeta = 10$.

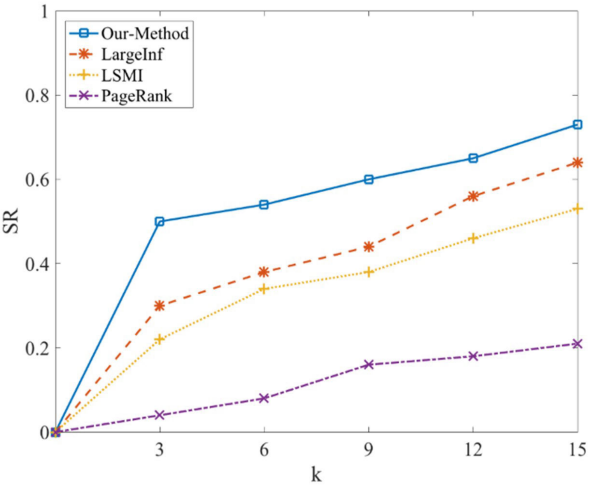



**FIGURE 8.** Runtime on $T_3$ training set when $\zeta = 10$.

22% when $k = 15$. It is because our method is related to the risk node level when looking for the truth node. In the

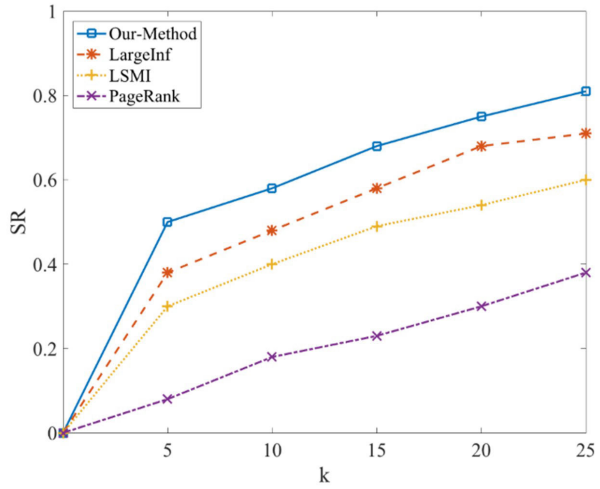



**FIGURE 9.** Runtime on $T_4$ training set when $\zeta = 10$.

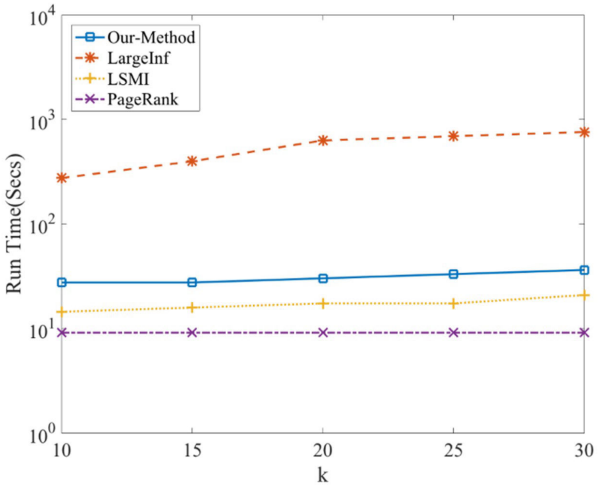



**FIGURE 12.** SR on $T_3$ training results when $\zeta = 10$.

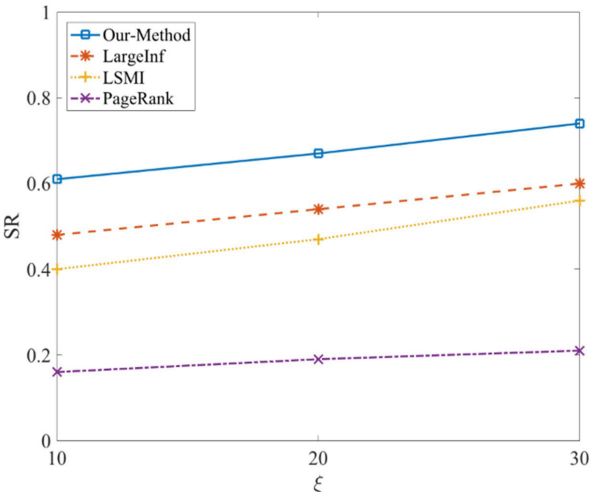



**FIGURE 10.** SR on $T_3$ training results when k=20.

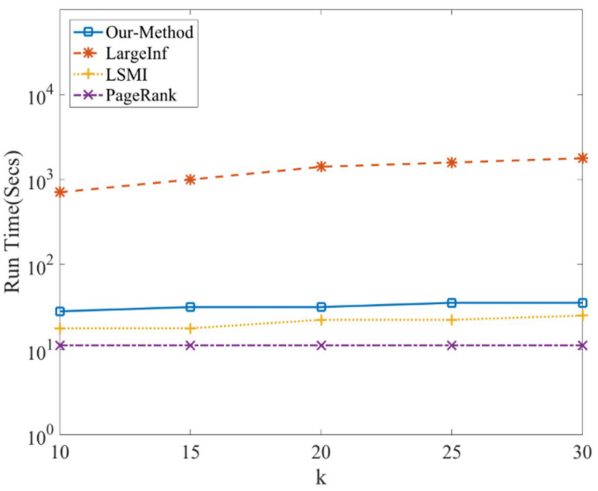



**FIGURE 13.** SR on $T_4$ training results when $\zeta = 10$.

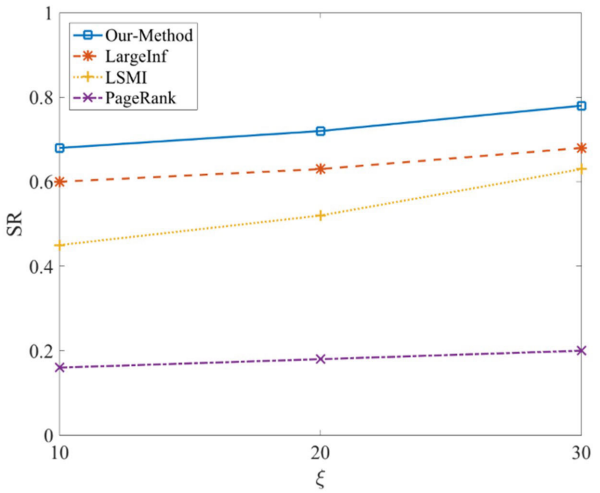



**FIGURE 11.** SR on $T_4$ training results when k=20.

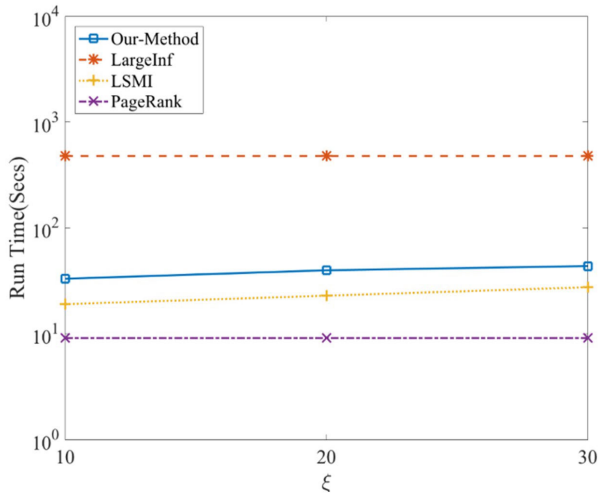



**FIGURE 14.** Runtime on $T_3$ training set when k=20.

high-density social network, the threat caused by the node with higher risk will be greater.

We compare the efficiency of the four methods in the two largest training sets, $T_3$ and $T_4$. Fig.10 and Fig.13 are the

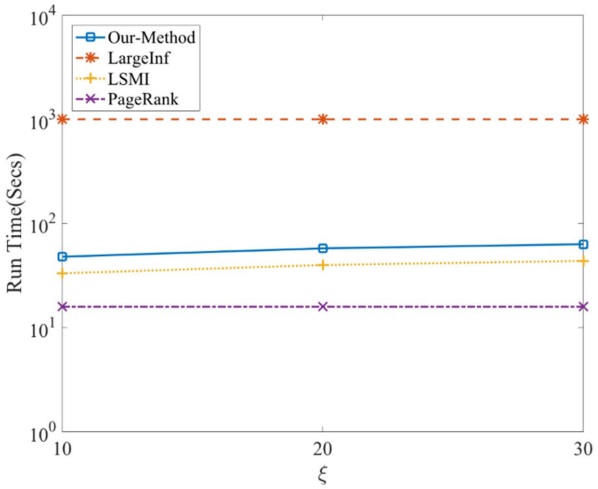



**FIGURE 15.** Runtime on $T_4$ training set when k=20.

runtime comparisons where we set the number of rumors $\zeta$ to 10 and set different truth node $k$. We can see that LargeInf takes the longest time among the four methods. It is because in the LargeInf algorithm, in order to select a truth node, it needs to run a large number of simulations to see how many other nodes the selected node can affect when it is infected by the rumor. In addition, selecting a node can greatly affect the propagation of rumors: Therefore, as $k$ increases in the runtime of LargeInf increase significantly.

*LSMI* runs slightly faster than ours because it only considers the local structure of the nodes in the network and focuses on finding the shortest path among them. PageRank is the fastest of the four methods because it is simply a topology between nodes, ignoring the weight of nodes and the initiators of rumors.

Fig.14 and Fig.15 show that we set $k$ to 20 and compare the four methods in the $T_3$ and $T_4$ training sets during different rumor propagation periods. As the propagation period increases, we can see that the four methods have only slightly increased their runtime.

## VI. CONCLUSION

Aiming at the problem of rumor propagation in social networks, we construct a multi-level propagation model based on entropy weight. By analyzing the propagation path of the rumor, we use the specific node as the root of the rumor path tree structure in the active period of the rumor. We construct a fractional function to evaluate the number of nodes that can potentially affect an arbitrary node as a truth node. By ranking the node scores, we can select the $top-k$ node with a high score as the truth initiator node. The experimental results show that our method is better than some existing methods in terms of effectiveness and efficiency.

In future work, we will classify the types of rumors, optimize the model parameters, and consider to apply our framework on some large scale real datasets to verify the efficiency.

## REFERENCES

[1] C. Song, W. Hsu, and M. L. Lee, "Node immunization over infectious period," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, 2015, pp. 831–840.

[2] Y. Zhang and B. A. Prakash, "Data-aware vaccine allocation over large networks," *ACM Trans. Knowl. Discovery from Data*, vol. 10, no. 2, pp. 1–32, Oct. 2015.

[3] B. Wang, G. Chen, L. Fu, L. Song, and X. Wang, "DRIMUX: Dynamic rumor influence minimization with user experience in social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 10, pp. 2168–2181, Oct. 2017.

[4] Y. Zhang and B. A. Prakash, "Scalable vaccine distribution in large graphs given uncertain data," in *Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manage.*, 2014, pp. 1719–1728.

[5] X. He, G. Song, W. Chen, and Q. Jiang, "Influence blocking maximization in social networks under the competitive linear threshold model technical report," 2011, *arXiv:1110.4723*. [Online]. Available: http://arxiv.org/abs/1110.4723

[6] J. Tsai, Y. Qian, Y. Vorobeychik, C. Kiekintveld, and M. Tambe, "Bayesian security games for controlling contagion," in *Proc. Int. Conf. Social Comput.*, Sep. 2013, pp. 33–38.

[7] C. Budak, D. Agrawal, and A. El Abbadi, "Limiting the spread of misinformation in social networks," in *Proc. 20th Int. Conf. World Wide Web*, 2011, pp. 665–674.

[8] F. Brauer, C. Castillo-Chavez, and C. Castillo-Chavez, *Mathmiscts models population Biol. epidemiology*, vol. 2. Cham, Switzerland: Springer, 2012.

[9] E. Beretta and Y. Takeuchi, "Global stability of an SIR epidemic model with time delays," *J. Math. Biol.*, vol. 33, no. 3, pp. 250–260, Dec. 1995.

[10] B. Shulgin, "Pulse vaccination strategy in the SIR epidemic model," *Bull. Math. Biol.*, vol. 60, no. 6, pp. 1123–1148, Nov. 1998.

[11] L. A. Meyers, M. E. J. Newman, and B. Pourbohloul, "Predicting epidemics on directed contact networks," *J. Theor. Biol.*, vol. 240, no. 3, pp. 400–418, Jun. 2006.

[12] H. W. Hethcote, "The mathematics of infectious diseases," *SIAM Rev.*, vol. 42, no. 4, pp. 599–653, 2000.

[13] B. Liu, G. Cong, D. Xu, and Y. Zeng, "Time constrained influence maximization in social networks," in *Proc. IEEE 12th Int. Conf. Data Mining*, Dec. 2012, pp. 439–448.

[14] H.-H. Chen, Y.-B. Ciou, and S.-D. Lin, "Information propagation game: A tool to acquire humanplaying data for multiplayer influence maximization on social networks," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 1524–1527.

[15] W. Chen, W. Lu, and N. Zhang, "Time-critical influence maximization in social networks with time-delayed diffusion process," in *Proc. AAAI Conf. Artif. Intell.*, 2012, pp. 1–10.

[16] X. He, M. Gao, M.-Y. Kan, Y. Liu, and K. Sugiyama, "Predicting the popularity of Web 2.0 items based on user comments," in *Proc. 37th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2014, pp. 233–242.

[17] C. Wang, W. Chen, and Y. Wang, "Scalable influence maximization for independent cascade model in large-scale social networks," *Data Mining Knowl. Discovery*, vol. 25, no. 3, pp. 545–576, Nov. 2012.

[18] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins, "Information diffusion through blogspace," in *Proc. 13th Conf. World Wide Web*, 2004, pp. 491–501.

[19] E. Cohen, D. Delling, T. Pajor, and R. F. Werneck, "Distance-based influence in networks: Computation and maximization," 2014, *arXiv:1410.6976*. [Online]. Available: http://arxiv.org/abs/1410.6976

[20] C. Song, W. Hsu, and M. L. Lee, "Targeted influence maximization in social networks," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2016, pp. 1683–1692.

[21] F. Huang, G. Yu, J. Zhang, C. Li, C. Yuan, and J. Lu, "Mining topic sentiment in micro-blogging based on micro-blogger social relation," *Ruan Jian Xue Bao/J. Softw.*, vol. 28, no. 3, 2017, Art. no. 694707.

[22] S. Liu, H. Zheng, H. Shen, X. Liao, and X. Cheng, "Learning sentimental influences from Users' behaviors," 2016, *arXiv:1608.03371*. [Online]. Available: http://arxiv.org/abs/1608.03371

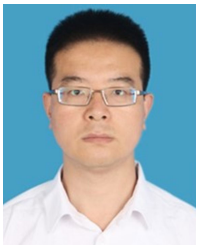
[23] D. Erdös, V. Ishakian, A. Lapets, E. Terzi, and A. Bestavros, "The filter-placement problem and its application to minimizing information multiplicity," 2012, *arXiv:1201.6565*. [Online]. Available: http://arxiv.org/abs/1201.6565

[24] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier, "Maximizing social influence in nearly optimal time," in *Proc. ACM-SIAM Symp. Discrete Algorithms*, Jan. 2014, pp. 946–957.

[25] H. T. Nguyen, M. T. Thai, and T. N. Dinh, "Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks," in *Proc. Int. Conf. Manage. Data*, 2016, pp. 695–710.
[26] C. Song, W. Hsu, and M. L. Lee, "Temporal influence blocking: Minimizing the effect of misinformation in social networks," in *Proc. IEEE 33rd Int. Conf. Data Eng. (ICDE)*, Apr. 2017, pp. 847–858.
[27] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the Web," Stanford Digit. Library Technol. Project, Tech. Rep. SIDL-WP-1999-0120, 1998.

**SONGTAO YE** received the Ph.D. degree from Hunan University, in 2011. His research interests include data mining, knowledge graph, and named entity extraction.

**JUNJIE WANG** is currently pursuing the master's degree with Xiangtan University. His research interest includes truth discovery.



**HONGJIE FAN** received the Ph.D. degree in computer science and engineering from Peking University, Beijing, in 2018. His research interests include data exchange, knowledge graph, and named entity extraction.