Computer Networks 120 (2017) 105–125

Contents lists available at ScienceDirect

Computer Networks

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

# Trust Traversal: A trust link detection scheme in social network

Zhang Bo, Zhang Huan, Li Meizi*, Zhao Qin, Huang Jifeng

*College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai, 200234, China*

ARTICLE INFO

*Article history:*
Received 14 May 2016
Revised 10 February 2017
Accepted 7 April 2017
Available online 8 April 2017

*Keywords:*
Social network
Trust Traversal
Trust
Reputation
Trust link

ABSTRACT

The nature of free communication results in increased challenges with respect to the reliability of user interaction in online social network. It is difficult to ensure that the users in a large-scale network are familiar with one another and that they interact without risk. Users' behaviors might therefore be vulnerable to attacks or frauds. Trust relationship detection scheme is a feasible solution. This study proposes an original trust link detection scheme, named Trust Traversal, which includes three aspects, i.e., trust scheme, trust link category, and traversal scheme, which are used to build up strong or weak trust links among users. Firstly, we address overview of Trust Traversal scheme and its related formal definitions. Then, trust scheme model is proposed for managing calculation methods of subjective trust, reputation, and indirect link. Subjective trust is calculated based on past interaction data, while reputation is defined as collective objective trust with three factors: effective accumulation, time attenuation, and intensity factor. Additionally, trust of indirect link in trust scheme is computed with three rules: trust transitivity rule, reputation weight rule, and level attenuation rule. To describe the relationship with both reliability and closeness, we present a trust link category, which comprise three kinds of trust links: strong trust link, weak trust link, and untrusted link, and meanwhile, gives the classifying method based on two factors: mutual trust factor and interaction active factor. Moreover, Trust Traversal scheme for both single and multiple target users is proposed to detect strong or weak trust links, which implies trustable and close relationships among users. In Trust Traversal scheme, each step is determined by trust based probability, and thus the user with the strongest trust link would have the highest probability of being the traversal user in the trust link. And also, terminal mechanism of Trust Traversal scheme is discussed. Finally, we provide an examination to determine further the performance of our proposed scheme.

## 1. Introduction

Social networks have yielded the greatest explosion of online information exchange in the Web 2.0 era [1]. People experience happiness from convenient interactions through social relationships, including friends, followers, and fans. Information is propagated through paths, which are composed of users' relationships [2,3]. This relation-oriented information dissemination plays a significant role in social network information services, such as online recommendation, item purchasing, and business consulting. There is no doubt that social relationships are among the key factors in the high achievement of social networks.

However, the relationship-based information propagation also boosts a new pattern, "*go vital*," which poses many problems for users in online social networks, for instance, spam information inundation, fake recommendation, malicious frauds, and even network crimes. That means that relationships, especially those that are unreliable, unfamiliar, or anonymous, are potential threats even though they play positive roles in social network interactions. This thus creates an urgent need for the discovery of trustworthy links among users for reliable daily communication in social networks.

Essentially, in such a large-scale network environment, it is impossible for a user to recognize millions of other individuals to select trustable links for their interactions. That is, users must select strangers without sufficient trust evidence to establish interconnections, and this brings potential threats to users. Trust has recently been suggested as an effective security mechanism to evaluate indirect reliability and to mitigate threats within networked environments [4–6]. By utilizing trust calculation, users can have indirect trust opinions towards strangers in accordance with the results of trust calculation.

Traditionally, trust mechanisms have focused on providing trust relationships (direct or indirect relationships) among users based on the global data of related users [2,6]. However, such mechanisms are not sufficient because social networks comprise a large number of users and users face complex relationships in social networks, which poses challenges in trustworthy link discovery [7,8].

* Corresponding author.
*E-mail address:* limeizi@shnu.edu.cn (L. Meizi).

http://dx.doi.org/10.1016/j.comnet.2017.04.016
1389-1286/© 2017 Elsevier B.V. All rights reserved.

It is therefore impossible for each user to obtain global data for his/her trust link detection since there might be various accessible paths from one user to another, which indicates that the links between them are too complex to evaluate. Accordingly, global trust link mining has a high cost of computation, and in addition, not all of these links can be regarded as reliable links for users. Only those links that reflect high trust degrees, especially mutual trust degrees, among users are reliable for user communication and interaction. Generally, a mutual trust relationship implies a strong link, which enables free, reliable, and authentic interaction and inspirit support. Even a one-way trust relationship, called a weak link, facilitates users to send or accept items from their trusted sources with little concern. Therefore, the main motivation of our study is to find strong or weak trustworthy links that connect users to ensure the reliability of their interactions without global user relationship data preparation.

In this work, we propose Trust Traversal, a random traversal scheme, to detect trust links for users in social networks. Our objective is to provide a low-cost retrieval scheme that not only promotes efficient, accurate, and deliverable collective trust calculation among indirect users but also takes social network features into account and avoids global network calculation. We consider the functioning of our proposed Trust Traversal to depend on the following factors: trust and reputation for indirect trust calculation, trust delivery through users' relationships, and traversal with probability through high-trust relationships to detect trust links iteratively. Consequently, we propose our Trust Traversal Scheme as comprising the following aspects: 1) formal descriptions and related definitions for the proposed scheme; 2) objective and subjective trust (reputation and trust among users) and calculation rules for indirect collective trust calculation in user links; 3) a category classification method for trust link, i.e., strong trust link, weak trust link, and untrusted link; and 4) a traversal scheme for detecting the trust link with single-target users and multiple-target users.

Compared with other trust management methods, our Trust Traversal scheme offers the following. (1) Three types of trust measurement are proposed: reputation (objective trustworthiness), direct trust (subjective trustworthiness) and indirect link trust (subjective trustworthiness). Compared with conventional trust methods, which provide limited trust aspects, such as trust relationship, reputation, and recommendation (at most two), our trust calculation methods enable each user to measure most other users' trustworthiness (direct trust for direct-interaction users, reputation for overall trustworthiness, and indirect link trust for any strange users). Further, we propose additional factors (effective accumulation, time attenuation, and intensity factor for reputation; the trust transitivity rule, reputation weight rule, and level attenuation rule for link trust) in trust calculation to improve the calculation accuracies. (2) Category of trust link is proposed in this study for identifying the trustable paths among users. Traditional methods focus on finding a trust path that can connect indirect users based on complex path composition between users. However, these trust paths in the above methods just reflect collective reliabilities, which cannot reveal the differences between all accessible paths. In our study, we aim to detect the strong or weak links that reflect high reliability and closeness relationships. (3) In contrast to trust computation schemes, this work addresses a traversal strategy that only detects links with high possibilities of being strong or weak trust links, and it thus avoids untrusted or aloof relationships in detected links. In traditional methods, trust between indirect connected users is evaluated comprehensively based on all possible paths among them (or pre-selected paths higher than a trust threshold), and thus great overhead in both time complexity and space complexity are yielded because of the complexity of social networks. In our scenario, Trust Traversal evaluates users serially with trust-based probability, which avoids global path evaluation between users. (4) The Trust Traversal Scheme only relies on partial topological information of the user relationship network in each step, e.g., neighbor users, trust values, and distances from the origin user, rather than a global topology of the social network as in other related models.

The remainder of the study is organized as follows. Section 2 gives a brief description of the typical related studies concerning trust management with respect to social network. In Sections 3, we give an overview of trust traversal scheme and related formal definitions. Section 4 addresses the trust calculation methods in our study, including three aspects: reputation, trust of link, and category of trust link. In Section 5, we propose the trust traversal scheme, which comprise two aspects: single target user and multiple target users in traversal. Section 6 lists the results of a set of simulations and analysis. Section 7 is the study's conclusions.

## 2. Related work

### 2.1. Trust and reputation

Trust, as an inherent willingness of human beings, shows the emotional and rational reliability in between people. It is derived from judging trustworthiness by evaluating various facts which can lead to either the trust or distrust. Traditionally, trust is described from subjective and objective perspectives, which comprises two aspects: trust relationship calculation and reputation rating [4,6]. Over the past few years, many works focused on subjective trust computation, which comprises two aspects: direct trust and indirect trust [4]. Direct trust is used for reflecting the trustworthiness between direct connected users [5], while indirect trust is widely in long path connected users and online recommendation systems [9].

See-To and Ho [10] propose a method to evaluate the influence of trust on oral comment in social network. Wang and Gui [11] select transaction users in social network and computes trust between them. Wu et al. propose an interval-valued fuzzy methodology based on social network analysis to represent the model of trust relationship between experts and then compute the trust degree of each expert in group decision making [12]. Perez et al. introduce trust as an important indicator for evaluating the contacts of online social network user based on smartphone [13]. Ortega et al. [14] propose a method to compute a ranking of the users in social network and propagate both positive and negative opinions of the users. Then the opinions from each user about others can influence their global trust score. Qureshi et al. [15] propose a decentralized framework and the related algorithms for trusted information exchanging and social interaction among users based on the dynamicity aware graph relabeling system. The Bellman–Ford algorithm (noted as BF in latter sections) computes trust based on direct witness interaction trust judgments [16]. The algorithm admits the most trustable link for trust computation; it deems a long link to be untrustworthy. Golbeck proposed TidalTrust that gets trust in social networks by utilizing the shortest link based on the breadth-first search [17]. There are many widely used methods of trust calculation. The trust model, named EigenTrust (ET), in the P2P environment is widely used based on the transitivity of indirect trust [38]. The rationale of EigenTrust is: If a node trusts any target node, it would also trust the nodes trusted by the target node. The local trust among nodes is calculated based on their interaction experience. The weighted average trust ratings method (WA) is a typical trust computation and has been widely used in trust computation [40]. In this method, all trust ratings about the target object are aggregated and then a weighted average of the aggregation is calculated as the new trust value for the target object. Technically, the average method of trust is easy to realize while witness information and ratings are available. But in such method, trust ratings aggregation from a long linked node path is not considered to be weakened. Bahtiyar et al. [41] presented an ultimate trust rating model (UT), concerning that the trust would be kept updating dynamically. To do that, two factors, risk factor and confidence factor, were given to record the failed and successful services. Then, each node broadcasts their trust information, including two factors and related interaction data, to other nodes. Therefore, all receiving nodes can use the information to update their trust records. In general, the ultimate trust mode admits the highest value of ratings for trust relationships and can keep changing with the trust factors dynamically. However, such dynamically trust updating and local trust factor broadcasting bring a relative high complexity.

Additionally, many researches aim to address indirect trust or trust computation method, such as recommended trust, in SNS [18,19]. Some indirect trust mechanisms pay attention to integrate trust computation methods into comprehensive indirect trust or trust computation algorithms, such as average trust rating model (AT), which calculates the trust by rating the feedbacks/judgments/reviews about items [20,21]. These methods are simple and easy to realize and have been proved effective in some network environments. In addition, there are many traditional recommending systems, such as collaborative filtering system, which are widely used for indirect trust evaluation [22]. But researches have testified that trust based information recommending methods will enhance and improve the performance of raising recommending accuracy, because trust based methods enable the trust or trust transmitting or propagating through users' relationships [23].

Reputation can be considered a collective measure of trustworthiness (in the sense of reliability) based on the referrals or ratings from other users [4]. In term of reputation measurement, there are two main types of reputation system: centralized [24] and distributed [4]. The former has a central authority to collect all the rating, and publish reputation score for every participant. Whereas in distributed reputation system, each member gets the belief about each experience with others, and submits the reputation on request from relying members. Different from subjective trust, reputation can be considered as a collective measure of trustworthiness (in the sense of reliability) based on the referrals or ratings from members in a community [4,25]. Therefore, reputation is aggregated from various members' joint decision. Many methods of reputation computation have been widely used, such as average judgment method of reputation (AJ) [26], Bayesian method based on previous reputation knowledge [27], fuzzy logic based reputation system [28], and a global reputation model of EigenRep (ER) [38], etc. However, many reputation calculation methods, such as EigenRep model, have a high time complexity and lower ability to resist risks.

However, in our consideration, the social nature of SNS, as is reflected by the formation of user relationships within, traceable linked routes and its topology, deserves more emphasis in trust computation based on links among users as it reinforces. In this study, we consider that there are following aspects for calculating trust in social network: 1) rating weight of voters, temporal dynamic property, and reliable confidence evaluation must be taken into account in reputation aggregation; 2) direct trust must reflect the strength of trust relationship between users more than past interaction judgment accumulation; 3) more importantly, links among users must be measured and classified into different categories according to the strength of their correlations. Overall, our proposed trust evaluation method can be seen as a hybrid trust model which uses both social network graph topological information (e.g. the direct path and indirect path) and interaction information (e.g. interaction factor and judgment scores) to compute trust and reputation according to the trust model classification in [6]. Meanwhile, our proposed method takes trust information from three main sources [6], i.e., the user attributes (e.g. community attribute for reputation aggregation), behaviors (e.g. interaction factor for trust calculation) and experiences (e.g. judgment score for reputation aggregation), into account for trust evaluation.

### 2.2. Link detection in social network

Relationship is a significant entity in online social network for connecting users [6,7,8,19,29]. Generally, a composite path which connects multiple users through their relationships is called as *link*. Link detection has its wide application prospects in online social networks, e.g. user link prediction in recommendation system, service selection with trust evaluation, and link detection for finding potential friends in social network [19,29–31]. However, due to the complexity of large-scale network nature, not all links are valuable for later mining. In our work, our motivation is to give scheme for detecting links that only reflect high trustworthiness, and mutual or one-way closeness relationships among users.

According to the view of Wang et al [44], the techniques of link prediction in social network can be classified as follows: node-based metrics evaluation, topology-based metrics, social theory based metrics, and learning-based methods. For node-based metrics methods, the similarity degrees between node pairs are evaluated based on node attributes to generate the similarity metrics [45,46], and then, the more similar the pair is, the more likelihood a link between them, and vice versa [44]; for topology-based metrics, topological information, e.g., neighbor information [47,48] and path information [49,50], can be used to establish metrics and then detect the possible links based on the topology characteristics; for social theory based metrics methods, many additional social information and social related characteristics, such as community, triadic closure, strong and weak ties, homophily, and structural balance [44], are taken into account to improve the performance of link prediction; the learning based methods use the classical or improved machine learning methods to predict links in social network [44].

There have been many other efforts for link detection in social network in past decades, such as link prediction by random walk (RAN), similarity calculation, probabilistic model and maximum likelihood evaluation [30,31,36]. Deng et al. [19] address a service recommendation method with trust enhancement, Relevant Trust Walker, under an extended random walk algorithm. Jamali and Martin [30] propose a random walk model for mining valuable item judgment by evaluating the trust relationships among users. However, the strength of user links is not taken into consideration. Liu et al. [31] propose a model of optimal trust path selection with multiple constraints by incorporating trust, social relationships and recommendation roles in social network. Fire et al. [33] give a set of structural features for identifying missing links in social network. Gupta et al. [34] propose a strategy for predicting missing links among users, which takes into account both the uncommon neighbors and common neighbors of given users. Valverde-Rebaza and Alneu de [35] address a simple and non-expensive method by combining structural with community information for predicting links in Twitter. And also, De Meo et al. [42] consider that trustworthiness can be seen as a quantitative factor for measuring the group compactness among users. Such work means that trust can be seen as a critical entity to reveal the relationships among users.

In our previous work, we present a trustworthy degree evaluation method for the complex links among users, which comprises two perspectives: reliability and strength evaluation [43]. The main goal of our previous work is to measure that whether a given set of existing complex links among users is trustworthy or not by taking the facts of link topological information (direct and indirect links) and social interactions among users into account. However, we in this work aim to propose a method to discover a set of unknown trustworthy links between two users by using the trust link detection method, which is not included in our previous work.

According to the number of tasks in link detection, link detection includes two types: single task in link detection and multiple tasks in link detection [31,32]. These two types can be cast as link detection with single target user and multiple target users in this study. Compared with existing works, in our scenario of trust link detection (named Trust Traversal Scheme), we have following differences: (1) a set of trust calculation methods, i.e., reputation, direct trust, and indirect link trust, is introduced previously for not only promoting the performance of trust evaluation but also taking into account features of user interactions and correlations in social network; (2) links among users are detected according to both the trust degree and the strength of the links; (3) trust link is found through a probabilistic traversal method that only relies on partial topological and trust information of user social graph (e.g. shortest distance from origin user to given target user, or trust information about the current user or the next user in traversal process) rather than the global topological trust information, which reduces the complexity of detection task.

## 3. Overview of the trust traversal scheme

### 3.1. Related definitions of trust links in SNS

If the users in social networks are deemed to be vertices, then we can associate two vertices with an edge if they have a direct association. Thus, individuals and their relationships in social networks form a relatively stable graph topology. To better explain the graph topology, we first introduce the following definitions, which are similar to definitions given in [43].

**Def. 1 Social network graph model**. A social network graph model is a graph model that describes the topology of individuals and their relationships, denoted as $UG =< U, E >$. The nonempty set $U = \{u_1, u_2, \ldots\}$ shows the users in the network, and $E = \{e_1, e_2, \ldots\}$ indicates the binary relations, which are described as links in this study, among users.

According to definition 1, the users' relationships are directional edges and have the following properties:

(1) Non-reflexivity. There is no relationship pointing to users themselves as $(u_i,\ u_i)$.
(2) Asymmetry. The binary relationship among users is directional, meaning $e(u_i,\ u_j) \neq e(u_j,\ u_i)$.
(3) Non-transitivity. The relationship denoted as $(\exists(u_i,\ u_j) \wedge \exists(u_j,\ u_k)) \rightarrow \exists(u_i,\ u_k)$ is not always true.

**Def. 2 Path.** A path is a direct or indirect route from an origin user to a target user composed of users' relationships (edges) in a social network graph model, shown as $path(u, v) = \{u, e(u, w_1), w_1, e(w_1, w_2), w_2, \ldots, e(w_n, v), v\}$, where $u$ and $v$ are the origin user and target user, $w_i$ is an intermediate user along the path, and $e(w_i, w_{i+1})$ is the direct path from $w_i$ to $w_{i+1}$.

In Def. 2, if there are no intermediate users in a path, the path is a direct one; otherwise, it is an indirect one. According to Def. 2, a path is a serial route from the origin user to the target user. If there is a path from user $u$ to user $v$, we denote that user $v$ is a reachable user of user $u$. Because a path can be seen as a directional route from the origin user to the target, we denote the directional path from the origin user to the target user as a "*positive path*" and the opposite directional path from the target to the origin user (if it exists) as an "*inverse path.*". Further, we give the following notion of the *Level* of a path as,

**Def. 3 Level**. The definition of the level of a path is as follows,

(1) For each user, his/her distance to himself/herself is zero.
(2) If there exists a direct path between two users, the level from the origin user to the target user is one.
(3) If there exists an indirect path between two users, the shortest distance from the origin user to the target user is their level in the indirect path.

Consequently, given an origin user (denoted as $rs$) and a target user (denoted as $rt$) in the social network, our work aims to detect multiple trustworthy paths for them. Therefore, we have the following definition of a *Link*,

**Def. 4 Link**. A link is a connected sub-graph consisting of paths from the origin user to the target user during information spread, shown as $L =< LU, LE >$. The nonempty set $LU = \{rs, RU, rt\}$ consists of the original user, a set of intermediate users and the target user in the link, while $LE = \{path(rs, rt)_1, path(rs, rt)_2, \ldots\}$ consists of one or more paths from the origin user to the target user.

It is worth noting that our definition of *Link* is the initial subgraph model that includes all existing paths from $rs$ to $rt$. That is, the set of paths $LE = \{path(rs, rt)_1, path(rs, rt)_2, \ldots\}$ includes direct paths (without intermediate users) and indirect paths (with intermediate users) without any filtering mechanism, such as trust evaluation. Such formalism is different from the definition of a link in our previous work [43]. Meanwhile, we have the following further justifications for definition 4,

(1) For the level of a link, we have the following justifications: the level of $rs$is 0; if there exists a directly accessible path from $rs$ to $ru_i$, the level of $ru_i$ is 1; if and only if there exists a directly accessible path from users in the $n$th level to $ru_i$ but there is no path from users in a level lower than the $n$th to $ru_i$, then the level of user $ru_i$ is $(n + 1)$. The level of user $ru_i$ is denoted as $lev(ru_i)$ with respect to the origin user in the link.
(2) $RU = \{ru_i^j | i = 1, 2 \ldots \wedge j = 1, 2, \ldots\}$ means that the intermediate user $ru_i^j$ is a reachable one (there is at least one path from $rs$ to $rt$ that includes $ru_i^j$), where the superscript $j (j = 1, 2, 3 \ldots)$ means the number of levels in the link and the subscript $i$ means the ordinal of the user is $i$ in this level in the link.
(3) For the edges in the link, there are three types of expressions: 1) $re_k^j = (ru_i^j, ru_l^{j+1})$ is the edge that links users in the $j$th level with ones in the $(j + 1)$th level, called a positive edge; 2) $re_k^j = (ru_i^j, ru_l^j)$ is the edge links user in the $j$th level with ones also in the$j$th level in the link, which is called an edge in the same level; and 3) $re_k^j = (ru_i^j, ru_l^{j-1})$ is the edge that links users in the $j$th level with ones in the $(j - 1)$th level, called an inverse level edge;
(4) For brevity, we signify the direct path $re_k^j = (ru_i^j, ru_l^{j+1})$ as $ru_i^j \rightarrow ru_l^{j+1}$.
(5) A link also retains the characteristics of the path in the graph.

To justify the definitions of path, level and link clearly, we give an example as shown in Fig. 1(a). We can see that there are four paths from origin user $A$ to target user $B$ ($A \rightarrow C \rightarrow B$, $A \rightarrow D \rightarrow B$, $A \rightarrow E \rightarrow B$, and $A \rightarrow F \rightarrow B$). And the link from $A$ to $B$ and the level values of users can be signified as,

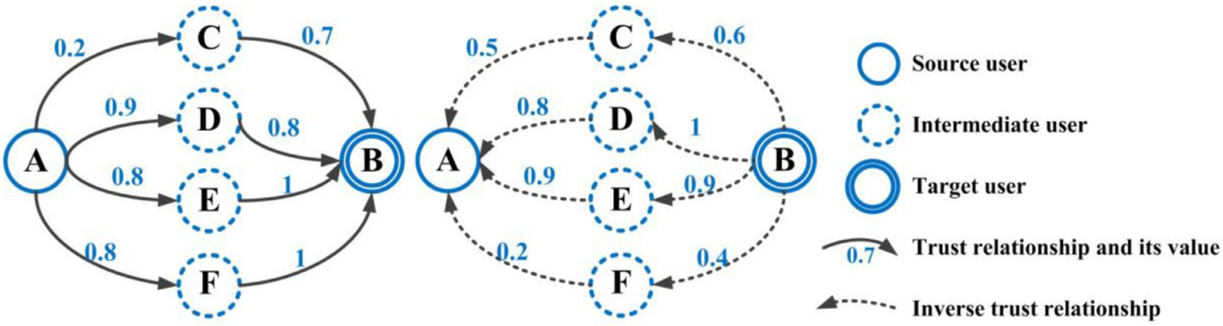


Fig. 1(a)

Fig. 1(b)

**Fig. 1.** Example of detecting a trust link.

$$L(A,B) = < LU, LE >$$

$$LU = \{rs = A, RU = \{C, D, E, F\}, rt = B\}$$

$$LE = \left\{ \begin{array}{l} path(A,B)_1 = \{A, e(A,C), C, e(C,B), B\}, \\ path(A,B)_2 = \{A, e(A,D), D, e(D,B), B\}, \\ path(A,B)_3 = \{A, e(A,E), E, e(E,B), B\}, \\ path(A,B)_4 = \{A, e(A,F), F, e(F,B), B\} \end{array} \right\}$$

$$lev(A) = 0; lev(C) = lev(D) = lev(E) = lev(F) = 1; lev(B) = 2$$

**Def. 5 Trust Schema.** A trust schema records the reputation and trust information locally for each user in a social network. The trust schema includes two types of information,

(1) Reputation. Reputation reveals the objective trustworthiness of a user, which results from social evaluation. In this work, if a user $u$ gains his reputation in his past experiences, then $reputation(u)$ denotes the value of this reputation as a real number in [0, 1]. Zero means no reputation, and one means full reputation.

(2) Trust. Trust describes the subjective trustworthy opinion of one user regarding another. In this work, if user $u$ trusts another user $v$, then $trust(u, v)$ denotes the value of this trust as a real number in [0, 1]. Likewise, zero means no trust and one means full trust.

Trust is a directional and binary value between two users due to different interaction experiences of users. Therefore, trust degrees of the positive and inverse links between two same users are asymmetrical. For example, user $A$ has high trust to user $B$, while $B$ has a very low trust to $A$. In such case, user $B$ would drift apart with user $A$, which means a weak link from $B$ to $A$. Therefore, links can be divided according to the closeness and mutual trustworthy degree of users. In this study, we define three types of trust links among users, strong trust links, weak trust links, and untrusted links:

**Def. 6 Trust link.** A trust link is a link among users that reveals their trustworthy and closeness degrees. A trust link can be described as follows,

(1) Strong trust link. A strong trust link reflects a highly mutual close relationship between two users, which includes high mutual trustworthiness and high frequency of mutual interactions. A strong trust link between two users, $d_i$ and $d_j$, is denoted as $d_i \Leftrightarrow d_j$.

(2) Weak trust link. A weak trust link reflects a one-way or mutual estranged relationship between two users, which includes one-way high trust and one-way low trust or infrequent mutual interactions. A weak trust link from user $d_i$ to $d_j$ is denoted as $d_i \Rightarrow d_j$.

(3) Untrusted link. An untrusted link denotes that both users in the link mistrust each other. An untrusted link from user $d_i$ to $d_j$ is denoted as $d_i \propto d_j$.

Compared with the definition of strong and weak links in our previous work [43], our proposed definition of trust links uses two aspects for defining a trust link: trust degree and interaction frequency. The former is taken into account in both definitions, while the latter was not included in our previous work. In detail, we propose a set of rules for classifying the above three different types of trust links. For users $d_i$ and $d_j$, they have trust degrees $trust(d_i, d_j)$ and $trust(d_j, d_i)$. Meanwhile, we use two factors for measuring interaction frequency: the mutual trust factor (denoted as $mtf(d_i, d_j)$) and the interaction active factor (denoted as $iaf(d_i, d_j)$). We have the following justifications for the two factors: (1) Mutual trust factor ($mtf(d_i, d_j)$): this factor reflects the two directional trust degrees between two users and the similarity degree of the two trust degrees; (2) Interaction active factor ($iaf(d_i, d_j)$): this factor aims to evaluate whether the two users have active interactions that reflect the closeness of relationships among users. These factors are given for evaluating the types of trust links among users. In our consideration, only those who have high mutual trust and interaction frequency have strong links; otherwise, the links are weak ones. We will address the detailed calculation methods for these two factors in a latter section.

Therefore, we give the following objective rules for trust link classification,

$$\begin{cases} d_i \Leftrightarrow d_j : trust(d_i, d_j) \geq \varpi_1 \wedge trust(d_j, d_i) \geq \varpi_1 \wedge mtf(d_i, d_j) \\ \quad \geq \varpi_2 \wedge iaf(d_i, d_j) \geq \varpi_3 \\ d_i \Rightarrow d_j : \big(trust(d_i, d_j) \geq \varpi_1 \wedge trust(d_j, d_i) \leq \varpi_1\big) \wedge (mtf(d_i, d_j) \\ \quad \geq \varpi_2 \vee iaf(d_i, d_j) \geq \varpi_3) \\ d_i \Leftarrow d_j : \big(trust(d_i, d_j) \leq \varpi_1 \wedge trust(d_j, d_i) \geq \varpi_1\big) \wedge (mtf(d_i, d_j) \\ \quad \geq \varpi_2 \vee iaf(d_i, d_j) \geq \varpi_3) \\ d_i \propto d_j : else \end{cases}$$

Here, we use three thresholds for classifying links as $\varpi_1, \varpi_2, \varpi_3 \in [0, 1]$.

For each user, it is essential to find a "*trust link*" which can connect to a target user for ensuring the reliability of the target one. Then, trust link denotes a link between users with the relative large value of trust. In addition, since mutual interaction is an important fact in daily communication for users in social network, trust link must mark a link which has the maximum opportunity to facilitate users' mutual interactions. That is the reason for trust link category in our work.

### 3.2. Problem of trust traversal for detecting trust links

In this work, trust traversal is proposed for finding a reachable path from an origin user to a target user with the highest trust, which is called a trust link as abovementioned. As defined in Def. 1, we have a set of users (vertices) $UG.U = \{u_1, u_2, ...\}$ and their relationships (edges)$UG.E = \{e_1, e_2, ...\}$. Each user has its trust schema, which records the reputation and trust information about other users. Then, we can construct a directional graph based on a social network graph model with trust as the weights of edges by using a trust evaluation method for weight measurement. Consequently, each user may obtain his trust links to other accessible users through our proposed scheme. To do that, we here denote the trust links in such a directional graph with trust weights as $TLink(u, v) =< L, trust >$, where *L* is the link from the origin user to the target user and *trust* is the trust value.

Then, the task of this work is as follows: Given a user in a social network graph model, shown as $u \in UG.U$ in Def. 1, and his trust schema, detect a trust link from user *u* to user *v*. Normally, the origin user only records trust information about those who have a direct relationship with him and has few recognitions regarding strange target users.

Traditional trust link evaluation methods estimate strange users' reliabilities based on indirect trust integration. Basically, they find all neighborhoods of the origin user and require the trust values regarding the target user. If the neighborhoods send back their trusts regarding the target user, the origin user aggregates this feedback comprehensively and then obtains the indirect trust regarding the target user. The neighborhoods are asked directly whether they know the trustworthiness of the target user if they are trusted by the origin user in the above method. If so, they return it; otherwise, they recursively ask their directly trusted neighbors.

Indirect trust aggregation works because of the effects of selection and social influence that have been postulated by sociologists for a long time. That means that people tend to trust a strange person through a trustable person, and due to social influence, related people in a social network influence each other to become more similar. Through indirect trust, most people would adopt transitive trust (including information propagation, recommendation, etc.), which confirms that trust can be delivered through a link from one user to another to improve the reliability estimation among users.

Exploiting the trust link among users relies on a better traversal strategy that allows the detection of a more reliable link for two users. Due to the complexity of the social network graph, there might be a complex reachable path (or a network) between users, which poses the problem of evaluating the most trustworthy link among all possible links. As shown in this Fig. 1(a), we have a social network graph from user *A* to user *B*. There are 4 possible reachable paths between them, while there is a path from *A* to *B* composed by the four paths. The origin user *A* wants a prediction trust about user *B*. Normally all paths can be taken into consideration in indirect trust aggregation. However, the path ($A \to C \to B$) might be meaningless since neighborhood *C* is not trusted by user *A*, and also the path get a very low trust value. Meanwhile, the paths ($A \to E \to B$ and $A \to F \to B$) bring a little larger trust value than other path ($A \to D \to B$) even though the neighbor user *D* gets higher trust from *A* than neighbor user *E* and *F*. Additionally, Fig. 1(b) shows the inverse trust relationships, and we can see that the inverse path ($B \to E \to A$) has higher trust degree than inverse path ($B \to F \to A$ and $A \to F \to B$) because user *B* has a low trust of *F* and *F* has also a low trust of *A*. That means the two paths shows different trust degrees from *A* to *B* because the path which passes user *E* implies a mutual trust relationship (also called a strong relationship) while the other one does not (also called a weak relationship). So we can see there are two strong trust links ($A \to D \to B$ and $A \to E \to B$), a weak trust link ($A \to F \to B$), and an untrusted link ($A \to C \to B$) between *A* and *B*. From Fig. 1, the goal of our work is to find a reliable trust link that is as strong as possible through a better traversal scheme for users to predict the more accurate indirect trust among them. Obviously, a strong trust link is the optimal solution, and a weak trust link is an alternative if a strong trust link does not exist.

## 4. Trust schema and its calculation method in trust traversal

In this section, we give the trust schema and its calculation methods. To distinguish the terms used in our following trust schema calculation, we first give a table justifying the descriptions of them, e.g., behaviors, interactions, or time slice, as listed in Table 1.

### 4.1. Reputation calculation method in SNS

Similar to the manner in which reputation is formed in natural society, reputation in a social network can be reliably computed through accumulating and synthesizing interactions, behaviors and experience sharing. In a social network, reputation is a key factor for evaluating individual reliability and is of particular importance for such applications as e-commerce and social services. Social networks, such as Taobao and eBay [37], have developed their own systems to compute reputation using strategies of, for example, summation, average, or weighted mean [38].

We propose a method for computing users' reputation based on their social information, taking into account the various factors in different types of social networks. This method mainly considers the following two aspects.

**Table 1**
Related terms justification in examination prototype.

| Term | Description | Value | Example (constructed based on the social network used in the experiment) |
|---|---|---|---|
| *judgment*($d_j$, $d_i$) | The score a user sending his feeling to another one for reputation aggregation or trust calculation | [0,1] | User A sends his feeling to user B's post with a positive comment. Then, the score of this judgment can be denoted as 0.9. |
| *Vote*($d_i$) | The set of users who send their judgment to a target user for reputation aggregation | 0, 1, 2, ... | If there are 12 users who have send their judgment to user A for reputation aggregation, the number of the set *Vote*(*A*) (denoted as \|*Vote*(*A*)\|) is 12. |
| *act*($d_i$, $d_j$) | Subjective trust opinion value of an interaction from rom user $d_i$ to user $d_j$ | [0,1] | If user *A* approves user B in an interaction, the value *act*(*A*, *B*) is 1; if user *A* forwards user B' post in an interaction, the value *act*(*A*, *B*) is 0.8; If user *A* sends his comment to user B's post in an interaction, the value *act*(*A*, *B*) is 0.4. |
| *Interact*($d_i$, $d_j$) | The number of interactions from user $d_i$ to user $d_j$ | 0, 1, 2, ... | If there are 50 responses from user A to user B, the value of *Feed*(*A*, *B*) is 50. |
| *Feed*($d_i$, $d_j$) | The number of responses from user $d_i$ to user $d_j$ | 0, 1, 2, ... | If there are 30 responses from user B to user A for user A's interactions, the value of *Feed*(*B*, *A*) is 30. |
| Time slice | The unit time for time attenuation in reputation aggregation | 1, 2, 3, ... days | If the reputation or trust is re-aggregated every week, the length of time slice is 7. |

(1) Effective accumulation. Individual reputation is built upon others' valid judgments and is finally formed through accumulation of all the valid comments.

We determine if a comment is effective based on the commenter's qualification for judgment by computing his/her reputation. We here use the term "*qualification*" to measure the weight of a commenter's judgments in reputation calculation. That is, if a commenter's reputation is high, his/her judgments would contribute more in reputation aggregation. Assuming that an individual $d_i$ has reputation ($reputation(d_i) \in [0,\ 1]$) and that the number of communities in the social network he/she belongs to is $|C(d_i)|$, then the qualification degree for effective judgment of this user can be computed as follows.

$$qu(d_i) = reputation(d_i) \times \beta^{\frac{1}{|C(d_i)|}} \tag{1}$$

where$\beta \in [0,\ 1]$ is a user-defined parameter. Additionally, we here use a notion, community, for our calculation in Eq. (1). Community can be seen as a group that is clustered by topic, interest, or other entity, in which users can communicate and share their information conveniently. In most traditional trustworthiness calculation methods, community is not included since it is not a common element in their computation environment. However, most users take part in multiple communities according to their different interests in a social network. Hence, community is here seen as a crucial factor for evaluating users' trust schema. In our consideration, recognizing or detecting community for users is a pre-processing task for reputation calculation. That is, the reputation calculation method in this work relies on the community information, which must be given in advance.

During reputation accumulation, we only consider effective comments and filter out ineffective ones. Meanwhile, the qualification for judgment should be considered as well. Based on these considerations, we calculate reputation as follows.

Assume that there is an individual $d_i$ and that the judgments of other individuals ($d_j$) in the social network regarding $d_i$ are expressed as a set of discrete variables $J(d_j, d_i) = \{judgment(d_j, d_i)_1, judgment(d_j, d_i)_2, \ldots\}(judgment(d_i, d_j) \in [0, 1])$. The qualification for judgment is denoted as $qu(d_j)$. Then, the reputation of $d_i$ is given by

$$\begin{aligned} &reputation(d_i) \\ &= \frac{\sum_{j=1}^{m} \sum_{k=1}^{|J(d_j,d_i)|} \big(judgment(d_j, d_i)_k \times qu(d_j)\big)/|J(d_j, d_i)|}{\sum_{j=1}^{m} qu(d_j)} \end{aligned} \tag{2}$$

where $m$ and $|J(d_j,\ d_i)|$ are the numbers of individuals ($d_j$) who provided judgment sets $J(d_j,\ d_i)$ to $d_i$ and the judgment set from individual $d_j$, respectively. It is worth noting that Eq. (2) is based on the values of $qu(d_j)$ computed by using Eq. (1), which, in turn, requires the value of $reputation(d_j)$ computed using Eq. (2). Thus, a working computation method for reputation requires an iterative approach. For the cold-start problem, we notice that most new users in social networks are honest user. Hence, we consider that most new users would obtain honest reputation values based on their later interactions in the social network by default. Therefore, we assign a random reputation value around the average value of an honest user in the social network for a new-arrival user.

For example, there is a user A, who obtains judgments from users, B and C, and the sets are $J(B, A) = \{0.85, 0.8, 0.9\} J(C, A) = \{0.9, 0.95, 0.8, 0.85\}$. Assume that user B and C have their reputation 0.9 and 0.85, and the numbers of communities in social network they belong to are 7 and 5, respectively. Then, we can get the effective accumulation of reputation as follows ($\beta = 0.8$ in this example),

$$\begin{aligned} &qu(B) = reputation(B) \times \beta^{\frac{1}{|C(B)|}} = 0.9 \times 0.8^{\frac{1}{7}} \approx 0.87 \\ &qu(C) = reputation(C) \times \beta^{\frac{1}{|C(C)|}} = 0.85 \times 0.8^{\frac{1}{5}} \approx 0.81 \\ &reputation(A) \\ &\quad = \frac{[(0.85+0.8+0.9) \times 0.87]/3+[(0.9+0.95+0.8+0.85) \times 0.81]/4}{0.81+0.87} \\ &\quad \approx 0.86 \end{aligned}$$

(2) Time attenuation. Each judgment has a validity duration. The longer the time interval is, the less effective the influence of judgments on reputation accumulation is. Here, we use a notion named *time slice* to signify a unit time for time attenuation. For example, the length of a time slice can be set as 7 days, and then the reputation of a user would be updated every other time slice (7 days) based on time attenuation.

Then, we introduce the factor of time attenuation to account for reputation time validity. Let $T(n)$, $n = 0, 1, 2\ldots.$, denote the *n-th* time slice. The reputation degrees that a user gains in a time slice are $reputation(d_i)^{T(0)}$, $reputation(d_i)^{T(1)}$.... After $k$ time slices, the reputation in the *(k + 1)-th* time slice is reduced to

$$reputation(d_i)^{T(k+1)} = reputation(d_i)^{T(0)} \times \sigma^k \tag{3}$$

where the time attenuation factor is $\sigma \in [0,\ 1]$.

According to Eq. (3), during *k-1* time slices, the user's reputation has respectively accumulated as $reputation(d_i)^{T(0)} \times \sigma^k, reputation(d_i)^{T(1)} \times \sigma^{k-1}\ldots.$ Then, in the *k-th* time slice, the user's real-time reputation can be computed by

$$reputation(d_i)^k = \frac{\sum_{j=0}^{k-1} reputation(d_i)^{T(j)}}{\sum_{j=0}^{k-1} \sigma^j} \tag{4}$$

For example, let there are 5 time slices in reputation aggregation of user $A$, and he gets his reputation values as 0.9, 0.85, 0.9, 0.9, and 0.8 respectively in every time slice. Then, user $A$ gets real time reputation as follows ($\sigma = 0.9$ in this example),

$$\begin{aligned} &reputation(A) \\ &\quad = \frac{[0.9 \times 0.9^4]+[0.85 \times 0.9^3]+[0.9 \times 0.9^2]+[0.9 \times 0.9^1]+[0.8 \times 0.9^0]}{0.9^4+0.9^3+0.9^2+0.9^1+0.9^0} \\ &\quad = 0.866 \end{aligned}$$

In addition to computing reputation, we introduce a reputation confidence factor, which represents the reliability of an individual reputation. We take into consideration the following aspects when computing this factor.

(1) Judgment consistency. Reputation is the aggregation of judgments that come from others in SNS. The consistency of judgments reflects whether users maintain similar reliable opinions towards the reputation keeper or not. Obviously, a consistent judgment manifests an authentic reputation. For example, suppose that user $A$ gets 0.8 reputation value from two judgments 0.8 and 0.8, while user $B$ gets 0.8 reputation value from two judgments 1.0 and 0.6 (effective accumulation and time attenuation are not considered here). However, reputation of user $A$ is more reliable than user $B$ because the judgments of $A$ are consistent while the judgments of $B$ are volatile near his/her reputation. That is, the more accordant the judgment divergence is, the greater the reputation reliability would be. Here, we use the standard deviation and convert it for judgment consistency calculation. Suppose that the set of users who vote regarding their judgments of $d_i$ for reputation aggregation is $Vote(d_i)$. Let the reputation of individual $d_i$ be $reputation(d_i)$, the *k-th* judgment for reputation aggregation from $d_j \in Vote(d_i)$ to $d_i$ be $judgment(d_j,\ d_i)_k$, the total number of judgments from $d_j$ to $d_i$ be $num_j$, and the total number of users voting on $d_i$ be $|Vote(d_i)|$. Then, the judgment consistency of the reputation, denoted as $jc(d_i)$, is

$$jc(d_i) = 1 - \sqrt{\frac{\sum_{j=1}^{|Vote(d_i)|}\sum_{k=1}^{num_j}\left(judgment(d_j, d_i)_k - reputation(d_i)\right)^2}{\sum_{j=1}^{|Vote(d_i)|} num_j}} \tag{5}$$

(2) Hit rate. Reputation can be used to forecast the future reliability of a user. Then, an authentic reputation would retain a high hit rate in near judgments. We set up an effect range for discerning hit rate. Let the reputation of individual $d_i$ be $reputation(d_i)_n$ after the $n$-th time slice. In the next time slice, the hit rate of the reputation is calculated as,

$$\begin{aligned} &hr(reputation(d_i)_{n+1}) = \frac{\sum_{T(n)}\sum_k l_k}{|T(n)| \times m} \\ &l_k = \begin{cases} p & (judgment(d_j, d_i)_k \in [reputation(d_i)_n \\ & \quad -\rho, reputation(d_i)_n + \rho]) \wedge (d_j \in Vote(d_i)) \\ 1-p & else \end{cases} \\ &p \in [0, 1] \end{aligned} \tag{6}$$

where $m$ is the total number of judgments for reputation in the $n+1$-th time slice, $|T(n)|$ is the total number of time slice, and $\rho \in [0, 1]$ is the error for discerning hit rate.

Thus, we introduce a confidence factor $\vartheta$ to express the reliability of reputation, and it is computed as follows.

$$\vartheta(d_i) = \frac{1}{2} \times [jc(d_i) + hr(reputation(d_i))] \tag{7}$$

### 4.2. Trust calculation for directly linked users

Direct paths reflect the familiar relationships among users in SNS. Through direct paths, users have interactions freely with others. Direct trust is the one of the basic opinions which includes in the paths of interpersonal interaction and thus, is of particular importance in computing reliable relationship. In contrast to traditional past judgment aggregation-based trust calculation, we here propose three factors for trust calculation: an interaction factor, common trust factor, and stability factor. In our work, we denote *trust*($d_i$, $d_j$) as the direct trust degree between two directly linked users, from $d_i$ to $d_j$.

- Interaction factor

The interaction factor reflects the subjective aspect to establish trustworthy feelings or experiences based on users' historical interactions. For each interaction, we use a numeric degree to signify the user's trustworthy opinion. Assume that the individual $d_i$has already interacted $n$ times with another individual $d_j$, that the $k$-th interaction has the trust opinion value of *act*($d_i$, $d_j$)$_k$ ∈ [0, 1] on $d_i$ and that $m$ out of those $n$ interactions are negative. Then, the interaction factor of trust from $d_i$ to $d_j$ is defined as follows:

$$if(d_i, d_j) = \frac{\sum_{k=1}^{n} act\left(d_i, d_j\right)_k}{n} \times \left(1 - \frac{m}{n}\right)^{\frac{1}{n-m}} \tag{8}$$

It can be concluded from the above equation that the trust among users has the following characteristics,

(1) The interaction factor among users is built upon past judgments using a cumulative average computation method.
(2) The interaction factor is one-way, indicating that *if*($d_i$, $d_j$) ≠ *if*($d_j$, $d_i$);
(3) Trust among users may be influenced by the number of judgments and the number of negative judgments. For example, if there are 4 negative judgments among 10, the influence factor is $(1-4/10)^{\frac{1}{10-4}} \approx 0.92$. While if there are 2 negative ones among 5 then the influence factor is $(1-2/5)^{\frac{1}{5-2}} \approx 0.84$. Again, it shows that the numbers of judgments may lead to different influence factors.

- Common trust factor

In our consideration, if two users have a set of common trust users and maintain similar trust degrees, they might be likely to have high trust for each other. Suppose that there are two individual $d_i$ and $d_j$ and that they have a common trust user set $Com = \{cu_1, cu_2, ...\}$. For $cu_k \in Com$, $d_i$ and $d_j$ have trust degrees *trust*($d_i$, $cu_k$) and *trust*($d_j$, $cu_k$), respectively. Then, the common trust factor of $d_i$ and $d_j$ is calculated as,

$$\begin{aligned} &ctf(d_i, d_j) = \frac{\sum_{cu_k \in Com}\left[trust(d_i, cu_k) \times trust(d_j, cu_k) \times w_{cu_k}\right]}{\sqrt{\sum_{cu_k \in Com} trust(d_i, cu_k)^2} \times \sqrt{\sum_{cu_k \in Com} trust(d_j, cu_k)^2}} \\ &w_{cu_k} = \begin{cases} \phi_1 & \left(d_i \Leftrightarrow cu_k \wedge d_j \Leftrightarrow cu_k\right) \vee \left(d_i : cu_k \wedge d_j : cu_k\right) \\ & \quad \vee\left(d_i \Rightarrow cu_k \wedge d_j \Rightarrow cu_k\right) \vee \left(d_i \Leftarrow cu_k \wedge d_j \Leftarrow cu_k\right) \\ \phi_2 & \left(d_i \Leftrightarrow cu_k \wedge \left(d_j \Rightarrow cu_k \vee d_j \Leftarrow cu_k\right)\right) \\ & \quad \vee\left(d_j \Leftrightarrow cu_k \wedge (d_i \Rightarrow cu_k \vee d_i \Leftarrow cu_k)\right) \\ \phi_3 & \text{else} \end{cases} \\ &\phi_1, \phi_2, \phi_3 \in [0, 1] \end{aligned} \tag{9}$$

In Eq. (9), we introduce the category of the trust relationship (strong trust link, weak trust link, or untrusted link) to describe the weight of the trust degree. The value of $w_{cu_k}$ is set according to experimental analysis, and details of the trust link category are discussed later in this study.

For instance, assume that two users, A and B, have their common trust user set as $Com = \{cu_1, cu_2, cu_3, cu_4\}$. And they have their trust degrees as 0.8 ($A \Leftrightarrow cu_1$), 0.6 ($A \Rightarrow cu_2$), 0.6 ($A \Leftarrow cu_3$), 0.9 ($A \Leftrightarrow cu_4$) and 0.8 ($B \Leftrightarrow cu_1$), 0.7 ($B \Leftarrow cu_2$), 0.9 ($B \Leftrightarrow cu_3$), 1.0 ($B \Leftrightarrow cu_4$), respectively. Then, the common trust factor of them is calculated as ($\phi_1 = 1, \phi_2 = 0.8, \phi_3 = 0.5$ in this example, respectively),

$$\begin{aligned} &ctf(A, B) \\ &= \frac{\sum_{cu_k \in Com}[trust(A, cu_k) \times trust(B, cu_k) \times w_{cu_k}]}{\sqrt{\sum_{cu_k \in Com} trust(A, cu_k)^2} \times \sqrt{\sum_{cu_k \in Com} trust(B, cu_k)^2}} \\ &= \frac{[0.8 \times 0.8 \times 1]+[0.6 \times 0.7 \times 0.5]+[0.6 \times 0.9 \times 0.8]+[0.9 \times 1 \times 1]}{\sqrt{0.8^2+0.6^2+0.6^2+0.9^2} \times \sqrt{0.8^2+0.7^2+0.9^2+1^2}} \\ &\approx 0.825 \end{aligned}$$

- Stability factor

Trust is a dynamic value since there might be various reasons behind changes of trust among users, and thus the stability factor of trust manifests the numeric fluctuation of trust between users. Here, we calculate the stability factor based on time slice aggregation. Let $T(k)$, $k = 0, 1, 2....$, denote the k-*th* time slice. Assume that *trust*($d_i$, $d_j$)$^{T(k)}$ denotes the trust value from $d_i$ to $d_j$ in the k-*th* time slice and that $\overline{trust}(d_i, d_j)$ is the average value of trust in all time slices. Then, the stability factor can be calculated as,

$$sf(d_i, d_j) = 1 - \sqrt{\frac{\sum_{k=1}^{n}\left[trust(d_i, d_j)^{T(k)} - \overline{trust}(d_i, d_j)\right]^2}{n}} \tag{10}$$

According to the above three factors, we can calculate the direct trust between two users as follows,

$$trust(d_i, d_j) = if(d_i, d_j) \times ctf(d_i, d_j) \times sf(d_i, d_j) \tag{11}$$

### 4.3. Trust calculation of indirectly linked users

Furthermore, users are connected by indirect paths, which present indirect trusts among users in the social network. Via Eq. (11), we provide the direct trust calculation method through users' direct interactions. However, there exists indirect trust, which exists in a path from the origin user to the target user through intermediate users. Here, we propose the trust calculation method for indirect paths. In our trust traversal, a traversed path is serial, which means a path from the origin user *rs* to the target user *rt* where each individual in this path is linked to only one reachable individual, which is expressed as $\exists(rs \to ru_1) \wedge \exists(ru_1 \to ru_2) \wedge \ldots \wedge \exists(ru_{i-1} \to ru_i) \wedge \exists(ru_i \to rt)$. This serial path is denoted as $\Phi(rs, rt)$. In addition, we extend the formulism of the serial path $\Phi(rs, rt)$to facilitate subsequent utilization in the proposed Trust Traversal. Let a given serial path be $\Phi(rs, rt)$. For an intermediate user $ru_i$ in $\Phi(rs, rt)$, the part of $\Phi(rs, rt)$ from the origin user *rs* to $ru_i$ can be denoted as $\Phi(rs, ru_{i-1}, ru_i)$.

Because, in the serial path, there is one and only one serial path $\Phi(rs, rt)$ between the origin user and target users, we use the following rules to compute trust.

- Trust transitivity rule. Assume that there exists a serial path $\Phi(rs, rt)$ and that we have already computed $trust(rs, ru_1)$, $trust(ru_{i-1}, ru_i)(i = 2, \ldots, n)$, and $trust(ru_n, rt)$; then,

$$trust(rs, rt) = trust(rs, r_1) + \sum_{i=2}^{n} trust(ru_{i-1}, ru_i) + trust(ru_n, rt) \tag{12}$$

- Reputation weight rule. In the serial path $\Phi(rs, rt)$, the reputation of a user (intermediate user or target user) is a weight parameter that can affect the trust value of the path. The weight parameter is denoted as $wr(u)$. The equation of the weight parameter is as follows.

$$wr(u) = reputation(u) \times \gamma^{1-\vartheta(u)} \tag{13}$$

In the above equation, $reputation(u)$ represents the reputation of user $u$ (intermediate user or target user), and $\vartheta(u)$ is the confidence factor. $\gamma$ is a regulation parameter, $\gamma \in [0, 1]$.
- Level attenuation rule. If there is a serial path $\Phi(rs, rt)$, then the trust of the path will be attenuated while the levels are increased. This means that, the more distant the target user is from the origin user, the less trust he/she gains. The attenuation factor is given by,

$$\kappa_{\Phi(rs,rt)} = \zeta^{(1-1/lev(rt))} \tag{14}$$

where parameter $\zeta \in [0, 1]$ is given in advance. For example, if *rt* is on the first level, i.e., $rs \to rt(lev(rt) = 1)$, the attenuation factor is $\kappa_{\Phi(rs,rt)} = (1/2)^{(1-1/1)} = 1$ ($\zeta = 1/2$ in this example). That means, there is a direct relationship between user *rs* and *rt* without attenuation. If *rt* is on the second level, i.e., $rs \to ru_1 \to rt(lev(rt) = 2)$, the attenuation factor is $\kappa_{\Phi(rs,rt)} = (1/2)^{(1-1/2)} \approx 0.71$. That is, there is an indirect relationship between user *rs* and *rt* with only one intermediate user, and then, the trust of path is attenuated. If *rt* is on the fifth level, then the attenuation factor is $\kappa_{\Phi(rs,rt)} = (1/2)^{(1-1/5)} \approx 0.57$. There is again an indirect relationship between user *rs* and *rt* with four intermediate users so that the trust is greatly attenuated. By that analogy the longer the path is, the less trust target user gains.

When we synthesize all three rules mentioned above, the trust in the serial path can be computed as follows.

$$trust_{\Phi(rs,rt)} = \Big\{[trust(rs, r_1) \times wr(r_1)] + \sum_{i=2}^{n}[trust(ru_{i-1}, ru_i) \times wr(ru_i)] + [trust(ru_i, rt) \times wr(rt)]\Big\} \times \kappa_{\Phi(rs,rt)} \tag{15}$$

### 4.4. Category of direct trust link

As mentioned above, trust of link is asymmetrical due to different interaction experiences of users. Therefore, links can be divided according to the closeness and mutual trustworthy degree of users. For example, user A has high trust to user B, while B has a very low trust to A. In such case, user B would drift apart with user A, which means an untrusted link from B to A. In this study, we define three types of trust links among users as strong trust link, weak trust link, and untrusted link as in definition 5. Here, we first propose the category of direct trust link to facilitate further trust link detection.

As mentioned in Section 2, we introduce two factors, the mutual trust factor and interaction frequency factor, for evaluating the type of trust links among users. Here, we give the two factors and their calculation methods as follows,

(1) Mutual trust factor. This factor is calculated by comparing their mutual trust degrees. Let the users $d_i$ and $d_j$ have trust degrees $trust(d_i, d_j)$ and $trust(d_j, d_i)$. The mutual trust factor can be calculated as,

$$mtf(d_i, d_j) = 1 - \frac{|trust(d_i, d_j) - trust(d_j, d_i)|}{trust(d_i, d_j) + trust(d_j, d_i)} \tag{16}$$

(2) Interaction active factor. In our opinion, obtaining more feedback on two users' interactions implies that they might have a closer relationship. That is, the greater the amount of feedback regarding two users' interactions, the higher the interaction active factor would be for them. Assume that $Interact(d_i, d_j)$ denotes the number of interactions from user $d_i$ to user $d_j$ and that $Feed(d_i, d_j)$ denotes the number of interactions that received feedback from user $d_j$ to $d_i$. Suppose that $Interact(d_i)$ denotes the total number of interactions from user $d_i$ and that $Feed(d_i)$ denotes the total number of pieces of feedback given from user $d_i$. Next, we calculate the interaction active factor as follows:

$$iaf(d_i, d_j) = \frac{1}{2} \times \left[\frac{Feed(d_i, d_j)}{Interact(d_i, d_j)} \times \left(\frac{Interact(d_i, d_j)}{Interact(d_i)}\right)^{\frac{1}{Feed(d_i, d_j)}} + \frac{Feed(d_j, d_i)}{Interact(d_j, d_i)} \times \left(\frac{Interact(d_j, d_i)}{Interact(d_j)}\right)^{\frac{1}{Feed(d_j, d_i)}}\right] \tag{17}$$

From Eq. (17), we can see that a higher ratio of interaction between users out of their total interactions and a greater the number of interactions with feedback imply a higher value of the interaction active factor. That is, frequent bi-directional interaction manifests that there is a close interaction relationship between users.

## 5. Trust traversal for link detection scheme-based trust

### 5.1. Single target user in trust traversal

Trust has been proven as being a factor in path detection [30,31]. Here, we first address the case of a single target user in trust traversal for finding the trust link by using trust as a probability evaluation factor. Starting from the origin user *rs*, we perform our trust traversal through users' direct paths with a probability of selecting the next user for traversal. Our proposed traversal method is similar to the *Random Walk* algorithm under a certain probability condition. At each step of trust traversal, we are at a current user (noted as *cur_u*) and need to make a decision regarding whether the traversal should continue on and which user should be selected as the next one. If the user *cur_u* is the target user, then we stop our traversal and calculate the final trust value of the existing link. If *cur_u* is not the target user, we have the following options:

(1) With probability $\chi_{forward}$, we continue our traversal and select a user (noted as $next\_u$) that satisfies the condition that $lev(next\_u) = lev(cur\_u) + 1$. That means it adds 1 to the length of the shortest link from the origin user to the selected next user.

(2) With probability $1 - \chi_{forward}$, we continue our traversal and select a user $next\_u$ that satisfies the condition that $lev(next\_u) = lev(cur\_u)$. That means that the length of the shortest link from the source to the selected user has not changed.

(3) If the out degree of user $cur\_u$ is 0, we need to backtrack our traversal to the previous user (noted as $pre\_u$) and then restart the trust traversal according to option (1) or (2).

Regarding the above options, option (1) implies that the trust link from the origin user to the target user would definitely be longer, and option (2) means that the length of the trust link would remain unchanged through link substitution (details of link substitution are discussed later). For option (3), we aim to prevent the traversal from entering an infinite loop if $cur\_u$ has no relationship to others (the out-degree is 0 in the social network graph), since there are many "*skeleton*" users in a social network.

Next, we discuss the former two options as follows,

(1) First, we need to decide the probability of the traversal selecting a user with a deeper level. We consider that the probability of $\chi_{forward}$ can be calculated based on level and the shortest distance between the next user and target user as follows,

$$\chi_{forward} = \begin{cases} \left(1 - \dfrac{1}{lev(cur\_u) + 1}\right)^{1 - \frac{1}{dis(Neighbor\_f(cur\_u), tar\_u)}} & cur\_u \neq rs \\ 1 \quad cur\_u = rs \end{cases} \tag{18}$$

where $Neighbor\_f(cur\_u)$ denotes the set of neighbors for forward traversal and $dis(Neighbor\_f(cur\_u), tar\_u)$ is the shortest length from users in $Neighbor\_f(cur\_u)$ to the target user. This implies that the traversal controller should query the current user to get a partial view of the network topology to obtain information regarding the shortest length to the target user. Then, the next node to visit is chosen on the basis of local information only.

Then, we address the method of selecting the next user. If we continue the traversal at the current user $cur\_u$ and select a user in the next level, we have to select one of the directly trusted neighbors of $cur\_u$ under a certain probability. Then, we have two rules for defining the probability calculation here: the trust link category and trust degree. That means that a strong trust link to a neighbor and a higher trust value imply a higher probability of being selected as the next traversal user. We define $next\_u \in Neighbor\_f(cur\_u)$ as the next user for selecting a user $v$ from $Neighbor\_f(cur\_u)$. Then, the probability of selecting user $v$ is

$$p(v) = \frac{(1-\eta)}{|Neighbor\_f(cur\_u)|} + \eta \times \frac{trust(cur\_u, v) \times wr(v)}{\sum_{w \in Neighbor\_f(cur\_u)} trust(cur\_u, w) \times wr(w)} \tag{19}$$

Here, we define a damping factor $\eta \in [0, 1]$ to denote the probability of randomly selecting a user as the next user. That means that each user has a probability if he/she is the direct neighbor of the current user. It can be used for selecting a new arrival neighbor as the next user in traversal under a certain probability even if he/she has relatively low trust with the current user.

In addition, because we aim to find a strong trust link between the origin user and target user in our trust traversal, a strong trust link between the current user and the next user is obviously more important than a weak trust link or untrusted link. Thus, we here propose a link factor $\tau(v)$ for evaluating the weight of user $v$ with his/her trust link category. In our study, the link factor $\tau(v)$ is calculated as

$$\tau(v) = \frac{trust(cur\_u, v) + trust(v, cur\_u) + mtf(cur\_u, v) + iaf(v, cur\_u)}{\sum_{w \in Neighbor\_f(cur\_u)} [trust(cur\_u, w) + trust(w, cur\_u) + mtf(cur\_u, w) + iaf(w, cur\_u)]} \tag{20}$$

Correspondingly, we have the final probability of selecting the next user as

$$p(next\_u = v) = p(v) \times \tau(v) \tag{21}$$

For preventing traversal to untrusted users, we define the probability from an untrusted link as follows,

$$p(next\_u = v) = 0(cur\_u \Leftarrow v) \vee (cur\_u \propto v) \tag{22}$$

(1) With probability $1 - \chi_{forward}$, we select a user as $next\_u \in Neighbor\_l(cur\_u)$ that has the same level as $cur\_u$ ($Neighbor\_l(cur\_u)$ denotes the set of neighbors who have the same level as $cur\_u$) based on the probability given by Eq. (21). That means that the length of the link would be longer, while the shortest distance from the origin user to the current user is not changed. Because the long path would bring trust damping, the traversal must evaluate whether such a selection is an optimized result or not. For the selected $next\_u$, the trust value of the link that passes over the current user$cur_u$ can be calculated based on Eq. (15), noted as $trust_{\Phi(rs, cur\_u, next\_u)}$. We now consider the following cases:

- There is a link from $pre\_u$ to $next\_u$, which forms a new link from the origin user to the next user. In such a case, the new link, noted as $\Phi(rs, pre\_u, next\_u)$, is shorter than the link $\Phi(rs, cur\_u, next\_u)$. Therefore, we need to evaluate which link is the optimized link with higher trust. Then, we have the following rule for the new $next\_u$:

$$\Phi(rs, next\_u) = \begin{cases} \Phi(rs, cur\_u, next\_u) & \text{if } trust_{\Phi(rs, cur\_u, next\_u)} \\ & \geq trust_{\Phi(rs, pre\_u, next\_u)} \\ \Phi(rs, pre\_u, next\_u) & \text{else} \end{cases} \tag{23}$$

- There is no link from $pre\_u$ to $next\_u$. The traversal scheme is allowed to select another temporary user (noted as $temp\_u$) that satisfies the condition that $lev(next_u) = lev(cur_u) + 1$with probability $p(temp\_u = v)$ given in Eq. (21). Then, we calculate the trust value of the link as $\Phi(rs, cur\_u, temp\_u)$. For the two links, $\Phi(rs, cur\_u, temp\_u)$ and $\Phi(rs, cur\_u, next\_u)$, we also evaluate which link is the optimized link with higher trust. As a result of the evaluation, the traversal scheme selects the link as

$$\Phi(rs, next\_u) = \begin{cases} \Phi(rs, cur\_u, next\_u) & \text{if } trust_{\Phi(rs, cur\_u, next\_u)} \\ & \geq trust_{\Phi(rs, temp\_u, next\_u)} \\ \Phi(rs, temp\_u, next\_u) & \text{else} \end{cases} \tag{24}$$

As shown in Fig. 2(a), the Trust traversal Scheme is at user C ($cur\_u = C$) and there are two options as: with probability of

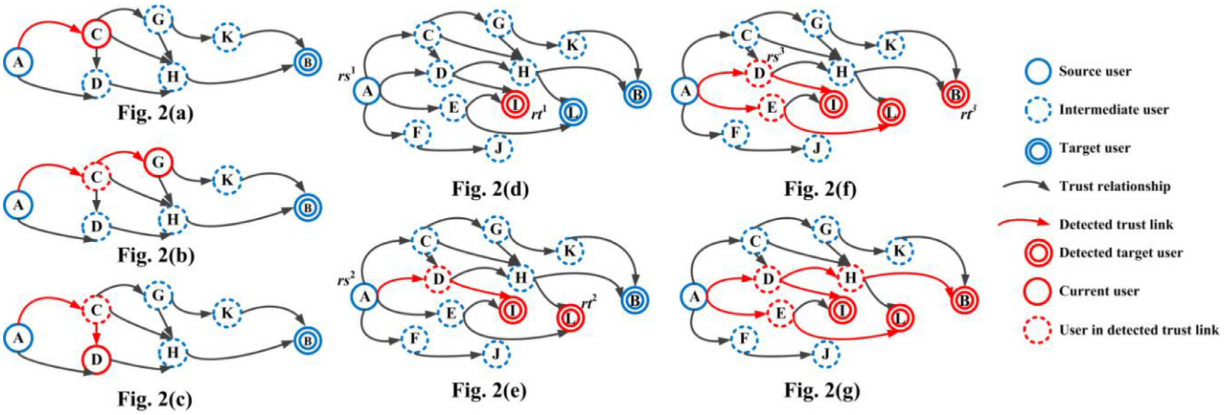


Fig. 2(a) Fig. 2(b) Fig. 2(c) Fig. 2(d) Fig. 2(e) Fig. 2(f) Fig. 2(g)

**Fig. 2.** An example of the Trust Traversal Scheme.

$\chi_{forward}$ for selecting a next user in set $Neighbor\_f(cur\_u) = \{G, H\}$ and with probability of $1 - \chi_{forward}$ for selecting a next user in set $Neighbor\_l(cur\_u) = \{D\}$. Then, we can get the calculations with above two probabilities as follows,

$$\begin{cases} \chi_{forward} = \left(1 - \dfrac{1}{lev(cur\_u)+1}\right)^{1-\frac{1}{dis(Neighbor\_f(cur\_u),tar\_u)}} = \left(1 - \dfrac{1}{1+1}\right)^{1-\frac{1}{2}} \approx 0.707 \\ p(next\_u = G) = p(G) \times \tau(G) \\ \quad = \left[\dfrac{(1-\eta)}{2} + \eta \times \dfrac{trust(C,G) \times wr(G)}{trust(C,G) \times wr(G) + trust(C,H) \times wr(H)}\right] \times \tau(G) \\ p(next\_u = H) = p(H) \times \tau(H) \\ \quad = \left[\dfrac{(1-\eta)}{2} + \eta \times \dfrac{trust(C,H) \times wr(H)}{trust(C,G) \times wr(G) + trust(C,H) \times wr(H)}\right] \times \tau(H) \end{cases}$$

$$\begin{cases} 1 - \chi_{forward} = 0.293 \\ p(next\_u = D) = p(D) \times \tau(D) \\ \quad = \left[\dfrac{(1-\eta)}{1} + \eta \times \dfrac{trust(C,D) \times wr(D)}{trust(C,D) \times wr(D)}\right] \times \tau(D) \end{cases}$$

If Trust traversal Scheme selects user G under above probability calculation, we can see the detected trust link is as in Fig. 2(b) ($cur\_u = G$ and detect trust link is $A \to C \to G$). If Trust traversal Scheme makes decision to select user D with probability of $1 - \chi_{forward}$, as shown in Fig. 2(c) ($cur\_u = D$ and $pre\_u = C$), we should consider two links of $\Phi(rs, pre\_u, next\_u) = A \to C \to H$ and $\Phi(rs, cur\_u, next\_u) = A \to C \to D \to H$. Then, we can get a detected trust link according to Eq. (23) by comparing the trust values of above two links.

### 5.2. Termination of a single target user in trust traversal

Under above trust traversal, we can find a trust link from origin user to target. Such traversal can be repeated iteratively for discovering all possible trust links between users. As a result of trust traversal scheme, there might be more than one trust link which can be detected among users with the times of traversal increasing.

For each traversal, there are the following possible alternatives in each traversal,

(1) The trust traversal reaches the target user in $n$ steps, and thus the length of the trust link is $n$. We denote such traversal processes as *active traversal*.
(2) The trust traversal cannot reach the target user in $n$ steps. We call these traversal processes *inactive traversal*.

Obviously, the links detected through active traversal must be considered as valid results, while the inactive traversal processes must be ignored in trust traversal. Based on the idea of *"six degrees of separation,"* which is mentioned widely [39], we set the max step of traversal as $n = 6$.

With the increasing times of trust traversal, we can obtain trust links in *active traversal* processes and their trust values. We use a factor $\overline{Trust}(rs, rt)$, which denotes the average value of trust links detected by *active traversal* processes. Then, for each trust link $\Phi_i(rs, rt)$ obtained through active traversal, we employ a filtering rule: 1) if $(rs \Leftarrow rt) \vee (rs \propto rt) \vee (|trust_{\Phi_i(rs,rt)} - \overline{Trust}(rs, rt)| \leq \varepsilon)$, the trust link $\Phi_i(rs, rt)$ is a valid trust link for trust traversal; 2) otherwise, $\Phi_i(rs, rt)$ is not critical for describing the trust link from the origin user to the target user and must be ignored in trust traversal.

After performing several instances of trust traversal, we can obtain trust links through active traversal from the origin user to the target user. Suppose that there are $m$ trust links in the trust link set $T\_\Phi(rs, rt)$, which is obtained by an active traversal process with the constraint of $(\forall \Phi_i(rs, rt), \Phi_j(rs, rt) \in T\_\Phi(rs, rt) \to \Phi_i(rs, rt) \neq \Phi_j(rs, rt))$. Then, we can terminate the trust traversal if there is no newly detected trust link $\Phi_k(rs, rt) \in T\_\Phi(rs, rt)$ that continuously satisfies the constraint $|trust_{\Phi_k(rs,rt)} - \overline{Trust}(rs, rt)| \leq \varepsilon$ in numerous instances of traversal. It is also noted that we need to set a minimum number of trust traversals to ensure that the trust link works well in the cold-start state. Meanwhile, if no trust link can be detected after a maximum threshold number of traversals from the origin user to the target user, we consider there to be no trust link between them.

### 5.3. Multiple target users in trust traversal

For origin user, if there is more than one target user in his/her trust traversal scheme, we should find trust links which connect origin user and all target users with high trust relationships. In such a case, trust traversal would not terminate until all target users are included in a trust link. Here, starting from the origin user $rs$, we perform the trust traversal with multiple target users through users' direct links with the probabilities of selecting the next user. Suppose that the set of target users is $Tar = \{rt_1, rt_2, \ldots\}$ and that, at each step of the trust traversal, the shortest distance from the current user *cur_u* to the target user $rt_i \in Tar$ is denoted as $dis(cur\_u, rt_i)$.

At each step of trust traversal, we are at a current user (noted as *cur_u*) and need to make a decision regarding whether the traversal should continue on and which user would be selected as the next one. If the user *cur_u* is a target user $rt_i \in Tar$, then we delete the target user as $Tar = Tar - \{rt_i\}$ and continue traversal until $Tar = \emptyset$. Therefore, the trust traversal with multiple target users can be composed of several rounds of trust traversal with a single target user. Because the problem of graph traversal is NP-hard, the rules of our trust traversal with multiple target users are as follows.

**Rule 1. Origin user selection.** In the *k-th* round, the origin user and target user are denoted as $rs^k$ and $rt^k$. For each round, the origin user is selected based on probability (the origin user in the first round is called the original origin user, and $rs^1 = rs$).

According to rule 1, we must make a decision regarding which user can be the origin user in the Trust Traversal Scheme in each round except the first round. In our consideration, there are the following aspects for selecting a new origin user in each round: trust relationship with the original origin user and the possibility of resulting in *active traversal* processing. Thus, we propose a selection method for determining a new origin user in the *k-th* round as follows.

(1) In first round, the trust traversal starts from the original origin user, while the user who is in the obtained trust link from the original origin user and the target user $rt_i \in Fin$, where the set *Fin* records traversed target users, can be selected as a new origin user with probability $p(rs^k = v)$ in other rounds;
(2) The probability $p(rs^k = v)$ is calculated as,

$$p(rs^k = v) = \frac{trust(rs^1, v) \times s(v)^{\frac{1}{dis(rs^1, v)+1}}}{\sum_{v_i \in \Phi(rs^1, Tar)} \left[ trust(rs^1, v_i) \times s(v_i)^{\frac{1}{dis(rs^1, v_i)+1}} \right]}$$
$$s(v_i) = \begin{cases} \varsigma_1 & v_i \Leftrightarrow rs^1 \\ \varsigma_2 & v_i \Leftarrow rs^1 \quad trust(v_i, v_i) = 1 \quad s(rs^1) = 1 \\ \varsigma_3 & \text{else} \end{cases} \tag{25}$$
$$\varsigma_1, \varsigma_2, \varsigma_3 \in [0, 1]$$

where we set a weight $s(v_i)$ to distinguish the importance of different trust links. Condition$v_i \in \Phi(rs^1$, *Tar*) denotes the user $v_i$ in the traversed trust links, which is obtained in the previous traversal round, from the origin user to any target user in *Fin*.

**Rule 2. Target user selection.** For all target users $rt_i \in Tar$, they are sorted as $lev(rt_1^k) \le lev(rt_2^k) \le lev(rt_3^k) \le \ldots$ at the beginning of the *k-th* round. There are the following rules for selecting the target user in each round:

(1) Before the traversal begins in each round, the target user $rt_i \in Tar$ is selected as $rt^k$ with the constraint $\min\{lev(rt_i^k)\}$ (if there are two or more $rt_i \in Tar$ that satisfy the constraint, we select one of them at random);
(2) At each step in the round, if a target user $rt_i \in Tar$ satisfies the constraint $\min(dis(cur\_u, rt_i)) \le lev(rt^k)$, it is selected as the new target user $rt^k = rt_i$ in this round.

**Rule 3. Termination of the Trust Traversal Scheme with multiple target users.** For traversal processing, we have the following alternatives for controlling termination.

(1) In the *k-th* round, if the constraint $(rs^k \Leftarrow cur\_u) \vee (rs^k \propto cur\_u)$ is satisfied, this traversal is terminated as an *inactive traversal* process. Then, the trust traversal selects another origin user with probability $p(rs^k = v)$ to restart this round of traversal;
(2) In the *k-th* round, if the traversal satisfies the termination of a single target user as $(cur\_u = rt_i) \wedge (rt_i \neq rt^k)$, this round is terminated and $Tar = Tar - \{rt_i\}$, $Fin = Fin \cup \{rt_i\}$;
(3) Each round is terminated based on the termination rule of trust traversal with a single target user, while trust traversal with multiple target users is terminated when $Tar = \emptyset$.

For example, we can see a trust traversal with multiple target users as shown in Fig. 2(d). In the first round, we can get the origin user as $rs^1 = A$ and target user as $rt^1 = I$ because user *I* has the minimum level of all target users. When the first round finishes, we can get a detected trust link (*A*→*D*→*I*) as shown in Fig. 2(e). In the second round, we can select users (users *A*, *D*, and *I*) as the origin user from users who are listed in the detected trust link (red link) according to Eq. (25). Then, target user L is selected as $rt^2 = L$ according to rule 2, and the result of the second round is as in Fig. 2(f) after traversal. In the third round, users in the detected links (*A*→*D*→*I* and *A*→*E*→*L*) can be selected as the origin user by Eq. (25). Finally, we can get the detected trust links (red links) as shown in Fig. 2(g). Note that the result in this example in Fig. 2(d)–(g) is not unique since the Trust Traversal Scheme would have different choices at each step according to the probability calculations, which would lead to an uncertain solution.

### 5.4. Confidence of a detected trust link

Traditionally, most trust computation methods aim to give users the trustworthiness for their future predictions. We can see that direct trust reflects the trustworthiness between two directly connected users in a social network. In a similar manner, a trust link in this study reveals the indirect trustworthiness between two or more users since the value of the trust link is calculated based on the trust relationship of each of the two directly connected users and their reputations in the trust link. Through a trust link, a user can obtain a detailed list of his/her possible strong and weak trust relationships with other users. However, it is still not sufficient for users to make decisions as to whether the relationships are reliable or not. That is, we should know how confident the detected trust links are. We consider there to be three aspects for evaluating the confidence of a detected trust link: reputation, standard deviation, and detected ratio in Trust Traversal.

(1) Because reputation reflects the objective trustworthiness of a user, we consider better reputation to imply greater confidence in a certain trust link. For a detected trust link $\Phi_i$(*rs, rt*), the reputation value of an intermediate user $ru_j \in \Phi_i$(*rs, rt*) in this trust link is *reputation*($ru_j$). Assume that the total number of intermediate users in $\Phi_i$(*rs, rt*) is *n* and that, for $ru_j \in \Phi_i$(*rs, rt*), he/she appears *num*($ru_j$) times in all detected trust links $\Phi$(*rs, rt*). Then, the confidence of the reputation in a trust link $\Phi_i$(*rs, rt*) is,

$$cor_{\Phi_i(rs,rt)} = \frac{\sum_{ru_j \in \Phi_i(rs,rt)} \left[ reputation(ru_j) \times s(ru_j)^{\frac{1}{num(ru_j)}} \right]}{n} \tag{26}$$

where $s(ru_j)$ is as defined in Eq. (25).

(1) Standard deviation aims to reflect the confidence in our detected trust links. The lower the standard deviation of a detected trust link, the more confident we are in the Trust Traversal Scheme results. For all $\Phi$(*rs, rt*), we denote their average as $\overline{Trust}(rs, rt)$, and the total number of detected

**Table 2**
Characteristics of five communities in the examination prototype.

| Community | Education | Financial | Sporting | Entertainment | Social |
|---|---|---|---|---|---|
| Number of IDs | 526 | 477 | 414 | 718 | 692 |
| Number of posts | 724 | 681 | 1107 | 1591 | 938 |
| Number of comments | 45,037 | 39,164 | 39,271 | 58,963 | 37,765 |

trust links is $|\Phi(rs, rt)|$. Then, the confidence of the standard deviation is

$$cos_{\Phi(rs,rt)} = \sqrt{\frac{\sum_{\Phi_i(rs,rt)} \left[trust_{\Phi_i(rs,rt)} - \overline{Trust}(rs,rt)\right]^2}{|\Phi(rs,rt)|}} \quad (27)$$

(2) In our consideration, the more times a trust link is detected in the Trust Traversal Scheme, the more confident we are in this trust link. Therefore, we introduce the detected ratio, noted as $cod_{\Phi_i(rs,rt)}$, in the confidence evaluation for the trust link.

According to the above three aspects, we give the confidence of a detected trust link, $\Phi(rs, rt)_i$, as follows,

$$confidence_{\Phi_i(rs,rt)} = cor_{\Phi_i(rs,rt)} \times \left(cod_{\Phi_i(rs,rt)}\right)^{cos_{\Phi(rs,rt)}} \quad (28)$$

## 6. Experimental results and analysis

In this section, we propose a set of experiments to verify the performance of our proposed scheme. In our experimental scenario, the data comes from the *Sina.com micro-blog* platform, which is very popular in China. We collected micro-blog data manually from *Sina.com*. Our data includes approximately 1251 IDs (some IDs are located in two or more communities) and more than 170,000 records (including posts and comments). In the dataset, there is a one-way direct link from one user towards another if he/she follows the user in a *Sina.com* micro-blog. All direct links are generated from the initial dataset and are fixed and invariable in our examination. All of these data are used to calculate the initial trust and form network topological information based on a real-world source.

Based on the micro-blog initial data, we developed a prototype for examining our proposed scheme in this study. Due to the dependency of community information on reputation and trust calculation, we here give the initial setting of five communities, i.e., education, financial, sporting, entertainment, and social. Detailed characteristics of the five communities in the prototype are shown in Table 2. In addition, because there is no reputation center in *Sina.com* micro-blogs, reputation values of IDs are initially set by following a normal distribution with mean 0.7 and variance 0.1 in our prototype. In addition, for further verification of trust effectiveness, we deploy an additional approximately 500 users (including honest users and malicious users) in the prototype, and the average out-degree of a user was 7.

Meanwhile, we have the following justifications for the parameter settings in our experimental scenario,

- Because the average reputation of honest users is larger than 0.6 according to our observations, we assign a random reputation value between 0.6 and 0.8 for a new-arrival user;
- There are the following value settings of parameters in equations: $p = 1$ (Eq. (6)); $\phi_1 = 1, \phi_2 = 0.8, \phi_3 = 0.5$ (Eq. (9)); $\zeta = 1/2$ (Eq. (14)), and $\varsigma_1 = 1, \varsigma_2 = 0.5, \varsigma_3 = 0$ (Eq. (25));
- The minimum number of trust traversals is set as 200 in a cold-start state, and the threshold for the maximum number of traversals is set as 1000 for traversal termination.

**Table 3**
Description of scoring for actions in the experiment.

| *Action* | *Score* | *Level* |
|---|---|---|
| *Approving* | 1 | The highest level of a trustworthy opinion |
| *Bookmarking* | 0.7 | Supportive opinions or holding interested attitudes, mostly with a positive opinion |
| *@+ID* | 0.6 | Explicit opinions expressed between users, many of them with a positive opinion |
| *forwarding* | 0.5 | Behaviors with neutral (or unknown, or implicit) opinions |

In addition, because there is no scoring system in Sina.com micro-blogs, we conducted a data pre-processing method, concerning different types of interactions listed in Table 1, to meet the requirement of scoring a number from the [0,1] interval in our experiment as follows,

- For the behavior $act(d_i, d_j)$ in Table 1, there are four types of actions that can be used for scoring: bookmarking, forwarding, approving and the "*@+ID*" action. Each action is seen as an independent behavior for scoring in our experiment. The score of each action is set according to the different level of the user's trustworthy opinion, as in Table 3,
- For behavior $judgment(d_i, d_j)$, we used the Chinese language processing method to get the sentiment contained in the post. We developed a sentiment recognition model to extract the sentiment of each judgment text with a Chinese sentiment dictionary (Hownet and NTUSD) and sentiment Corpus. This model can recognize the polarity of judgment text (positive, neutral, and negative) and calculate the degree of the sentiment of the judgment. We defined the score intervals of different sentiments as [0.6–1], [0.4–0.6], and [0–0.4] for positive, neutral, and negative sentiments, respectively. The degrees of judgment were calculated based on polarity evaluation and were obtained according to the facts of the frequencies and weights of sentiment words in text.

### 6.1. *Trust calculation analysis*

#### 6.1.1. *Performance comparison of reputation computation*

We implemented the following methods and compared their performance in terms of reputation calculation.

- EigenRep method (ER): We used the method of EigenRep [38] as discussed in Section 2.1, which first establishes the local trust, i.e., EigenTrust, between directly linked users and then calculates the global reputation based on the local trust through iterations between neighbor users;
- Average judgment method of reputation (AJ): We used the average of all judgments from all users voting on the target user for the target user's reputation aggregation. This method is easy to realize and widely used in many existing reputation systems;
- Our proposed reputation aggregation without time attenuation (Non-TA): We used this method in comparison to examine the impact of time attenuation on reputation aggregation. In this method, the effect of time attenuation was not taken into account for reputation. That is, equations of 3 and 4 were not included in reputation aggregation;

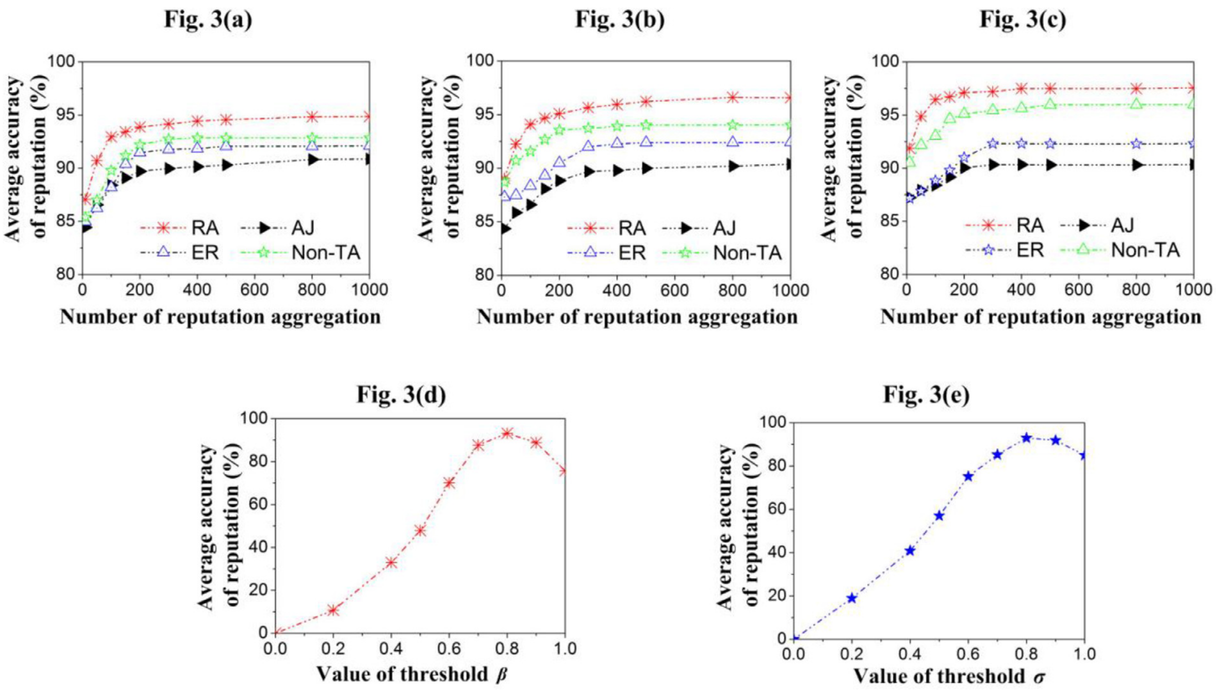


**Fig. 3.** Performance of reputation computation without malicious users.

- Our proposed reputation aggregation (RA). This method is the same as the method that we proposed in Section 4.1, which calculates the reputation based on Eqs. (1)–(4);
- Our proposed reputation aggregation with additional reputation confidence factors (RA-CF). This method includes our proposed additional confidence factors, i.e., judgment consistency and hit rate, as calculated by Eqs. (5)–(7).

In our experimental scenario, we use a factor named average accuracy of reputation aggregation to measure the performance of the proposed reputation aggregation. If an honest user gets a reputation value larger than 0.6 or a malicious user gets a reputation value lower than 0.4, the reputation aggregation is denoted as an accurate reputation; otherwise, the method gets an error reputation for a user. In addition, a user who gets a reputation value between 0.4 and 0.6 is seen as an unidentified one, which also causes an error and decreases the accuracy of reputation aggregation. Therefore, we formulate the term of accuracy as 1) *AH*: the set of honest users who are identified as honest ones accurately through our method; 2) *FH:* the set of honest users who are falsely identified as malicious ones; 3) *AM*: the set of malicious users who are accurately identified as malicious ones; 4) *FM*: the set of malicious users who are falsely identified as hones ones, and 5) *UN*: the set of users who cannot be identified as honest or malicious through our method. Therefore, the accuracy term is calculated as follows in our experiment:

$$accuracy = \frac{|AH| + |AM|}{|AH| + |FH| + |AM| + |FM| + |UN|} \times 100\%$$

where $|AH|$, $|FH|$, $|AM|$, $|FM|$, and $|UN|$ denote the numbers of users in the corresponding sets.

(1) Reputation calculation comparison without malicious users

In this examination, we aim to reveal the performance of the proposed method of reputation in this study. We select 100 users at random to calculate their reputation values and record the average accuracy of reputation aggregation. For performance evaluation, we introduce the following methods for comparison: the EigenRep trust method (ER), average judgment method of reputation (AJ), proposed reputation aggregation without time attenuation (Non-TA), and our proposed reputation aggregation (RA). In this experiment, we set three types of experimental environments: users only belonging to 1 community, users belonging to 3 communities, and users belonging to 5 communities. In our examination, all users interacted with others under above three types of experimental environments, respectively. The average accuracies of reputation calculations are shown in Fig. 3(a)–(c) as follows, and we can see that our reputation aggregation (RA) has the best performance in all methods. By comparison, the average accuracy of our proposed reputation aggregation is approximately 96.32%, which is approximately 6.8%, 4.1%, and 2.1% higher than the average accuracies of AJ, ER, and Non-TA methods, respectively. In our consideration, we think that the reasons behind the results are: users' reputations in the average judgment method (AJ) received all feedback judgments without any discrimination and were calculated equally; and factors of community and time-oriented dynamics are not taken into account in the methods of EigenRep and the proposed reputation aggregation without time attenuation. In our proposed method, all feedback judgments were impacted by qualification values for effective judgments. That means that our computation method can better reflect the weight and validity of each feedback in reputation computation. In addition, reputation is an attenuation value based on Eqs. (3) and (4), which reflect the dynamic feature of reputation rather than compute it statically as in other methods. The results reveal that users can obtain more accurate reputations when their interaction partners are located in multiple communities, and thus their reputations and positions are authoritative and widely admitted. Such advantages would bring more contributions in reputation computation.

Moreover, we evaluate the impact of two thresholds, $\beta$ and $\sigma$, in Eqs. (1) and (3), respectively. In Fig. 3(d) and (e), we can see that threshold values approximately 0.7–0.9 are reasonable based on the test validation. The values of these two thresholds are set as 0.85 and 0.85, respectively, in our following examinations.

(2) Reputation effectiveness with malicious users

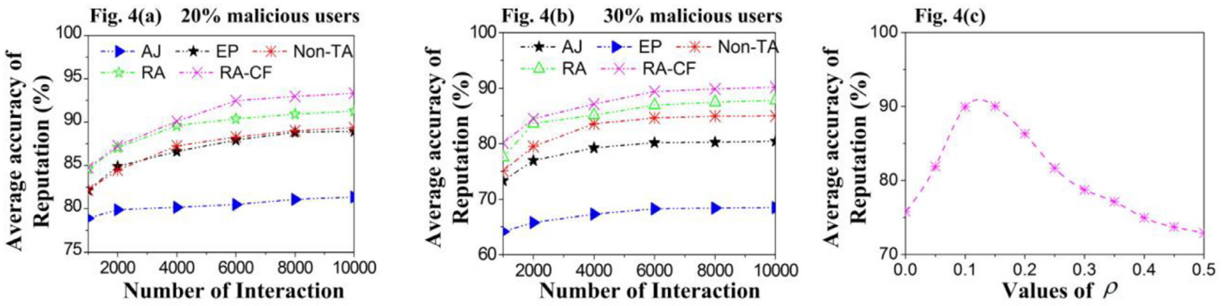


**Fig. 4.** Performance of reputation computation with malicious users.

Furthermore, we examine the average accuracy of reputation aggregation with malicious users. For comparison, we still use the same methods as in the above examination for reputation aggregation: the EigenRep trust method (ER), average judgment method of reputation (AJ), proposed reputation aggregation without time attenuation (Non-TA), our proposed reputation aggregation (RA), and additionally the proposed reputation aggregation with reputation confidence factor (RA-CF). In this examination, we conduct two groups of examinations with 20% and 30% additional malicious users, respectively, and then record the average accuracies of different methods after 10,000 iterations of reputation aggregation. From Fig. 4(a) and (b), we can see that the average accuracies were decreased as the malicious users increased. Our proposed method yields the best performance in all comparison methods. We find that a rating from a user is weighted by its qualification effectiveness value such that poorer reputation implies that the rating receives less significance in reputation calculation. Additionally, the accumulated reputation is also attenuated as time passes. Moreover, our reputation confidence factor reveals the reliability of a reputation by evaluating the stability of the reputation voting process and the effectiveness of predicting the future trustworthiness of the user. Therefore, most users can obtain reputation values that they deserve in our proposed reputation aggregation method.

In addition, we examine the impact of the value setting of error $\rho$ in discerning the hit rate for RA-CF. We record the average accuracy of reputation for RA-CF with different values of $\rho$. As shown in Fig. 4(c), setting the value of $\rho$ approximately 0.1–0.15 is reasonable.

### 6.1.2. Performance comparison of trust computation

We utilize examinations for verifying the effectiveness of our proposed trust calculation by evaluating whether our proposed trust computation method can accurately establish direct or indirect trustworthy opinions among users in social networks to distinguish between honest and malicious users.

(1) Direct trust computation

First, we examine the performance of direct trust calculation proposed in this study. For comparison, we implemented the following methods and compared their performance in terms of direct trust calculation.

- EigenTrust method (ET): We used users' past interaction records to calculate the local trust among them as proposed in [38]. As defined in [38], this method can also be seen as a transitive trust among indirect users: if a user trusts another target user, it would also trust the users trusted by the target user;
- weighted average trust rating (WA): As we discussed in Section 2.1, we used the reputation value of the judgment of the origin user as a weight and then calculated the average of all judgments from the origin user to the target user for their direct trust [40];
- ultimate trust rating (UT): we used the ultimate trust rating method [41], which was discussed in detail in Section 2.1;
- Our proposed trust computation method (TC): we implemented the method of trust computation given in Section 4.2 for direct trust computation based on Eqn. (8)–(11).

We record the trust values among users to evaluate whether the trust values reflect the authentic trustworthiness between users, and we employ approximately 10,000 interactions for users. If an honest user gets a belief value greater than 0.6 for another honest user or a trust value lower than 0.4 for a malicious user, his/her trust regarding the target user is considered accurate. Fig. 5 shows the average trust accuracy; we conduct three tests with 10%, 20% and 40% additional malicious users, and the results are depicted in Fig. 5(a)–(c).

We can see that the ultimate trust rating method gets the best performance among all methods, while our proposed trust computation method in this study yields performance that is near that of the UT method and better than the methods of WA and ET. The ultimate trust method provides trust calculations via its dynamic adjusting of risk factors and confidence factors based on interactions, and thus it enables all users to maintain trustworthy knowledge

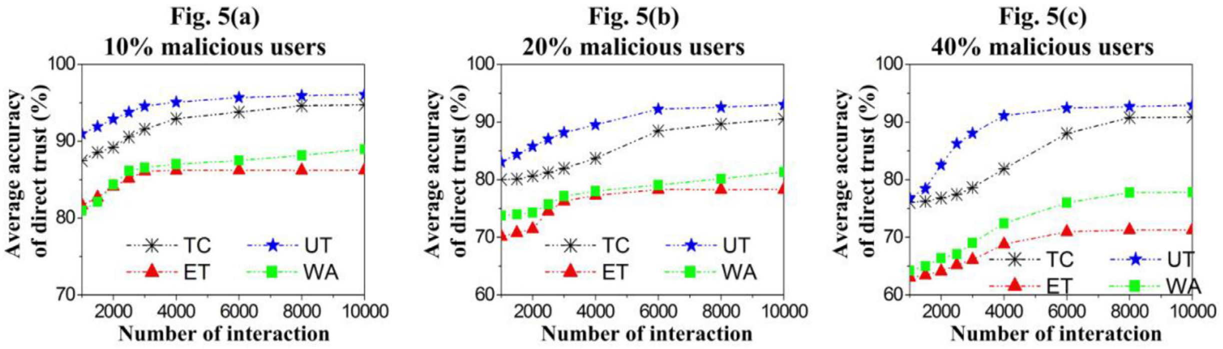


**Fig. 5.** Performance of direct trust computation.

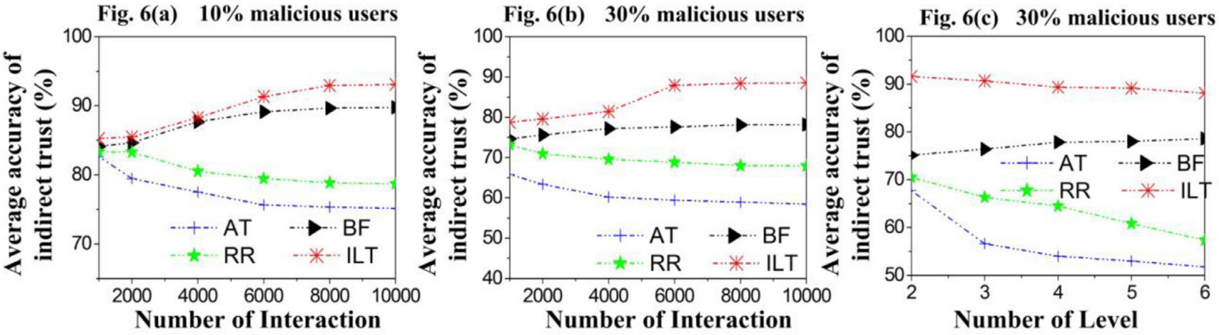


**Fig. 6.** Performance of indirect trust computation.

about others to detect malicious behaviors in interactions. Our proposed method can provide a more comprehensive evaluation of trust relationships among users through the proposed factors. In a weighted average trust rating, reputation is the weight in trust calculation such that higher reputation users can obtain more accurate trust relationships, but there is no punishment method in the ET method. As a result, if a malicious user has a high reputation, it cannot be detected perfectly by WA and ET. Additionally, there are a few users that calculated trust values as 0.4–06, and these trust relationships are regarded as inaccurate in our examination. Therefore, the average accuracy of trust computation is decreased as the number of malicious users increases.

(2) Indirect trust computation

Here, we test the performance of indirect trust computation. All indirect trust values are calculated in serial links between users because the Trust Traversal Scheme obtains trust values among users through a serial link. We select users who have no direct interactions at random and then calculate their indirect link trust values through our proposed indirect link trust (ILT) computation method. For comparison, we implemented the following four methods,

- Trust proposed in the Bellman-Ford algorithm (BF): We implemented the Bellman-Ford algorithm-based trust [16] as discussed in Section 2.1;
- Average trust rating (AT): In this method, we first calculated the direct trust values among users in the indirect serial path through the average of direct judgments, and then the indirect trust of the serial path was calculated based on the average of all direct trust values among users in the serial path;
- Indirect trust based on reputation aggregation (RR): In this method, we also first calculated the direct trust values among users in the indirect serial path through the average of direct judgments, and then the indirect trust of the serial path was calculated based on the weighted average of all direct values among users in the serial path by using the reputation values of intermediates and target user as weights;
- Our proposed indirect trust computation method (ILT): We implemented the proposed trust computation method as in Section 4.2 based on Eqs. (12)–(15).

The results are shown in Fig. 6(a) and (b). After 10,000 instances of comparison, the average accuracy of our proposed trust method is better than other methods under 10% and 30% malicious users. We think that the reasons are (1) the long distance of link is punished in BF and our methods by introducing the level value as an attenuation factor for improving the trust accuracy; and (2) the trust value is impacted by the reputation weight in our proposed method such that the trust would be attenuated if there were poor-reputation users in the link.

Moreover, we verify the impacts of the numbers of levels. Fig. 6(c) shows that, as the number of levels increases, the accuracies decrease. In the BF method, accuracy is nearly stable because long links are deemed untrustworthy. In our method, trust is attenuated as the level increases according to level attenuation rules. Because of this, most untrustworthy impacts of high-level users are avoided in our method.

### 6.2. Experiment for categorization of direct trust links

In this experiment, we aim to reveal the performance of the direct trust link classification method in this study. Because the dataset is collected from a microblog platform, we here use the interaction data in the microblogs to evaluate the performance of trust links. In the experiment, we evaluate and obtain a trusted set that denotes the users who are trusted by the origin user and then calculate a set of indicators, which are given in Table 4.

In the examination, we record the average values of indicators for all users in our initial dataset. For comparison, we give the following methods: the EigenTrust method (ET), weighted average trust rating (WA), ultimate trust rating (UT), proposed direct trust computation method (TC), strong trust link method (STL), and weak trust link method (WTL) for evaluation. The results are shown in Table 5. We can see that the performances of UT, TC, and WTL, i.e., *Followed rate, Hit rate, Forwarding rate*, and *Agreement rate*, are relatively similar to each other and obviously higher than the performances of the ET and WA methods. Meanwhile, STL obtains the best performance except for the indicator of coverage, and we analyze that the reasons for such a result are (1) the high strength of a strong trust link implies high mutual trust and maintains a high interaction feature between users, which leads to a high followed rate, forwarding rate, and agreement rate and a low error rate; and (2) a strong trust link requires a high mutual trust degree between users, which reduces the coverage rate by excluding low inverse trust relationships.

In addition, we evaluate the impact of parameters on the categorization of direct trust links. We test the different values of $\varpi_1$, $\varpi_2$, $\varpi_3$ for accuracy, as shown in Fig. 7(a)–(c). The results in Fig. 7 show that the values of three thresholds approximately 0.6–0.7, 0.7–0.8, and 0.5–0.6 are approximate reasonable compromises. In our consideration, an overly low threshold value can result in untrusted links that cannot be filtered or weak trust links that are identified as strong trust links, while an overly high threshold can result in weak or strong trust links that are classified incorrectly. In our other examinations, the default values of thresholds $\varpi_1$, $\varpi_2$, $\varpi_3$ are set as 0.65, 0.75, and 0.55, respectively.

### 6.3. Performance evaluation of the trust traversal scheme

In this examination, we select origin users and target users (single user or multiple users) at random to detect indirect trust links

**Table 4**
Indicator descriptions for performance evaluation of the categorization of trust links.

| Indicator | Description |
|---|---|
| *Followed rate* | The rate of a user being followed by others who he/she trusts. |
| *Hit rate* | The rate of a user's posts being hit by others who he/she trusts. |
| *Forwarding rate* | The rate of a user's posts being forwarded by others who he/she trusts. |
| *Agreement rate* | The rate of a user's opinions being agreed with by others who he/she trusts. |
| *Coverage* | The rate of a user's trust links with others who he/she deserves to trust. |
| *Error rate* | The rate of a user's trust links with others who he/she establishes with mistakes. |

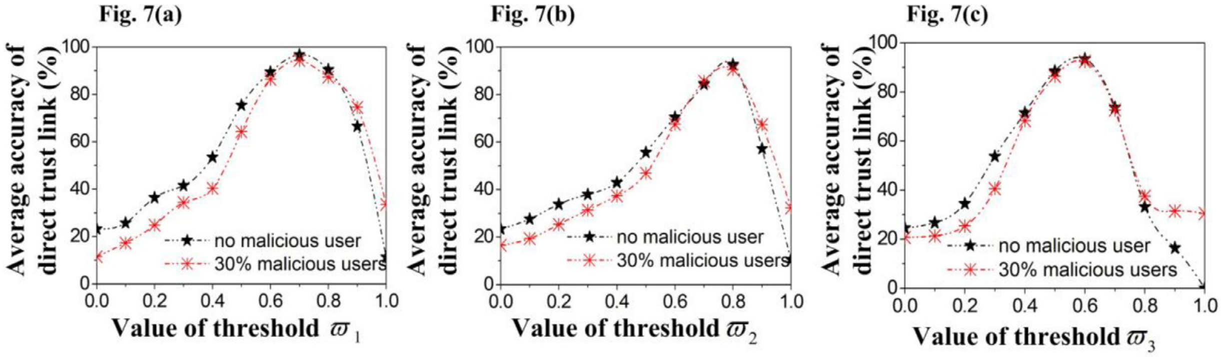


**Fig. 7.** Impact evaluation of thresholds $\varpi_1$, $\varpi_2$, $\varpi_3$ for direct trust links.

**Table 5**
Experimental results for direct trust links.

| Indicator | ET (%) | WA (%) | UT (%) | TC (%) | WTL (%) | STL (%) |
|---|---|---|---|---|---|---|
| *Followed rate* | 47.41 | 47.95 | 65.34 | 64.34 | 64.81 | 91.45 |
| *Hit rate* | 63.98 | 65.91 | 83.84 | 80.04 | 84.63 | 93.21 |
| *Forwarding rate* | 30.87 | 35.47 | 48.34 | 45.24 | 48.54 | 68.67 |
| *Agreement rate* | 47.58 | 48.24 | 62.34 | 60.68 | 60.27 | 83.64 |
| *Coverage* | 82.45 | 83.78 | 95.54 | 90.34 | 71.35 | 55.37 |
| *Error rate* | 10.87 | 11.54 | 1.28 | 5.38 | 4.01 | 0.84 |

between them. We also utilize the indicators listed in Table 2 to evaluate the performance of our proposed Trust Traversal Scheme with a single target user (STTS). All indicator values of STTS are obtained from interactions along with trust links detected by STTS. We compare the results for different methods. Following is the description of methods we use for comparison in this experiment: an algorithm (BF), average trust rating (AT), indirect trust of received rating aggregation (RR), random walking method with 6 steps (RAN), and our proposed trust calculation (STTS). In these above methods, the indirect paths from the origin user to target user, which have trust values larger than the threshold of 0.6, are noted as valid trust paths, and then the indicators are collected. In our proposed Trust Traversal Scheme, we record the indicators from all trust links returned by STTS. The results are shown in Fig. 8(a), and we can see that our proposed STTS obtains the best performance. In addition, we use the confidence factor to evaluate the error of our method. For comparison, the calculation of confidence in other methods is also based on $cor_{\Phi(rs,rt)_i}$ and $cos_{\Phi(rs,\ rt)}$, while $cod_{\Phi(rs,rt)_i}$ is calculated as the weight of the trust path in the returned trust path set. From Fig. 8(b), the average confidence value of STTS is larger than that of other methods. That means that the trust links detected by STTS have high reliability based on the confidence factor.

We examined the performance of avoiding the establishment of untrusted links with malicious users or untrustworthy users through STTS by first setting the additional malicious users at 30%. In our examination, we used two thresholds, *threshold_trust*, and *threshold_confidence*, and defined any detected trust link through STTS that satisfied the condition $(trust_{\Phi_i(rs,rt)} \leq threshold\_trust) \vee (confidence_{\Phi_i(rs,rt)} \leq threshold\_confidence)$ as an invalid trust link with a malicious or untrustworthy user. We select an honest user as the origin user and an additional malicious or honest user as the target user randomly, and then execute STTS for performance evaluation. In Fig. 8(c), STTS yields the best performance with *threshold_trust*and *threshold_confidence* approximately 0.6–0.7 and 0.4–0.5, independently. Lower values of thresholds result in malicious or untrustworthy links that cannot be excluded as invalid links, while higher values of thresholds filter valid trust links between honest users, both of which decrease the accuracy of STTS. Fig. 8(d) shows that, for different values of $\langle threshold\_trust, threshold\_confidence\rangle$, the accuracy for excluding malicious or untrustworthy trust links through STTS is at least 86.7%, and the accuracy is approximately 91.5% in the strictest threshold setting with an average accuracy of approximately 89.2%. In our consideration, the values of our thresholds, *threshold_trust* and *threshold_confidence*, are set as 0.7 and 0.5 for a reasonable compromise. Fig. 8(e) shows the impact of parameter $\varepsilon$ on the termination of STTS. We can see that both low and high values of $\varepsilon$ result in poor performance of STTS. Then, we can see that a value of $\varepsilon$ approximately 0.05–0.1 is an approximately reasonable compromise.

Because each round of the Trust Traversal Scheme with multiple target users can be regarded as STTS, which implies a similar traversal strategy with STTS in each round, we here mainly reveal the feasibility and effectiveness of our proposed Trust Traversal Scheme with multiple target users (MTTS). We select 300 honest users as origin users from the initial dataset and set 2–10 target users randomly for each of them so that they can detect trust links. By MTTS, we obtain the set of trust links from each origin user to their appointed target users. Then, we record the indicators: the average number of traversal steps for connecting the origin user and each target user, the coverage rate of connecting the origin user and target users, and the average accuracy of trust link detection. The results are shown in Fig. 9(a)–(b). Fig. 9(a) shows that MTTS can establish trust links for users with lower average numbers of traversal steps because it allows the traversal scheme to selects users from the detected trust link as a new origin user and then restart a new round of traversal, which results in a reduction

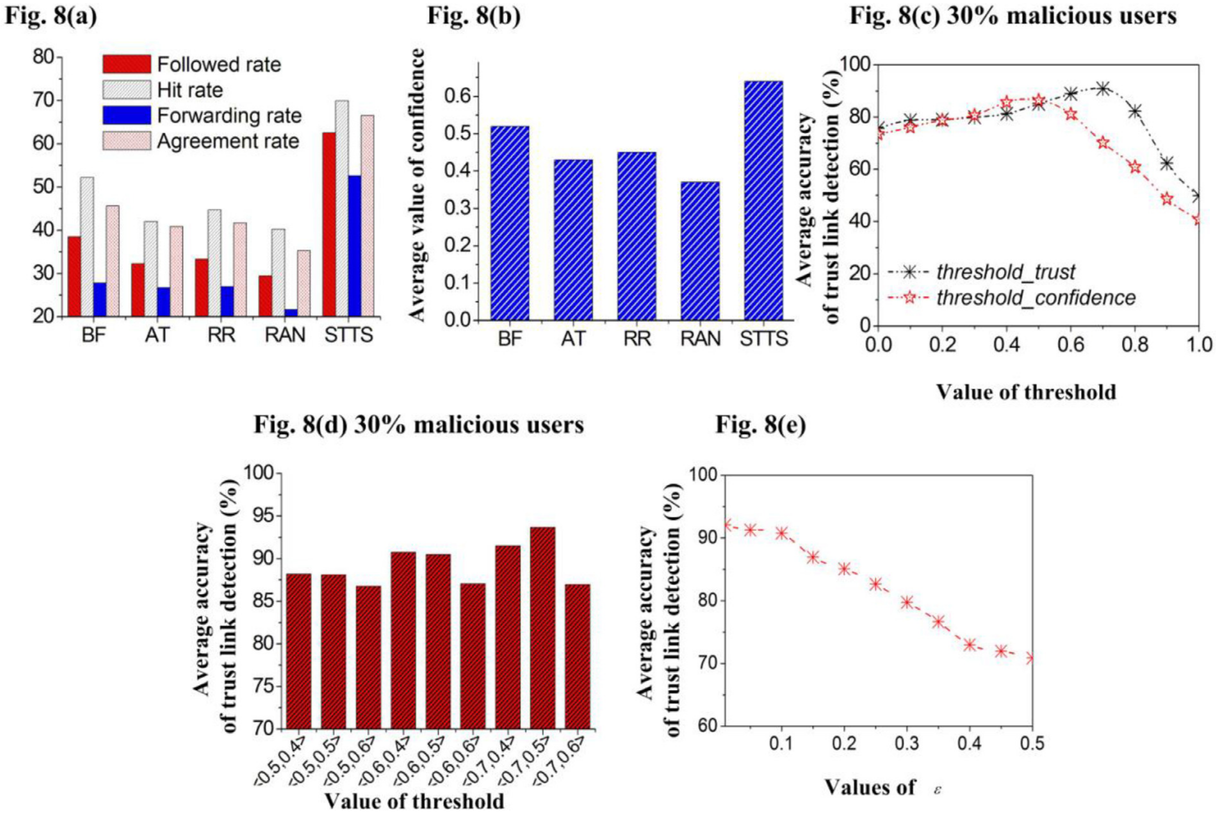


**Fig. 8.** Performance of the Trust Traversal Scheme.

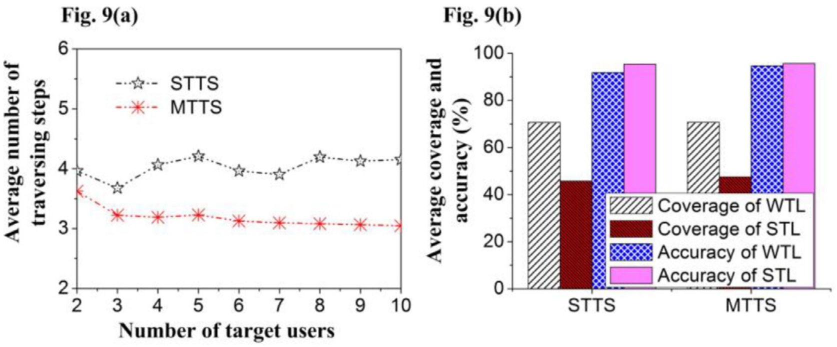


**Fig. 9.** Performance comparison of STTS and MTTS.

of traversal steps. In Fig. 9(b), we can see that the coverage and trust link detection accuracies of STTS and MTTS are similar.

To conclude, there are the following aspects of our proposed Trust Traversal Scheme under our experimental scenario,

(1) Cost. According to our observation, the average number of trust traversals from an origin user to a particular target user is nearly 238, which is only slightly more than the minimum number of traversals setting. That means that our proposed scheme has a low traversal cost. Meanwhile, the average numbers of traversal steps under different settings are smaller than 4.5, as shown in Fig. 9(a). That means that our proposed scheme avoids lengthy evaluation in traversal, which also results in a relatively low cost. Meanwhile, due to the setting of the maximum number of traversals as the termination condition, our proposed scheme also avoids endless loops.

(2) Efficiency. As shown in Figs. 8(a) and 9(b), our proposed scheme has the best performance in all groups and yields a relatively high coverage factor. That means that our proposed scheme can effectively find the trustworthy users in a large-scale network. In addition, we can see that the proposed scheme can recognize malicious users in traversal in Fig. 8(b)–(d). The results verify that our proposed scheme can work efficiently in both aspects, i.e., ensuring trustworthy user detection and achieving large-scale traversal.

(3) Accuracy. In our experiments, the accuracies of our proposed methods, including the reputation aggregation, trust calculation, and trust traversal scheme, have been verified through comparison with other related works. Our proposed methods obtain better performances than the other works. Additionally, our proposed Trust Traversal Scheme yields relatively high accuracy in establishing trust links, as shown in Fig. 9(b).

(4) Deliverable trust calculation. The trust calculation method is deliverable in indirect links, and the effect of our proposed deliverable trust calculation is shown in Fig. 6. The re-

sults reveal that there are additional rules, as mentioned in Section 4.3, and they play an important role in trust traversal.

## 7. Conclusions

A link is a significant entity for connecting strange users in a social network to facilitate their recognition, communication, and further interaction. It is essential for ensuring the reliability of links that potential threats for users be avoided. However, traditional trust-based methods only evaluate the trustworthiness among users and pay less attention to the strength of the detected links. In this study, we propose a trust link detection method, Trust Traversal Scheme, for users in SNS, which comprises the following aspects: (1) methods of reputation, direct trust and indirect trust aggregation; (2) categories of trust links, i.e., strong trust links, weak trust links, and untrusted links, among users; and (3) a trust-based traversal scheme for detecting strong or weak trust links with a single target user and multiple target users. In our proposed scheme, trust is used to calculate the probability of selecting a traversal target at each step, and the strength evaluation is taken into account to enhance the reliability of the trust link.

The results have shown that, under our dataset from a real social network scenario, the performances of our proposed scheme are better than those of comparison groups. The trust calculation methods, i.e., reputation and trust aggregation, can establish accurate subjective and objective trust by introducing factors based on social network features and prevent malicious threats. The trust link category has been analyzed, showing that the trust links, including strong and weak trust links, reflect a trustworthy and closeness relationship among users and contribute to frequent communication and interaction. The proposed TTS relies on users' trust degree to determine the traversal process and thus detects a set of trust links that maintain relative strength degrees to ensure reliable and feasible interactions among users. To conclude, our experimental results have shown that, in terms of performances of the trust aggregation, trust links, and TTS, our proposed TTS can detect trust links for users while ensuring the trustworthiness and strength degree of connecting paths, which would enhance the communication security in social networks and improve the interaction experience for users. For the practical significance, our proposed trust link detection method can be used in further community detection because the trust links detected through TTS manifest the trustworthy relationships among users, and the mutually trusted users can also be clustered as a community. Our future works will focus on further analysis of complex link composition impacts for our proposed approach in this study, and furthermore, we plan to introduce a cluster detection method to detect trust-intensive communities based on more real-world social network environments.

## Acknowledgments

This work is funded by National Natural Science Foundation of China (61572326, 61103069), Innovation Program of Shanghai Municipal Education Commission (C160049, 14zz125), and the Shanghai Committee of Science and Technology (16JC1403000).

## Reference

[1] Leonard Reinecke, Sabine Trepte, Authenticity and well-being on social network sites: a two-wave longitudinal study on the effects of online authenticity and the positivity bias in SNS communication, Compu. Hum. Behav. 30 (2014) 95–102.

[2] Luarn Pin, Jen-Chieh Yang, Yu-Ping Chiu, The network effect on information dissemination on social network sites, Comput. Hum. Behav. 37 (2014) 1–8.

[3] P. Groenewegen, C. Moser, Online communities: challenges and opportunities for social network research, Res. Sociol.Organizations 40 (2014) 463–477.

[4] Audun Jøsanga, Roslan Ismailb, Colin Boydb, A survey of trust and reputation systems for online service provision, Decis. Support Syst. 43 (2) (2007) 618–644.

[5] Jiang Wenjun, Guojun Wang, Wu. Jie, Generating trusted graphs for trust evaluation in online social networks, Future Gener. Comput. Syst. 31 (2014) 48–58.

[6] Sherchan Wanita, Surya Nepal, Cecile Paris, A survey of trust in social networks, ACM Comput. Surv. (CSUR) 45 (4) (2013) 47.

[7] Huang Jin, et al., Social trust prediction using heterogeneous networks, ACM Trans. Knowl. Discovery Data (TKDD) 7 (4) (2013) 17.

[8] Fire Michael, et al., Computationally efficient link prediction in a variety of social networks, *ACM Trans. Intell. Syst. Technol.* (TIST) 5 (1) (2013) 10.8.

[9] H. Liu, et al., TruCom: exploiting domain-specific trust networks for multicategory item recommendation, IEEE Syst. J. (2015) 1–10.

[10] E.W.K See-To, KKW. Ho, Value co-creation and purchase intention in social network sites: the role of electronic Word-of-Mouth and trust-a theoretical analysis, Comput. Hum. Behav. 31 (2014) 182–189.

[11] Wang Gang, Gui Xiaolin, Selecting and trust computing for transaction nodes in online social networks, Chin. J. Comput. 36 (2) (2013) 368–383.

[12] Jian Wu, Chiclana Francisco, A social network analysis trust consensus based approach to group decision-making problems with interval-valued fuzzy reciprocal preference relations, Knowl.-Based Syst. 59 (2014) 97–107.

[13] Charles Perez, Babiga Birregah, Marc Lemercier, A smartphone-based online social network trust evaluation system, Soc. Netw. Anal. Min. 3 (4) (2013) 1293–1310.

[14] F. Javier Ortega, JoséA. Troyano, FermínL. Cruz, CarlosG. Vallejo, Fernando Enríquez. Propagation of trust and distrust for the detection of trolls in a social network, Comput. Netw. 56 (12) (2012) 2884–2895.

[15] Qureshi Basit, Min Geyong, Kouvatsos Demetres, Trusted information exchange in peer-to-peer mobile social networks, Concurrency Comput.-Pract. Experience 24 (17) (2012) 2055–2068.

[16] Huanyu Zhao, Li. Xiaolin, VectorTrust: trust vector aggregation scheme for trust management in peer-to-peer networks, in: Computer Communications and Networks, 2009. ICCCN 2009. *Proceedings of 18th Internatonal Conference on*, IEEE, 2009.

[17] J. Golbeck, Trust and nuanced profile similarity in online social networks, ACM Trans. Web 3 (4) (2009) [Article 12].

[18] Ma Hao, King Irwin, R. Lyu Michael, Learning to recommend with explicit and implicit social relations, ACM Trans. Intell. Syst. Technol. 2 (3) (2011).

[19] Shuiguang Deng, Huang Longtao, Xu. Guandong, Social network-based service recommendation with trust enhancement, Expert Syst. Appl. 41 (18) (2014) 8075–8084.

[20] Tang, Mingdong, et al. "Combining global and local trust for service recommendation." *Web Services (ICWS), 2014 IEEE International Conference on*. IEEE, 2014

[21] P. Victor, C. Cornelis, M.D. Cock, A. Teredesai, Key figure impact in trust-enhanced recommender systems, AI Commun. 21 (2-3) (2008) 127–143.

[22] Xiwang Yang, et al., A survey of collaborative filtering based social recommender systems, Comput. Commun. 41 (2014) 1–10.

[23] Kim Young Ae, Phalak Rasik, A trust prediction framework in rating-based experience sharing social networks without a web of trust, Inf. Sci. 191 (5) (2012) 128–145.

[24] Sokratis Vavilis, Milan Petković, Nicola Zannone, A reference model for reputation systems, Decis. Support Syst. 61 (2014) 147–154.

[25] Samah Al-Oufi, Heung-Nam Kim, AbdulmotalebEl Saddik, A group trust metric for identifying people of trust in online social networks, Expert Syst. Appl. 39 (18) (2012) 13173–13181.

[26] Ferry Hendrikx, Kris Bubendorfer, Ryan Chard, Reputation systems: a survey and taxonomy, J.Parallel Distrib. Comput. (2014).

[27] Krishnaprasad Thirunarayan, et al., Comparative trust management with applications: bayesian approaches emphasis, Future Gener. Comput. Syst. 31 (2014) 182–199.

[28] Siyuan Liu, et al., A fuzzy logic based reputation model against unfair ratings, in: Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems, International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[29] D. Wang, D. Pedreschi, C. Song, F. Giannotti, A. Barabasi, Human mobility, social ties, and link prediction, KDD, 2011.

[30] Mohsen Jamali, Ester Martin, TrustWalker: a random walk model for combining trust-based and item-based recommendation, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2009.

[31] Guanfeng Liu, et al., Finding the optimal social trust path for the selection of trustworthy service providers in complex social networks, Serv. Comput., IEEE Trans. 6 (2) (2013) 152–167.

[32] YuLin He, et al., OWA operator based link prediction ensemble for social network, Expert Syst. Appl. 42 (1) (2015) 21–50.

[33] Michael Fire, et al., Computationally efficient link prediction in a variety of social networks, *ACM Trans. Intell. Syst. Technol. (TIST)* 5 (1) (2013) 10.

[34] Naveen Gupta, Anurag Singh, A novel strategy for link prediction in social networks, in: Proceedings of the 2014 CoNEXT on Student Workshop, ACM, 2014.

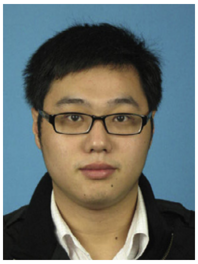
[35] Jorge Valverde-Rebaza, Lopes Alneu de Andrade, Exploiting behaviors of communities of twitter users for link prediction, Soc. Netw. Anal. Min. 3 (4) (2013) 1063–1074.

[36] Linyuan Lü, Zhou Tao, Link prediction in complex networks: a survey, Physica A 390 (6) (2011) 1150–1170.
[37] Qiang Ye, et al., In-depth analysis of the seller reputation and price premium relationship: a comparison between eBay US and Taobao China, J. Electron. Commerce Res. 14 (1) (2013) 1–10.
[38] SD Kamvar, MT Schlosser, EigenRep: reputation management in P2P networks, in: Proceedings of the 12th International World Wide Web Coneference, 2003, pp. 123–134.
[39] O. Rolim, Carlos, et al., Six degrees of separation to improve routing in opportunistic networks, Int. J.UbiComp 4 (3) (2013) 11–22.
[40] AliAydin Selcuk, Ersin Uzun, MarkResat Pariente, A reputation based trust management system for P2P networks, CCGrid 2004, IEEE International Symposium on Cluster Computing and the Grid,2004, IEEE, 2004.
[41] Serif Bahtiyar, Murat Cihan, MehmetUfuk Caglayan, A model of security information flow on entities for trust computation, 2010 IEEE 10th International Conference on Computer and Information Technology (CIT), IEEE, 2010.
[42] P. De Meo, E. Ferrara, D. Rosaci, G.M.L. Sarnè, Trust and Compactness in Social Network Groups, IEEE Trans. Cybern. (TOC). 45 (2) (2015) IEEE Press.
[43] M Li, Y Xiang, B Zhang, et al., A trust evaluation scheme for complex links in a social network: a link strength perspective, Appl. Intell. (2016) 1–19.
[44] Peng Wang, et al., Link prediction in social networks: the state-of-the-art, Sci. China Inf. Sci. 58 (1) (2015) 1–38.
[45] G Akcora C, B Carminati, E. Ferrari, User similarities on social networks, Soc. Netw. Anal. Min. 3 (2013) 475–495.
[46] A Anderson, D Huttenlocher, J Kleinberg, et al., Effects of user similarity in social media, in: Proceedings of the 5th ACM International Conference on Web Search and Data Mining (WSDM'12), Seattle, USA, 2012, pp. 703–712.
[47] A Adamic L, E. Adar, Friend and neighbors on the web, Soc. Netw. 25 (2003) 211–230.
[48] MEJ. Newman, Clustering and preferential attachment in growing networks, Phys. Rev. Lett. E 64 (2001) 025102.
[49] G Jeh, J. Widom, SimRank: a measure of structural-context similarity, in: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD'02), Edmonton, Canada, 2002, pp. 538–543.
[50] L Lu¨, H Jin C, T. Zhou, Similarity index based on local paths for link prediction of complex networks, Phys. Rev. E 80 (2009) 046122.

**Bo Zhang** received the Ph.D. degree in College of Electronics and Information Engineering, Tongji University in 2009. And he finished his postdoc research work in Tongji University in 2011. He is now associate professor of College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai, China. His current research interests include trust computation and social network analysis. He is now a director of trust computation project, which is funded by National Nature Science Foundation of China.

**Huan Zhang** is now a master candidate in College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai, China. Her research interests include relationship mining in social network and trust evaluation based on network environment.

**Meizi Li** is now the lecturer of College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai, China. Her current research interests include the social network analysis, trust and reputation computation.



**Qin Zhao** received the Ph.D. degree in College of Electronics and Information Engineering, Tongji University in 2016. He is now the lecturer of College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai, China. His current research interests include the social network analysis, machine learning and data mining.

**Jifeng Huang** is now a professor of College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai, China. His current research interests include data mining and pattern recognition.