Arabian Journal for Science and Engineering (2022) 47:9615–9628
https://doi.org/10.1007/s13369-021-06332-w

RESEARCH ARTICLE-COMPUTER ENGINEERING AND COMPUTER SCIENCE

An Interaction-Based and Graph-Based Hybrid Approach to Evaluate Trust in Online Social Networks (OSNs)

Gordhan Jethava[1] · **Udai Pratap Rao**[2]

Received: 19 June 2021 / Accepted: 21 October 2021 / Published online: 13 November 2021

Abstract
With the digital revolution and the Web 2.0 era, web-based social networks such as Facebook and others have become popular mediums for users to do various activities. While social networks are becoming increasingly popular, concerns about trust and trust-related issues are also growing among users. There are many applications where trust plays a vital role in users' decision-making, requiring trust evaluation. There are several trust evaluation approaches for online social networks in the literature. However, the existing approaches focus only on certain aspects and believe that direct trust between participants is known. Thus, there is a need for a comprehensive trust evaluation approach that infers indirect trust and strives to measure direct trust. This paper proposes an interaction-based and graph-based hybrid approach that attempts to measure direct trust and infer indirect trust among users. Our direct trust measure method utilizes the most important features and similarities between users to measure direct trust. The proposed indirect trust inference method uses the graph theory concept to infer indirect trust. We implement the friend-request identification and the Sybil attack detection applications using the proposed direct trust measure method. Both the applications are evaluated on synthetic and real-world datasets. The empirical results show that the friend-request identification application achieves a high accuracy of 96.17%, and the Sybil attack detection application obtains a high detection rate of 93.20%. The false rates of both applications are very low. The proposed indirect trust inference method is efficient, and it outperforms the existing approach.

Keywords Online social networks · Trust evaluation · Direct trust measure · Indirect trust inference · A hybrid approach

1 Introduction

Millions of people join online social networks and perform a wide range of activities such as making new friends, online purchasing, sharing opinions, posting photographs, disseminating information, commenting, and more. More and more people depend on online social networks to get relevant information, find news, decide on their online purchases, and more. While millions of users communicate with other users on online social networks, they have limited awareness of other users as they do not have face-to-face interaction, and most of them are anonymous. Trust plays a vital role in many aspects, e.g. identifying individual users, discovering the most appropriate products or services, building a trustworthy recommendation system, and more, needing trust evaluation. Moreover, users are increasingly concerned about the privacy of their personal information. It is essential to develop a trust system that allows members to share their ideas, views, and experiences without being concerned about their privacy and fear of being evaluated. Trust relationship among users is the foundation for robust social networking. Generally speaking, trust is a measure of faith that an individual or entity will act in an anticipated way.

There are many types of trust relationships in online social networks. Figure 1 depicts four types of trust relationships: (A) A trust relationship between explicitly connected members, also known as explicit or direct trust relationship. (B) A trust relationship between members who do not have a direct link between them; and trust value is inferred with the help of known members of the network, also known as implicit or

✉ Udai Pratap Rao
upr@coed.svnit.ac.in

Gordhan Jethava
g.jethava@gmail.com

[1] Department of Computer Science and Engineering, Parul Institute of Technology, FET, Parul University, Vadodara, Gujarat, India

[2] Department of Computer Science and Engineering, Sardar Vallabhbhai National Institute of Technology, Surat, Gujarat, India

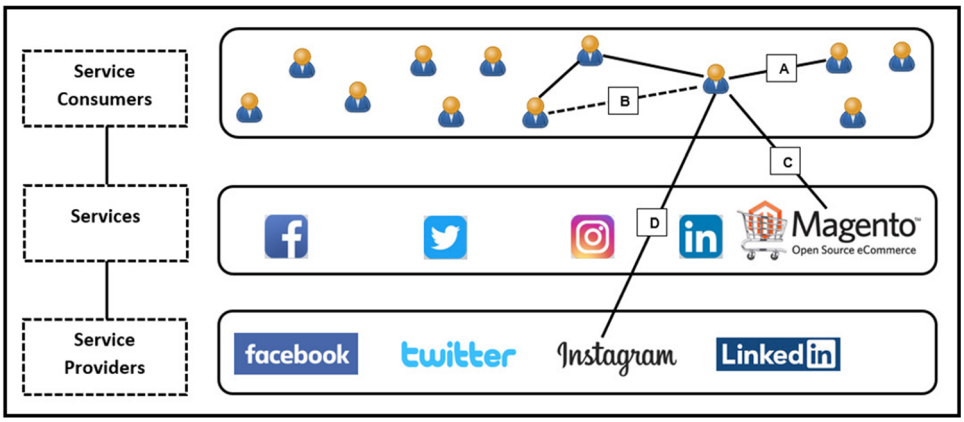


Fig. 1 Trust relationships in online social networks (OSNs)

indirect trust relationship. (C) A trust relationship between a member and a service offered (e.g. third-party e-commerce services on online social networks). (D) A trust relationship between a member and a social network service provider.

There are many applications where trust evaluation can play an important role. For instance, users receive many friend requests on online social networks. These friend requests may be from genuine users, or they may be from malicious users. The direct trust measure can help users distinguish friend requests and prevent them from being victims of security threats. Trust evaluation can also be useful in detecting Sybil or fake users in online social networks. Friend recommendation, however, is the popular service offered by most of the social networks, and measuring trust between users can enhance the quality of friend recommendation [1] and offer a better service to users. Many e-commerce activities (e.g. as shown in Fig. 1, Magento, the third-party e-commerce service on OSNs) are happening on online social networks where trust plays a vital role in consumer buying decisions. Trust assessment can defend users' sensitive information and preserve users' privacy by protecting their personal information. Trust evaluation can verify the authenticity of any information by evaluating that information source's trust.

In summary, trust assessment is useful in many applications such as friend-request identification, detecting Sybil/fake users, making friend recommendation trustworthy, helping users in their e-commerce transactions, verifying the trustworthiness of information by checking the authenticity of the source of information, protecting users' sensitive data, and preserving the privacy of users by protecting their personal information.

We aim to propose an approach that can evaluate the trustworthiness of the target participant (trustee) for the user(trustor) in online social networks (OSNs). The target participant may be a direct (or 1-hop) neighbour of the user (trustor) or may have an indirect link to the user (trustor). The target participant may be a user, a service, or a service provider.

1.1 Our Contributions

The significant contributions of our work are as follows:

- We propose an interaction-based and graph-based hybrid trust evaluation approach for OSNs using dynamic features and similarities. The proposed approach consists of two phases: the direct trust measure and indirect trust inference.
- The direct trust measure phase computes the trust scores between each directly connected node in the network. It uses the interaction-based dynamic features (relationship trust, location trust) and similarities (mutual-friend similarity, likes similarity, group-joined similarity) to measure direct trust between each directly linked node.
- The indirect trust inference phase utilizes direct trust values, pre-processes the social network graph, and generates the trusted graph. It applies the graph theory concept(breadth-first search technique) to the trusted graph and finds trusted paths between the source node and the non-neighbour target node. Each trusted path between the source node and the target node and its trust values are considered for the indirect trust inference.
- We use the proposed methods to implement the friend-request identification and the Sybil attack detection applications. We evaluate both the applications on synthetic and real-world datasets and measure various evaluation parameters.

The remainder of this paper is arranged as follows. Section 2 discusses the theoretical background of trust models, explores the various existing trust evaluation models, and provides our findings from the literature review. Section 3 explains our problem scenario, defines the problem statement, presents the proposed system architecture, and

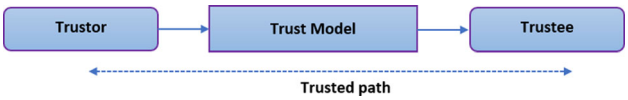


Fig. 2 Elements of social network trust system

discusses each phase of the proposed architecture in detail. We identify the real-world datasets, discuss the performance metrics, and provide the experimental results in Sect. 4. We conclude and provide the future scope of the work in Sect. 5.

2 Background and Related Work

2.1 Background

2.1.1 Key Concepts of Social Networks Trust System

Figure 2 illustrates some of the key terms widely used in various trust systems of online social networks.

Trustor: Trustor is a social network participant who tries to know or evaluate another participant's trustworthiness or trust degree in the network.

Trustee: A participant in an online social network whose trustworthiness or trust degree is being assessed is called a trustee.

Trust model: A trust model is an intermediary tool that helps a trustor to evaluate the trust of a directly connected or indirectly connected trustor in online social networks.

Trusted path: A trusted path from the trustor to the trustee consists of various elements such as a trustor, several intermediary recommenders, a trustor, and trust relationships among them.

2.1.2 Trust Definitions

There are many definitions of trust in the literature on trust assessment. Some of the trust definitions are as follows:

- Golback et al. [2] defined trust as "trust in a person is a commitment to an action based on a belief that the future actions of that person will lead to a good outcome".
- Jøsang et al. [3] define trust as "the subjective probability by which one user expects that another user performs a given action".
- According to Sherchan et al. [4], "trust is a measure of faith that an individual or entity will act in an anticipated way".
- JH Cho et al. [5] summarized the concept of trust as "trust is the ability of the trustee to take risks on the basis of a subjective expectation that the trustee will show a trustworthy conduct to maximize the interest of the trustee in the volatility of a given situation, based on a cognitive evaluation of previous experience with the trustee".

2.1.3 Types of Trust

There are different types of trust [6]. Some of them are mentioned below.

Explicit/direct trust: An explicit/direct trust is the trust relationship between directly connected entities. The direct interaction between entities, experiences, or similarity between the entities may help to measure an explicit trust between the entities.

Implicit/indirect trust: Implicit/indirect trust is the trust relationship between indirectly connected entities. The trust score of the target(trustee), which has not a direct link with the user(trustor) can infer using the pairwise trust score of the entities in the network. The intermediate trusted entities can help to infer the implicit trust.

Global trust: The global trust is measured by taking into account all participants' experiences, views, and all their trust relationships.

In our work, we focus on measuring direct trust and inferring indirect trust. We measure direct trust between each 1-hop neighbour node and use it to infer indirect trust between any two non-neighbour nodes in the network.

2.2 Related Work

We categorize the existing trust evaluation approaches into graph-based approaches, interaction-based approaches, behaviour-based approaches, and statistical approaches. The graph-based trust models [7] [8] [9] [10] [11] [12] [13] explore a network topology, use the structural properties of a network, model a social network as the social network graph, and apply graph theory concepts to assess trust. The interaction-based schemes [14] [15] [16] [17] examine user interaction features like commenting, posting, liking, sharing, forwarding, mutual friends, and more, and measure connection strength between users to assess trust among users. The behaviour-based approaches [18] [19] [20] analyse various user behaviour-based features and evaluate trust among users. Some of the behavioural features are frequency of use, social affiliation, user activities with other users, social investigation, social boldness, self-orientation, information discloser on social networks, and more. The statistical approach [21] aims to provide a sound mathematical model useful in trust management.

The current graph-based approaches modelled social networks as the social network graph where nodes are participants and edges are trust relationship among participants. The existing approaches used various graph theory concepts (e.g. random walks, breadth-first traversal, depth-first traversal) to evaluate trust among users. The existing approaches Tidal-

Trust [7], MoleTrust [8], and SWTrust [11] first simplified the social network graph by discarding some nodes or edges. The approaches then performed some operations on the graph like graph reduction, graph adjustment, graph weighted average, and extracted simplified the trusted graph to assess trust. Some of the graph-based schemes like FlowTrust [10], GFTrust [12] directly dealt with the original social network graph and considered each trust relationships for trust inference.

Golbeck [7] suggested the TidalTrust. TidalTrust aimed to quantify trust value from the trustor to the indirectly connected trustee in a social network graph where there exist many paths from the trustor to the trustee. The proposed technique found trusted paths from the trustor to trustee based on the breadth-first search way. Eventually, it considered only the shortest and strongest path to perform the trust values aggregation from the trustor to the trustee. For example, any source user v_s wants to evaluate the trust value of the target user v_t; the trust value calculation formula in TidalTrust is as follows:

$$T_{v_s->v_t} = \frac{\sum_{x \in N_{v_s}, T_{v_s->x} \geq \max} T_{v_s->x} * T_{x->v_t}}{\sum_{x \in N_{v_s}, T_{v_s->x} \geq \max} T_{v_s->x}} \quad (1)$$

where N_{v_s} is neighbours of source user v_s and max is the threshold used to restrict the number of paths in trust aggregation. The shortcoming of TidalTrust is that, in the trust value aggregation, it is considered only the shortest and strongest route, while trust from multiple routes could be better than that of a single route. Trust from multiple routes could avoid being biased and selective.

Avesani et al. [8] proposed the MoleTrust, which includes two steps. In the first step, MoleTrust modifies the graph by removing cycles and convert it into a directed acyclic graph (DAG). Based on the shortest distance from the source user v_s, MoleTrust sorts users and distinguishes all users that can be reached from the source user. In step two, MoleTrust computes the trust value of all users who are at a distance one, two, three, and so from the source user. The trust value of any user at a distance d only depends on trust values of the users who are at a distance $d-1$. Using MoleTrust, we can calculate the trust value of any destination trustee v_d by aggregating all incoming trust values from $v'_d s$ neighbours to v_d using the weighted average as follows:

$$T_{v_d} = \frac{\sum_{x \in \text{pred}(v_d)} (\text{trust}(x) * \text{edge}(x, v_d))}{\sum_{x \in \text{pred}(v_d)} (\text{trust}(x))} \quad (2)$$

where T_{v_d} is trust value of trustee v_d, $pred(v_d)$ is the predecessors or incoming trusted neighbours of trustee v_d, and $edge(x, v_d)$ is edge weight value of edge between nodes x and v_d.

Wang and Wu [9] proposed a trust management framework named MeTrust. MeTrust used multi-trusted paths with multi-dimensional evidence to evaluate trust in any random multifaceted trusted graph. The authors conducted trust calculation at three layers: the node layer, the path layer, and the graph layer. The node layer considered multi-dimensional trust and allowed users to assign a weight to each dimension on their own for trust calculation. At the path layer, authors used the Frank t-norm to manage the trust decay rate for trust combination. The graph layer simplified multifaceted trusted graph using GraphReduce, GraphAdjust, and WeightedAverage algorithms.

Jiang et al. [11] proposed SWTrust. Most graph-based trust evaluation approaches believed that a small trusted graph was already available. The main objective of SWTrust was to generate a small trusted graph from a large OSN so it could be integrated into existing trust evaluation approaches to make them more practical and efficient. The framework includes three steps: (I) pre-processing a large social network (PSN), (II) building trust network (BTN), and (III) generating trusted graph (GTG). SWTrust pre-processed a social network and found a trusted acquaintance chain using users' domain knowledge. In the BTN step, SWTrust explored all possible paths between a given trustor and trustee using the breadth-first search technique. The GTG step measured each path's trust values, compared them with the pre-defined threshold, and discarded the paths with less trust value than the pre-defined threshold.

Jiang et al. [12] proposed a modified flow-based trust assessment approach named GFTrust. GFTrust addressed two open challenges: path dependency and trust decay. The authors assumed that the trusted graph is available in which a direct trust relationship between each directly connected node is available. The scheme used a generalized network flow concept to tackle path dependency and model trust decay with the leakage associated with each node. The authors used a modified network flow model with leakage and evaluated trust of non-neighbour destination nodes, resolving path dependency and trust decay issues together.

Hamdi et al. [13] proposed the trust inference model called TISoN. The authors presented a new trust path searching algorithm based on transitivity properties, in which they pre-processed a large social network, generated the trusted network, and handled the trusted path discovery problem. The trust inference measure (TIM) method for indirect trust inference was introduced, and it was based on the Trust Path Search algorithm and the trust aggregation function. The authors hypothesized that the trust scores between each directly connected node were known.

The authors in [22] assumed that the direct trust relationships between each directly connected participant were known and that by utilizing them and employing the uncertainty theory, the indirect or recommended trust had been inferred.

In [23], the authors designed the trust evaluation framework named *Guardian* based on graph convolutional neural networks for online social networks (OSNs). The framework included social network structures and explicit trust relationships and inferred the indirect trust between users. The framework was tested with the real-world datasets Advogato and PGP, and it achieved an F1-score of 74.3% with the Advogato dataset and 87.1% with the PGP dataset.

The authors in [18] developed the method named CoRank. The method analysed behaviours of users and tweets and assessed the trustworthiness of users and tweets on the Twitter social network. The authors exploited the complex features and relations of users and tweets and measured their trust scores. A series of experiments were carried out on real data extracted from Twitter and demonstrated the method's efficacy.

In [24], the authors proposed an integrated time-aware similarity-based trust prediction approach called iSim, leveraging user similarity. Several methods had been employed in order to improve the time complexity of iSim and thus its efficiency.

2.2.1 Findings from the Literature Survey

Our findings from the literature survey are as follows:

- The approaches [7] [8] [12] [13] [22] assumed that **the direct trust relationship between each directly connected participants is known**, and they utilized the available direct trust relationships to infer indirect trust. **The approaches did not strive to measure direct trust.** There is a need for an integrated trust evaluation approach that **infers indirect trust and further seeks to measure direct trust.**
- The TidalTrust algorithm [7] considered only the shortest and strongest path to infer the trust value of indirectly connected trustee. Hence, the algorithm's efficiency was affected because **trust from multiple paths could be better than that of a single path.** Moreover, trust from multiple paths could also avoid being biased and selective. Thus, **trust evaluation approaches should consider trust values of as many paths as possible between a trustor and a trustee to maximize inferred trust quality.**
- The current approaches concentrated only on specific issues. For example, SWTrust [11] aimed to generate a small trusted graph from a large OSN, and GFTrust [12] addressed the path dependency and trust decay problem.

There is still a **need for a systematic and comprehensive trust assessment approach** to measure direct trust accurately and infer indirect trust effectively.

3 The Proposed Trust Evaluation Approach

3.1 Overview

When we want to interact with any unknown entity on a social networking site like Facebook, we need to assess the unknown entity's trustworthiness in order to ensure the security and privacy of users and their data. We propose an interaction-based and graph-based hybrid approach that aims to measure direct trust and infer indirect trust among participants in OSNs. Our approach encompasses two integrated modules, namely, a direct trust measure and an indirect trust inference. The direct trust measure module utilizes the interaction-based dynamic features (relationship trust, location trust) and the similarities (mutual-friend similarity, likes similarity, and groups joined similarity) and measures the direct trust between each directly connected node in the network. In the indirect trust inference module, we pre-process the network graph, generate the trust trusted network graph, apply the graph theory concept, and infer the indirect trust between any two non-neighbour nodes in the network. Thus, our approach effectively evaluates the trust score of target participants (trustee) that are directly linked with a source participant (trustor), and it also infers the trust score of the participants that are not directly reachable.

3.2 Problem Scenario

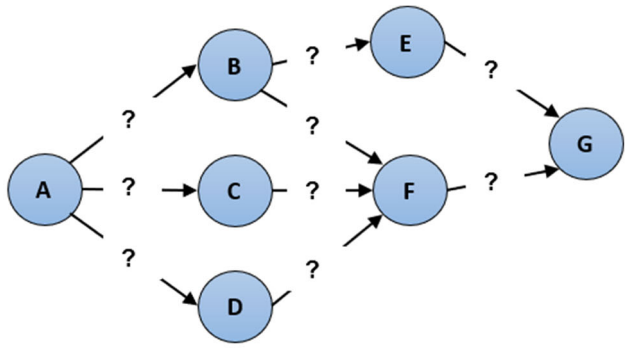
Figure 3 illustrates the problem scenario of our work. On online social networks, users may directly connect to some 1-hop neighbour users or have indirect links with non-neighbour users.

The objectives of our work are to address two research questions: (1) How to measure trust values between each directly connected user in the network (as shown in Figure 3a)? (2) How to infer trust value between any two non-neighbour users in the network (e.g. as shown in Figure 3b, trust inference between nodes A and G, or between nodes A and F)?

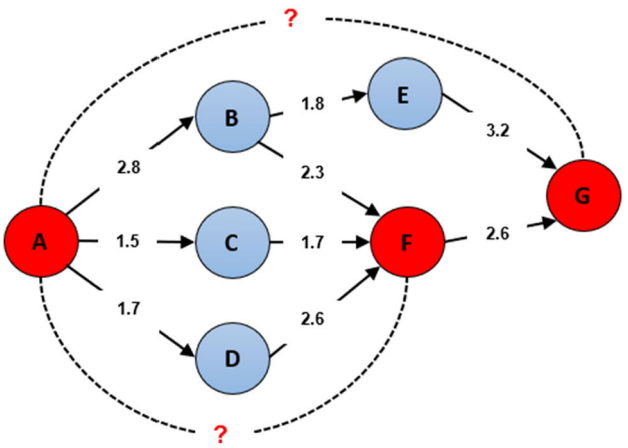
3.3 Problem Definition

Table 1 lists the mathematical symbols used in our work.

We model social networks as the directed social network graph $G = (V, E)$, where vertex V is a set of nodes representing individual participants, and edge E is a set of directed edges or connections representing participants' relationships. For all directed edges $e_{i->j} = (v_i, v_j) \in E$, $v_i, v_j \in V$, we



(a) Direct trust calculation



(b) Indirect trust inference

Fig. 3 Illustration of the problem scenario

calculate the trust scores and assign it as a label of edges. The label of edges reflects direct trust between nodes, and edge direction signifies that which node has quantified the trust value for which node. We define our direct trust assessment and indirect trust inference problem as follows:

Definition 1 (*Direct Trust Measure:*) Calculating the trust score between two directly connected nodes using some dynamic features and similarities is the direct trust measure in the social network graph.

Definition 2 (*Indirect Trust Inference:*) Using the direct trust values and the graph theory concept, inferring the trust score between any two non-neighbour nodes is the indirect trust inference in the social network graph.

3.4 Proposed System Architecture

Figure 4 demonstrates the system architecture of our proposed hybrid trust evaluation approach.

As shown in Fig. 4, our approach works in three phases: Data collection, measuring direct trust, and indirect trust inference. We extract users' data and their interactions data from online social networks during the data collection process. We model users and their interactions information as the directed social network graph, where nodes represent individual participants and edges are the relationships between two participants. We use the interaction-based dynamic features and similarities to measure the direct trust score between each connected node. We assign a pair of nodes'

Table 1 The Notations

Symbol	Description
$G = (V, E)$	A directed social network graph
rty	Relationship or Friendship type
$RT_{v_i->v_j}$	The relationship trust between nodes v_i and v_j
$LT_{v_i->v_j}$	The location trust between nodes v_i and v_j
$v_i.FL, v_j.FL$	Friend list of nodes v_i and v_j
$MFS_{v_i->v_j}$	The mutual-friend similarity between neighbour nodes v_i and v_j
$\|v_i.LI\|, \|v_j.LI\|$	No. of likes of nodes v_i and v_j
$LIS_{v_i->v_j}$	The likes similarity between neighbour nodes v_i and v_j
$v_i.GJ, v_j.GJ$	Groups joined by nodes v_i and v_j
$GJS_{v_i->v_j}$	The groups joined similarity between neighbour nodes v_i and v_j
$DT_{v_i->v_j}$	The direct trust value between neighbour nodes v_i and v_j
v_s, v_t	A source node, and a target node
τ	Pre-defined threshold to identify suspicious links
T_{route}	Trust score of the trusted path
$IT_{v_s->v_t}$	To a non-neighbour destination node v_d The inferred or indirect trust value between a source node v_s and a non-neighbour destination node v_d
τ_1	The pre-defined threshold for the friend-request identification application
τ_2	The pre-defined threshold for the Sybil attack detection procedure

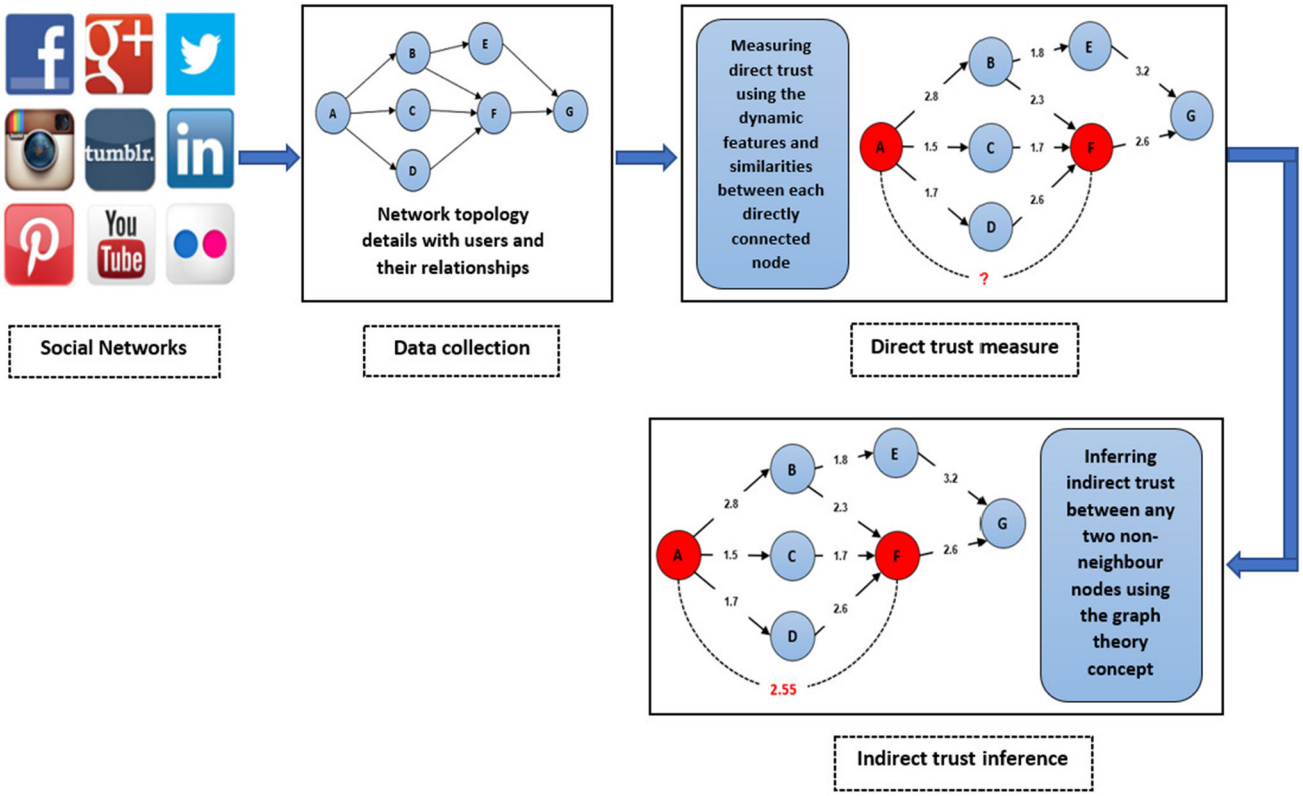


Fig. 4 Proposed system architecture

trust scores as the edge weight in the social network graph. The weight of edges is the direct trust relationship between two directly connected users. The directed weighted social network graph, the source node, and the target node are the input to the indirect trust inference module. In the indirect trust inference, we pre-process the social network graph and generate the trusted graph. We apply the graph theory concept (breadth-first search technique) to the trusted graph and find trusted paths between the source and target nodes. We measure each path's trust value and consider each trusted path's trust values to infer the indirect trust value between two non-neighbour nodes.

3.5 Measuring Direct Trust

In the direct trust measure phase, we calculate the direct trust value for each directly connected node using the interaction-based features and the similarities. To quantify direct trust values for each pair of directly linked nodes, we consider the dynamic features such as the relationship trust and the geographical location trust. We also incorporate the similarities such as the mutual-friend similarity, the like similarity, and the groups joined similarity between two directly connected nodes to measure the direct trust.

The type of relationship between users has a significant influence on the evaluation of trust between them. Among Facebook users, there are mainly three kinds of friendships: family members, close friends, and acquaintances. We assume that users trust and interact more with known users on online social networks (OSNs). Users usually do not trust unknown users on OSNs. Hence, in our approach, we assign trust scores to users based on the type of relationship. We give more trust value to the family member relationship type than the acquaintance relationship type because users trust more to their family members than acquaintances. We assign zero trust values to the unknown relationship types as users do not trust unknown users on OSNs. Algorithm 1 describes the method of calculating the relationship trust between two directly linked nodes v_i and v_j.

Users typically trust more to those familiar to them, and the geographical location is similar to that of the users. We hypothesis that it is more likely that users will trust unknown users if there are some geographical location similarities between them. There are different geographical location relationships between users, such as neighbour users, users from the same hometown, users living in the same current city, from the same province, from the same country, and the same region. As shown in Algorithm 2, we weight each location

Algorithm 1 Calculating the relationship trust

Input: Pair of neighbour nodes (v_i, v_j).
Output: Relationship trust value between neighbour nodes v_i and v_j.
for each neighbour nodes v_i, v_j in the social network graph **do**
  $rty \leftarrow extractRelationship(v_i, v_j)$;
  if $rty == Family\ member$ **then**
    $RT_{v_i->v_j} = 1$;
  else if $rty == close\ friend$ **then**
    $RT_{v_i->v_j} = 0.90$;
  else if $rty == acquaintance$ **then**
    $RT_{v_i->v_j} = 0.70$;
  else
    $RT_{v_i->v_j} = 0$;
  end if
end for

relationship and measure the location trust value for each pair of directly connected nodes.

Algorithm 2 Calculating the location trust

Input: Pair of neighbour nodes (v_i, v_j).
Output: Location trust value between neighbour nodes v_i and v_j.
for each neighbour nodes V_i, V_j in the social network graph **do**
  Extract location information of nodes v_i and v_j;
  if v_j is a neighbour of v_i **then**
    $LT_{v_i->v_j} = 0.80$;
  else if v_i and v_j are from the same hometown **then**
    $LT_{v_i->v_j} = 0.50$;
  else if v_i and v_j belong to the same current city **then**
    $LT_{v_i->v_j} = 0.40$;
  else if v_i and v_j are from the same province **then**
    $LT_{v_i->v_j} = 0.30$;
  else if v_i and v_j are from the same country **then**
    $LT_{v_i->v_j} = 0.20$;
  else if v_i and v_j are from the same region **then**
    $LT_{v_i->v_j} = 0.10$;
  else
    $LT_{v_i->v_j} = 0$;
  end if
end for

Along with the relationship trust and the location trust, we also consider the mutual-friend similarity, the likes similarity, and the groups joined similarity as a component of direct trust. If two directly connected users have mutual friends in their friend list, it is common for users to trust each other because they have similarities in their friends' choice. The Likes similarity between users indicates their affinity to the same interest. The like-minded and same domain knowledge users join the same groups and may trust and assist each other in that domain. Thus, the mutual-friend similarity, the likes similarity, and the groups joined similarity are the crucial components to measure the direct trust value between two directly connected users. We quantify these components using the Sorensen similarity metric [25]. The Sorenson similarity metric finds common elements between sets and divides them by the sum of the elements in each set. The Sorenson metric is local structural similarity and can help measure various similarities between users.

For two directly connected users(nodes) v_i, v_j and their friend list $v_i.FL, v_j.FL$, respectively, we calculate the mutual-friend similarity score $MFS_{v_i->v_j}$ using the Sorensen similarity metric as follows:

$$\mathrm{MFS}_{v_i->v_j} = \frac{|v_i.\mathrm{FL} \cap v_j.\mathrm{FL}|}{|v_i.\mathrm{FL}| + |v_j.\mathrm{FL}|}. \tag{3}$$

Two directly linked users(nodes) v_i, v_j and their likes $v_i.LI, v_j.LI$, respectively, we calculate the likes similarity $\mathrm{LIS}_{v_i->v_j}$ using the Sorensen similarity metric as follows:

$$\mathrm{LIS}_{v_i->v_j} = \frac{|v_i.\mathrm{LI} \cap v_j.\mathrm{LI}|}{|v_i.\mathrm{LI}| + |v_j.\mathrm{LI}|}. \tag{4}$$

For two directly connected users(nodes) v_i, v_j and the groups joined by those users are $v_i.GJ$, $v_j.GJ$, respectively, we measure the group-joined similarity $GJS_{v_i->v_j}$ using the Sorensen similarity metric as follows:

$$\mathrm{GJS}_{v_i->v_j} = \frac{|v_i.\mathrm{GJ} \cap v_j.\mathrm{GJ}|}{|v_i.\mathrm{GJ}| + |v_j.\mathrm{GJ}|}. \tag{5}$$

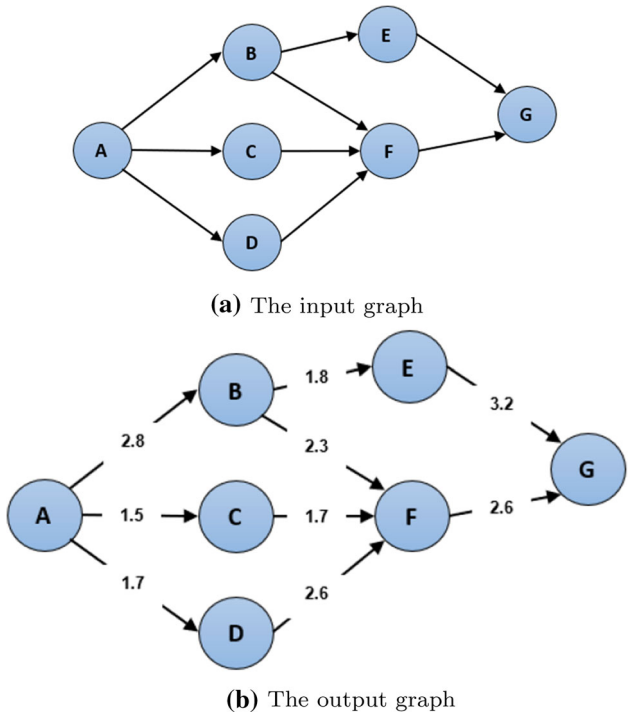
We sum up all the above components and measure the direct trust values $DT_{v_i->v_j}$ for each pair of directly connected nodes as follows:

$$\begin{aligned}\mathrm{DT}_{v_i->v_j} = {} & \mathrm{RT}_{v_i->v_j} + \mathrm{LT}_{v_i->v_j} + \mathrm{MFS}_{v_i->v_j} \\ & + \mathrm{LIS}_{v_i->v_j} + \mathrm{GJS}_{v_i->v_j}.\end{aligned} \tag{6}$$

Figure 5 shows the input–output graphs of the direct trust calculation phase, and Algorithm 3 summarizes the direct trust measure procedure.

3.6 Inferring Indirect Trust

The main problem of trust inference in social networks is that trust networks are sparse. Many users' explicit trust relationships are unknown in online social networks. So, if we manage n nodes, the expected number of edges (and so of trust relationships that we can use to our benefit) is much less than n^2. The usage of paths to infer trust values can be an effective substitute for compensating the lack of an explicit trust relationship. Our indirect trust inference module uses the breadth-first search technique to find trusted paths between a source node and a destination node and infers the indirect trust between them. In the indirect trust inference phase, we utilize the direct trust values of directly connected nodes. We input the source node v_s and the indirectly connected target node v_t between which we want to infer the indirect trust. We first pre-process the weighted social network graph and generate a trusted graph. In the



(a) The input graph

(b) The output graph

Fig. 5 The sample input–output graphs of the direct trust calculation phase

pre-processing, we identify all the edges having trust value less than the pre-defined threshold, consider them the suspicious links, and delete them from the social network graph. We apply the graph theory concept(breadth-first search technique) on the trusted graph to find all paths between the source node and the destination node. For trustor, the number of nodes in the trusted path is crucial in evaluating the trustworthiness of the trustee because as the number of nodes increases in the trusted path, trust decay from trustor to trustee [12]. So, we are considering the number of nodes in the trust evaluation of trusted paths. We calculate the trust value of each trusted path between the source and the destination by considering the number of nodes in that trusted path as follows:

$$T_{\text{route}} = \frac{\sum_{x=1}^{\text{Number of nodes in the route}} A_{x,x+1}}{\text{Number of nodes in the route}} \tag{7}$$

We find all possible routes from a source node v_s to a target node v_t using the breadth-first search (BFS) method. We aggregate each route's trust values and compute the indirect trust $IT_{v_s->v_t}$ between a source node v_s and a target node v_t as follows:

$$\text{IT}_{v_s->v_t} = \frac{\sum_{y=1}^{\text{Number of routes from } v_s \text{ to } v_t} T_{route(y)}}{\text{Number of routes from } v_s \text{ to } v_t}. \tag{8}$$

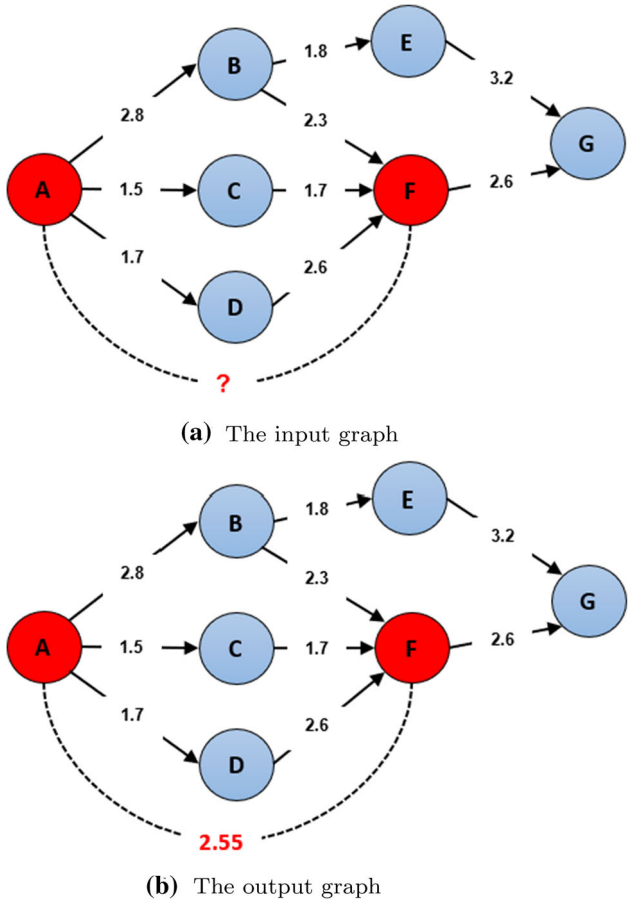
Algorithm 3 Measuring direct trust

Input: Directed social network graph $G = (V, E)$ with network topology information, users and their relationship information, and users' location information.
Output: Directed weighted social network graph $G = (V, E, W)$, where $W = e_{i->j} = (v_i, v_j) \in E$ is the direct trust score of each pair of nodes $v_i, v_j \in V$.
for each pair of neighbour nodes v_i, v_j in the social network graph $G = (V, E)$ **do**
  Calculate the relationship trust $RT_{v_i->v_j}$ values for each neighbour nodes v_i, v_j using ***algorithm 1***;
  Calculate the location trust $LT_{v_i->v_j}$ values for each neighbour nodes v_i, v_j using ***algorithm 2***;
  Measure the mutual-friend similarity score $MFS_{v_i->v_j}$, the Likes similarity score $LIS_{v_i->v_j}$, and the group-joined similarity score $GJS_{v_i->v_j}$ using the Sorensen similarity metric [25];
  $MFS_{v_i->v_j} = \frac{|v_i.FL \cap v_j.FL|}{|v_i.FL|+|v_j.FL|}$;
  $LIS_{v_i->v_j} = \frac{|v_i.LI \cap v_j.LI|}{|v_i.LI|+|v_j.LI|}$;
  $GJS_{v_i->v_j} = \frac{|v_i.GJ \cap v_j.GJ|}{|v_i.GJ|+|v_j.GJ|}$;
  Measure the direct trust $DT_{v_i->v_j}$ values for each neighbour nodes pair (v_i, v_j) and assign it to each corresponding edges as edge weight;
  $DT_{v_i->v_j} = RT_{v_i->v_j} + LT_{v_i->v_j} + MFS_{v_i->v_j} + LIS_{v_i->v_j} + GJS_{v_i->v_j}$;
  $W = DT_{v_i->v_j}$;
end for

Figure 6 depicts the input–output scenario, and Algorithm 4 describes the procedure of indirect trust inference.

Algorithm 4 Inferring indirect trust

Input: The directed weighted social network graph $G = (V, E, W)$, a source node v_s, and a target node v_t.
Output: TrustRoutes and the indirect trust $IT_{v_s->v_t}$ value from a source node v_s to a target node v_t.
Generate the trusted graph from the social network graph G;
for all weighted edges $e_{i->j} \in E$ in the social network graph $G = (V, E, W)$ **do**
  if $W(e_{i->j}) < \tau$ **then**
    Delete the edge $e_{i->j}$ from the social network graph G;
  end if
end for
Find all trusted routes from a source node v_s to a non-neighbour target node v_t in the trusted graph using the breadth-first search technique;
Calculate trust values T_{route} of each trusted routes;
$T_{route} = \frac{\sum_{x=1}^{Number\ of\ nodes\ in\ the\ route} A_{x,x+1}}{Number\ of\ nodes\ in\ the\ route}$;
Aggregate trust values of each route and measure the indirect trust $IT_{v_s->v_t}$ between a source node v_s and a non-neighbour target node v_t;
$IT_{v_s->v_t} = \frac{\sum_{y=1}^{Number\ of\ route\ from\ v_s\ to\ v_t} T_{route(y)}}{Number\ of\ route\ from\ v_s\ to\ v_t}$;



(a) The input graph

(b) The output graph

Fig. 6 The input–output scenario of the indirect trust inference module

4 Experimental Results and Discussion

We use the proposed approach to implement various application scenarios and validate our approach. We first evaluate the application scenarios with the synthetic dataset, and then we test them using the two real-world social network datasets, *soc-Advogato* [26] and *soc-Epinions* [26]. We have downloaded both the real-world datasets from the *networkrepository.com* site and modify them as per our requirement using Gephi software. Table 2 shows the details of the datasets.

Our proposed solution contains two phases: the direct trust measure and indirect trust inference. The direct trust measure can be useful to distinguish friend requests, whether it is from genuine users or it is from malicious users. We consider the accuracy parameter to check how accurately our direct trust measure identifies friend requests in real time. We have implemented the friend-request identification scenario using the direct trust measure module in the python programming language using the NETWORKX package. We have generated the synthetic dataset by creating nodes and linking nodes with each other by edges. In the synthetic dataset, nodes are users, and edges represent relationships among users. Table 3 shows the details of the synthetic dataset.

We manually labelled nodes as genuine nodes or fake nodes and sent friend requests from genuine nodes and suspicious nodes to the manually identified genuine node. Based on the trust value between the requesting node and the genuine node, and using the pre-defined threshold τ_1, the friend-request identification module decides whether the requesting node is a genuine node or a suspicious node and assist the genuine node to accept or reject that friend request. We sent several friend requests from genuine and fake nodes to the genuine node and measured the friend-request identification application's accuracy as follows:

$$\text{Accuracy} = \frac{\text{TPR} + \text{TNR}}{\text{TPR} + \text{FPR} + \text{TNR} + \text{FNR}}. \tag{9}$$

In Eq. 9, the number of genuine requests from the total genuine requests identified as genuine requests is the true positive rate (TPR). The number of fake requests from the total fake requests identified as fake requests is the true negative rate (TNR). The number of fake requests from the total fake requests identified as genuine requests is the false positive rate (FPR). The number of genuine requests from the total genuine requests identified as fake requests is the false negative rate (FNR).

Table 4 gives the performance results of the friend-request identification application. A value of the pre-defined threshold is set to $\tau_1 = 0.94$.

Figure 7 shows the performance parameters TPR, FPR, TNR, and FNR results of the friend-request identification application with different numbers of friend requests. Figure 8 depicts the accuracy of the friend-request identification scenario with a different number of requests.

The direct trust measure can also be useful in detecting Sybil attack in ONSs. We have implemented the Sybil attack detection application using the direct trust measure in the python programming language (NETWORKX package). We have updated the synthetic and the real-world datasets by adding 10% Sybil nodes in them manually. We linked the Sybil nodes with genuine nodes using the edges (we called them the attack edges). Table 2 shows the original datasets' details, and Table 5 depicts details of the modified datasets with the Sybil nodes and attack edges.

We have tested the Sybil attack detection application with the synthetic and real-world datasets and measure **the detection rate**. The Sybil attack detection application is based on our direct trust measure procedure. The application measures the direct trust score between each directly connected nodes. It compares the direct trust scores with the pre-defined threshold τ_2, finds the suspicious links, and detects Sybil attack in the network. We define and evaluate the detection rate of the Sybil attack detection module as follows:

Table 2 The datasets information

	Synthetic dataset	soc-Advogato	soc-Epinions
#Nodes	250	6541	26588
#Edges	600	51127	100120
Type	Directed	Directed	Directed
Vertex type	User	User	Consumer
Edge type	Trust	Trust	Trust
Weight of edges	Positive weights	Positive weights	Positive weights

Table 3 The synthetic dataset details

#Nodes	#Requests	#Genuine requests	#Fake requests
50	50	42	8
100	100	85	15
150	150	127	23
200	200	170	30
250	250	212	38

Table 4 Performance results of the friend-request identification scenario

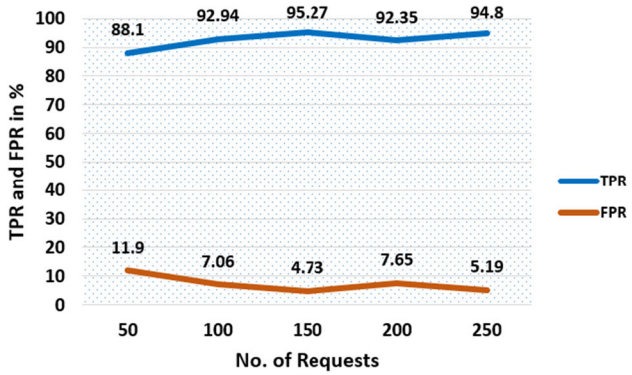
#Requests	TPR	TNR	FPR	FNR	Accuracy
50	88.1%	87.5%	11.9%	12.5%	87.80%
100	92.94%	86.66%	7.06%	13.34%	89.80%
150	95.27%	95.65%	4.73%	4.35%	95.46%
200	92.35%	90%	7.65%	10%	96.17%
250	94.81%	94.73%	5.19%	5.27%	94.45%

Definition 3 (**Detection Rate:**) *Detection Rate is a ratio of the number of detected Sybil nodes to the total number of Sybil nodes in the network.*

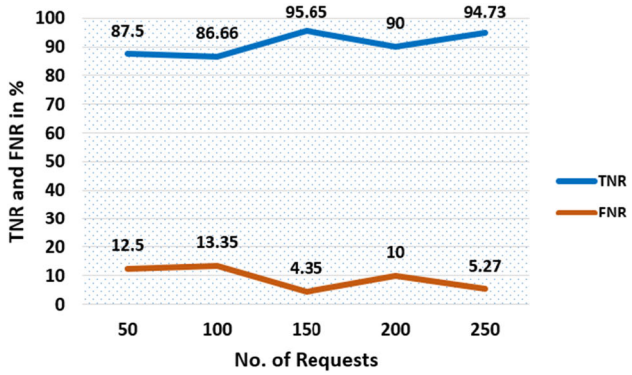
$$\text{Detection Rate} = \frac{\text{No. of Sybil nodes detected}}{\text{Total Sybil nodes in the network}}. \quad (10)$$

The Sybil attack detection module calculates each directly connected node's direct trust value and compares it with the pre-defined threshold τ_2. Edges having the direct trust value less than the pre-defined threshold τ_2 are considered the suspicious edges, and nodes connected by those edges are considered the Sybil nodes. Figure 9 presents the detection rate of our Sybil attack detection module for various datasets. We have set the threshold $\tau_2 = 0.24$.

Along with the detection rate, we consider three evaluation parameters: precision, recall, and F1-score. These parameters are measured as follows:



(a) The TPR and FPR measure



(b) The TNR and FNR measure

Fig. 7 The performance results of the friend-request identification application

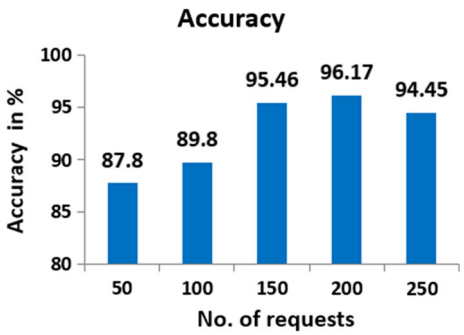


Fig. 8 An accuracy of the friend-request identification application

Table 5 The modified datasets details

Datasets	#Nodes	#Edges	#Sybil nodes	#Attack edges
Synthetic dataset	250	600	25	34
soc-Advogato	6541	51127	654	1837
soc-Epinions	26588	100120	2658	9090

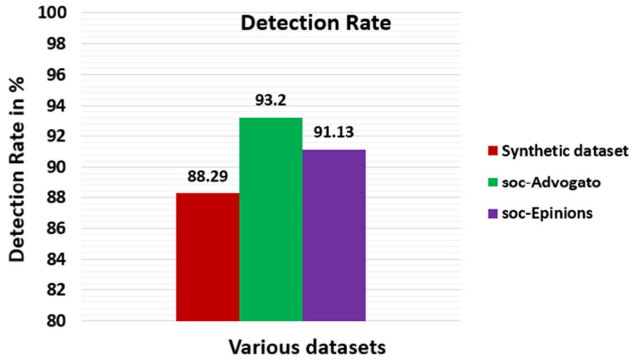


Fig. 9 Detection rate of the Sybil attack detection module

Table 6 The performance results of the Sybil attack detection module

Datasets	Precision	Recall	F1-score
Synthetic dataset	91.00%	89.16%	90.07%
soc-Advogato	93.67%	93.36%	93.51%
soc-Epinions	91.75%	91.58%	91.66%

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{11}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{12}$$

$$\text{F1} - \text{Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{13}$$

The Sybil detection application has been tested with both synthetic and real-world datasets. The performance results of the application are shown in Table 6.

We compare our scheme's results to those of the current methods [13] and [23]. Figure 10 compares the proposed approach's performance metrics results using the soc-Advogato dataset with the state-of-the-art approach TISoN [13].

Figure 11 compares the proposed approach's performance parameter F1-score value to the current scheme Guardian [23]. Figures 10 and 11 demonstrate that the proposed method outperforms the existing methods [13] [23].

We have implemented two applications using the proposed trust evaluation method. In our implementation, we have normalized the direct trust values into the range of [0, 1]. We employ the threshold τ_1 for the friend-request identification application and τ_2 for the Sybil attack detection application. We have carried out a set of experiments in which we varied values of the thresholds τ_1, τ_2 to see how they affect the accuracy and detection rate of the applications. In the friend-request identification application, when we set the threshold τ_1 value high, some genuine requests are identified as fake, resulting in a high false negative rate and lower accuracy. If the threshold τ_1 value is set too low, some fake requests will go undetected, resulting in a high false positive rate and lower accuracy. When we set the threshold τ_2 value to a high value, some genuine nodes are identified as Sybil nodes in the Sybil attack detection application, resulting in a high false negative rate and a lower detection rate. If the τ_2 threshold is set too low, some Sybil nodes will not be detected, resulting

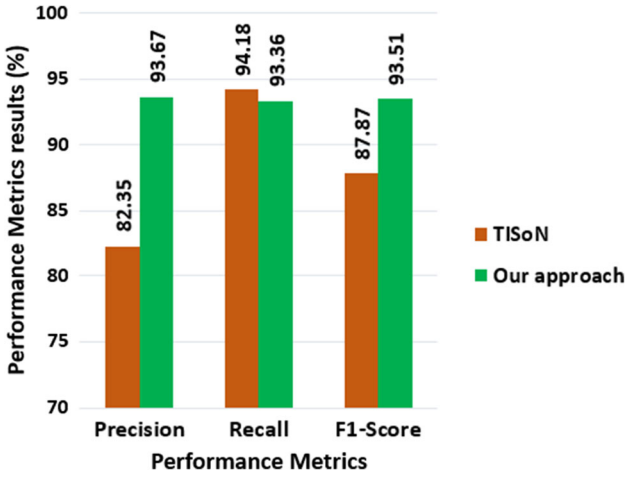


Fig. 10 Comparing the proposed approach's performance with the TISoN [13]

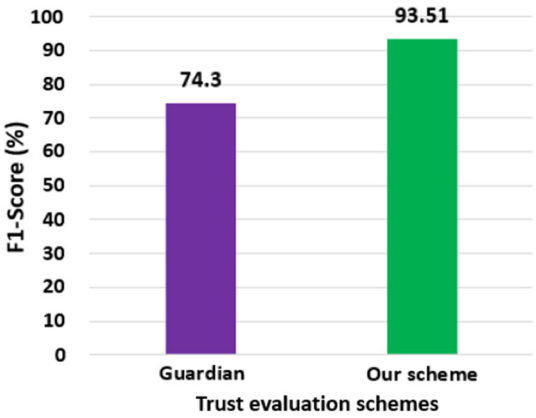


Fig. 11 Comparing our approach's F1-score with the existing scheme Guardian [23]

in a high false positive rate and a lower detection rate. In our approach, we keep the thresholds adjustable.

Our indirect trust inference method measures the indirect trust effectively between any two non-neighbour nodes in the network. In the indirect trust inference, we first pre-process the social network graph and then apply the BFS technique to find all paths between a source node and a destination node. In pre-processing, the direct trust values of each link are compared with the pre-defined threshold. The edges with less direct trust value than the pre-defined threshold are identified as the suspicious edges and are discarded. The BFS technique checks every vertex and edge once; hence its time complexity is $\mathbf{O}(\mathbf{V}+\mathbf{E})$. Our indirect trust inference method removes the edges with less direct trust value; it requires fewer edges to walk to find paths from a source node to a destination node. Thus, **the traversal cost** of our indirect trust inference method is less than $\mathbf{O}(\mathbf{V}+\mathbf{E})$. The existing approach, e.g. the TidalTrust algorithm [7] considered only the shortest and strongest path to infer indirect trust, decreasing its efficiency and also low the quality of inferred trust. Our indirect trust inference module considers all paths' trust values between a given source node and a target node, maximizing inferred trust quality. Thus, the indirect trust inference module infers indirect trust between any two non-neighbours nodes efficiently and outperforms the TidalTrust algorithm [7].

The proposed approach is effective at measuring the direct trust and inferring the indirect trust between participants in online social networks (OSNs). Our approach is also efficient in terms of traversal cost because it discards suspicious edges, resulting in fewer edges to traverse and thus a lower traversal cost. Our approach outperforms the current approaches [7] [13] [23]. Hence, the proposed approach can play an important role in users' decision-making in several scenarios in OSNs.

5 Conclusion and Future Scope

This paper proposed the hybrid trust evaluation approach for OSNs that includes two integrated modules: direct trust measure and indirect trust inference modules. We implemented the friend-request identification and the Sybil attack detection applications using the proposed approach and tested them using synthetic and real-world datasets. The experimental results validated the effectiveness of the proposed direct trust measure method. The false rates of both applications are also very low. The proposed approach outperforms the current methods [13] [23]. Hence, the proposed direct trust measure method is useful in many applications. The indirect trust inference module discarded the suspicious links, requiring fewer edges to walk to find paths between the trustor and the trustee. Thus, the traversal cost of the proposed indirect trust inference module is low. Moreover, the module also incorporated all the paths' trust values between the trustor and the trustee; therefore, it maximized the inferred trust quality. The proposed indirect trust inference module efficiently determines the trust score of any non-neighbour trustees in the network, and it also outperforms the existing approach [7]. As future work, the proposed direct trust measure and indirect trust inference methods can be improved by extracting and incorporating important multifaceted features as elements of trust.

References

1. Massa, P.; Avesani, P.: Trust-aware recommender systems. In: Proceedings of the 2007 ACM conference on Recommender systems, pp. 17–24 (2007)
2. Golbeck, J.; Hendler, J.: Inferring binary trust relationships in web-based social networks. ACM Trans. Internet Technol. (TOIT) **6**(4), 497–529 (2006)
3. Jøsang, A.; Ismail, R.; Boyd, C.: A survey of trust and reputation systems for online service provision. Decision Support Syst. **43**(2), 618–644 (2007)
4. Sherchan, W.; Nepal, S.; Paris, C.: A survey of trust in social networks. ACM Comput. Surveys (CSUR) **45**(4), 1–33 (2013)
5. Cho, J.H.; Chan, K.; Adali, S.: A survey on trust modeling. ACM Comput. Surveys (CSUR) **48**(2), 1–40 (2015)
6. Jiang, W.; Wang, G.; Bhuiyan, M.Z.A.; Wu, J.: Understanding graph-based trust evaluation in online social networks: methodologies and challenges. ACM Comput. Surveys (CSUR) **49**(1), 1–35 (2016)
7. Golbeck, J.A.: Computing and applying trust in web-based social networks. Ph.D. thesis, University of Maryland (2005)
8. Avesani, P.; Massa, P.; Tiella, R.: Moleskiing. It: a trust-aware recommender system for ski mountaineering. Int. J. Infonomics **20**(35), 1–10 (2005)
9. Wangl, G.; Wu, J.: Multi-dimensional evidence-based trust management with multi-trusted paths. Fut. Gener. Comput. Syst. **27**(5), 529–538 (2011)
10. Wang, G.; Wu, J.: Flowtrust: trust inference with network flows. Frontiers Comput. Sci. China **5**(2), 181 (2011)
11. Jiang, W.; Wang, G.; Wu, J.: Generating trusted graphs for trust evaluation in online social networks. Fut. Gener. Comput. Syst. **31**, 48–58 (2014)
12. Jiang, W.; Wu, J.; Li, F.; Wang, G.; Zheng, H.: Trust evaluation in online social networks using generalized network flow. IEEE Trans. Comput. **65**(3), 952–963 (2016)
13. Hamdi, S.; Gancarski, A.L.; Bouzeghoub, A.; Yahia, S.B.: Tison: trust inference in trust-oriented social networks. ACM Trans. Inf. Syst. (TOIS) **34**(3), 1–32 (2016)
14. Li, M.; Xiang, Y.; Zhang, B.; Huang, Z.; Zhang, J.: A trust evaluation scheme for complex links in a social network: a link strength perspective. Appl. Intell. **44**(4), 969–987 (2016)
15. Rahangdale, R.; Thakar, U.: A user action based approach to determine trustworthiness among users in social network. In: 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), pp. 148–152. IEEE (2017)
16. Kiliroor, C.C.; Valliyammai, C.: Trust analysis on social networks for identifying authenticated users. In: 2016 Eighth International Conference on Advanced Computing (ICoAC), pp. 37–41. IEEE (2017)
17. Al-Garadi, M.A.; Varathan, K.D.; Ravana, S.D.; Ahmed, E.; Mujtaba, G.; Khan, M.U.S.; Khan, S.U.: Analysis of online social network connections for identification of influential users: survey

and open research issues. ACM Comput. Surveys (CSUR) **51**(1), 1–37 (2018)
18. Li, P.; Zhao, W.; Yang, J.; Sheng, Q.Z.; Wu, J.: Lets corank: trust of users and tweets on social networks. World Wide Web **23**(5), 2877–2901 (2020)
19. Meo, P.D.: Trust prediction via matrix factorisation. ACM Trans. Internet Technol. (TOIT) **19**(4), 1–20 (2019)
20. Khaksari, A.; Keyvanpour, M.: Tp-ta: a comparative analytical framework for trust prediction models in online social networks based on trust aspects. Artif. Intell. Rev. **52**(3), 1929–1960 (2019)
21. Jøsang, A.; Hayward, R.; Pope, S.: Trust network analysis with subjective logic. In: Proceedings of the 29th Australasian Computer Science Conference-Volume 48, pp. 85–94 (2006)
22. Gong, Z.; Wang, H.; Guo, W.; Gong, Z.; Wei, G.: Measuring trust in social networks based on linear uncertainty theory. Inf. Sci. **508**, 154–172 (2020)
23. Lin, W.; Gao, Z.; Li, B.: Guardian: Evaluating trust in online social networks with graph convolutional networks. In: IEEE INFOCOM 2020-IEEE Conference on Computer Communications, pp. 914–923. IEEE (2020)
24. Gao, X.; Xu, W.; Liao, M.; Chen, G.: Trust prediction for online social networks with integrated time-aware similarity. ACM Trans. Knowl. Discov. Data (TKDD) **15**(6), 1–30 (2021)
25. Sorensen, T.A.: A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons. Biol. Skar. **5**, 1–34 (1948)
26. Rossi, R.; Ahmed, N.: The network data repository with interactive graph analytics and visualization. In: Proceedings of the AAAI Conference on Artificial Intelligence (2015). http://networkrepository.com