IRIS: A Novel Method of Direct Trust Computation for Generating Trusted Social Networks

Sana Hamdi1,2, Alda Lopes Gancarski2, Amel Bouzeghoub2, Sadok Ben Yahia1
1 Department of Computer Science, Faculty of Sciences of Tunis, 2092 EL Manar, Tunis, Tunisia
2 Institut Mines-TELECOM, TELECOM SudParis, UMR CNRS Samovar, 91011 Evry Cedex, France
{sana.hamdi; sadok.benyahia}@fst.rnu.tn, {alda.gancarski; amel.bouzeghoub}@ir-sudparis.eu

Abstract—Improving trust in social networks appears as the first step toward addressing the existing confidence and privacy concerns related to online social networks. Direct trust is used to develop different trust-based methods such as transitivity and access control, however how to compute direct trust levels is rarely discussed in the literature. To address some of the current limitations, we introduce a novel approach for generating trusted social networks and we compute trust levels between users having direct relationships. Experimental results with data extracted from FOAF files show that our work presents high accuracy.

Keywords—Social networks; direct trust; trusted relationships.

I. INTRODUCTION AND MOTIVATIONS

Online Social Networks (OSN) became more and more popular thanks to the features that they offer in terms of sharing different information types as images, video and bookmarks, exchanging ideas, experiences and knowledge, establishing new relationships between users, etc. The decision to exchange information or to create a new relationship is partially based on the establishment of a trust relationship. In addition, this trust exerts an enormous impact on decisions whether to believe or disbelieve information asserted by other users. In fact, it provides information about with whom we should share information, from whom we should accept information, and what consideration to give to information from people when aggregating or filtering data.

Trust relationship is considered as a fundamental concept in social activities. It has been used in different applications. For instance, trust-based access control uses trust as a metric to identify users and authorize resources access in [1], [6] and [7]. Trust has been used as a motivation for online interaction and recommendation systems [12]. Trust is a measure of the quality of a peer in P2P systems [15]. The trust management problem was identified as a distinct and important component of security in network services [9].

In this paper we are interested in trust in OSN. In fact, establishing trust among the OSN users plays a vital role in improving the quality of services and enforcing security for the social activities. Our work is aiming to introduce how direct trust, i.e. trust between two users directly connected, can be associated between users.

A. Main issue

Most of the existing trust computing algorithms are based on direct trust values to manage their trust models. However, they suppose that these values already exist and they do not present how to compute these trust levels. Consequently, there is a large gap in the trust models definition. Our work in this paper aims to bridge this gap.

B. Our Contributions

Our goal is to generate a trusted social network from an OSN. Our key ideas and contributions are as follows:
1) We propose a novel approach for computing direct trust degrees between OSN users. It considers the Interactions between users, their existing Relationship types and their Interest Similarity (IRIS).
2) We generate the trusted social network to distinguish between malicious, controversial and benevolent users.
3) To validate the effectiveness of IRIS, we conducted several experiments with a data set extracted from FOAF files.

The remainder of the paper is organized as follows. Section II reviews the related work. Section III presents the FOAF vocabulary and how we exploit it. Section IV introduces the IRIS direct trust computation method. Section V describes the evaluation procedure as well as the results obtained from FOAF files. The final section sketches our contributions and points out avenues of future work.

II. RELATED WORK AND PROBLEM DEFINITION

Once two users are directly connected in a social network, they can compute mutual trust ratings. Indeed, in Web-based social networks, direct trust values are the most important and basic information to propose trust based approaches for protecting user data or inferring trust relationships. In fact, given the direct trust relationships among users of a social network, several methods have been developed to: (i) recursively compute the transitive trust (indirect trust) between two non-neighbor users; (ii) to set the access control policies; (iii) to compute the reputation of a member in the network or (iv) even to compute recommended ratings. However, defining how to compute direct trust is rarely discussed in the literature. So, in the following we review
different works that used direct trust to develop their trust-based methods.

Authors in [18] present an indepth analysis of trust propagation based on the notion of transitivity: if an agent A trusts agent B, and agent B trusts agent C, then, to some extent, the agent A will also trust agent C. Based on this simple concept, a trust metric to assess the reliability of any reachable agent may be inferred. They consider that direct trust is defined a priori and they exploit it to define inferred trust. In the same context of computing indirect trust, authors in [14] propose a new approach that gives an explicit probabilistic interpretation for confidence, i.e., trust, in social networks. They describe SUNNY, a trust inference algorithm that uses a probabilistic sampling technique to assess the user’s confidence in the trust information from some designated sources. In addition, to solve the problem of “To which extent can a user trust another one on a service in a social network setting”, authors in [13] propose the SWTrust framework, in which they focus on generating small trusted graphs from large OSN. To tackle the key challenge of efficiently discovering short trusted paths, they propose an algorithm for processing a large social network.

Authors in [11] also propose a relationship trust computing method, called Tidal, and how those relationships can be used in designing interfaces. As a matter of fact, they present FilmTrust, a website that relies on trust in Web-based social networks to provide predictive movie recommendations. Indeed, trust values assigned for users who rated a film are used as weights to compute the film average rating. Being so, this weighted average of movie ratings reflects the users’ opinion, since the direct or indirect trust values reflect how much the user should trust the opinions of other users rating the movie.

An access control mechanism for Web-based social networks was proposed in [5]. In this work, authors adopt a rule-based approach for specifying access policies on the resources owned by network participants, and where authorized users are denoted in terms of the type, depth, and trust level existing between nodes directly connected in the network. Similarly, another work presented in [1] proposes a reachability-based access control model for OSNs where access control policies are expressed as reachability queries in terms of constraints on the type, direction, distance, and on the trust level according to a given utility between nodes. Yet in the same context, authors in [2] introduce a social access control strategy inspired by multi-level security [3] for protecting data on social networks. Instead of clearance levels, they use trust levels to annotate objects and subjects. The trust level of an object is specified by the creator. The trust level of a subject is obtained from an existing trust level of an object. The computed direct trust values belong to the continuous 0-1 range where 0 denotes complete distrust and 1 absolute trust between the trustor to the trustee user.

In this work, we introduce a novel approach, IRIS, that aims at computing direct trust in OSN. Thus, a management trust model will be able to apply its trust computing algorithms with real, not random, direct trust values that make these algorithms more reliable. Moreover, our method considers users’ interactions or interests as well as their relationship types. The computed direct trust values belong to the continuous 0-1 range that would its straightforward applicability by the surveyed methods. Furthermore, we consider FOAF profiles as the input social network and exploit and extract its necessary informations allowing to compute and generate trust between these OSN users. Worth of mention, our approach could be easily adapted to take other social networks as input and generate its correspondent trusted social network.

At a glance, Table I sketches the surveyed approaches based on direct trust values using the following criteria:

- **Approach’s Aim**: This criterion presents the aim of using direct trust. It can be used to provide access control, to compute indirect trust or to manage personal reputation.
- **Direct Trust Computing**: This criterion checks whether, in the given approach, authors compute direct trust or not.
- **Direct Trust Metrics**: This criterion gives the metrics considered to compute the direct trust.
- **Direct Trust Range Values**: This criterion gives the possible values the direct trust has in the appropriate approach.

As depicted in Table I, several approaches managing trust in social networks are based on direct trust values. Some works used the direct trust to solve trust-inference problems ([13], [14] and [18]), others are interested in managing the access control policies ([1], [2] and [5]), while [11] focuses on creating predictive movie recommendations and [16] concentrates on managing users’ reputations. Only [13] and [16] paid attention to computing direct trust values; in fact, all the remaining works consider direct trust as defined a priori or with random values. However direct trust computation in [13] is only based on users’ interest metric, as well as on past interactions of users in [16]. Regarding the range values of direct trust, most of the reviewed works used a continuous 0-1 range where 0 denotes complete distrust and 1 absolute trust between the trustor to the trustee user.

<table>
<thead>
<tr>
<th>Approach’s Aim</th>
<th>Direct Trust Computing</th>
<th>Direct Trust Metrics</th>
<th>Direct Trust Range Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>To which extent can</td>
<td>Computes direct trust</td>
<td>Computes indirect</td>
<td>Possible values</td>
</tr>
<tr>
<td>a user trust another</td>
<td>in the given approach</td>
<td>or to manage personal</td>
<td>the direct trust has</td>
</tr>
<tr>
<td>one on a service in</td>
<td>to compute indirect</td>
<td>reputation;</td>
<td>in the appropriate</td>
</tr>
<tr>
<td>a social network</td>
<td>trust or not;</td>
<td>using direct trust;</td>
<td>approach.</td>
</tr>
<tr>
<td>setting.</td>
<td></td>
<td>to manage personal</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>reputation.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Direct trust metrics</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Direct trust range</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>values</td>
<td></td>
</tr>
</tbody>
</table>

As depicted in Table I, several approaches managing trust in social networks are based on direct trust values. Some works used the direct trust to solve trust-inference problems ([13], [14] and [18]), others are interested in managing the access control policies ([1], [2] and [5]), while [11] focuses on creating predictive movie recommendations and [16] concentrates on managing users’ reputations. Only [13] and [16] paid attention to computing direct trust values; in fact, all the remaining works consider direct trust as defined a priori or with random values. However direct trust computation in [13] is only based on users’ interest metric, as well as on past interactions of users in [16]. Regarding the range values of direct trust, most of the reviewed works used a continuous 0-1 range where 0 denotes complete distrust and 1 absolute trust between the trustor to the trustee user.

In this work, we introduce a novel approach, IRIS, that aims at computing direct trust in OSN. Thus, a management trust model will be able to apply its trust computing algorithms with real, not random, direct trust values that make these algorithms more reliable. Moreover, our method considers users’ interactions or interests as well as their relationship types. The computed direct trust values belong to the continuous 0-1 range that would its straightforward applicability by the surveyed methods. Furthermore, we consider FOAF profiles as the input social network and exploit and extract its necessary informations allowing to compute and generate trust between these OSN users. Worth of mention, our approach could be easily adapted to take other social networks as input and generate its correspondent trusted social network.
### III. Exploitation of FOAF Vocabulary Informations

These last years, semantic Web researchers have focused on social and structural relationships. In addition, we have seen a dramatic increase in the amount of published RDF documents using the Friend of a Friend (FOAF) vocabulary, providing a valuable resource for investigating how early semantic Web adopters use this technology as well as build social networks. In this section, we describe the FOAF vocabulary and present its necessary informations that we exploit to put on value our proposed method.

#### A. FOAF

The FOAF (Friend-of-a-Friend) vocabulary [4] describes users information and their social connections through concepts and properties using the semantic Web technologies. The FOAF vocabulary reveals basic information of users such as name, surname as well as personal information about the people that a user "knows" and his interest area. It also depicts user information regarding his social relationships by OnlineAccounts such as YahooChatID, msnChatID and users membership information in different groups and organizations. Users resources such as images, thumbnails or logo are included in Documents and Images of the user’s informations.

**Example 1:** The following code example contains a simple FOAF description of a person:

```xml
<foaf:Person rdf:ID="GP">
  <foaf:givenname>Gautier</foaf:givenname>
  <foaf:familyname>Poupeau</foaf:familyname>
  <foaf:weblog rdf:resource="http://www.lespetitescases.net/"/>
  <foaf:img rdf:resource="http://www.lespetitescases.net/got.jpg"/>
  <foaf:gender>male</foaf:gender>
  <foaf:knows>Christian Faure, David Larlet, Emmanuelle Berms</foaf:knows>
  <foaf:interests>Semantic Web, comics, Science fiction</foaf:interests>
</foaf:Person>
```

From this snippet, a program that understands OWL and RDF is able to process the information. Using the FOAF vocabulary, the program can recognize that there is a person named "Gautier Poupeau" with a weblog and a picture online who knows "Christian Faure", "David Larlet", and "Emmanuelle Berms" and is interested in the semantic Web, comics and the science fiction.

#### B. Defining Users’ Relationships

The property "knows" from the FOAF vocabulary can be useful to create social links between users (i.e. one user knows another one). This property possesses several, more specific, sub-properties defined in RELATIONSHIP vocabulary [8]. This last describes the different relationships that can exist between users. The list of the relationships is \{Acquaintance Of, Antagonist Of, Apprentice To, Child Of, Close Friend Of, Collaborates With, Colleague Of, Employed By, Employer Of, Enemy Of, Engaged To, Friend Of, Grandchild Of, Grandparent Of, Has Met, Influenced By, Knows In Passing, Life Partner of, Lives With, Lost Contact With, Mentor Of, Neighbor Of, Parent Of, Sibling Of, Spouse Of, Works With, Would Like To Know\}. More details about each property and its use can be found in [8].

**Example 2:** An online social network example is depicted in Figure 1, where users are connected with multiple direct relationship types. Alice (A), for instance, has a direct relationship of type FriendOf with Bob (B) and a direct relationship of type ParentOf with Carl (C).

After defining the relationships, we can consider a social network as a directed labeled graph. Each node of the graph

<table>
<thead>
<tr>
<th>Approach Aim</th>
<th>Direct Trust Computation</th>
<th>Direct Trust Metrics</th>
<th>Direct Trust Range Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kuter and Golbeck, 2007 [14]</td>
<td>Indirect Trust</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Richters and Peixoto, 2010 [18]</td>
<td>Indirect Trust</td>
<td>No</td>
<td>[0, 1]</td>
</tr>
<tr>
<td>Abdessalem and Ben Dhia, 2011 [1]</td>
<td>Access Control</td>
<td>No</td>
<td>[0, 1]</td>
</tr>
</tbody>
</table>

### Figure 1. An example of an online social network

![An example of an online social network](image-url)
denotes a user in the network, whereas edges represent the existing relationships between users. An edge is directed from the node specifying the relationship to the node for which the relationship has been specified, whereas the label associated with each edge denotes the type of the relationship.

We can formally redefine a social network as follows.\footnote{Definition Definition (Social Network)}

A social network $\mathcal{SN}$ is a graph $\mathcal{SN} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$ where $\mathcal{V}$ is the set of users, $\mathcal{R}$ is the set of possible relationship types between them and $\mathcal{E}$ is the set of directed links between users labeled with relationship types.

C. Presenting Users’ Interests

Many properties (“interest”, “Topic interest”, “theme”) from the FOAF vocabulary are used to define the users’ interests. These informations enrich the social network by assigning the users to different groups according to their interests. In fact, a group is composed of a set of users sharing interests in the same domain.

Example 3: A classification example of users according to their interests from the FOAF informations is presented in Figure 2. For instance, Alice (A), Bob (B), Carl (C) and David (D) are all interested in the Computer Science domain.

IV. Direct Trust Computation

We model the direct trust between a user $v$ and a user $v'$, $DT(v,v')$, as a real value ranging within the unit interval. On the one hand, if $DT(v,v') = 0$, the degree of trust $v$ has in $v'$ is the minimum, i.e. $v$ totally distrusts $v'$. On the other hand, if $DT(v,v') = 1$, then this implies that $v$ gives a total trust to $v'$.

Our approach for computing trust values between directly connected users, $IRIS$, considers the nature of Interactions between them, the Relationships’ type connecting users, as well as their Interests Similarity.

A. Direct Trust Metrics

1) The Friendship Trust: Such relationship type, denoted $r_{v \rightarrow v'}$, is a direct one since $v$ and $v'$ are directly connected through an edge $v \rightarrow v'$. The direct trust level assigned to a friend should be different from the one assigned to an enemy or a close friend.

We define a friendship function, $F : \mathcal{V} \times \mathcal{R} \times \mathcal{V} \rightarrow [0,1]$, that computes the trust degree $ft$ in $[0,1]$ between two vertices (users) according to their friendship relation. $F(v, r, v') = ft$ underlies that the user $v$ assigns a friendship trust degree $ft$ to his friend $v'$. As presented above, there are many relationship types connecting users in a social network. To be able to capture this potentially large amount of relationship information, we need to generalize relationship types. In our method, we use relationship categories to represent which aspect of closeness and proximity we are referring to, and trust values for the different levels of friendship within each category. The relationship categories are given in Table II below.

![Figure 2. Classification of users according to their interests](image)

<table>
<thead>
<tr>
<th>Category</th>
<th>Relationships</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close relationships</td>
<td>{Child Of, Close Friend Of, Engaged To, Grandchild Of, Grandparent Of, Life Partner of, Lives With, Parent Of, Sibling Of, Spouse Of}</td>
</tr>
<tr>
<td>Friendships</td>
<td>{Colleague Of, Employed By, Employer Of, Friend Of, Mentor Of, Works With}</td>
</tr>
<tr>
<td>Acquaintance</td>
<td>{Collaborates With, Influenced By, Neighbor Of}</td>
</tr>
<tr>
<td>Superficial acquaintance</td>
<td>{Has Met, Knows In Passing, Lost Contact With}</td>
</tr>
<tr>
<td>Bad acquaintance</td>
<td>{Antagonist Of, Enemy Of}</td>
</tr>
</tbody>
</table>

Table II

RELATIONSHIP CATEGORIES

Algorithm 1 returns the trust values corresponding to the direct relationships. It takes as input a user $v$ and the set of his neighbors $V$. In Line 2-3, the algorithm iteratively searches for the relationship type $r$ between the user and each of his neighbors. From Line 4 to 18, it returns the corresponding friendship trust values $ft$ according to the category of this friendship type.

2) The Interactions’ Trust: The trust a social network user $v$ has in his friend $v'$ may refer to the reputation of $v'$ based on his interactions within the social network. Examples of interactions include downloading files, posting informations, replying to another user, etc. The existing interactions between two users $v$ and $v'$ can influence the trust between them. Thus, we first need to identify metrics that can be used to model this interactions’ trust. Our metrics include the number of positive and negative feedbacks given from $v$ to $v'$.

---

**Figure 2. Classification of users according to their interests**
Data: 1: \( v \), the user
2: \( V \), the set of \( v \)'s neighbors

Result: The friendship trust value \( ft \)

```plaintext
foreach neighbor \( v' \) ∈ \( V \) do
  \( r \leftarrow \text{getRelationship}(v, v') \);
  switch \( r \) do
    | case \( r \in \text{Close relationships} \) then \( ft_{v \rightarrow v'} = 1; \)
    | break;;
    | \( ft_{v \rightarrow v'} \) = 0.75;;
    | break;;
  \endsw
case \( r \in \text{Friendships} \) then \( ft_{v \rightarrow v'} = 0.5; \)
  break;;
endsw
case \( r \in \text{Acquaintance} \) then \( ft_{v \rightarrow v'} = 0.5; \)
  break;;
endsw
case \( r \in \text{Superficial acquaintance} \) then \( ft_{v \rightarrow v'} = 0.25; \)
  break;;
endsw
otherwise then \( ft_{v \rightarrow v'} = 0; \)
endsw
return (\( ft \))
```

Algorithm 1: FRIENDSHIPTRUSTCOMPUTING

We define a satisfaction function \( S : V \times I \times V \rightarrow \{0, 1\} \) where \( S(v, i, v') \) implies the satisfaction value a user \( v \) gives to his neighbor \( v' \), based on their mutual interaction \( i \). If \( S(v, i, v') = 0 \), then \( v \) is not satisfied by the interaction \( i \), otherwise, \( S(v, i, v') = 1 \) means that \( v \) is satisfied by the interaction \( i \).

We define, in Eq 1, a trust value assigned by user \( v \) to user \( v' \) after \( n \) interactions.

\[
\text{it}_{v \rightarrow v'}^{n} = \begin{cases} 
1 - \frac{\text{Neg}_{v \rightarrow v'}}{\text{Pos}_{v \rightarrow v'}} & \text{if } \text{Pos}_{v \rightarrow v'} > \text{Neg}_{v \rightarrow v'} \\
0 & \text{otherwise}
\end{cases} \tag{1}
\]

For each user \( v \), the local value \( \text{Pos}_{v \rightarrow v'} \) presents the sum of \( p \) interactions between \( v \) and \( v' \) considered as positive by \( v \) and \( \text{Neg}_{v \rightarrow v'} \) the sum of \( n-p \) interactions between \( v \) and \( v' \) considered as negative by \( v \). These values are computed recursively in Eq 2 and Eq 3.

\[
\text{Pos}_{v \rightarrow v'} = \sum_{j=1}^{n} S(v, i_j, v') \text{ if } S(v, i_j, v') = 1 \tag{2}
\]

\[
\text{Neg}_{v \rightarrow v'} = n - \text{Pos}_{v \rightarrow v'} \tag{3}
\]

3) The Interests' Similarity Trust: Having the same interests increases trust between two users. Indeed, generally, recommendation systems based on users’ collaboration, compute the recommendations by measuring resemblance between users. Golbeck also highlighted, through the analysis of data in FilmTrust website [11], that there is a correlation between similarity of users and trust between them [10].

Each user \( v \) in a social network is interested in \( N \) different domains, where \( N = \{\text{domains}_{v}\} \). Thus, we propose a similarity trust degree, as shown in Eq 4, that allows a user \( v \) to assess to which extent a user \( v' \) is similar to him.

\[
\text{st}_{v \rightarrow v'} = \frac{|\text{domains}_{v} \cap \text{domains}_{v'}|}{|\text{domains}_{v}|} \tag{4}
\]

B. Generating the Trusted Social Network

1) Direct Trust: Considering the parameters described above, we can compute the direct trust assigned to \( v' \) by \( v \) as presented in Eq 5, where \( \alpha, \beta \) and \( \gamma \) are the normalized factors of weights assigned respectively for the friendship trust, the interactions’ trust and the similarity trust, with \( \alpha + \beta + \gamma = 1 \) and \( \{\alpha, \beta, \gamma\} \subseteq [0, 1] \). Worth of mention here that we assign the three parameters equal weights, so we consider \( \alpha = \beta = \gamma = 1/3 \).

\[
\text{DT}_{v \rightarrow v'} = \alpha \text{it}_{v \rightarrow v'} + \beta \text{it}_{v \rightarrow v'}^{n} + \gamma \text{st}_{v \rightarrow v'} \tag{5}
\]

By associating a trust value with each directed edge linking two users, we obtain a trusted social network as defined below.

(Trusted Social Network)

A trusted social network is the trusted graph \( \text{TSN} = (SN, DT) \), where \( SN \) is the social network and \( DT \) is the value function that describes the weights of the direct trust relationships between two participants in the social network.

Example 4: Figure 3 depicts the generated social network after the computation of the direct trust. Each relationship type is associated with a trust level denoting the direct trust between the two users participating in the given relationship. Considering the trust level, \( \text{DT}(A, B) = 0.7 \), existing between Alice (A) and Bob (B). The relationship type assigned to Bob is FriendOf, so the friendship level is \( \text{ft}_{A \rightarrow B} = 0.75 \). We consider their interactions’ trust is \( \text{it}_{A \rightarrow B} = 0.35 \) and, since they share the same interests, their similarity trust \( \text{st}_{A \rightarrow B} = 1 \). Thus, the obtained direct trust value is \( \text{DT}(A, B) = 1/3 \times (0.75 + 0.35 + 1) = 0.7 \).

2) Enriching the Trusted Network: Giving a total trust to a user equals to issuing a trust statement in him \( (\text{DT}(v, v') = 1) \) while giving him a total distrust corresponds to issuing a distrust statement in him \( (\text{DT}(v, v') = 0) \). This trust computation allows to: (i) to know and consequently to isolate malicious users from the network; (ii) know and encourage benevolent users by rewarding them with good reputation. Therefore, the trust values will be of
less use in preventing malicious users from giving negative interactions.

We now redefine, in Eq 6, the controversialityPercentage quantity introduced in [12]. A user with 1 as controversialityPercentage is totally trusted by all his judges, i.e. $|\text{receivedDistrust}(v)| = 0$, and is judged as benevolent. On the other side, a user with -1 is totally distrusted by all his judges, i.e. $|\text{receivedTrust}(v)| = 0$ and is judged as malicious. Users at these two extremes are non-controversial since all the remaining users agree on their opinions about them. Conversely, a user $v$ with 0 as controversiality percentage, i.e. $|\text{receivedTrust}(v)| = |\text{receivedDistrust}(v)|$, is highly controversial since other users split into two same-sized opinions group about this user, i.e. one half of users trusts him and the other half distrusts him.

$$\text{controversialityPercentage}(v) = \frac{|\text{receivedTrust}(v)| - |\text{receivedDistrust}(v)|}{|\text{receivedTrust}(v)| + |\text{receivedDistrust}(v)|} \quad (6)$$

**Example 5:** consider the generated trusted social network depicted in Figure 3. The network can thus be enriched by defining benevolent, controversial and malicious peers. For instance in Figure 3, John (J) is totally trusted by all his judges, he is then considered as benevolent user. Whereas Mark (M) and Frederic (F) are totally distrusted by all their judges, so they are considered as malicious. Others like Alice (A), Bob (B) and Carl (C) are controversial users, with different levels of controversiality.

We modify the controversialityLevel quantity, introduced in [12], as shown in Eq 7. The controversialityLevel of a user is the number of users who disagree with the majority in issuing a statement about that user. For example, a user who received 21 distrust statements and 14 trust statements has a controversialityLevel of 2/5.

$$\text{controversialityLevel}(v) = \min(|\text{receivedTrust}(v)|, |\text{receivedDistrust}(v)|) \div (|\text{receivedTrust}(v)| + |\text{receivedDistrust}(v)|) \quad (7)$$

A user who has a controversialityLevel of $\alpha$ is called controversial. $0$ – controversial users received only trust or distrust statements and they are non controversial. A user who has a controversialityLevel not less than $\alpha$ is defined as at least $\alpha$ – controversial user.

V. EXPERIMENTAL EVALUATION

We are interested in evaluating the IRIS trust-based approach. So, in this section, we first describe the used data set. Second, we define the accuracy metrics needed to the evaluation. Third, we show the tests’ results and finally we compare our method vs surveyed methods.

A. Data Set

We use the foafPub data set from http://ebiquity.umbc.edu/ which is a dataset of information extracted from FOAF files collected during the Fall of 2004. A total of 201,612 RDF triples with provenance information are included. The dataset is distributed as a zip file containing SQL commands to create three tables: dict-host, dict-url and triple-person. The SQL commands were generated from the original MySQL database using the export command.

B. Accuracy Metrics

To test the accuracy, we adapted the accuracy metrics in [13] and [19] including absolute error, precision, recall, and F-score.

1) Absolute Error: It is the difference between the actual value of trust and the computed value from the proposed method.

$$\text{Absolute error} = |\text{calculatedTrust} - \text{actualTrust}| \quad (8)$$

The actual value of trust is obtained by using Richardson’s technique [17], which uses the concept of quality of users assigning a trust value to each node, and we make a little modification (the same method as [13] and [19]). Each user has a quality measure $q_v \in [0, 1]$. A user’s quality determines the probability that a statement given by the user is true and complete. The higher the user’s quality, the more likely to be trusted he is. Therefore, for any pair of users $v$ and $v'$ where $v$ trusts $v'$:

$$\text{actualTrust} \in [\max(q_v - \delta_{vv'}, 0), \min(q_v + \delta_{vv'}, 1)] \quad (9)$$

In Eq 9, $q_v$ is the quality of the user $v$ and $\delta_{vv'}$ is a noise parameter that determines how accurate users were at estimating the quality of the user they were trusting. We suppose that a user with low quality is bad at estimating trust, so for these experiments we set $\delta_{vv'} = \frac{1 - q_v}{2}$.
2) Precision, Recall and Fscore: Accuracy represents the ability of predicting a user to be trusted or not. Based on the defined criterion for accuracy, making a right decision is the ultimate metric for comparison. We set a threshold = 0.5 for deciding to trust or not to trust. If the calculated DT value is equal to or greater than 0.5, we trust, otherwise we distrust. We use precision and recall metrics to compare the accuracy of methods in making the trust decision. Precision and recall are defined for three states as follows.

**Trust State:**

\[ \text{Precision}_t = \frac{|X_t \cap Y_t|}{|Y_t|}, \quad \text{Recall}_t = \frac{|X_t \cap Y_t|}{|X_t|} \] (10)

**Distrust State:**

\[ \text{Precision}_d = \frac{|X_d \cap Y_d|}{|Y_d|}, \quad \text{Recall}_d = \frac{|X_d \cap Y_d|}{|X_d|} \] (11)

**General State:**

\[ \text{Precision}_{\text{total}} = \frac{|X_t \cap Y_t| + |X_d \cap Y_d|}{|Y_t| + |Y_d|} \] (12)
\[ \text{Recall}_{\text{total}} = \frac{|X_t \cap Y_t| + |X_d \cap Y_d|}{|X_t| + |X_d|} \] (13)

We use the Fscore metric to measure the accuracy using recall and precision jointly.

\[ Fscore = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \] (14)

C. Results for the Proposed Method

The programs were run for a range of threshold values \( \text{th} \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.6, 0.7, 0.8, 0.9\} \). Figures 4 and 5 show the behavior of this parameter with respect to accuracy metrics.

In Figure 4, as far as \( \text{th} \) varies, the accuracy decreases significantly in trust state while it increases in distrust state, and steadily increases in general state. This finding indicates that whenever we are interested in just predicting to trust or to distrust, then \( \text{th} = 0.5 \) would be the best choice. Nevertheless, Figure 5 shows the behavior of the mean of error with \( \text{th} \). It has the best values when \( \text{th} \) ranges between 0.5 and 0.6. From these two figures, the optimum value of \( \text{th} \) could be 0.5.

<table>
<thead>
<tr>
<th>Method</th>
<th>Fscore</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max-Mean</td>
<td>0.49</td>
</tr>
<tr>
<td>Tidal Trust</td>
<td>0.67</td>
</tr>
<tr>
<td>Max-Min</td>
<td>0.72</td>
</tr>
<tr>
<td>Max-*</td>
<td>0.73</td>
</tr>
<tr>
<td>Max-weight</td>
<td>0.81</td>
</tr>
<tr>
<td>IRIS</td>
<td>0.82</td>
</tr>
</tbody>
</table>

Table III

Fscore for different methods

D. Comparison with Other Methods

The results of Tidal trust [11] and fuzzy based composition methods such as Max-min, Max-*, Max-mean and Max-weight were shown in previous works [19]. Table III compares Fscore for the proposed method with the other methods; it is observed that the accuracy is high and reaches 82% for IRIS. Figure 6 shows that IRIS has the best mean of error (it is lower than other methods and is around 0.098).
VI. CONCLUSION

In this paper a new method for direct trust computation and evaluation is proposed. It aims at generating trusted social networks, what is useful to develop different trust-based methods for computing indirect trust or for providing the access control policies. This new method is tested experimentally using a data set extracted from FOAF files. The tests showed that our work presents high accuracy. The obtained results are compared and contrasted with those obtained from other methods. Experimental results demonstrate that the proposed approach is well comparable.

Our future work includes creating a general trust management model. It will be based on our IRIS method to provide access control policies and compute trust even between users that do not have direct relationships. Besides, we aim to find a generally accepted data collection for different social networks to evaluate the model.

REFERENCES


