IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 18 November 2024, accepted 22 December 2024, date of publication 25 December 2024, date of current version 3 January 2025.

Digital Object Identifier 10.1109/ACCESS.2024.3522560

RESEARCH ARTICLE

Using Trust and Reputation for Detecting Groups of Colluded Agents in Social Networks

MARIANTONIA COTRONEI[1], SOFIA GIUFFRÈ[1], ATTILIO MARCIANÒ[1], DOMENICO ROSACI[1], AND GIUSEPPE M. L. SARNÈ[2], (Senior Member, IEEE)

[1]Department of Information Engineering, Infrastructure and Sustainable Energy (DIIES), Mediterranea University of Reggio Calabria, 89124 Reggio Calabria, Italy

[2]Department of Psychology, University of Milano-Bicocca, 20126 Milan, Italy

Corresponding author: Attilio Marcianò (attilio.marciano@unirc.it)

This work was supported in part by the Project CALabria HUB Ricerca Innovativa Avanzata (CAL.HUB. RIA) funded by Italian Ministry of Health under Project CUP: F63C22000530001 and Project CUP: C33C22000540001, and in part by the Piano Nazione di Ripresa e Resilienza (PNRR) Project TECH4YOU–Technologies for Climate Change Adaptation and Quality of Life Improvement funded by Italian Ministero dell'Università e della Ricerca (MUR) under Grant CUP: C33C22000290006.

ABSTRACT One of the most common types of malicious behavior in social networks is represented by *collusion*, which consists of a secret cooperation between two or more agents providing mutual, highly positive feedback to each other. This collusion creates misleading advantages for the involved agents, deceiving others and distorting the actual reputation perception of the colluding members. Although the well-known EigenTrust algorithm can be fruitfully used to detect colluded agents, two important issues arise which limit its effectiveness: 1) it requires input information about which agents can be a-priori considered particularly trustworthy; and 2) it is not designed to handle situations in which we have several, different groups of colluded agents. These problems lead EigenTrust, to produce a significant number of false positives in some real situations. In this paper, we address the aforementioned issues. We introduce an automatic procedure to provide EigenTrust with the necessary inputs, and we propose an appropriate algorithm that combines EigenTrust with a clustering process. This procedure groups agents based on their reputation scores to tackle the presence of different groups of colluded agents. Through experiments, we demonstrate that our method, while maintaining the same effectiveness as EigenTrust in detecting malicious agents, is significantly more capable of avoiding the generation of false positives.

INDEX TERMS Multi-agent systems, recursive models, reputation, social networks, trust.

I. INTRODUCTION

Nowdays, social networks are configured as communication spaces having different purposes, where people operate with different activities as writing posts and talking (Facebook [1], X [2]), sharing images (Instagram [3]) and videos (TikTok [4]), making business transactions (Amazon [5], eBay [6]), dealing with job activities (Linkedin [7]) or research topics (ResearchGate [8]) etc., with the purpose of providing the community members with the possibility to share knowledge, insights and ideas. As it is easy to understand, a common, crucial problem in all the social network environments is represented by the necessity to guarantee an high level of trustworthiness about the members of the communities, which can be either human beings or software entities acting on the behalf of humans to support them in their activities. Generally, we call these social network members as *agents* (regardless of whether they are human or software) and we use the term *trust* to denote the quantitative measure of the trustworthiness perceived by an agent in another one, and the term *reputation* referring to the global measure of trustworthiness that the whole community has in a given agent. This problem becomes particularly important when the social network involves agents which compete with each other or desire to have mutual collaborations [9], [10]. In both these situations, each agent has the necessity to accurately choose its partners, since its incomes and outcomes depend on this choice. However,

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

there is already the possibility to operate in presence of malicious agents performing deceptive behaviors or even operating frauds in order to reach personal purposes. One of the most common type of malicious behavior is represented by *collusion*. It consists of a secret cooperation between two or more agents which provides mutual, high positive feedback to each other, to generate incorrect advantages for themselves deceiving others of perceiving the actual reputation of the colluded agents. In the related scientific literature, several approaches to detect malicious agents in social networks have been presented (see Section II-B), and one of the most effective is represented by the EigenTrust [11] algorithm which, in P2P Networks, should support sureness and authenticity of traffic flows. From a conceptual point of view, this algorithm is a variation of PageRank [12], the well-known algorithm used by the Google search engine to assign the ranking to URL based on the number of links that the other pages have with that specific web page. In particular, the EigenTrust algorithm assigns a measure of the reputation to an agent accordingly to the trust that other agents address with that specific target.

Although EigenTrust can be fruitfully used to detect agents performing several different malicious behaviors in a social network, three important issues arise to limit its effectiveness when the particular malicious behavior is the collusion: (*i*) EigenTrust algorithm needs to yield as inputs the information about those agents that can be a-priori considered as particularly trustworthy, and detecting such agents is not a trivial task in the particular case of collusion; (*ii*) Eigentrust detects malicious agents by artificially rewarding, in terms of trust, non honest agents, without considering the reliability of these agents (i.e., the effectiveness in performing their tasks and this generates an artificial modification of the actual reputation values); (*iii*) finally, EigenTrust is not designed to face the situation in which we have several, different groups of colluded agents. In this latter case, the presence of different clusters of malicious agents leads Eigentrust to confuse malicious agents, which have low reputation, with non-malicious but not very effective ones, which also have low reputation. This confusion generates, in these particular cases, a significant number of false positive. We have addressed the first two issues in a previous paper [13], by introducing an automatic procedure to provide EigenTrust with the necessary inputs mentioned in (*i*) and adopting a strategy that detects malicious agents without modifying reputation values of the other agents. Here we refer to our previous method as Efficient Reputation-Eigentrust (ER-EingenTrust). In this paper, we provide a contribution to face the issue (*iii*). To this purpose, we will introduce a specific algorithm that combines EigenTrust with a clustering procedure, which groups agents based on their reputation scores. We experimentally show that our method, besides maintaining the same effectiveness of Eigentrust in detecting malicious agents (also considering the improvement introduced by ER-EigenTrust), is significantly more capable of avoiding the presence of false positives. We highlight that we have compared our approach with the well-known EigenTrust algorithm recognized as one of the most effective methods for detecting malicious users, but not designed to handle scenarios involving several, different groups of colluding agents. In such cases, it often produces a significant number of false positives in real-world situations. To the best of our knowledge, no other algorithm for malicious user detection, has been proposed in the literature to specifically deal with a situation involving different groups of colluded agents.

II. PROBLEM SETTING AND STATE OF THE ART

A. THE MULTI-AGENT SOCIAL SCENARIO

The possibility of modeling trust-based activities in social networks by means of Social Multi-Agent Systems (SMASs) has been widely considered in the literature as a promising solution to compute effective measures of trustworthiness in these communities [14], [15]. In fact, it is able to improve the quality of social interactions [16], limiting or preventing malicious behaviors as the collusion.

Several issues regarding social behaviors in MAS have been investigated in the literature. For example, [17] explores the flocking dynamic behavior of multi-agent networks with limited communication resources, where there are both cooperative and antagonistic relationships among agents. As another examples, in [18] the robust bipartite tracking consensus problem for second-order multi-agent systems has been addressed in the presence of lumped disturbances and unknown velocity information, while in [19] the practical predefined-time consensus (PTC) tracking problem is studied for nonlinear multiple ground vehicles (MGVs) via dynamic event-triggered control (ETC) and self-triggered control (STC).

More in particular, SMASs [20], [21], [22], [23], [24] introduce the possibility to quantitatively measure the trustworthiness of an agent, be it either a human being or a software, and using such measures to detect malicious agents [25], [26], [27], [28], [29], [30], [31].

In this setting, SMASs are specific frameworks designed to introduce the computation of trust and reputation measures by using the feedback of the agents themselves. Several different SMASs have been developed and in these environments there is always the possibility that some malicious agents can engage in misleading behaviors [32], [33], [34] for deceiving others to obtain some undue advantages. In this perspective, SMASs are generally equipped with specific engines to be resilient to such kind of attacks [35], [36]. In this context, in a SMAS, the agent interactions are modeled as e-services provided by a provider agent (i.e. the *trustee*) to a client agent (i.e. the *trustor*), where the trust is defined as:"the quantified belief by the trustor with respect to the competence, honesty, security and dependability of the trustee within a specified context" [37]. Generally,

trust involves different dimensions under which the mutual interaction between the two agents is evaluated. Some usual trust dimensions are *reliability* (i.e. ability of effectively and efficiently providing requested services), *honesty* (i.e. willing to correctly operating, avoiding misleading behaviors) and *security* (i.e. preventing undesired activities as, for instance, unauthorized accessing to private data).

This paper only deals with reliability and honesty, which strictly depend only on the agent's behavior, and we focus only on a particular malicious behavior, namely collusion. Consequently, we here use the term *trust* as a measure of reliability and honesty, and the term *collusion* as the common willing of a group of agents to jointly operate in order to falsify trust measures.

While trust is a subjective measure (formed by the contribution of *reliability* and *reputation*, with reliability directly assessed by a trustor in a peer-to-peer interaction with a trustee), *reputation* is a measure of trustworthiness evaluated by the entire community in relation to a specific agent. The concept of reputation assumes a relevant role in all those situations where an agent x is not provided with a sufficient knowledge about another agent y. In those cases, x might use the y's reputation to decide if y can be considered as a trustworthy interlocutor.

We highlight an important issue arising when it is necessary to compute reputation measures in presence of several, distinct groups of colluded agents. In this situation, the agents of each group operate independently from the colluded agents of each other group. The result will be that the effect of the malicious activities of a given group of colluded agents could also falsify the reputation of another group of colluded agents, making harder to detect all the malicious agents that in a given time are present in a community.

B. RELATED WORK

There are a number of studies and surveys of Trust-Reliability-Reputation (TRR) systems in the literature that analyze the robustness of TRRs to a variety of malicious behaviors, although sometimes it can be complicated to compare systems designed for different specific application contexts. This section will present those TRR systems that, in our opinion, appear as the more relevant for our aims.

An early and well-known analysis of TRRs is presented in [36] where various TRRs are analyzed as well as their defense mechanisms towards the most common types of attacks. The studies [38], [39] are also worthy of mention in this area. A common feature of these earlier studies is the absence of a clear and quantitative evaluation methodology about TRRs strengths and weaknesses.

Later, TRRs began to be evaluated by simulating some scenario populated by honest and malicious actors competing with each other to gain some advantage (economic or otherwise) as, for example, in [40], [41]. A generalization of this mechanism is found in testbeds designed to match multiple TRRs against each other. Probably, the best known of these testbeds is ART (Agent Reputation and Trust testbed) [42] used by several authors and also resulted in the past in a kind of annual competition. Other examples of testbeds for TRRs can be found in [43], [44], [45].

In any case, simulation-based approaches are unable to autonomously identify the worst scenario and, inherently, may lead to errors in TRRs evaluations (remember that most TRRs are designed for a well-defined context and relocating them may not be so easy). Differently, verification techniques based on mathematical/analytical approaches [46], [47] allow a more comprehensive analysis of TRRs, but suffer of a lack of generality. Moreover, this approach is more complex than the previous ones.

This overview can only begin by presenting the Reputation System (RS) of eBay [6], the well-known online auction site. This RS, one of the most studied in the literature [48], is very popular and equally simple. Although its simplicity leads it to be exposed to a number of malicious attacks [48], [49], [50], its simplicity has made it easy for users to be understood and accepted. This RS computes reputation based on positive feedback released by users. In particular, the minimum value of the reputation is 0, which is also the initial reputation assigned to each newcomer, this value is increased or decreased according to the feedback received. Note that it is left to the users to interpret the reputation rate. Over time, this RS has been slightly updated to increase its resilience to malicious behavior, but these changes have not changed its basic characteristics.

PeerTrust [51] is a distributed transaction-based reputation model, designed to work over a structured peer-to-peer overlay network. PeerTrust combines direct feedback, number of the transactions performed by each peer, credibility of indirect feedback sources with transition and context factors information to be more resilient against malicious behaviors and different threats. The goal of PeerTrust is to allow identifying those peers suitable to collaborate on the basis of their reputation-based trustworthiness.

In [52] Hypertrust is described, it is a decentralized solution for discovering and allocating trusted resources into competitive, large scale, federations of utility computing infrastructure. The HyperTrust metric considers reliability and reputation (derived from recommendations) information and allows any user to exploit an efficient finding process to use its trust model for limiting the search of collaborators to an admissible region. A decentralized procedure provides to construct an overlay network including all the nodes of the federation, by exploiting the peers featuring to be interconnected by means of the links of the overlay thus forming clusters.

Obviously, the goal of all TRRs is to classify the actors on the basis of their trustworthiness but some of TRRs can result only a theoretic exercise. In this viewpoint, FIRE [53] is a TRR that, in a multi-agent environment, in a idealistic way imposes the agent benevolence (e.g., each agent freely exchange information with each other agent) and honesty

in exchanging information. Its model considers more type of trust and sources of reputation feedback, namely: the *interaction trust* represents the direct experience of an agent; the *role-based trust* is referred to the nature of agents' relationships; the witness reputation is that of the source releasing the feedback released about the behaviors of other agents; the certified reputation of an agent is that given by a third party suggested by the rated agent. Unfortunately, benevolence and honesty assumptions make this TRR vulnerable to malicious attacks (e.g., collusion, alternate, complaining, etc.) [54] and, moreover, FIRE requires the setting of a large number of parameters, which is another critical aspect of this TRR. Therefore, notwithstanding its versatility, given by the use of different information sources, FIRE is representative of all those TRRs unfitting with the real world, where malicious actors there exist.

Finally, EigenTrust [11], originally designed to operate in peer-to-peer file sharing networks, is one of the most well-known and widely used reputation algorithms. In Eigentrust, each peer evaluates every other peer to construct its own trust representation, called local trust. Then, the global reputation of each peer is computed by aggregating into a matrix the normalized local reputations of all peers, appropriately weighted by the trustworthiness assigned by each source (i.e. peer). To achieve this, EigenTrust imposes trust transitivity. The authors have shown how trust values, arranged in a trust matrix, converge asymptotically to its eigenvalues. Conversely, this requires the presence of trusted mentors in the system to minimize the influence of colluding activities of malicious peers. From a robustness viewpoint EigenTrust is susceptible to manipulations that peers may perform on their feedback [55]. A significant number of modifications of the original algorithm have been introduced by many authors to increase its robustness [56], [57], [58] and/or adapt it to new scenarios, for example in [59], [60].

The main drawbacks of Eigentrust have been highlighted in the introductiory section. They have been almost overcome with the new strategy proposed in [13]. The aim of this paper is to go a little bit further, considering the situation where the network is populated by several groups of colluded agents. In this case, we show how clusterization techniques can effectively been used to detect such groups and how this information can be iteratively leveraged to assess reliable final reputation scores.

III. A NEW STRATEGY FOR COMPUTING THE REPUTATION

A. THE EIGENTRUST-BASED MODEL

Suppose that the social network is composed by n members, each of them being uniquely identified by an integer i, $1 \leq i \leq n$. The *trust* perceived by the member j with respect to the member i is represented by the real number t_{ij}, $0 \leq t_{ij} \leq 1$, $\forall i, j = 1, \ldots, n$.

The *reputation* r_i of each member i of the social network can be seen as the sum of all the trust values $t_{ij}, j = 1, \ldots, n$, corresponding to a trustor j, weighted by the reputation r_j of j. In other words, r_i can be viewed as the barycenter of all the trust values expressed for i by the trustors.

Let us denote with $T = [t_{ij}]$ the *trust matrix*.

The *reputation vector* $r = (r_1, \ldots, r_n)^T$ can then be computed by solving the following eigenvector problem:

$$Tr = r, \quad \|r\|_1 = 1, \tag{1}$$

where the 1-norm is defined as the sum of the values of the vector components and with the assumption

$$\sum_{i=1}^{n} t_{ij} = 1, \tag{2}$$

which is equivalent to say that the matrix T is column-stochastic.

A modified version of the eigensystem (1) is represented by the the well-known *PageRank model*:

$$\left(\alpha T + (1-\alpha) v u^T\right) r = r, \tag{3}$$

where $0 \leq \alpha \leq 1$, u is the vector of all ones and v is a non-negative vector with unitary 1-norm that is $u^T v = 1$. The vector v is usually called *teleportation vector*.

The original PageRank algorithm adopts the uniform choice $v = \frac{1}{n}u$. In our framework, this approach fails because the final reputation of honest agents can be affected by the behavior of malicious one. To solve such an issue, a modified model, known as *Eigentrust algorithm* has been proposed [11]. The idea is to choose v in such a a way that the final reputation of malicious users can be mitigated. In particular, the entry v_i is set to zero if the i-th agent is not considered *pre-trusted*; in other words, it cannot be considered as particularly trustworthy.

Nevertheless,the Eigentrust algorithm requires a priori knowledge of pre-trusted users, it does not provide a procedure for distinguishing honest from malicious agents beforehand. Recognizing this limitation, a strategy proposed in [13] involves extracting this information directly from the trust matrix T and incorporating it into the computation of the final reputation vector.

However such a procedure fails in the presence of ''groups'' of malicious agents.

To address this issue, a modified definition of a *malicious users* is introduced, classifying a group of agents as *malicious* when they *collude*. Collusion in this context refers to a scenario where agents engage in a bi-directional exchange of high trust values while concurrently receiving low trust values from other members. This definition provides a basis for identifying and addressing collective malicious behavior within the network and using clusterization strategies for their identification

B. DETECTION OF MALICIOUS USERS THROUGH GRAPH CLUSTERING

An appropriate model for a social network can be given in terms of a graph $G = (V, E)$, where each vertex (or node) v_i

corresponds to a user. In our scenario G is a directed weighted graph, the value t_{ji} (i.e. the trust value assigned by the member i to the member j) representing the weight of the edge (i, j). Note that the adjacency matrix A of such graph corresponds to the transpose of the trust matrix T.

A common feature of social networks is the presence of community structure properties, so that the graph can be often considered organized into subgraphs, each of them representing a *cluster*.

The concept of *similarity* is fundamental for grouping together items that share similar characteristics or neighboring behaviors and an appropriate definition of similarity is the crucial aspect for every clustering process. Similarity can be defined through different metrics, depending on the type of data or characteristics being considered. In our scenario, similarity is associated with a group tendency to assign high trust values within the group, while receiving low or high trust values from the other users, depending on their level of ''maliciousness''.

Our aim is to use techniques for graph clustering for detecting malicious behaviors in order to use such an information for a more reliable computation of the final reputation.

The problem of identifying clusters over a network have been addressed in several contributions (see, for example, [61], [62] and references therein). However, the case of undirected graphs has been mainly considered. Finding clusters in directed graphs is a more challenging task and it is the topic of an increasing research activity (see [63] for a review).

For the purpose of our work, we found out that the use of spectral clustering techniques, originally designed for undirected graphs, can be adopted by means of a proper reformulation of the problem.

Since spectral clustering only operates on symmetric adjacency matrices, a proper redefinition of such matrix must be given.

Typical transformation to the adjacency matrix in order to obtain a symmetric one, include $\tilde{A} = AA^T$, $\tilde{A} = A^TA$.

Nevertheless, due to its simplicity, most of the algorithms use $\tilde{A} = A+A^T$. This approach almost ignores the orientation of edges, except that in the case of pairs of nodes with directed edges in both directions, which is indeed the situation we are facing in this work (malicious pairs). On the other hand, the information about the low trust values received by malicious agents by the other members risks to be lost.

We thus propose to adopt the following transformation for the elements in the new adjacency matrix

$$\tilde{a}_{ij} = \frac{a_{ij} + a_{ji}}{2(0.1 + |a_{ij} - a_{ji}|)} \tag{4}$$

By construction, $\tilde{A} = [\tilde{a}_{ij}]$ is a symmetric matrix, whose elements $\tilde{a}_{ij}$ assume high values if the users i and j are ''similar'', low values if the users i and j are ''not similar'', thus satisfying the definition of similarity given above.

Spectral clustering is a strategy for clustering graph data which operates on the Laplacian matrix L, or on some normalized version of it, performing dimensional reduction before of using a standard clustering procedure, as k-means.

In particular, we make use of the *normalized spectral clustering* procedure described in [64]. Our input data consist in the similarity matrix $\tilde{A}$ as in (4) and the number k of clusters.

Thus the process consists of the following steps:

1) Construct the *normalized graph Laplacian* matrix
$$L = D^{-1/2}(D - \tilde{A})D^{1/2} \tag{5}$$
where D is the *degree matrix* defined as the diagonal matrix with diagonal elements $d_{ii} = \sum_{j=1}^{n} \tilde{a}_{ij}$.
2) Evaluate the k eigenvectors corresponding to the k smallest eigenvalues of L.
3) Obtain a reduced-dimensional representation of the data arranging such eigenvectors in the columns of a matrix V of size $n \times k$.
4) Normalize the each row $v_i \in R^k$ of V so that its 2-norm is equal to 1.
5) Use k-means to cluster the rows v_i, $i = 1, \ldots, n$, of V, into the clusters $C_1, \ldots, C_k$.

An important role in the above described procedure is played by the parameter k connected to the number of clusters, which must be judiciously chosen according to the number of users and the type of social network. A possible automatic choice can be performed in terms of *eigengap heuristic* in the following way: choose k such that the first k eigenvalues of L are very small and the following one is relatively large. This idea works well in case of very distinct clusters, but fails in case there is some kind of ''overlapping''. After several experiments, we empirically found out that, for our purposes, a good choice, in the case of large networks, is a value k approximately equal to $\log(n)$.

Once the clusters $\mathcal{C} = \{C_1, \ldots, C_k\}$ have been obtained, we proceed recursively as follows:

1) Compute the reputation vector r as in (1).
2) Compute the j-th cluster reputation as
$$\bar{r}_j = \frac{\sum_{i \in C_j} r_i}{|C_j|}, \quad \forall j = 1, \ldots, k \tag{6}$$
3) Fixing $\delta > 0$, identify as ''colluding clusters'' all those that possess a reputation less or equal to δ.
4) Remove from the network all the agents belonging to such clusters, i.e. delete the corresponding rows and columns from the matrix T. Let $\mathcal{C} \leftarrow \mathcal{C} \setminus \{C_j : \bar{r}_j \le \delta\}$, $k = |\mathcal{C}|$.
5) Repeat from step 1 and stop when all clusters in $\mathcal{C}$ have a reputation greater than δ.

C. COMPUTATIONAL ISSUES

We highlight that the major computational load in our algorithm is associated with solving the system of linear equations 3. Our code uses the MATLAB function eig to solve it by finding the eigenvector corresponding to

the eigenvalue 1, with a computational cost that is $O(n^3)$. In our approach, we just refer to a centralized architecture, that requires a re-computation of the reputation values whenever some users are added or removed from the system. However, we note that on large-scale communities centralized architectures are more expensive than distributed ones in terms of costs for data storage, retrieval tasks, computational and communication overheads.Differently, distributed architectures may perform better than centralized approaches in terms of computational costs, although they are more complex to implement and more vulnerable to malicious attacks and inconsistencies (e.g., users may perceive the trustworthiness of a target differently). It is of course possible to implement the algorithm in a distributed architecture fashion, to improve efficiency, but this is out of the scope of our paper.

IV. AN EXPLANATORY EXAMPLE

We propose an example to illustrate our approach in the particular situation of a social network of 8 agents. We consider the original trust matrix

$$T = \begin{pmatrix} 0 & 0.9 & 0.7 & 0.7 & 0.2 & 0.2 & 0.8 & 0.9 \\ 0.9 & 0 & 0.8 & 0.7 & 0.3 & 0.1 & 0.9 & 0.8 \\ 0.25 & 0.25 & 0 & 0.9 & 0.95 & 0.95 & 0.2 & 0.2 \\ 0.25 & 0.25 & 0.9 & 0 & 0.95 & 0.95 & 0.15 & 0.2 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0 & 0.9 & 0.3 & 0.25 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.9 & 0 & 0.25 & 0.3 \\ 0.4 & 0.4 & 0.4 & 0.4 & 0.4 & 0.4 & 0 & 0.4 \\ 0.4 & 0.4 & 0.4 & 0.4 & 0.4 & 0.4 & 0.4 & 0 \end{pmatrix}.$$

The suspected malicious colluding group are the ones corresponding to the agents $\{3, 4\}$ and $\{5, 6\}$.

We assume that $t_{ii} = 0, \forall i = 1, \ldots, 8$, so that the trust assigned by each member to itself is not considered and compute similarity matrix $\tilde{A}$.

$$\tilde{A} = \begin{pmatrix} 0 & 9 & 0.86 & 0.86 & 0.75 & 0.75 & 1.2 & 1.08 \\ 9 & 0 & 0.8 & 0.86 & 0.66 & 1 & 1.08 & 1.2 \\ 0.86 & 0.8 & 0 & 9 & 0.55 & 0.55 & 1 & 1 \\ 0.86 & 0.86 & 9 & 0 & 0.55 & 0.55 & 0.78 & 1 \\ 0.75 & 0.66 & 0.55 & 0.55 & 0 & 9 & 1.75 & 1.3 \\ 0.75 & 1 & 0.55 & 0.55 & 9 & 0 & 1.3 & 1.75 \\ 1.2 & 1.08 & 1 & 0.78 & 1.75 & 1.3 & 0 & 4 \\ 1.08 & 1.2 & 1 & 1 & 1.3 & 1.75 & 4 & 0 \end{pmatrix}.$$

We apply the spectral clustering procedure, after choosing the appropriate number k of clusters, in this case $k = 4$.

Agents $i = 1, \ldots, 8$ are clustered as follows:

$$C_1 = \{1, 2\},\ C_2 = \{3, 4\},\ C_3 = \{5, 6\},\ C_4 = \{7, 8\}.$$

By computing the reputation score of users, we obtain the average reputation $\tilde{r}(C_k)$ associated with each cluster:

$$\tilde{r}(C_1) = 0.20,\ \tilde{r}(C_2) = 0.12,\ \tilde{r}(C_3) = 0.06,\ \tilde{r}(C_4) = 0.12$$

Fixing $\delta = 0.11$, we have $\tilde{r}(C_3) \leq \delta$, then the users 5 and 6 are classified as malicious colluding agents and removed from the network.

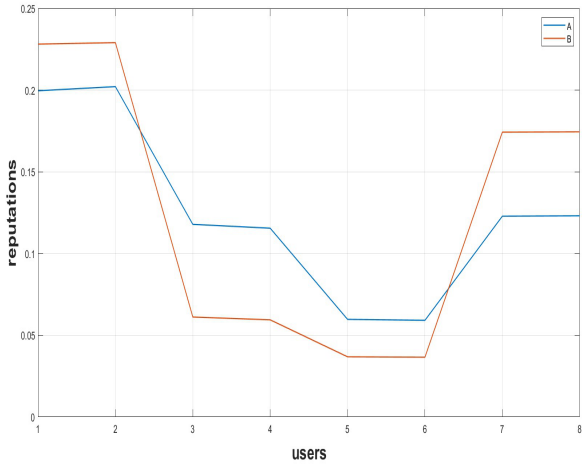

FIGURE 1. Reputations obtained by not using (A) and using (B) the information on the malicious users.

We recalculate the new reputation score of the remaining users and the new average reputation of the remaining clusters:

$$\tilde{r}(C_1) = 0.25 \quad \tilde{r}(C_2) = 0.10 \text{ and } \tilde{r}(C_4) = 0.14$$

We note that $\tilde{r}(C_2) \leq \delta$, so the users 3 and 4 are classified as malicious colluding agents and removed from the network. By iterating the process, we observe that the two remaining clusters (C_1 and C_4) have a average reputation greater than δ, so we stop the algorithm.

In Figure 1, we illustrate the results. In case A, no information on the malicious users is taken into account, while in case B we incorporated the information on the malicious users obtained as above.

It is evident from this toy example that the influence of malicious users on a social network significantly affects the overall reputations of its members. Recognizing this impact underscores the critical need for a method that effectively identifies malicious users in advance, before the computation of final reputations. This approach ensures the trustworthiness and reliability of the entire reputation assessment process.

V. EXPERIMENTAL RESULTS

We present the results of an experimental campaign of simulations for testing our method, comparing it with our previous algorithm. Our numerical code has been implemented by using MATLAB Release 2023a following the steps summarized in section III-B. Our numerical experiments were run on a 64-bit workstation with a Intel(R) Core(TM) i7-10875H CPU @ 2.30 GHz and 128 GB of RAM. We considered different cases, for different values of n, i.e. the dimension of the social network, and different ratios of malicious users in the network. For each case we have performed several simulations. As shown in Table 1 and Table 2, to evaluate the performance of our algorithm, we used the well-know metrics *precision*, *recall* and *F-score*,

making a comparison with the values obtained with the threshold strategy in our previous algorithm.

$$precision = \frac{TruePositives}{TruePositives + FalsePositives} \quad (7)$$

$$recall = \frac{TruePositives}{TruePositives + FalseNegatives} \quad (8)$$

$$F - score = \frac{2 * (Precision * Recall)}{Precision + Recall} \quad (9)$$

To clarify and make more accessible the fundamental concepts used in the above-mentioned evaluation metrics, remember that:

- *True Positives* are the positive elements that have been correctly classified as positive by the model, in our case the malicious colluding agents;
- *False Positives* are the negative elements that have been wrongly classified as positive by the model, in our case the honest agents;
- *True Negatives* are the negative elements that have been correctly classified as negative by the model;
- *False Negatives* are the positive elements that have been misclassified as negative by the model.

In particular, *precision* measures how much of the cases classified as positives are actually positives, while *recall* measures how much of the true positives have been identified by the model. Instead, the *F-score* provides a single number that balances *precision* and *recall*. This metric is particularly useful in situations where classes are unbalanced, as it takes into account both type I errors (false positives) and type II errors (false negatives). A high *F-score* indicates a good balance between *precision* and *recall*.

It is very clear that our method having a very high *recall* maintains the same effectiveness as Eigentrust in detecting malicious agents (also considering the improvement introduced by ER-EigenTrust). But the advantage is mainly in terms of *precision*: it is significantly better at avoiding false positives.

For example, as shown in the Figure 2, as n increases, leaving the percentage of malicious stable at 15%, the *precision* of Cluster Method is high, in many cases with values of 1 for small communities. Instead, the *precision* of ER-EigenTrust Method is low, varying with values between 0.27 and 0.55.

In Figure 3 we analyze the *precision* obtained in a social network with 500 users, varying the percentage of malicious from 5% to 25%. Regarding the Cluster Method, we note that as the percentage of maliciousness varies, the *precision* remains stable at high values, while as regards ER-EigenTrust Method, the *precision* is very low but increases as the percentage of malicious increase.

The experiments conducted on communities of users with medium-high dimensions produced highly satisfactory results that were very similar to the previous cases. The Cluster method also proved adaptable to large communities, showing high precision and recall both as the size of the community increased and as the percentage of malicious

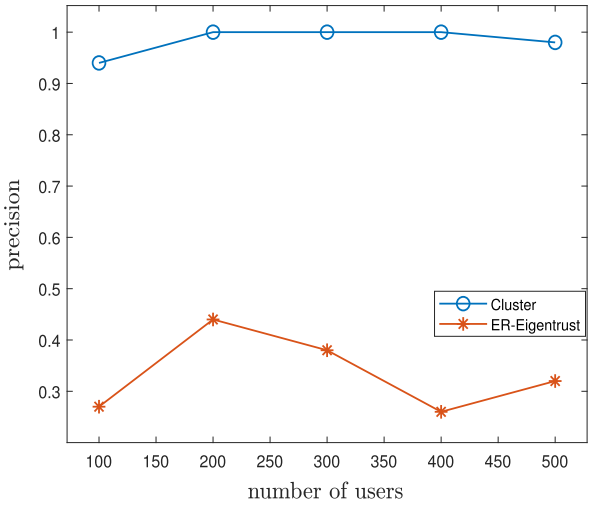

FIGURE 2. Precision of Cluster and ER-EigenTrust methods as users increase.

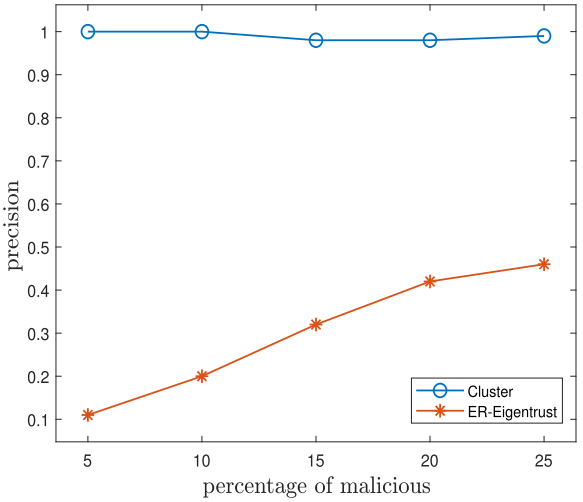

FIGURE 3. Precision of Cluster and ER-EigenTrust methods as malicious agents increase.

users increased. At the same time, the ER EigenTrust Method continues to prove effective especially in the presence of a high percentage of malicious users, showing a significant improvement in precision as the size of communities increases. As an example, in the case of 10000 agents, the precision of the Cluster method varies between 0.89 and 0.98 as the percentage of malicious users changes, while the precision of the ER EigenTrust method increases rapidly from 0.15 to 0.77. The execution times of the algorithms used to detect malicious users vary considerably. The Cluster method takes around 1700 seconds to complete the process while ER EigenTrust method takes just over 100 seconds. The significantly different execution times between the two detection methods should not be misinterpreted as a direct indicator of the overall efficiency of the two approaches. Despite the longer time involved, the cluster method demonstrates significantly higher precision in detecting malicious

TABLE 1. Results of common evaluation metrics for small communities.

n	% of malicious	Cluster method			ER-EigenTrust method		
		precision	*recall*	*F-score*	*precision*	*recall*	*F-score*
100	5%	1	1	1	0.13	1	0.23
	10%	1	1	1	0.2	1	0.3
	15%	0.94	1	0.96	0.27	1	0.42
	20%	1	0.8	0.88	0.34	1	0.51
	25%	1	0.91	0.95	0.38	1	0.55
200	5%	1	1	1	0.16	1	0.27
	10%	1	1	1	0.34	1	0.51
	15%	1	0.8	0.88	0.44	1	0.61
	20%	1	0.85	0.91	0.55	1	0.71
	25%	1	0.92	0.95	0.51	1	0.68
300	5%	1	0.87	0.93	0.12	1	0.21
	10%	1	0.8	0.88	0.19	1	0.32
	15%	1	0.82	0.9	0.38	1	0.55
	20%	1	0.83	0.9	0.4	1	0.57
	25%	1	0.82	0.9	0.57	1	0.73
400	5%	1	1	1	0.13	1	0.23
	10%	1	0.8	0.88	0.17	1	0.29
	15%	1	1	1	0.26	1	0.41
	20%	1	0.87	0.93	0.39	1	0.56
	25%	0.98	0.94	0.96	0.46	1	0.63
500	5%	1	0.92	0.96	0.11	1	0.20
	10%	1	0.88	0.93	0.2	1	0.33
	15%	0.98	0.97	0.98	0.32	1	0.49
	20%	0.98	0.91	0.94	0.42	1	0.60
	25%	0.99	0.91	0.95	0.46	1	0.63

TABLE 2. Results of common evaluation metrics for medium communities.

n	% of malicious	Cluster method			ER-EigenTrust method		
		precision	*recall*	*F-score*	*precision*	*recall*	*F-score*
1000	5%	1	1	1	0.16	1	0.28
	10%	1	0.96	0.97	0.25	1	0.40
	15%	0.98	1	0.99	0.31	1	0.47
	20%	0.96	0.89	0.92	0.42	1	0.59
	25%	0.96	0.94	0.95	0.51	1	0.68
2500	5%	0.92	1	0.96	0.14	1	0.25
	10%	0.96	0.99	0.97	0.27	1	0.42
	15%	0.97	0.98	0.98	0.34	1	0.51
	20%	0.96	0.66	0.78	0.28	1	0.43
	25%	0.95	0.95	0.93	0.61	1	0.54
5000	5%	0.92	1	0.96	0.13	1	0.23
	10%	0.96	0.98	0.97	0.26	1	0.42
	15%	0.97	0.99	0.98	0.37	1	0.54
	20%	0.96	0.85	0.90	0.78	1	0.87
	25%	0.93	0.59	0.72	0.55	1	0.71
7500	5%	0.94	0.99	0.97	0.14	1	0.24
	10%	0.97	0.99	0.98	0.26	1	0.42
	15%	0.98	0.97	0.97	0.4	1	0.57
	20%	0.98	0.88	0.92	0.86	1	0.92
	25%	0.99	0.89	0.94	0.87	1	0.93
10000	5%	0.89	1	0.94	0.15	1	0.27
	10%	0.98	0.98	0.98	0.27	1	0.43
	15%	0.98	0.97	0.97	0.35	1	0.52
	20%	0.93	0.37	0.53	0.75	1	0.86
	25%	0.97	0.74	0.84	0.77	1	0.87

users and in not generating false positives, underlining the trade-off between execution time and the accuracy of the results obtained.

The significant improvement in *precision* achieved through the application of the Cluster Method is attributable to its ability to aggregate malicious users into appropriate clusters and honest users into others. This subdivision facilitates a distinct disparity in average reputations between individual clusters, mitigating the risk of misclassifying honest users as malicious. This is particularly relevant since honest users are generally expected to maintain a higher level of reputation.

In contrast, the ER-EigenTrust Method directly classifies malicious users based on the trust matrix, determining that all users, including honest ones, who receive low trust ratings are tagged as malicious. This process leads to a situation in which any honest users who receive low trust ratings and,

consequently, possess a low reputation, are also wrongly identified as malicious. This procedure increases the number of false positives and ultimately reduces the *precision* of the model.

VI. CONCLUSION

In this paper, we have focused on the problem of detecting colluded agents in a social network and marginalize their weight in the community.

In this context, several TRRs have been proposed in the literature, and here we have examined the particular case represented by the well-known algorithm Eigentrust, that is recognized as one of the most effective solution to measure the reputation in a set of social agents. We had already faced, in a previous paper, two important problems affecting Eigentrust, i.e. the use of some additional information about agents that can be a-priori considered particularly trustworthy, and the strategy that EigenTrust uses of rewarding these trustworthy agents in terms of trust, while the other agents are penalized, producing the side effect to flattening the differences, in terms of reliability, between honest agents. However, we have highlighted as ER-EigenTrust, similarly to the original version of EigenTrust, presents an important limitation in terms of false positives that are generated when colluded agents are partitioned in different groups. We have proposed a new algorithm for detecting colluded agents, which combines EigenTrust with a clustering procedure, grouping agents based on their reputation scores. Our experimental campaign, described in the previous section, shows that our method, besides of maintaining the same effectiveness of Eigentrust in detecting malicious agents (also considering the improvement introduced by ER-EigenTrust), is significantly more capable of avoiding the presence of false positives. This advantage, in terms of precision, increases from a 77% achieved by ER-EigenTrust to a 97% reached by our proposed method an a social network having 10.000 users and a 25% of malicious agents, and this difference is even higher when the percentage of malicious agents is smaller.

It is important to highlight two current limitations of our approach: First, the reputation threshold under which a group should be considered as colluded is arbitrarily decided from the human administrator of the social network. Although it can be reasonable to take into account the personal opinion of the administrator about the level of reputation that a group of agents must have to be considered as colluded, we argue that some more deep considerations should be made about the possibility to help the administrator with additional information automatically extracted from the social network. Secondly, the clustering algorithm we have used needs as input the number of clusters to be formed. Currently, we are heuristically determining this number by an analysis of sensitivity, testing different values and using that value leading to the best results. However, an apposite procedure to automatically determining the reputation threshold in a more efficient way would increase the efficiency of the approach. We are currently studying both the aforementioned issues in our ongoing research and they are subject of our future work. Finally, it is important to highlight that the study we proposed carries out a first evaluation of the performances of our approach based on a simulation applied to small-medium sized social networks, which shows promising results in terms of improvement compared to our previous algorithm, but which needs to be extended to larger social networks and real data collections. Our ongoing research is focusing on fine-tuning applications of our new approach to scenarios involving multiple malicious groups and large social networks, a challenge further complicated by the absence of established benchmarks for such scenarios. Moreover, in our paper we have investigated only the sensitivity of our approach to the size of the social network and to the percentage of malicious. Our ongoing research is now devoted to also study the sensitivity of the approach to the number k of different clusters, and this will be a subject of our future work.

REFERENCES

[1] (2023). *Facebook*. [Online]. Available: http://www.facebook.com
[2] (2023). *Twitter*. [Online]. Available: http://www.x.com
[3] (2023). *Instagram*. [Online]. Available: http://www.instagram.com
[4] (2023). *Tiktok*. [Online]. Available: http://www.tiktok.com
[5] (2023). *Amazon*. [Online]. Available: http://www.amazon.com
[6] (2023). *Ebay*. [Online]. Available: http://www.ebay.com
[7] (2023). *Linkedin*. [Online]. Available: http://www.linkedin.com
[8] (2023). *ResearchGate*. [Online]. Available: http://www.researchgate.com
[9] Y. Rizk, M. Awad, and E. W. Tunstel, "Decision making in multiagent systems: A survey," *IEEE Trans. Cognit. Develop. Syst.*, vol. 10, no. 3, pp. 514–529, Sep. 2018.
[10] Y. Zhang, D. Shi, Y. Wu, Y. Zhang, L. Wang, and T. Xu, "Multi-agent feature learning and integration for mixed cooperative and competitive environment," in *Proc. IEEE 32nd Int. Conf. Tools Artif. Intell. (ICTAI)*, Nov. 2020, pp. 9–14.
[11] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in P2P networks," in *Proc. 12th Int. Conf. World Wide Web*, 2003, p. 640.
[12] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bring order to the web," Stanford Univ., Stanford, CA, USA, Technical 1998. [Online]. Available: http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf
[13] M. Cotronei, S. Giuffrè, A. Marciano, D. Rosaci, and G. M. L. Sarnè, "Improving the effectiveness of eigentrust in computing the reputation of social agents in presence of collusion," *Int. J. Neural Syst.*, vol. 34, no. 2, Feb. 2024, Art. no. 2350063.
[14] A. Dorri, S. S. Kanhere, and R. Jurdak, "Multi-agent systems: A survey," *IEEE Access*, vol. 6, pp. 28573–28593, 2018.
[15] Fabien Michel, Jacques Ferber, and Alexis Drogoul, "Multi-agent systems and simulation: A survey from the agent community's perspective," in *Multi-Agent Systems*. Boca Raton, FL, USA: CRC Press, 2018, pp. 17–66.
[16] P. E. Petruzzi, J. Pitt, and D. Busquets, "Electronic social capital for self-organising multi-agent systems," *ACM Trans. Auto. Adapt. Syst.*, vol. 12, no. 3, pp. 1–25, Sep. 2017.
[17] L. Shi, Z. Ma, S. Yan, and T. Ao, "Flocking dynamics for cooperation-antagonism multi-agent networks subject to limited communication resources," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 71, no. 3, pp. 1396–1405, Mar. 2024.
[18] W. Li, K. Qin, G. Li, M. Shi, and X. Zhang, "Robust bipartite tracking consensus of multi-agent systems via neural network combined with extended high-gain observer," *ISA Trans.*, vol. 136, pp. 31–45, May 2023.
[19] J. Liu, J. Shi, Y. Wu, X. Wang, and C. Sun, "Self-triggered predefined-time consensus tracking control of nonlinear multiple ground vehicles," *IEEE Trans. Veh. Technol.*, vol. 73, no. 12, pp. 18031–18042, Dec. 2024.
[20] A. Ahmed, K. Abu Bakar, M. I. Channa, K. Haseeb, and A. W. Khan, "A survey on trust based detection and isolation of malicious nodes in ad-hoc and sensor networks," *Frontiers Comput. Sci.*, vol. 9, no. 2, pp. 280–296, Apr. 2015.

[21] P. De Meo, F. Messina, M. N. Postorino, D. Rosaci, and G. M. L. Sarné, "A reputation framework to share resources into IoT-based environments," in *Proc. IEEE 14th Int. Conf. Netw., Sens. Control (ICNSC)*, May 2017, pp. 513–518.
[22] G. Fortino, F. Messina, D. Rosaci, and G. M. L. Sarne, "ResIoT: An IoT social framework resilient to malicious activities," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 5, pp. 1263–1278, Sep. 2020.
[23] H. Jnanamurthy and S. Singh, "Detection and filtering of collaborative malicious users in reputation system using quality repository approach," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Aug. 2013, pp. 466–471.
[24] S. M. Sajjad, S. H. Bouk, and M. Yousaf, "Neighbor node trust based intrusion detection system for WSN," *Proc. Comput. Sci.*, vol. 63, pp. 183–188, Apr. 2015.
[25] M. Akintunde, V. Yazdanpanah, A. S. Fathabadi, C. Cirstea, M. Dastani, and L. Moreau, "Formal specification of actual trust in multiagent systems," in *Frontiers in Artificial Intelligence and Applications*. Amsterdam, The Netherlands: IOS Press, 2024.
[26] R. Barbosa, R. Santos, and P. Nováis, "A trust model for informed agent collaboration in complex tasks," in *Proc. Sci. Inf. Conf.*, Jan. 2024, pp. 61–76.
[27] G. Alwhishi, J. Bentahar, A. Elwhishi, and W. Pedrycz, "MV-checker: A software tool for multi-valued model checking intelligent applications with trust and commitment," *Expert Syst. Appl.*, vol. 245, Jul. 2024, Art. no. 123113.
[28] A. Nazari, M. Kordabadi, and M. Mansoorizadeh, "Scalable and data-independent multi-agent recommender system using social networks analysis," *Int. J. Inf. Technol. Decis. Making*, vol. 23, no. 2, pp. 741–762, Mar. 2024.
[29] S. He, J. Chen, P. Zhang, and Z. Fu, "Multi-source trust model based on blockchain and IoT edge task collaboration," in *Proc. IEEE 49th Conf. Local Comput. Netw. (LCN)*, Oct. 2024, pp. 1–7.
[30] C. Dhasarathan, M. Shanmugam, M. Kumar, D. Tripathi, S. Khapre, and A. Shankar, "A nomadic multi-agent based privacy metrics for e-health care: A deep learning approach," *Multimedia Tools Appl.*, vol. 83, no. 3, pp. 7249–7272, Jan. 2024.
[31] Q. Shen and Y. Shen, "Endpoint security reinforcement via integrated zero-trust systems: A collaborative approach," *Comput. Secur.*, vol. 136, Jan. 2024, Art. no. 103537.
[32] W. Fang, W. Zhang, W. Chen, T. Pan, Y. Ni, and Y. Yang, "Trust-based attack and defense in wireless sensor networks: A survey," *Wireless Commun. Mobile Comput.*, vol. 2020, pp. 1–20, Sep. 2020.
[33] F. G. Mármol and G. M. Pérez, "Security threats scenarios in trust and reputation models for distributed systems," *Comput. Secur.*, vol. 28, no. 7, pp. 545–556, Oct. 2009.
[34] B. Pourghebleh, K. Wakil, and N. J. Navimipour, "A comprehensive study on the trust management techniques in the Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9326–9337, Dec. 2019.
[35] A. J. Bidgoly and B. T. Ladani, "Benchmarking reputation systems: A quantitative verification approach," *Comput. Hum. Behav.*, vol. 57, pp. 274–291, Apr. 2016.
[36] K. Hoffman, D. Zage, and C. Nita-Rotaru, "A survey of attack and defense techniques for reputation systems," *ACM Comput. Surveys*, vol. 42, no. 1, pp. 1–31, Dec. 2009.
[37] T. Grandison and M. Sloman, "Trust management tools for Internet applications," in *Proc. Int. Conf. Trust Manage.*, Jan. 2003, pp. 91–107.
[38] A. Jøsang and J. Golbeck, "Challenges for robust trust and reputation systems," in *Proc. 5th Int. Workshop Secur. Trust Manage.*, Jan. 2009, pp. 1–20.
[39] S. Vavilis, M. Petković, and N. Zannone, "A reference model for reputation systems," *Decis. Support Syst.*, vol. 61, pp. 147–154, May 2014.
[40] G. Lax and G. M. L. Sarné, "CellTrust: A reputation model for C2C commerce," *Electron. Commerce Res.*, vol. 8, no. 4, pp. 193–216, Dec. 2008.
[41] Y.-F. Wang, Y. Hori, and K. Sakurai, "Characterizing economic and social properties of trust and reputation systems in P2P environment," *J. Comput. Sci. Technol.*, vol. 23, no. 1, pp. 129–140, Jan. 2008.
[42] K. K. Fullam, T. B. Klos, G. M'uller, J. Sabater, A. Schlosser, Z. Topol, K. S. Barber, J. S. Rosenschein, L. Vercouter, and M. Voss, "A specification of the agent reputation and trust (ART) testbed: Experimentation and competition for trust in agent societies," in *Proc. 4th Int. Joint Conf. Auto. Agents Multiagent Syst.*, Jul. 2005, pp. 512–518.
[43] A. A. Adamopoulou and A. L. Symeonidis, "A simulation testbed for analyzing trust and reputation mechanisms in unreliable online markets," *Electron. Commerce Res. Appl.*, vol. 13, no. 5, pp. 368–386, Sep. 2014.
[44] R. Kerr and R. Cohen, "TREET: The trust and reputation experimentation and evaluation testbed," *Electron. Commerce Res.*, vol. 10, nos. 3–4, pp. 271–290, Dec. 2010.
[45] F. G. Marmol and G. M. Perez, "TRMSim-WSN, trust and reputation models simulator for wireless sensor networks," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2009, pp. 1–5.
[46] A. J. Bidgoly and B. T. Ladani, "Modelling and quantitative verification of reputation systems against malicious attackers," *Comput. J.*, vol. 58, no. 10, pp. 2567–2582, Oct. 2015.
[47] S. A. Ghasempouri and B. Tork Ladani, "Modeling trust and reputation systems in hostile environments," *Future Gener. Comput. Syst.*, vol. 99, pp. 571–592, Oct. 2019.
[48] S. C. Hayne, H. Wang, and L. Wang, "Modeling reputation as a time-series: Evaluating the risk of purchase decisions on eBay*," *Decis. Sci.*, vol. 46, no. 6, pp. 1077–1107, Dec. 2015.
[49] L. Cabral and A. Hortaçsu, "The dynamics of seller reputation: Evidence from eBay," *J. Ind. Econ.*, vol. 58, pp. 54–78, Jan. 2006.
[50] P. Resnick and R. Zeckhauser. (2002). *Trust Among Strangers in Internet Transactions: Empirical Analysis of Ebay's Reputation System* (The Economics of the Internet and E-commerce). Emerald Group Publishing Limited, 2002, pp. 127–157.
[51] L. Xiong and L. Liu, "PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 7, pp. 843–857, Jul. 2004.
[52] F. Messina, G. Pappalardo, D. Rosaci, C. Santoro, and G. M. L. Sarné, "A trust-aware, self-organizing system for large-scale federations of utility computing infrastructures," *Future Gener. Comput. Syst.*, vol. 56, pp. 77–94, Mar. 2016.
[53] T. D. Huynh, N. R. Jennings, and N. R. Shadbolt, "An integrated trust and reputation model for open multi-agent systems," *Auto. Agents Multi-Agent Syst.*, vol. 13, no. 2, pp. 119–154, Sep. 2006.
[54] A. Salehi-Abari and T. White, "DART: A distributed analysis of reputation and trust framework," *Comput. Intell.*, vol. 28, no. 4, pp. 642–682, Nov. 2012.
[55] R. Jansen, T. Kaminski, F. Korsakov, A. S. Croix, and D. Selifonov, "A priori trust vulnerabilities in eigentrust," Univ. Minnesota, Minneapolis, MN, USA, Tech. Rep., 2008. [Online]. Available: http://www-users.cs.umn.edu/jansen/papers/fet-csci5271
[56] Z. Abrams, R. McGrew, and S. Plotkin, "A non-manipulable trust system based on EigenTrust," *ACM SIGecom Exchanges*, vol. 5, no. 4, pp. 21–30, Jul. 2005.
[57] X. Fan, L. Liu, M. Li, and Z. Su, "EigenTrustp++: Attack resilient trust management," in *Proc. 8th Int. Conf. Collaborative Computing: Netw., Appl. Worksharing*, Oct. 2012, pp. 416–425.
[58] H. A. Kurdi, "HonestPeer: An enhanced EigenTrust algorithm for reputation management in P2P systems," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 27, no. 3, pp. 315–322, Jul. 2015.
[59] S. Gao, T. Yu, J. Zhu, and W. Cai, "T-PBFT: An EigenTrust-based practical Byzantine fault tolerance consensus algorithm," *China Commun.*, vol. 16, no. 12, pp. 111–123, Dec. 2019.
[60] K. Lu, J. Wang, and M. Li, "An eigentrust dynamic evolutionary model in P2P file-sharing systems," *peer-to-peer Netw. Appl.*, vol. 9, no. 3, pp. 599–612, May 2016.
[61] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, nos. 3–5, pp. 75–174, Feb. 2010.
[62] S. E. Schaeffer, "Graph clustering," *Comput. Sci. Rev.*, vol. 1, no. 1, pp. 27–64, Aug. 2007.
[63] F. D. Malliaros and M. Vazirgiannis, "Clustering and community detection in directed networks: A survey," *Phys. Rep.*, vol. 533, no. 4, pp. 95–142, Dec. 2013.
[64] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. 14th Int. Conf. Neural Inf. Process. Systems, Natural Synth.*, vol. 14, Jan. 2001, pp. 849–856.

MARIANTONIA COTRONEI received the Ph.D. degree in mathematics, in 1996. She is currently an Associate Professor of numerical analysis with the Department of Information Engineering, Infrastructures and Sustainable Energy, University of Mediterranea University of Reggio Calabria, Italy. Her research interests include approximation theory, wavelets, subdivision schemes, and signal and image processing.

SOFIA GIUFFRÈ received the Ph.D. degree in applied mathematics and computer sciences, in 2001. She is currently an Associate Professor of mathematical analysis at the Department of Information Engineering, Infrastructures and Sustainable Energy, Mediterranea University of Reggio Calabria, Italy. Her research interests include equilibrium problems, variational inequalities, infinite-dimensional duality theory, boundary value problems for linear and nonlinear elliptic, and parabolic systems with discontinuous coefficients.

ATTILIO MARCIANÒ received the Ph.D. degree in information engineering, in 2024. He is currently a Research Fellow with the Department of Information Engineering, Infrastructures and Sustainable Energy, Mediterranea University of Reggio Calabria, Italy. His research interests include equilibrium problems, variational inequalities, infinite-dimensional duality theory, and trust and reputation systems.

DOMENICO ROSACI received the Ph.D. degree in electronic engineering, in 1999. He is currently an Associate Professor of computer science with the Department of Information Engineering, Infrastructures and Sustainable Energy, Mediterranea University of Reggio Calabria, Italy. His research interests include distributed artificial intelligence, multi-agent systems, and trust and reputation in social communities. He is a member of a number of conference PCs. He is an Associate Editor of *Journal of Universal Computer Science* (Springer).

GIUSEPPE M. L. SARNÈ (Senior Member, IEEE) is currently an Associate Professor of computer science with the Department of Psychology, University of Milano-Bicocca, Italy. His main research interests include distributed artificial intelligence, multi-agent systems, and trust and reputation systems. He is a member of a number of conference PCs and the IEEE Technical Committee on Hyper-Intelligence. He is an Associate Editor of *Electronic Commerce Research and Applications* (Elsevier) and a member of the editorial board of *Big Data and Cognitive Computing* (MDPI).

Open Access funding provided by 'Univ Mediterranea di Reggio Calabria' within the CRUI CARE Agreement