Leveraging Node Properties in Random Walks for Robust Reputations in Decentralized Networks

Dimitra Gkorou
Delft University of Technology
Delft, the Netherlands
Email: D.Gkorou@tudelft.nl

Tamás Vinkó
University of Szeged
Szeged, Hungary
Email: tvinko@inf.u-szeged.hu

Johan Pouwelse and Dick Epema
Delft University of Technology
Delft, the Netherlands
Email: {J.A.Pouwelse,D.H.J.Epema}@tudelft.nl

***Abstract*—Reputation systems are essential to establish trust and to provide incentives for cooperation among users in decentralized networks. In these systems, the most widely used algorithms for computing reputations are based on random walks. However, in decentralized networks where nodes have only a partial view of the system, random walk-based algorithms can be easily exploited by uncooperative and malicious nodes. Traditionally, a random walk only uses information about the adjacency of nodes, and ignores their structural and temporal properties. Nevertheless, the properties of nodes indicate their reliability, and so, random walks using much richer information about the nodes than simple adjacency may achieve higher robustness against malicious exploitations. In this paper, we introduce the properties of nodes that are indicative of their reliability, and we propose a scheme to integrate these properties into the traditional random walks. Particularly, we consider two common malicious exploitations of random walks in decentralized networks, uncooperative nodes and Sybil attacks, and we show that integrating node properties into random walks results in much more robust reputation systems. Our experimental evaluation in synthetic graphs and graphs derived from real-world networks covering a significant number of users, shows the effectiveness of the resulting biased random walks.**

I. Introduction

Reputation systems establish trust and provide incentives for cooperation among users in many decentralized networks such as P2P networks [1], [2], distributed social networks [3], and markets on mobile devices [4]. These systems combine the history of node interactions to one reputation score for each node. For the computation of reputations, random walks constitute the core of the most widely used algorithms such as EigenTrust [5], PageRank [6], and TrustRank [7], because of their simple decentralization, their ability to take advantage of the sparsity of networks, and their computational efficiency. However, random walk-based algorithms can be easily exploited by uncooperative and malicious nodes using various self-serving and self-promoting strategies. Particularly in decentralized systems, nodes only have a partial view of the system and hence, their protection against uncooperative and malicious nodes is challenging. Traditionally, nodes visited during a random walk treat all their neighbors equally, ignoring any properties they may have. Nevertheless, properties of nodes, such as their age, may be indicative of their reliability, and thus, by integrating them into random walks, we can design more robust reputation systems. In this paper, we identify the properties of nodes that are indicative of their reliability, and we bias random walks towards the most reliable nodes.

Random walk-based reputation algorithms compute the reputation of a node as the probability of visiting that node in a random walk. In most implementations, random walks try to achieve resilience against uncooperative and malicious nodes based on a uniformly random selection of the next node to be visited. As a result, such *simple* random walks are vulnerable to many types of self-serving and self-promoting strategies [8], [9]. In a self-serving strategy, nodes abuse the system by first behaving properly for some time, and by then letting their reputations decrease in order to achieve a short-term gain [10]. We consider the most common form of self-serving misbehavior, which is a lack of cooperativeness as exhibited by free-riders, who passively abuse the system by consuming its resources without contributing to it. In a self-promoting strategy, nodes try to falsely increase their reputations using a variety of techniques such as web spamming and link farming [9], collusion [8], and Sybil attacks [11]. From the self-promoting strategies, we consider only the Sybil attack since most self-promoting strategies can be seen as special cases of it. In a Sybil attack, a malicious node boosts its reputation by controlling fake identities (its sybils), which report fake interactions with each other and with the malicious node.

In this paper, we show that the properties of a node indicate accurately its reliability, and that random walks exploiting these properties are more resilient than simple random walks. We model reputation systems in growing synthetic random and scale-free graphs, and in real-world graphs derived from the Bartercast reputation system [12] which is used in the BitTorrent client Tribler [13], from the citation network of Physical Review E journal, and from Facebook [14]. Each node in a graph initiates its own random walks and computes its own personalized reputations for the other nodes. Through an extensive analysis on our graphs, we introduce the node properties indicative of their behavior and we bias random walks with those properties. We observe that those properties depend on the characteristics and the construction of the graph.

In the case of uncooperative nodes, we evaluate biased random walks in growing graphs based on the observation that the ranking of nodes according to their reputations is more important than the actual values of reputations. In the case of sybil attacks, we evaluate the escape probability of a random

978-1-4799-0521-8/13/$31.00

walk to the sybil area with the number of the attack edges, since it has been already shown that the effectiveness of a sybil attack depends on this number [15]. Due to the size of our graphs, it is prohibitive to evaluate our biased random walks after the entry of each new node or edge and so, we use properly chosen time windows. Our experiments reveal that biased random walks are very robust in comparison with simple random walks, especially in real-world graphs.

II. Problem Statement

Random walk-based algorithms have been widely used for decentralized reputation systems since they have a low computational cost and resilience against noisy input [16]. However, they are vulnerable to malicious behaviors. Most implementations of random walks ignore the structural properties of nodes such as their centrality, clustering, and age, while these properties are indicative of their reliability. Our goal is to identify these properties of nodes and integrate them into random walks for building reputation systems that are more robust against exploitations. According to our approach, each node initiates its own random walks and we do not assume the existence of pre-trusted nodes. Therefore, our approach is suitable for decentralized networks.

A. Motivation

There are many exploitations of random walks in decentralized reputation systems. In this paper, we study the two most common exploitations: the uncooperative nodes and the Sybil attacks. Nevertheless, our method can be generalized to improve the robustness against other exploitations as well.

In reputation systems such as online markets or collaboration networks (e.g., eBay, eLance and Wikipedia), where nodes have to behave constantly according to the protocol, many users can be exploited by traitors or uncooperative nodes. Even simple uncooperative nodes can degrade significantly the system's performance. For instance, Sopcast, a P2P live streaming network, has on average around 87% uncooperative nodes degrading its performance [17]. Reputations predicting the behavior of nodes give incentives to the nodes to behave continuously according to the protocol. Computations of reputation predicting the behavior of nodes have been proposed in the context of Wikipedia authors [18], in P2P file-sharing systems [19] and P2P live streaming [17]. When using simple random walk, the predictive ability of the system is very low and some attempts to integrate properties of nodes have already improved it for the link prediction problem [20], [21]. In distributed systems and especially in unstructured P2P networks, biased random walks have been used for searching content [22], [23] but not in the context of reputation systems.

Reputation systems based on random walks are also sensitive to Sybil attacks [24]. Specially in the context of Pagerank, this observation is very common [25], [9]. Many users perform Sybil strategies, such as link farming [9] where fake links point to both the Sybils and the malicious node. In networks where the creation of links among nodes is easy, such as WWW and Facebook, random walk performs very poorly against Sybil attacks. Therefore, biased random walk with node properties indicating trust between two nodes decreases radically the effect of Sybil attack. A first study of the effect of biased random walks on algorithms against Sybil attacks explored the use of node similarity and strength of interactions [11]. Unlike [11], our study for sybil attacks evaluates the escape probability of random walks into the sybil area, and we use random walks with restarts biased with structural and temporal parameters.

B. Definitions and Network Model

We model a reputation network as a weighted directed graph $G = (V, E)$ whose vertices V correspond to the nodes, and whose edges E correspond to the interactions among nodes. A weighted edge $e_{ij} \in E$ connects two vertices $i, j \in V$ in the direction $i \to j$ with weight w_{ij}. Depending on the context, weights may represent the amount of data transferred across edges (in a P2P network), or the number of citations among authors (in a citation network). Computing the reputation of nodes with a random walk-based algorithm implies that the past interactions between nodes are interpreted as trustiness, in a similar way that links between web pages are interpreted as votes in a search engine like Google. The transition matrix P of a random walk in G is defined by $p_{ij} = w_{ij} / \sum_{k \in N_i} w_{ik}$, where N_i denotes the set of neighbors of node i.

We will use random walks with restarts [26], which means that each node visited by a random walk decides to direct the random walk back towards its initiator with the *teleportation probability* α. Then, the transition matrix becomes $P' = (1-\alpha)P + \alpha\mathbf{1}$, where $\mathbf{1}$ is the matrix with all its entries equal to 0 except for the elements of the column corresponding to the initiator, which are equal to 1. A random walk with restarts is *personalized* and represents better the inherent trust in a network, since each node trusts itself more than the other nodes and its trust towards the other nodes decreases with the increase of their distance. The vector π_i with the reputations computed by node i is the solution of the eigenvector equation $\pi_i = \pi_i P'$.

In an *unbiased random walk (simple RW)*, the weights w_{ij} represent simply the adjacency of the nodes in G, that is $w_{ij} = 1$ if $e_{ij} \in E$ and $w_{ij} = 0$ otherwise. In a *biased random walk (bRW)*, the weight w_{ij} is equal to some actual weight of the corresponding edge. In that case, we have to assign a weight w_{ij} to each edge e_{ij} in G in order to have a bRW visit more often the most reliable nodes. For each edge e_{ij}, we consider a vector ψ_{ij} with the values of properties of node j as perceived by node i, and we combine its elements to one weight $w_{ij} = f(\psi_{ij})$ for some function f. The function f can be defined by using the normalized product of the node properties or it can be learned in a supervised way. We refer to the former walk as *naive RW (nRW)* and to the latter walk as *supervised random walk (sRW)*.

For training our sRW, we use a slightly modified version of the method described in [21] adapted to our problem. We assume that the function f has an exponential form $f = \exp(u_i \psi_{ij})$, where u_i is the vector learned by node i

with the coefficients of ψ_{ij}. We formalize the problem of determining the vector u_i as a nonlinear optimization problem:

$$\min_{u_i} \|u_i\|^2 + \sum_{d \in D_i, l \in L_i} \frac{1}{1+\exp(-(\pi_i(l)-\pi_i(d)))}$$
$$\text{s.t.} \quad \pi_i(j) = \sum_{k \in V} \pi_i(k) P'_{kj} \quad (\forall j \in V),$$

where D_i and L_i are the sets of the top-30 best-behaved nodes and the top-30 worst-behaved nodes from the perspective of a node i, respectively. This objective function is highly multimodal, so an optimizer can easily get trapped in a local minimum. In order to avoid this, we let every node perform the following iterative process: we make only a small number of steps (up to 5) with the optimizer, then we compute the values of w_{ij} using the current value of u_i and solve the equation $\pi_i = \pi_i P'$ with power-iteration; using these u_i and π_i values as starting points we get back to the optimization problem again. We proceed such iterations until the solution vector u_i converges.

Our computation of reputations can be easily implemented in decentralized reputation systems where each node stores locally its own view of the reputation network, such as Bartercast [12], the system proposed by Piatek et al. [1], and MobID [3]. In these systems, when a node interacts with another node, they both store the weight of their interaction and the identity of the corresponding node. Nodes exchange information about their interactions using a gossip-like protocol. Based on its own interactions and the interactions gossiped about other nodes, each node builds locally its own partial view of the reputation network. Each node i performs the computation of π_i and the properties of other nodes on its own partial view.

For our analysis, we assume full-gossip in which nodes forward all their interactions, and eventually, their partial views converge to the global reputation network G. In a real system, the partial views of nodes may not convergence to G due to their resource limitations or high churn. Nevertheless, the reputations, as computed by random walks with restarts, are only slightly affected. In random walks with restarts, an interaction between two nodes occurring in the neighbourhood of the initiator of a random walk, contributes more on the computed reputations and gossip protocols propagate fast information in the neighbourhood of a node.

III. Datasets

In order to evaluate bRWs, we consider synthetic and real-world graphs which are defined below. When a graph is not connected, we proceed in our analysis using its largest weakly connected component.

Both our synthetic and real-world graphs grow over time. During the construction of the synthetic graphs, in each time step, with probability p_c a new node enters the system, or with probability $1-p_c$ already existing nodes interact and create new edges. The value of probability p_c depends on the dynamics of the system. Particularly, in highly dynamic systems the appearance of new nodes is dominant. For our synthetic graphs, we assume moderate system dynamics and so, we choose p_c equal to 0.5. Moreover, we allow the occurrence of multiple edges between a pair of nodes and we consider the number of occurrences of an edge as the weight of that edge. In real-world graphs, the addition of new nodes and edges is based on the timestamps available on the corresponding datasets and it is expressed in terms of actual time. In the synthetic graphs, no notion of actual time exists. For the construction of the synthetic graphs, time is divided into time steps during which new edges and nodes are added.

Table I: The diameter, the average path length (L) and the clustering coefficient (cc) of the largest weakly connected component of our graphs.

Graph	# Nodes	# Edges	Diameter	L	cc
Bartercast	$10,364$	$44,796$	13	2.64	0.00074
Citation	$31,238$	$110,638$	15	7.66	0.20
Facebook	$63,392$	$1,545,309$	15	4.32	0.15

A **random graph**, denoted by $R(n, p_r)$, is composed of n nodes, and each potential edge connecting two nodes occurs independently with probability p_r. We start from a single node, and in each time step, with probability p_c we add a node with each of its potential directed edges existing with probability p for some value of p, and with probability $1-p_c$ we add pn_t directed edges adjacent to existing nodes chosen uniformly at random. It has been shown that $p_r \sim p/2p_c$ [27]. In our experiments, we use a graph $R(5000, 0.02)$.

Scale-free graphs, denoted by $S(m)$, are characterized by their degree distribution following a power law. We create a growing directed scale-free graph based on the BA model [28]. We start with a small seeding connected triangular graph, and in each time step, with probability p_c we add a node with m directed edges. The end point of each of these edges is adjacent to an already existing node i with probability $\Pi(i) = d_i / \sum_j d_j$, where d_i is the degree of node i. With probability $1-p_c$ we add m directed edges, each of which is adjacent to an existing node i with probability $\Pi(i)$. One can show that S is scale-free with power-law exponent equal to $\gamma = 1+2/(2-p_c)$ [27]. For our evaluation, we use a graph $S(3)$ of 5000 nodes.

The **Bartercast graph**, denoted by B, is derived from the distributed reputation mechanism called Bartercast [12] of a BitTorrent-based client, Tribler [13]. The Tribler system was crawled from September 1, 2010 to January 31, 2011, collecting information from 29,716 nodes. As a deployed system, the Bartercast graph has a high population turnover, and so, the derived graph consists of a dense core with very few long living and active nodes and a periphery with many loosely connected nodes of low activity (small average path length and small clustering coefficient, see Table I).

The author-to-author **Citation graph**, denoted by C, is derived from the citation network of 32,584 papers published in Physical Review E from January 1993 to November 2011. Its vertices represent the authors of papers and edges represent the citation relationship between two authors (or coauthors). The weight of an edge indicates multiple citations from one author to another. In Table I, we can see that graph C exhibits small-world behavior because of its small average path length

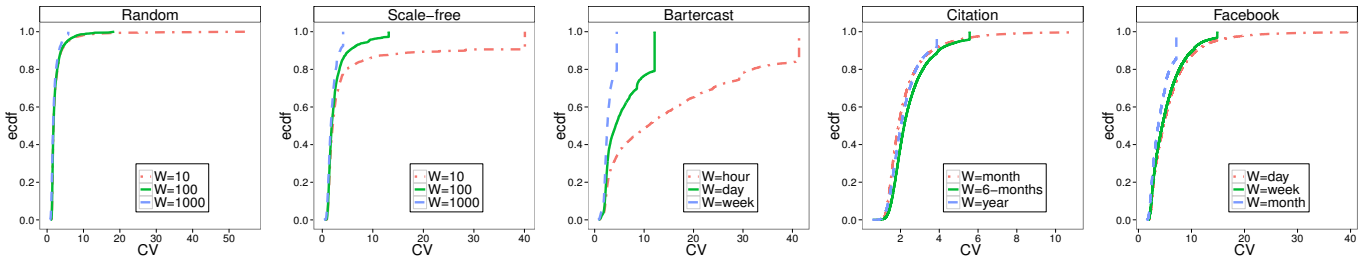


Figure 1: The ecdf of the coefficient of variation of the ranking stability of nodes over time for different time windows W[1].

and its large clustering coefficient. Its degree distribution has a power-law tail with exponent $\gamma = 2.55$. This dataset is available to us upon request to American Physical Society.

The **Facebook graph**, denoted by F, derives from the Facebook network in New Orleans with 63,732 users and it contains information about the interactions of users from September 26, 2006 to January 22, 2009 [14]. Its vertices represent Facebook's users and its edges represent friendships between two users. The weights of edges represent the number of interactions between two users and each edge has a timestamp indicating the time of this interaction. Graph F is a small-world graph like graph C (see Table I).

IV. Choosing the time window

Computing the properties of nodes such as centrality, clustering coefficient, and similarity, and the reputations of nodes in large graphs is very computationally intensive. Particularly in large growing graphs, updating those properties after every entry of a new edge or a new node is unrealistic. Fortunately, in most graphs neither the properties nor the reputations change very much with a small growth of the graph. Therefore, we choose an appropriate time window W for updating the properties and the reputations of nodes, so that we can reduce the cost of their update but we can still keep track of the dynamics of the reputations of nodes. Since we use personalized RWs, in principle, each node can use a different time window W according to its resources. For simplicity, however, we study the case that all nodes use the same value for W, but our method can be easily generalized for different durations of W across different nodes.

Usually in reputation systems, we are interested in the relative values of the reputations of the nodes rather than in their actual values. Therefore, as a metric for selecting a good value for W, instead of simply using reputations, we use the so-called *ranking stability* of nodes [16]. In order to compute this metric, we define the global reputation of a node as the average of its reputations computed by all the other nodes. Then, denoting the global reputation of node i at a certain time t in the evolution of the graph by $\pi(i)$, the ranking stability of node i at time t is defined as $(\pi(i) - \pi(j))/\sigma(\pi(i))$; here j is the node ranked immediately after node i in the ranking of nodes according to their decreasing reputation values at time t, and $\sigma(\pi(i))$ is the standard deviation of the set of values of node i's global reputation computed at different time instances up to and including time t. The rank of a node i is considered stable if its ranking stability is high.

In order to find the appropriate time window W, we choose a few different values of W and we keep track of the reputation and the ranking stability of each node over time as the graph grows. The reputation and the ranking stability of each node is recomputed at the end of every time window, that is, at the time points $t \cdot W$ for $t = 1, 2, \ldots$. Then, in order to observe the change of the ranking stability, we compute the *coefficient of variation* (CV) of the ranking stability of each node at these time points. A similar approach for computing the appropriate time window W has been used in [17] but that approach focused only on the actual reputation values and not on the ranking stability. The chosen W should result in a CV of the ranking stability that is neither too large nor too small, so that we are able to observe the dynamics of the ranking of reputations without needing to update them very often. For the Bartercast graph we evaluate W equal to one hour, one day, and one week, for the Citation graph one month, 6 months, and one year, and for the Facebook graph one day, one week, and one month. In the synthetic graphs, time is divided into time steps during which new edges and nodes are added (Section III), and we use W equal to 10, 100, and 1000 time steps.

In Figure 1, we present the empirical cdf (ecdf) of the CV of the ranking stability of the nodes for the chosen values of W. We observe that the ranking stability of the nodes in our graphs is sensitive to W, the shorter the window W, the higher the variation of the ranking stability of nodes. Moreover, a short duration of W implies a frequent update of the reputations of nodes, fluctuating ranking stability, and noisy observations. On the other hand, a larger duration of W makes our observations smoother because of the aggregative effect of node interactions.

In real-world graphs, the variation of the ranking of nodes is smaller than in synthetic graphs, which implies that the ranking of their nodes is more stable. The Citation graph has the lowest variation of the ranking because the creation of an edge between two nodes requires more time and effort in comparison with the other graphs. In the Bartercast and Facebook graphs, the variation of the ranking is closer to those of the synthetic graphs because its nodes interact easier and so, their ranking is more dynamic. The highest variation of the rankings is observed in random graphs because node interactions follow random patterns. As a result, there are no

[1] All our figures are better viewed in color

nodes having a relatively stable behavior over time and being able to stabilize their ranking. In this paper we choose W in such a way that the variation of the ranking is neither too small nor too large. Specifically, we choose W equal to one day for the Bartercast graph, to one year for the Citation graph, to one month for the Facebook graph, and to 100 edges for the random and scale-free graphs.

V. Identifying Properties of Nodes Indicative of their Behavior

In this section we define the behavior of nodes and we introduce the properties of nodes that are indicative of their future behavior. A reputation system whose calculated reputations predict the quality of future interactions reduces the effect of uncooperative nodes which do not contribute to the network resources without abusing the protocol. Such a reputation system needs to be able to predict the behavior of nodes and to rank higher the nodes with better future behavior.

A. Introducing the Properties of Nodes

We take the behavior of a node to be the difference between the resources it contributes to the network minus the network resources it consumes. We define the behavior $B(i,t)$ of a node i at time $t \cdot W$ as $B(i,t) = \sum_{j \in N_i}(s_{ji} - s_{ij})$, where the s_{ij} and s_{ji} are the strengths of the incoming and outgoing edges of node i at time $t \cdot W$. In the Bartercast graph, the strength of a link is the amount of data transferred across the link, and the behavior of a node corresponds to its cooperation level. In the Citation graph, the strength of an edge is the number of citations and in the Facebook graph, the strength of an edge is the number of interactions between two friends.

The properties of a node that may be predictive of its behavior can be divided into three categories based on the information needed for their computation: local, global, and temporal. The *local properties* can be naturally integrated in a RW since their computation does not need access to global information and they are computationally simple. The local properties of a node i that we use are:

- Its degree, which represents its activity.
- Its ego-betweenness centrality (ego-BC), which is its betweenness centrality in its ego-network, namely the network containing that node, its neighbours, and all the links among them [29].
- Its clustering coefficient, which is defined as the fraction of links among its neighbors that actually exist.

The computation of *global properties* demands high cost and global information. However, it is interesting to observe their predictive ability on the behavior of a node. The global properties of a node i that we use are:

- Its eigenvector centrality, whose basic idea is that interactions with highly reputed nodes contribute more to the reputation of a node.
- Its betweenness centrality (BC), which is defined as the sum of the fractions of shortest paths among all pairs in the graph that pass through this node and it indicates the amount of flow passing through that node.
- Its closeness centrality, which is the inverse of the sum of its distances from every other node in the network.

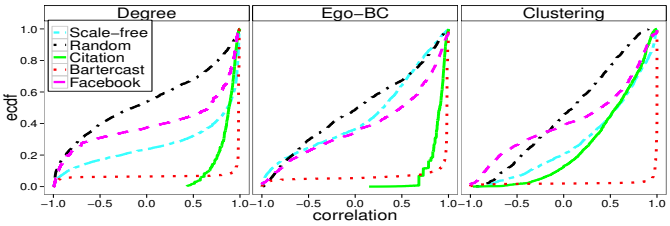
Finally, we use the *temporal properties* of node i to predict its behavior. Temporal properties require only local computation and can be easily integrated into RW. The temporal properties of a node i that we use are:

- Its average interaction time, which is the average time interval between successive interactions of node i.
- The time occurrence of the last interaction of a node i.
- Its age, which is expressed as $\tau(i) = t_c - t(i)$ where t_c is the current time and $t(i)$ is the time instance node i joined the system.

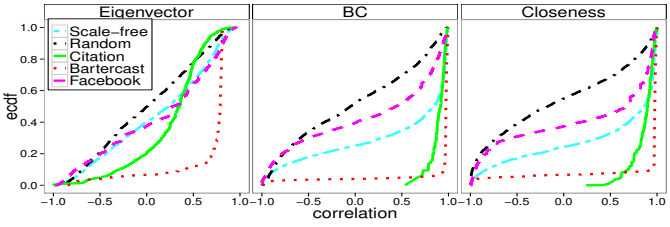
B. Evaluation

In order to assess to what extent a property of nodes is predictive of their future behavior, we compute the correlation between the node properties and the behavior of nodes over time as the graph grows. More precisely, for each node i and for each property, we compute the correlation between the sequence of values of that property of node i at time $t = 1, 2, \ldots$ and the sequence of values of its behavior at the next time step $B(i, t+1)$, for all t available from our datasets. For all the correlations, we use the Spearman correlation, which assesses the monotonic relationship between two sequences.

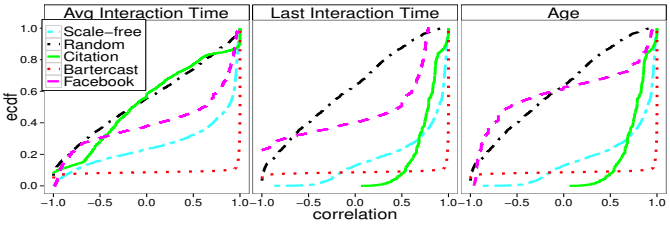
In Figure 2, we present the ecdf of these correlations for all the nodes in each graph. In our real-world graphs, the properties of a node are strongly correlated with its future behavior, particularly in Bartercast where the correlation is



(a) local properties



(b) global properties



(c) temporal properties

Figure 2: The ecdf of the correlations between the properties of each node and its future behavior over time.

Table II: The local, global and temporal properties of nodes exhibiting the highest correlation with their future behavior.

Graph	Local	Global	Temporal
Random	none	none	none
Scale-free	degree	BC, closeness	age
Citation	degree	BC	last interaction time
Bartercast	clustering	BC, closeness	all
Facebook	degree	closeness	avg interaction time

almost perfect. In these graphs, there are a few nodes attracting the majority of links. In Bartercast, these nodes are the nodes with high upload speeds that share many files, while in the Citation network, they are the authors of papers with high impact. As their degree, clustering coefficient, and centrality increase, these highly connected nodes improve their behavior as well. Nevertheless, temporal properties are also indicative of their future behavior because as has been observed in many real-world networks, the nodes gradually reduce their activity with time until they become inactive [30]. Facebook exhibits correlations similar to those in scale-free graphs.

In scale-free graphs, the future behavior of nodes is correlated mostly with their degree, BC, and age, due to the way they are constructed. In a scale-free graph, a new node connects with higher probability to nodes with high degrees, and so, a few older nodes obtain higher degrees and exhibit better behavior while the majority of nodes have much smaller degrees. In such graphs, the nodes with higher degree participate in the majority of the paths between the other nodes, and as a result they have high BC and high closeness centrality. Besides age, the other temporal properties are not correlated with the future behavior of nodes. In random graphs, all nodes have uniform connectivity and the interactions between the nodes are random. Therefore, there is almost no correlation between the properties of nodes and their future behavior.

In Table II, we present for all our graphs the properties of nodes having the highest correlations with their future behavior. For most graphs, the degree of nodes, even though it is the simplest local property, exhibits the highest correlation in comparison with the other local properties. Only for Bartercast, the clustering coefficient of nodes is more correlated with their future behavior because of its high churn. In Bartercast, a high clustering coefficient indicates that a node participates in the core of the network where its neighbors are active and interact with each other. A node having a low clustering coefficient is located in the periphery of the network. Nevertheless, in Bartercast also the degree of nodes predicts very well their future behavior.

The global properties of nodes that are based on shortest paths, namely BC and closeness, exhibit much higher correlations than eigenvector centrality which is based on random walks. In the Citation graph, the flow of information passing through an author influences his future connections. As a result, an author with high BC has a higher probability to contribute more in the network. In Facebook, a node within a short distance from other nodes has better access to their wall-post and vice versa. Therefore, this node having higher closeness centrality, has a higher probability to have a good behavior. In scale-free graphs and Bartercast, BC and closeness perform equally well. In these graphs, the nodes having high BC also have high closeness centrality because the clustering of these graphs is low. As a result, the nodes having many shortest paths passing through them are closer to the other nodes in the network. As our experiments show eigenvector centrality does not predict well future behavior of nodes.

The temporal property of nodes having the highest correlations depends on the construction process of each graph. In scale-free, the age of nodes predicts better their behavior since older nodes attract the majority of links. In Citation graph, the time of last interaction is more predictive because it indicates that an author is still active. In Facebook, the average time between two interactions performs better since it reveals the tendency of a node to participate in conversations. For Bartercast, all the temporal properties perform almost equally well. In a network with high churn like Bartercast, the nodes that stay for a long time in the system tend to interact more often with other nodes and contribute to the system. Thus, all the temporal properties of these nodes are equally good predictors of their behavior.

VI. Biasing Random Walks in the Face of Uncooperative Nodes

After having observed the correlations between the properties of nodes and their future behavior, we bias two types of RW with those properties: the naive RW (nRW) and the supervised RW (sRW). We assess to what extent the reputations computed by both types of bRWs predict the behavior of nodes and rank lower the nodes with bad future behavior.

A. Naive Random Walks

Naive RWs are implemented in a similar way as the simple RW but now the edge weights, and so the transition probabilities, depend on the node properties presented in Table II. The only additional cost of nRWs is the computation of these properties. We consider four types of nRWs: local nRW, global nRW, temporal nRW,and mixed nRW, in which we bias each walk with the corresponding local, global, temporal property of nodes, and with the combination of all these properties, respectively. In each case, the transition probabilities are proportional to the property of the targeted node. If according to Table II, more than one node properties correspond to a random walk, we chose the property with the lowest computational cost.

We evaluate whether the reputations of nodes predict their future behaviors, considering that in most reputation systems we are interested in the ranking of nodes according to their reputations. At each time $t \cdot W$, we compute the correlation between the sequence of the reputations of all the nodes in the graph and the sequence of their behaviors at the next time step $(t+1) \cdot W$, and we observe this correlation over consecutive time windows. We note that correlating the reputations at time t with the corresponding behaviors at time $t+2$ is equivalent

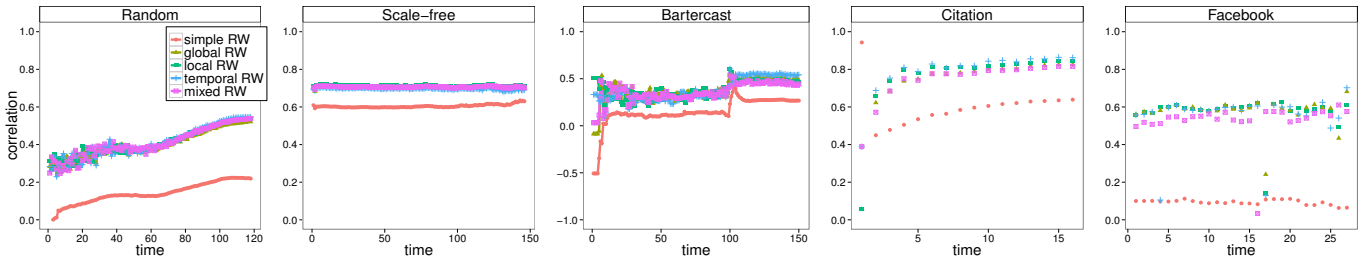


Figure 3: The correlation of the reputations of nodes as computed with naive Random Walks and their future behavior for consecutive time windows (note: the scale of the vertical axis of the Bartercast plot is different).

to choosing a W of double size. In Figure 3, we present the result of our evaluation for the random walks with teleportation probability $\alpha = 0.15$, a commonly used value for teleportation [6]. The presented result is the average of all the nodes. We found that the value of α does not affect much the correlation and so, we present only the values for $\alpha = 0.15$.

In all graphs in Figure 3, the reputations computed by the nRWs achieve much higher correlations with the future behaviors of nodes than the simple RW. Therefore, all nRWs are able to predict the nodes with the best future behavior. Nevertheless, the performance of the nRWs depends on the topology of the graph. In graphs such as scale-free, Citation and Facebook, where the creation of links follows specific patterns almost stable over time, all RWs exhibit higher correlations than in random graphs and Bartercast. Furthermore, temporal and global RWs exhibit similar correlations implying that the global and temporal properties of a node are highly dependent. For instance, in most cases, an old node with small average interaction time has high centrality.

Table III: The size of the intersection of the top-5 most highly reputed nodes at time $t \cdot W$ and the top-5 best behaved nodes at time $(t+1) \cdot W$ averaged over all t.

	simple RW	local RW	global RW	temporal RW	mixed RW
Random	0.64	0.95	0.91	0.42	0.94
Scale-free	3.64	4.87	4.75	2.37	4.79
Bartercast	2.15	3.22	2.94	2.67	3.16
Citation	3.36	4.64	4.23	3.77	4.58
Facebook	2.90	3.40	2.90	3.00	3.20

In many applications of reputation systems, such as recommendation of friends in Facebook or recommendation of papers in Citation graphs, we are interested only in the top ranked nodes. In Table III, we show the size of the intersection of the set of the top-5 most highly reputed nodes at time $t \cdot W$ and the set of the top-5 nodes with the best behavior at time $(t+1) \cdot W$, averaged for all t. In all graphs, the nRWs rank the top-5 nodes with the best future behavior higher than simple RW does, with local nRW achieving the highest number of common nodes and temporal nRW the smallest. In all graphs other than the random graph, the number of common nodes is high. In random graphs, the nodes follow a random pattern of interactions and so, we cannot predict accurately even the top-5 nodes with the best future behavior.

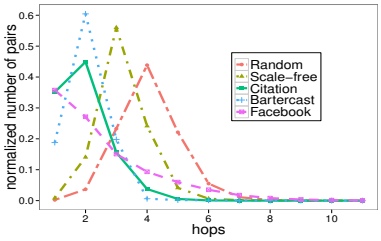


Figure 4: The distance in number of hops between a pair of nodes before they interact.

B. Supervised Random Walks

Although naive RWs are able to predict the best behaved nodes with high accuracy, the weights they use combine node properties into transition probabilities in a rather arbitrary way. For further evaluation of the ability of random walks to predict the best behaved nodes, we use supervised RWs (sRWs) where the weights assigned to each edge are learned and optimized during the previous time window. We compare sRW with simple RW and naive mixed RW. For each edge e_{ij}, we assume that the vector ψ_{ij} (see Section II-B) keeps all the properties of node j presented in Section V.

The computation of the optimal weights for sRW starting from a node i includes the computation of the vector u_i, which is the solution of the multimodal optimization problem presented in Section II-B. Due to its multimodality, this optimization problem is very computationally expensive and so, we need to further reduce the cost of computation. We observe that the vast majority of nodes in our graphs interact with other nodes that are only a few hops away. In Figure 4, we present the probability of interaction between two nodes in our graph as a function of their distance just before they interact. As we see, our graphs exhibit a high locality of interaction, which implies that we can reduce the cost of the computation of reputations by pruning the graph which is traversed by the random walks started at an initiator node without losing much on the performance. We observe that in all graphs but random graphs, more than 90% of pairs of interacting nodes have a distance of at most 3 hops just before they interact. Therefore, here, we use random walks with length of 3 hops.

In Figure 5, we present the correlation between the reputations of nodes as computed by RW, nRW, and sRW, and their future behavior. For sRW, we start the random walks from

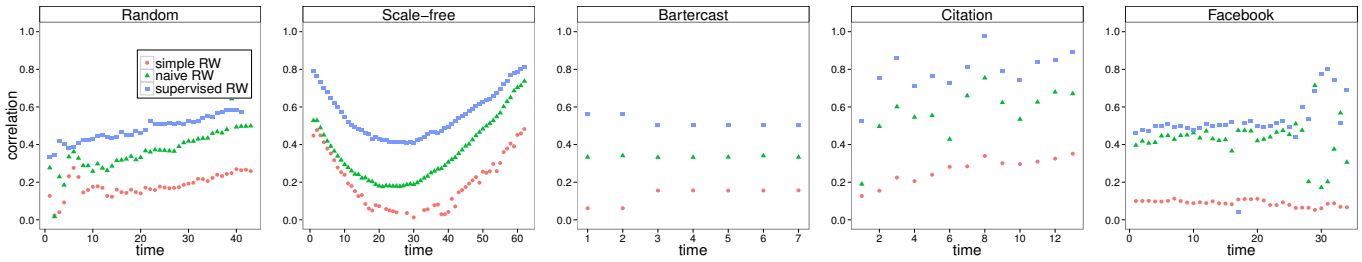


Figure 5: The correlation between the reputations of nodes as computed by simple, naive and supervised Random Walks and their future behavior for consecutive time windows.

Table IV: The size of the intersection of the top-5 most highly reputed nodes at time $t \cdot W$ and the top-5 best behaved nodes at time $(t+1) \cdot W$ averaged over all t.

	simple RW	naive RW	supervised RW
Random	0.09	0.24	0.26
Scale-free	2.5	3.10	3.14
Bartercast	1.75	3.00	3.00
Citation	3.18	4.21	4.22
Facebook	1.24	1.53	2.14

the two most well connected nodes in each graph, due to the high computational complexity of sRW. Our sRW outperforms nRW and RW in all graphs. However, the computational cost of sRW is much higher than nRW.

In Table IV, we present the size of the intersection of the set of the top-5 most highly reputed nodes and the set of the top-5 best behaved nodes in the next time window averaged over all consecutive time windows. In most graphs, the sRW identifies the best behaved nodes only slightly better than nRW does. Therefore, if we are only interested in the top ranked nodes, nRW constitutes a good compromise between accuracy and computational cost.

VII. Biasing Random Walks in the Face of Sybil Attacks

In this section, we bias RWs with node properties in order to increase their resilience against sybil attacks. Our aim is to make RWs stay away from malicious nodes and sybils so that the reputations assigned to such nodes are low. We bias only nRWs because for sRWs we cannot have a meaningful training set [31], since we have not observed any sybil attack in our datasets. Nevertheless, our experimental evaluation shows that even nRWs drastically reduce the effect of sybil attacks.

Most of the schemes proposed against sybils attacks in the literature [15], [32] are based on the observation that the sybil nodes can create only a limited number of edges to honest nodes because interacting with honest nodes requires a high social engineering cost [33]. As a result, the honest nodes form a region that is well separated from the sybil region containing the sybil nodes. The sybil nodes connect with each other and with the malicious nodes in an arbitrary way. The two regions are connected by the *attack edges* that link nodes in the sybil region to *victim* nodes in the honest region. The probability that an RW escapes to the sybil region depends on the number of attack edges and the visit ratios of the RW to the victims, but not on the topological characteristics of the sybil region [15]. In our experiments, we take as the honest region our initial graph $G = (V, E)$. Since the topology of the sybil graph is not important, we create a sybil graph $G_s = (V_s, E_s)$ using the BA model [28]. Then, we chose some sybil nodes from G_s and some prespecified victim nodes from G, and connect them through the corresponding attack edges E_a. The resulting graph is $G' = (V', E')$ where $V' = \{V \cup V_s\}$ and $E' = E \cup E_s \cup E_a$. To chose the victim nodes in G we use two approaches. Either the victims are chosen uniformly at random, or the malicious nodes try to increase their impact and attack highly reputed nodes by choosing the victims with probabilities proportional to their reputations. The latter selection of victims is also known as centrality attack [34].

The properties of nodes used to bias RWs must not depend on the topological properties of the sybil region. Therefore, we do not use global properties of nodes, but we use the following *local* and the *temporal properties*:

- The similarity of two nodes i and j with neighborhoods N_i and N_j, respectively, defined by the Jaccard similarity ($|N_i \cap N_j|/|N_i \cup N_j|$), which assumes that two nodes are similar if they have many common neighbors.
- The weight of an edge e_{ij} connecting two nodes i and j, indicating the strength of the corresponding interaction, as mentioned in Section II-B.
- The inverse log-weighted similarity between two nodes i and j, defined as the number of their common neighbors weighted by the inverse logarithm of their degrees ($\sum_{k \in |N_i \cap N_j|} (1/\log[d(k)])$, where $d(k)$ is the degree of node k). It assumes that two nodes are similar if they have low-degree common neighbors [35].
- The time t_{ij} that an edge e_{ij} is created.

The nodes in the sybil region can claim any values for these properties without affecting the probability of a RW escaping from an honest node to the sybil region. Since in order to escape to the sybil region, the RW has to traverse an attack edge, the properties of the nodes adjacent to the attack edges determine the probability that an RW escapes into the sybil region. We assume that it is more costly for an attacker to create an attack edge with a large weight than an attack edge of a low weight and so, attacks edges of low weights are more common. Therefore in our experiment, we assign

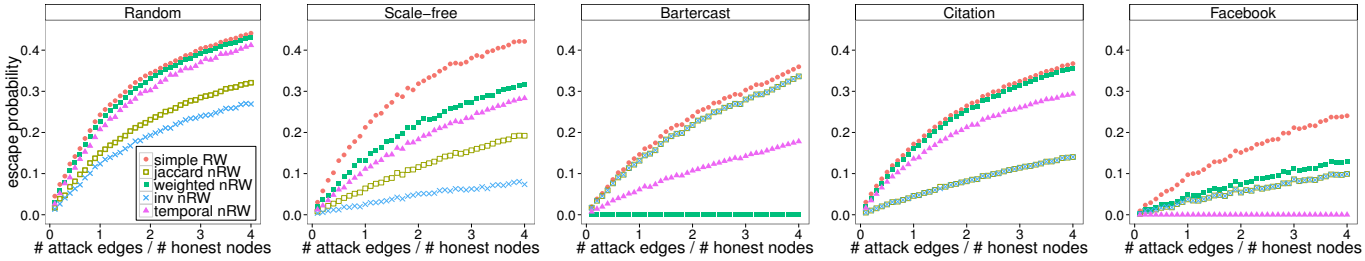


Figure 6: The escape probability to the sybil region of the simple and the biased Random Walks versus the ratio of the number of attack edges and the number of honest nodes when the victims are chosen uniformly at random.

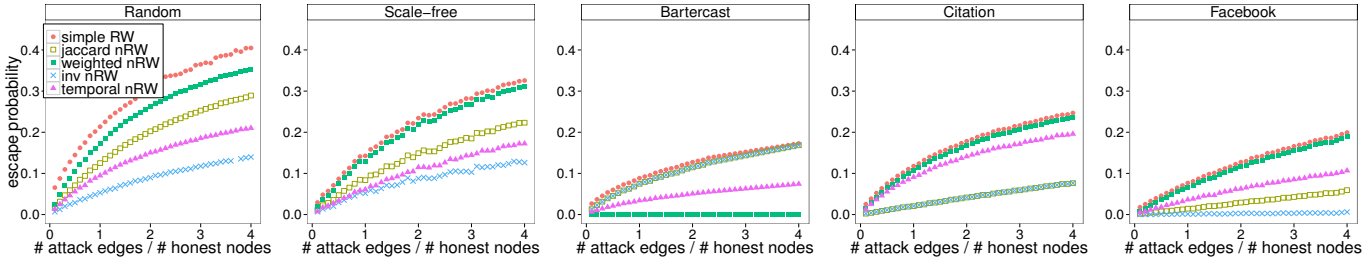


Figure 7: The escape probability to the sybil region of the simple and the biased Random Walks versus the ratio of the number of attack edges and the number of honest nodes when the victims are chosen with probabilities proportional to their reputations.

probabilistically a weight to each attack edge so that, attack edges with small weights are more common. For the time the attack edges have been created, we assume that they are uniformly distributed over time. We bias the nRW with each of the properties defined above, and we correspondingly have four types of nRWs: Jaccard nRW, weighted nRW, inverse nRW, and temporal nRW.

In Figure 6, we show the escape probability of the different RWs versus the ratio of the number of attack edges and the number of honest nodes when victims are chosen uniformly at random. Due to the large size of most of our graphs, the results are the average escape probability with 500 nodes performing the corresponding RW with teleportation parameter $\alpha = 0.15$. The effect of parameter α on the escape probability is not assessed in this paper. However, there is a first study on this effect in [11].

The real-world graphs where the honest nodes form a well connected region, have the smallest escape probability for all types of RWs, while the synthetic graphs have the largest. The type of nRW giving the smallest escape probability depends on the topology and the characteristics of the graph. In the random, scale-free, and Citation graphs, the weights take values in a small range and so, the weighted nRWs perform similarly to simple RW. In these graphs, inverse nRW results in the smallest escape probability, especially in the Citation graph, which has a large clustering coefficient indicating that nodes share many neighbors. On the other hand, in Bartercast, where the weights can vary from a few KB to several MB, the weight of an edge indicates accurately the trust between the interacting nodes and so, weighted nRW results in an escape probability that is almost zero, even though the number of attack edges is relatively high. On the contrary, due to its small clustering coefficient which is smaller than the corresponding random graph, the Jaccard and inverse nRWs result in escape probabilities similar to that of simple RW. Nevertheless, the Jaccard and inverse nRWs result in small escape probabilities for all the graphs but Bartercast. The temporal nRW performs better in Facebook, resulting in an escape probability that is almost zero because two nodes with many fresh interactions between them trust each other.

In Figure 7, we show the escape probabilities of different RWs when the victims are chosen with probabilities proportional to their reputations. Counter-intuitively, when malicious nodes use this targeted attack instead of randomly choosing victims, the escape probability is smaller for all types of RWs. The most highly reputed nodes attract the majority of the attack edges while they also have many edges from honest nodes connected to them. As a result, an RW visiting them has a lower probability to traverse an attack edge even though highly reputed nodes are visited with a higher probability by RWs. Nevertheless, in a random graph the difference between the impact of the two types of sybil attack on the escape probability is very small due to the homogeneity of its nodes. Furthermore, in all graphs, inverse nRW gives a smaller escape probability than Jaccard nRW because low-degree nodes usually have low reputations and are not targeted by the malicious nodes.

VIII. Concluding Discussion

Our evaluations indicate that using node properties improves a lot the resilience of RWs against uncooperative nodes and Sybil attacks. Concluding, our results imply the following.

First, the time window used for observing a graph depends on the characteristics of the graph. In graphs such as Citation,

where the creation of an edge requires large effort and time, the time window can be large, for example a year, and still follow accurately the dynamics of nodes. On the contrary, in graphs such as Bartercast graph, where the creation of edges is easier, we need a smaller time window, for example a day.

Secondly, the prediction of the behavior of nodes depends on the characteristics of the graph. Predicting the behavior of nodes is very accurate in graphs with both specific construction patterns and nodes with heterogenous properties, such as the scale-free, Citation and Facebook graphs. In graphs of nodes with uniform properties and highly dynamic behavior, biased random walks predict less accurately the behavior of nodes but still much better than simple RW.

Furthermore, the appropriate node properties to bias random walks against Sybils depend on the characteristics of the graph. In graphs with large clustering coefficient, such as our scale-free, Facebook and Citation, RW biased with node similarities, especially inverse log-weighted similarity, are very effective. In graphs with edges with heterogenous strengths, such as Bartercast and Facebook, biasing RW with the strength of an edge is very effective while using temporal properties is effective in graphs with strong temporal patterns.

In most of our graphs, random walks biased with very simple node properties with low computational cost such as, the degree, the weights, and the age, perform very well against uncooperative nodes or sybil strategies. Biasing random walks does not necessarily add a lot of extra computational cost and as a result, biased random walks can be easily used in decentralized systems where nodes have limited resources.

In this paper, we have shown that node properties enhance a lot the robustness of RW against exploitative nodes. Nevertheless in a distributed environment, nodes do not necessarily have access to the properties of other nodes, nor the information to compute them. Nodes can exchange their properties using a gossip-like protocol, but this is not reliable due to potential misreporting by some nodes. A reliable alternative is the use of a system like Bartercast [12], where the nodes store locally their own perception of the graph and then they can compute the properties of the nodes in their locally stored graph. Moreover, directing most of the RWs through nodes with particular properties results in overloading those nodes. This overload might cause even the failure of some highly reputed nodes and thus, it must be studied before adopting biased RW.

ACKNOWLEDGEMENT

This work was partially supported by the European Union and the European Social Fund through project FuturICT.hu (grant no.: TAMOP-4.2.2.C-11/1/KONV-2012-0013).

REFERENCES

[1] M. Piatek, T. Isdal, A. Krishnamurthy, and T. Anderson, "One hop reputations for peer to peer file sharing workloads," in *NSDI'08*, 2008.
[2] M. Feldman, K. Lai, I. Stoica, and J. Chuang, "Robust incentive techniques for peer-to-peer networks," in *ACM EC*, 2004.
[3] D. Quercia and S. Hailes, "Sybil attacks against mobile users: friends and foes to the rescue," in *INFOCOM*, 2010.
[4] R. Chakravorty, S. Agarwal, and S. Banerjee, "Mob: A mobile bazaar for wide-area wireless services," in *ACM MobiCom*, 2005.
[5] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in *WWW*, 2003.
[6] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," Technical Report 1999-66, 1999.
[7] Z. Gyongyi, H. Garcia-Molina, and J. Pedersen, "Combating web spam with trustrank," in *VLDB*, 2004.
[8] J. Hopcroft and D. Sheldon, "Manipulation-resistant reputations using hitting time," in *Conference on Algorithms and models for the web-graph*, 2007.
[9] Z. Gyongyi and H. Garcia-Molina, "Web spam taxonomy," in *AIRWeb*, 2005.
[10] K. Hoffman, D. Zage, and C. Nita-Rotaru, "A survey of attack and defense techniques for reputation systems," *ACM Comput. Surv.*, 2009.
[11] A. Mohaisen, H. N, and K. Y., "Keep your friends close: Incorporating trust into social network-based sybil defenses," in *INFOCOM*, 2011.
[12] R. Delaviz, N. Andrade, and J. A. Pouwelse, "Improving accuracy and coverage in an internet-deployed reputation mechanism," in *P2P*, 2010.
[13] J. A. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D. H. J. Epema, M. Reinders, M. R. van Steen, and H. J. Sips, "Tribler: a social-based peer-to-peer system," *Concurr. Comput.: Pract. Exper.*, 2008.
[14] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi, "On the evolution of user interaction in facebook," in *ACM WOSN*, 2009.
[15] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao, "Sybillimit: A near-optimal social network defense against sybil attacks," in *IEEE Symposium on Security and Privacy*, 2008.
[16] G. Ghoshal and A.-L. Barabási, "Ranking stability and super-stable nodes in complex networks," *Nature Communications*, 2011.
[17] G. D. Gonçalves, A. Guimarães, A. B. Vieira, Í. S. Cunha, and J. M. Almeida, "Using centrality metrics to predict peer cooperation in live streaming applications," in *IFIP Networking*, 2012.
[18] B. T. Adler and L. de Alfaro, "A content-driven reputation system for the wikipedia," in *WWW*, 2007.
[19] K. Walsh and E. G. Sirer, "Experience with an object reputation system for peer-to-peer filesharing," in *NSDI*, 2006.
[20] R.-H. Li, J. X. Yu, and J. Liu, "Link prediction: the power of maximal entropy random walk," in *CIKM*, 2011.
[21] L. Backstrom and J. Leskovec, "Supervised random walks: Predicting and recommending links in social networks," in *WSDM*, 2011.
[22] C. Gkantsidis and M. Mihail, "Hybrid search schemes for unstructured peer-to-peer networks," in *IEEE INFOCOM*, 2005.
[23] K. Ronasi, M. Firooz, M.-R. Pakravan, and A. Nasiri Avanaki, "An enhanced random-walk method for content locating in p2p networks," in *ICDCSW*, 2007.
[24] J. R. Douceur, "The Sybil attack," in *IPTPS*, 2002.
[25] A. Cheng and E. Friedman, "Manipulability of pagerank under sybil strategies," in *NetEcon*, 2006.
[26] H. Tong, C. Faloutsos, and J.-Y. Pan, "Fast random walk with restart and its applications," in *ICDM*, 2006.
[27] D. Gkorou, T. Vinkó, N. Chiluka, J. A. Pouwelse, and D. H. J. Epema, "Reducing the history in decentralized interaction-based reputation systems," in *IFIP Networking*, 2012.
[28] A. L. Barabási and R. Albert, "Emergence of Scaling in Random Networks," *Science*, pp. 509–512, 1999.
[29] M. Everett and S. P. Borgatti, "Ego network betweenness," *Social networks*, vol. 27, no. 1, pp. 31–38, 2005.
[30] L. A. N. Amaral, A. Scala, M. Barthélémy, and H. E. Stanley, "Classes of small-world networks," *PNAS*, 2000.
[31] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *IEEE Symposium on Security and Privacy*, 2010, pp. 305–316.
[32] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman, "Sybilguard: defending against sybil attacks via social networks," in *SIGCOMM*, 2006.
[33] B. Viswanath, A. Post, K. P. Gummadi, and A. Mislove, "An analysis of social network-based sybil defenses," in *SIGCOMM*, 2010.
[34] N. Chiluka, N. Andrade, D. Gkorou, and J. A. Pouwelse, "Personalizing eigentrust in the face of communities and centrality attack," in *AINA*, 2012.
[35] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Social Networks*, 2003.