Random Walk based Fake Account Detection in Online Social Networks

Jinyuan Jia, Binghui Wang, Neil Zhenqiang Gong
ECE Department, Iowa State University
{jinyuan, binghuiw, neilgong}@iastate.edu

***Abstract*—Online social networks are known to be vulnerable to the so-called *Sybil attack*, in which an attacker maintains massive fake accounts (also called Sybils) and uses them to perform various malicious activities. Therefore, Sybil detection is a fundamental security research problem in online social networks. Random walk based methods, which leverage the structure of an online social network to distribute reputation scores for users, have been demonstrated to be promising in certain real-world online social networks. In particular, random walk based methods have three desired features: they can have theoretically guaranteed performance for online social networks that have the fast-mixing property, they are accurate when the social network has strong homophily property, and they can be scalable to large-scale online social networks. However, existing random walk based methods suffer from several key limitations: 1) they can only leverage either labeled benign users or labeled Sybils, but not both, 2) they have limited detection accuracy for weak-homophily social networks, and 3) they are not robust to label noise in the training dataset.**

In this work, we propose a new random walk based Sybil detection method called SybilWalk. SybilWalk addresses the limitations of existing random walk based methods while maintaining their desired features. We perform both theoretical and empirical evaluations to compare SybilWalk with previous random walk based methods. Theoretically, for online social networks with the fast-mixing property, SybilWalk has a tighter asymptotical bound on the number of Sybils that are falsely accepted into the social network than all existing random walk based methods. Empirically, we compare SybilWalk with previous random walk based methods using both social networks with synthesized Sybils and a large-scale Twitter dataset with real Sybils. Our empirical results demonstrate that 1) SybilWalk is substantially more accurate than existing random walk based methods for weak-homophily social networks, 2) SybilWalk is substantially more robust to label noise than existing random walk based methods, and 3) SybilWalk is as scalable as the most efficient existing random walk based methods. In particular, on the Twitter dataset, SybilWalk achieves a false positive rate of 1.3% and a false negative rate of 17.3%.

I. Introduction

Online social networks (OSNs) are important platforms for people to interact with each other, to process information, and to diffuse social influence. For instance, Facebook owned 1.65 billion monthly active users as of April 2016 [1]. Moreover, according to Alexa (a web service ranking popularities of websites) [2], Facebook was the third most visited website, just below the giant search engine Google.com and video sharing site Youtube.com. However, OSNs–like many other distributed systems–are open to the so-called *Sybil attacks*. In a Sybil attack, an adversary registers and maintains massive fake (or Sybil) accounts, often using computer software. These Sybil accounts can subvert the security and privacy of OSNs. For instance, an attacker can use Sybils to manipulate presidential election and stock market via fake news [3, 4], as well as disseminate spams, phishing URLs, and malware [5]. Therefore, Sybil detection in OSNs is a fundamental and important security research problem.

Indeed, Sybil detection has attracted much attention from multiple research communities including dependable systems, cybersecurity, networking, as well as data mining. A particular category of Sybil detection methods [6–13] leverage the structure of an OSN and distribute reputation scores to users via *random walks*. We call these methods *random walk based methods*. For instance, SybilRank [10] distributes *benignness scores* from a set of labeled benign users to the rest of users via a random walk, while CIA [11] distributes *badness scores* from a set of labeled Sybil users to the rest of users via a random walk. These scores can be used to classify users to be benign or Sybil, or rank all users such that top-ranked users are more likely to be Sybil. In practice, OSN operators often hire human workers to manually inspect users and flag Sybils. The ranking can be used as a priority list to guide human workers to detect Sybils more efficiently. Specifically, a human worker can only inspect a limited number of Sybils within a given time period. Therefore, inspecting the top ranked users in the priority list can help human workers detect more Sybils within the same period of time.

Random walk based methods have demonstrated promising results in certain real-world OSNs [10, 12]. Specifically, random walk based methods have three promising features. First, for OSNs that have the fast-mixing property, some random walk based methods have theoretically guaranteed performance. For instance, SybilRank guarantees that the number of Sybils that are ranked lower than certain benign users is asymptotically bounded as $O(g \log n)$, where g is the number of attack edges (an edge is an attack edge if it connects a benign user and a Sybil user) and n is the number of users in the OSN. Second, for OSNs that have a strong *homophily* property, random walk based methods can accurately detect Sybils. An OSN has a strong homophily property if any two linked users are highly likely to have the same label. For instance, SybilRank [10] can accurately detect top-ranked Sybils in Tuenti, the largest OSN in Spain that has the homophily property. Third, state-of-the-art random walk based methods are scalable to large-scale OSNs.

However, existing random walk based methods suffer from several key limitations: 1) they only leverage either labeled benign users or labeled Sybils, but not both, 2) they have limited detection accuracy for weak-homophily social networks (many OSNs have weak homophily), and 3) they are not robust to label noise in the training dataset. For instance, we will demonstrate that SybilRank and CIA have limited detection accuracy on a Twitter network that has a weak homophily.

Our work: In this work, we propose SybilWalk, a new random walk based method, to perform Sybil detection in OSNs. SybilWalk overcomes the limitations of existing random walk based methods while maintaining their advantages. Specifically, given a social graph, we augment the graph with two additional nodes. The two nodes represent the two labels, i.e., benign or Sybil; and we call them *benign label node* (denoted as l_b) and *Sybil label node* (denoted as l_s), respectively. Given a training dataset, we create an edge between each labeled benign node and l_b, and we create an edge between each labeled Sybil node and l_s. Then, for each remaining node, we start a random walk from the node; and we treat the probability that this random walk reaches l_s before reaching l_b as the *badness score* for the node. A larger badness score indicates a higher likelihood of being a Sybil. Finally, SybilWalk uses the badness scores to classify users or rank them to be a priority list. The intuition of SybilWalk is that a node is more likely to be a Sybil if it is structurally closer to the labeled Sybils than the labeled benign nodes in the OSN.

Computing the badness scores defined by our SybilWalk is non-trivial. For instance, one way to compute the badness score for a node is to simulate r random walks that all start from the node. If r_s of them reach l_s before reaching l_b, then we can approximate the badness score as r_s/r. However, this method is inefficient, because 1) we often need to simulate a large number of random walks in order to obtain a confident approximate of the badness score, and 2) we need to simulate random walks for each node. To address the challenge, we design an iterative method to efficiently compute the badness scores. Our method computes the exact badness scores and computes them for all nodes simultaneously.

We compare SybilWalk with previous random walk based methods both theoretically and empirically. Theoretically, for OSNs that are fast mixing, we show that SybilWalk can bound the number of Sybils, whose badness scores are lower than certain benign nodes, to be $O(g \log n/d(s))$, where g is the number of attack edges, n is the number of users, and $d(s)$ is the average number of Sybils that a Sybil node is connected to. A larger $d(s)$ indicates more dense connections between Sybil nodes. In contrast, the tightest bound of existing random walk based methods is $O(g \log n)$ [7, 10]. Moreover, we demonstrate that SybilWalk has almost the same computational complexity as the most efficient existing random walk based methods.

Empirically, we compare SybilWalk with SybilRank and CIA using 1) social networks with synthesized Sybils and 2) a large-scale Twitter network with real Sybils. Our results demonstrate that 1) SybilWalk is substantially more accurate than SybilRank and CIA when the social network has weak homophily, 2) SybilWalk is substantially more robust to label noises than SybilRank and CIA, and 3) SybilWalk is as scalable as SybilRank and CIA. For instance, on the Twitter dataset, in the ranking list produced by SybilWalk, 99% of the top-80,000 nodes are Sybils. However, in the ranking lists produced by SybilRank and CIA, only 0.3% and 30% are Sybils, respectively.

In summary, our key contributions are as follows:

- We propose a new random walk based method called SybilWalk to detect Sybils in OSNs.
- We theoretically analyze the performance of SybilWalk. SybilWalk achieves a tighter bound on the number of falsely accepted Sybils than all existing random walk based methods.
- We empirically compare SybilWalk with existing random walk based methods on both social networks with synthesized Sybils and a Twitter dataset with real Sybils. Our results demonstrate that SybilWalk is more accurate and more robust to label noises than existing random walk based methods, while it is as scalable as the most efficient existing random walk based methods.

II. Related Work

A. Random Walk based Methods

Random walk based methods aim to leverage social structure [6–13]. The key intuition is that, although an attacker can control the connections between Sybils arbitrarily, it is harder for the attacker to manipulate the connections between benign nodes and Sybils, because such manipulation requires actions from benign nodes. Therefore, there is a structural gap between benign nodes and Sybils. Random walk based methods aim to leverage such structural gap.

Example random walk based methods include SybilGuard [6], SybilLimit [7], SybilInfer [8], SybilRank [10], Criminal account Inference Algorithm (CIA) [11], and Íntegro [12]. Specifically, SybilGuard [6] and SybilLimit [7] assume that it is easy for short random walks starting from a labeled benign user to quickly reach other benign users, while hard for short random walks starting from Sybils to reach benign users. SybilGuard and SybilLimit use the same random walk lengths for all nodes. SmartWalk [13] leverages machine learning classifiers to predict the appropriate random walk length for different nodes, and can improve the performance of SybilLimit via using the predicted (different) random walk length for each node. SybilInfer [8] combines random walks with Bayesian inference and Monte-Carlo sampling to directly detect the bottleneck cut between benign users and Sybils. SybilRank [10] uses short random walks to distribute benignness scores from a set of labeled benign users to all the remaining users. CIA [11] distributes badness scores from a set of labeled Sybils to other users. With a certain probability, CIA restarts the random walk from the initial probability distribution, which is assigned based on the set of labeled Sybils. Íntegro [12] improves SybilRank by first leveraging victim prediction (a victim is a user that connects to at least one Sybil) to assign weights to edges of a social network and then performing random walks on the weighted social network.

All existing random walk based methods require 1) the OSN (in particular the benign region) is *fast mixing*, which roughly means that a random walk in the OSN will converge to its stationary probability distribution quickly; and 2) the OSN has a strong *homophily* property, which means that if we sample an edge from the OSN uniformly at random, then the two corresponding nodes have the same label with a high probability. An OSN is said to have a weaker homophily if the two nodes of the sampled edge have the same label with a smaller probability. Íntegro further requires the number of victims to be small.

Existing random walk based methods suffer from several key limitations: 1) they can only leverage either labeled benign users or labeled Sybils, but not both, 2) they have limited detection accuracy for weak-homophily social networks, and 3) they are not robust to label noise in the training dataset. Specifically, SybilGuard, SybilLimit, SybilInfer, and SmartWalk only leverage one labeled benign node, making their performance limited [10] and them not robust to label noise. Moreover, they are not scalable to large-scale OSNs because they need to simulate a large number of random walks. SybilRank and Íntegro were successfully applied to detect a large amount of Sybils in Tuenti, the largest OSN in Spain. The reason of such success is that Tuenti has a strong homophily property [10]. However, they can only leverage the labeled benign users in the training dataset, limiting their performance in weak-homophily OSNs, as we will demonstrate in our experiments. CIA only leverages labeled Sybils. As we will demonstrate in our experiments, CIA also achieves limited performance for weak-homophily OSNs and is not robust to label noises. Our new random walk based method can tolerate a much weaker homophily and is more robust to label noises than existing ones.

Summary: Existing random walk based methods 1) can only leverage either labeled benign users or labeled Sybils, but not both, 2) have limited detection accuracy for weak-homophily social networks, and 3) are not robust to label noise in the training dataset.

B. Markov Random Fields based Methods

Markov Random Fields (MRF) based methods also leverage the structure of the OSN [14–17]. In particular, SybilBelief [14] associates a binary random variable with each node in the OSN; a random variable has a value of 1 if the corresponding node is Sybil, otherwise the random variable has a value of -1. Then, SybilBelief models the joint probability distribution of all these binary random variables as a pairwise Markov Random Field. Given a set of labeled benign nodes and (optionally) a set of labeled Sybil nodes, SybilBelief estimates the conditional probability of being Sybil for each node via the standard Loopy Belief Propagation (LBP) method [18]. The conditional probabilities are then used to detect Sybils. Gao et al. [15] and Fu et al. [16] demonstrated that SybilBelief can achieve better performance when learning the node and edge priors using local graph structure analysis. SybilBelief and its variants are not scalable. Moreover, they are iterative algorithms, but the iterative processes are not guaranteed to converge. The fundamental reason is that they rely on LBP to perform inference, which maintains messages on edges (so it is not scalable) and is not guaranteed to converge on loopy graphs [18]. Wang et al. [17] proposed SybilSCAR, a general framework to unify random walk based methods and MRF based methods. SybilSCAR is much more scalable than SybilBelief and is guaranteed to converge.

Summary: The key limitation of MRF based methods is that they do not have theoretical guarantees on the number of Sybils that can be falsely accepted into an OSN. We believe it is an interesting future work to generalize our theoretical analysis to derive guarantees for MRF based methods.

C. Other Methods

Other Sybil detection methods aim to leverage a user's content, a user's behavior, as well as a user's local graph structure (i.e., friends and connections between them) [5, 19–28]. For instance, contents could be tweets and hashtags on Twitter, news feeds and wall posts on Facebook, and clickstreams (e.g., a sequence of HTTP/HTTPS requests made by users). User behaviors could be the frequency of sending tweets on Twitter. These approaches span a variety of schemes, including blacklisting, whitelisting, URL filtering, as well as machine learning methods. In particular, most studies in this direction [5, 19–22, 24, 27] treat Sybil detection as a supervised learning problem; they extract various features from user-generated contents, behaviors, and local graph structure, and then learn machine learning classifiers using a training dataset consisting of a large number of labeled benign users and Sybils; the learnt classifiers are then used to predict the label (i.e., benign or Sybil) of each remaining user. The major challenge of these approaches is that attackers can mimic benign users and produce similar content, behavior, and local graph structure, making these methods less effective.

Summary: Attackers can easily evade the methods that use content, behavior, or local graph structure.

III. Problem Definition

We formally define our structure-based Sybil detection problem, introduce our design goals, and describe the threat model we consider in the paper.

A. Structure-based Sybil Detection

Suppose we are given an OSN $G = (V, E)$, where a node $v \in V$ represents a user and an edge $(u, v) \in E$ indicates a mutual relationship between u and v. For instance, on Facebook, an edge (u, v) could mean that u is in v's friend list and vice versa. On Twitter, an edge (u, v) could mean that u and v follow each other. In the OSN, each user has a *label*, which can be *benign* or *Sybil*. We say a user is *labeled* if we already know its label, e.g., via manual inspection, otherwise we say it is *unlabeled*. Labeled users form a *training dataset*. Our structure-based Sybil detection is defined as follows:

Definition 1 (Structure-based Sybil Detection): Suppose we are given a social network, and a training dataset consisting of some labeled benign nodes and some labeled Sybils. Structure-based Sybil detection is to predict the label of each remaining node by leveraging the structure of the social network.

B. Design Goals

We target a method that satisfies the following goals:

1) Leveraging both labeled benign users and labeled Sybils: OSN providers often have a set of labeled benign users and labeled Sybils. For instance, verified users on Twitter or Facebook can be treated as labeled benign users; users

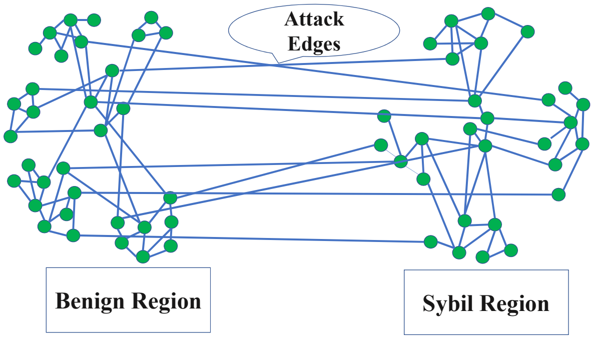


Fig. 1: Benign region, Sybil region, and attack edges.

spreading spam or malware can be treated as labeled Sybils, which can be obtained through manual inspection [10] or crowdsourcing [29]. Our method should be able to leverage both labeled benign users and labeled Sybils to enhance detection accuracy.

2) Robust to label noise: A given label of a user is noisy if it does not match the user's true label. Labeled users may have noisy labels. For instance, an adversary could compromise a labeled benign user or make a Sybil whitelisted as a benign user. In addition, labels obtained through manual inspection, especially crowdsourcing, often contain noises due to human mistakes [29]. We target a method that is robust when a minority fraction of given labels in the training dataset are incorrect.

3) Scalable: Real-world OSNs often have hundreds of millions of users and edges. Therefore, our method should be scalable and easily parallelizable.

4) Theoretical guarantee: Our method should have a theoretical guarantee on the number of Sybils that can be falsely accepted into an OSN. This theoretical guarantee is important for security-critical applications that leverage social networks, e.g., social network based Sybil defense in peer-to-peer and distributed systems [6], and social network based anonymous communications [30].

Existing random walk based methods SybilGuard [6] and SybilLimit [7] [8] do not satisfy requirements 1), 2), and 3). SybilInfer [8] satisfies none of these requirements. SybilRank [10] and Íntegro [12] do not satisfy requirements 1) and 2). CIA [11] does not satisfy requirements 1), 2), and 4).

C. Threat Model

We call the subgraph containing all benign nodes and edges between them the *benign region*, and call the subgraph containing all Sybil nodes and edges between them the *Sybil region*. Edges between the two regions are called *attack edges*. Fig. 1 illustrates these concepts. Note that both the benign region and the Sybil region can consist of multiple communities. Once we have a labeled node in each community, our method is able to detect Sybils accurately. We consider the following threat model.

One basic assumption under structure-based Sybil detection methods is that the benign region and the Sybil region are sparsely connected (i.e., the number of attack edges is relatively small), compared with the edges among the two regions. In other words, most benign users would not establish trust relationships with Sybils. We note that this assumption is equivalent to requiring that the OSNs have the *homophily* property, i.e., two linked nodes share the same label with a high probability. For an extreme example, if the benign region and the Sybil region are separated from each other, then the OSN has a perfect homophily, i.e., every two linked nodes have the same label. Note that, it is of great importance to obtain OSNs that satisfy this assumption, otherwise the detection accuracies of structure-based methods are limited. For instance, Yang et al. [28] showed that RenRen *friendship* social network does not satisfy this assumption, and thus the performance of structure-based methods are unsatisfactory. However, Cao et al. [10] found that Tuenti, the largest OSN in Spain, satisfies the homophily assumption, and SybilRank can detect a large amount of Sybils in Tuenti.

Generally speaking, there are two ways for OSN providers to construct a social network that satisfies homophily. One way is to approximately obtain trust relationships between users by looking into user interactions [31], predicting tie strength [32], asking users to rate their social contacts [33], etc. The other way is to preprocess the network structure so that structure-based methods are suitable to be applied. Specifically, human analysts could detect and remove compromised benign nodes (e.g., front peers) [34], or employ feature-based classifier to filter Sybils, so as to decrease the number of attack edges and enhance the homophily. For instance, Alvisi et al. [35] showed that if the attack edges are established randomly, simple feature-based classifiers are sufficient to enforce Sybils to be suitable for structure-based Sybil detection. We note that the reason why the RenRen friendship social network did not satisfy homophily in the study of Yang et al. is that RenRen even didn't deploy simple feature-based classifiers at that time [28].

Formally, we measure homophily as the fraction of edges in the OSN that are not attack edges. For the same benign region and Sybil region, more attack edges indicate weaker homophily. As we will demonstrate in our empirical evaluations, our SybilWalk can tolerate weaker homophily than existing random walk based methods, i.e., SybilWalk is more accurate than existing random walk based methods when the number of attack edges gets larger. This is because SybilWalk incorporates both labeled benign users and labeled Sybils in the training dataset via a novel random walk.

Apart from the homophily property, we also require the benign region to be fast mixing. We stress that fast mixing is not contradictory to community structure, i.e., having rich community structures does not necessarily mean slow mixing. Moreover, the fast mixing assumption is mainly used to derive SybilWalk's theoretical bound. In practice, SybilWalk still accurately detects Sybils even if the benign region is not fast mixing. Specifically, Mohaisen et al. [36] measured the mixing time for some OSNs and found that they have relatively large mixing time. Our SybilWalk is still accurate in such OSNs, once we have labeled nodes in each community in the training dataset.

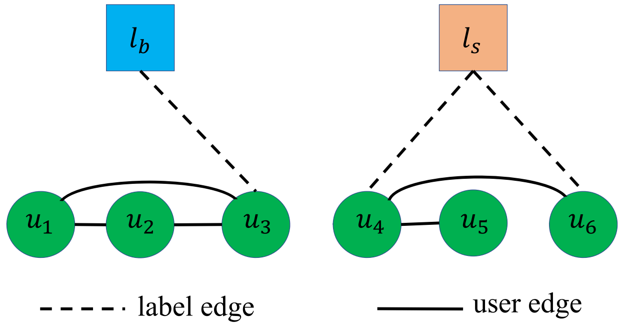


Fig. 2: An example of label-augmented social network.

IV. SybilWalk

We first introduce *label-augmented social network* to integrate labels and social network structure. Second, we define *badness score* for nodes using a novel random walk on the label-augmented social network. Third, we develop an iterative method to compute the badness scores efficiently. Fourth, we present a variant of SybilWalk.

A. Label-augmented Social Network

Leveraging random walks to incorporate both labeled benign nodes and labeled Sybils in the training dataset is challenging. For instance, no existing random walk based Sybil detection methods can incorporate both labels. To address this challenge, we design a *label-augmented social network* (LASN), on which we can gracefully incorporate both labels. Fig. 2 illustrates an example LASN. Specifically, we add two additional nodes into an existing social network; one node represents the label benign and the other node represents the label Sybil. We call the two nodes *benign label node* and *Sybil label node*, respectively. Moreover, we denote them as l_b and l_s, respectively. Then, given a training dataset, we create an edge between each labeled benign node and the benign label node l_b; and we create an edge between each labeled Sybil node and the Sybil label node l_s.

We call a node corresponding to a user *user node* and a node corresponding to a label *label node*. We call an edge between two user nodes *user edge*, while we call an edge between a user and a label node *label edge*. We can assign weights to different edges, which balance the importance of different edges. For instance, weights on user edges could be tie strengths, characterizing the closeness between two users. We use w_{uv} to represent the weight between nodes u and v. We note that our label-augmented social network can also be viewed as a Social-Attribute Network [37] or Social-Behavior-Attribute network [38], where the two label nodes are treated as attributes.

B. Defining Badness Scores Using Random Walks

The badness score of a node is the node's likelihood of being a Sybil. A larger badness score means that the node is more likely to be a Sybil. Intuitively, a node has a larger badness score if the node is structurally closer to the labeled Sybils than the labeled benign nodes among the social network. To capture such intuition, we define badness scores using random walks on the label-augmented social network.

Badness scores for label nodes: For the benign label node l_b, we define its badness score to be 0; and we define the badness score for the Sybil label node l_s to be 1.

Badness scores for user nodes: For a user u, we initiate a random walk from u and the random walk spreads among the label-augmented social network. We define the badness score of u as the probability that this random walk reaches l_s before reaching l_b. In particular, imagine we have a particle, which can stay on nodes of the label-augmented social network. Initially, the particle stays on u. In the next step, the random walk picks a neighbor v of u with a probability that is proportional to w_{uv}, and the particle moves the v. Formally, the particle moves to v with a probability $\frac{w_{uv}}{\sum_{t\in\Gamma_u} w_{ut}}$, where Γ_u is the set of neighbors of u. This pick-and-move process is repeated many times until the particle reaches either l_s or l_b. Since each pick-and-move is a random event, it is also random regarding whether the particle reaches l_s first or l_b first. However, if u is structurally closer to the labeled Sybils than the labeled benign nodes in the training dataset, then the particle in the random walk is more likely to first reach l_s than to first reach l_b. Therefore, our random walk based badness scores capture the structural information of the social network as well as incorporate both labeled benign nodes and labeled Sybil nodes.

C. Computing Badness Scores Using an Iterative Method

Computing our random walk based badness scores is nontrivial. For instance, one way to compute the badness score for a node u is to simulate r random walks that all start from the node. If r_s of them reach l_s before reaching l_b, then we can approximate the badness score as $\frac{r_s}{r}$. However, this method is inefficient, because 1) we often need to simulate a large number of random walks in order to obtain a confident approximate of the badness score, and 2) we need to simulate different random walks for each node. To address the challenge, we design an iterative method to efficiently compute the badness scores. Our method computes the *exact* badness scores and computes them for all users simultaneously.

Notations: We denote the badness score of a node u as p_u. We denote by Γ_u the set of neighbors of u. Moreover, we denote by d_u the weighted degree of u, i.e., $d_u = \sum_{v\in\Gamma_u} w_{uv}$, where w_{uv} is the weight of edge (u, v).

Representing a node's badness score using its neighbors' badness scores: We show that a node's badness score can be represented as a linear combination of its neighbors' badness scores. We first use an example to illustrate this linear relationship, and then we describe the relationship formally. Suppose we want to compute the badness score p_{u_1} of u_1 in the example label-augmented social network shown in Fig. 2. u_1 has two neighbors u_2 and u_3. Recall that u_1's badness score is the probability that a random walk, which starts from u_1, reaches the label node l_s before reaching the label node l_b.

Initially, the particle in the random walk stays on u_1. In the next step, the particle moves to a neighbor u_2 with a probability of $\frac{w_{u_1u_2}}{w_{u_1u_2}+w_{u_1u_3}}$, and the particle moves to u_3 with

Algorithm 1 SybilWalk

```
Input: A label-augmented social network, $\epsilon$, and $T$.
Output: $p_u$ for every user node $u$.
  Initialize $p_u^{(0)} = 0.5$ for every user node $u$.
  Initialize $p_{l_b}^{(0)} = 0$.
  Initialize $p_{l_s}^{(0)} = 1$.
  Initialize $t = 1$.
  while $\sum_u (p_u^{(t)} - p_u^{(t-1)})^2 \geq \epsilon$ and $t \leq T$ do
    for each user $u$ do
      $p_u^{(t)} = \sum_{v \in \Gamma_u} \frac{w_{uv}}{d_u} p_v^{(t-1)}$.
    end for
    $t = t + 1$.
  end while
  return $p_u$ for every $u$.
```

a probability of $\frac{w_{u_1 u_3}}{w_{u_1 u_2} + w_{u_1 u_3}}$. If the particle moves to u_2, then the probability that the particle reaches l_s before reaching l_b is p_{u_2}, the badness score of u_2. If the particle moves to u_3, then the probability that the particle reaches l_s before reaching l_b is p_{u_3}, the badness score of u_3. Therefore, we can represent u_1's badness score using u_2's and u_3's badness scores. Specifically, we have $p_{u_1} = \frac{w_{u_1 u_2}}{w_{u_1 u_2} + w_{u_1 u_3}} p_{u_2} + \frac{w_{u_1 u_3}}{w_{u_1 u_2} + w_{u_1 u_3}} p_{u_3}$. In other words, a node's badness score is a linear combination of its neighbors' badness scores. More formally, we have the following linear equation for each node u:

$$p_u = \sum_{v \in \Gamma_u} \frac{w_{uv}}{d_u} p_v. \quad (1)$$

Our SybilWalk algorithm: We leverage Equation 1 to design an iterative algorithm to compute the badness scores for all user nodes. We initialize the badness score of every user node u to be 0.5, i.e., $p_u^{(0)} = 0.5$. Note that the badness scores of the label nodes l_b and l_s are initialized and fixed to be 0 and 1, respectively. Then, in the tth iteration, we update the badness score for every user node as follows:

$$p_u^{(t)} = \sum_{v \in \Gamma_u} \frac{w_{uv}}{d_u} p_v^{(t-1)}. \quad (2)$$

The iterative process halts when the change of the badness scores of all user nodes in two consecutive iterations is smaller than a given small threshold ϵ (i.e., 10^{-3}) or the number of iterations has reached a predefined maximum number of iterations T. Algorithm 1 shows our SybilWalk algorithm.

D. A Variant of SybilWalk (SybilWalk-Var)

An alternative way to incorporate both labeled benign nodes and labeled Sybil nodes in the training dataset is to define the badness score of a node u as the probability that the random walk, which starts from u, reaches a labeled Sybil node before reaching any labeled benign node in the social network. This alternative formulation does not require the creation of the additional label nodes, and the random walks can be performed on the original social network.

We can adapt SybilWalk algorithm to compute such badness scores. Specifically, we initialize the badness score of every labeled benign node to be 0 and the badness score of every labeled Sybil node to be 1, and we fix the badness scores of these labeled nodes. Moreover, for each unlabeled user, we initialize its badness score to be 0.5. In each iteration, we apply Equation 2 to update the badness score of each unlabeled user. The process is repeated until the change of the badness scores of all unlabeled users in two consecutive iterations is smaller than a given small threshold (i.e., 10^{-3}) or the number of iterations has reached a predefined maximum number of iterations. We denote the adapted version of SybilWalk as SybilWalk-Var. We note that SybilWalk-Var can be viewed as a semi-supervised learning method, which is known as *label propagation* [39] in the machine learning community.

TABLE I: Summary of theoretical guarantees of various random walk based methods. g is the number of attack edges, n is the number of users in the social network, and $d(s)$ is the average node degree in the Sybil region. SybilGuard requires $g = o(\sqrt{n}/\log n)$. The symbol "–" means the corresponding bound is unknown.

Method	#Accepted Sybils
SybilGuard [6]	$O(g\sqrt{n}\log n)$
SybilLimit [7]	$O(g \log n)$
SybilInfer [8]	–
SybilRank [10]	$O(g \log n)$
CIA [11]	–
SybilWalk	$O(\frac{g \log n}{d(s)})$
SybilWalk-Var	$O(\frac{g \log n}{d(s)})$

In Section V, we will demonstrate that SybilWalk-Var has the same theoretical guarantees with SybilWalk. However, as we will show in our empirical evaluations in Section VI, SybilWalk is more accurate than SybilWalk-Var when the social network has a weaker homophily (i.e., the number of attack edges is larger). Moreover, SybilWalk is robust to a larger amount of label noises in the training dataset than SybilWalk-Var.

V. Theoretical Evaluation

We first analyze the ranking accuracy of SybilWalk and SybilWalk-Var. Then, we analyze their computational complexity.

A. Ranking Accuracy

Our theoretical guarantee of SybilWalk and SybilWalk-Var is summarized in the following theorem.

Theorem 1: Suppose the benign region is fast mixing and the attacker randomly establishes g attack edges. Then, the total number of Sybils whose badness scores are lower than certain benign nodes in SybilWalk (or SybilWalk-Var) is bounded by $O(\frac{g \log n}{d(s)})$, where n is the number of users in the social network and $d(s)$ is the average node degree in the Sybil region.

Proof: See Appendix A. ■

Our results imply that when Sybils are more densely connected among themselves (i.e., the average degree $d(s)$ is larger), it is easier for SybilWalk to detect them. We note that, when considering edge weights, the average node degree is the average weighted node degree. Table I summarizes the theoretical guarantees of various random walk based methods. For SybilRank, CIA, SybilWalk, and SybilWalk-Var, the metric *#accepted Sybils* means the number of Sybils that are ranked lower than certain benign nodes. For the rest of methods, #accepted Sybils means the number of Sybils are classified as benign. As we can see, our SybilWalk achieves the tightest bound on the number of accepted Sybils.

B. Computational Complexity

SybilWalk: Each iteration of SybilWalk traverses each edge in the label-augmented social network. Therefore, one iteration of SybilWalk has a time complexity of $O(m_u+m_l)$, where m_u is the number of edges between users and m_l is the number of edges between users and the label nodes. In other words, m_l is the number of labeled benign nodes and labeled Sybil nodes in the training dataset, since each labeled node has a label edge. Therefore, the total complexity of SybilWalk is $O(t(m_u + m_l))$, where t is the number of iterations.

SybilWalk-Var: Each iteration of SybilWalk-Var essentially traverses each edge in the original social network. Therefore, one iteration of SybilWalk has a time complexity of $O(m_u)$, and the total time complexity is $O(tm_u)$, where t is the number of iterations.

Both SybilRank and CIA have a time complexity of $O(tm_u)$, where t is the number of iterations. Although SybilWalk theoretically has a higher time complexity than SybilRank and CIA, we expect that they are almost the same efficient in practice. This is because the number of label edges m_l is negligible compared to the number of edges between users in practice. Indeed, our empirical evaluation results demonstrate that SybilWalk, SybilWalk-Var, CIA, and SybilRank have almost identical scalability. Other random walk based methods including SybilGuard, SybilLimit, and SybilInfer are known to be inefficient, because their time complexity is at least $O(n^2)$, where n is the number of users in the social network.

VI. Empirical Evaluation

We compare our methods with previous random walk based methods with respect to: 1) detection accuracy, 2) robustness to label noise, and 3) scalability.

A. Experimental Setup

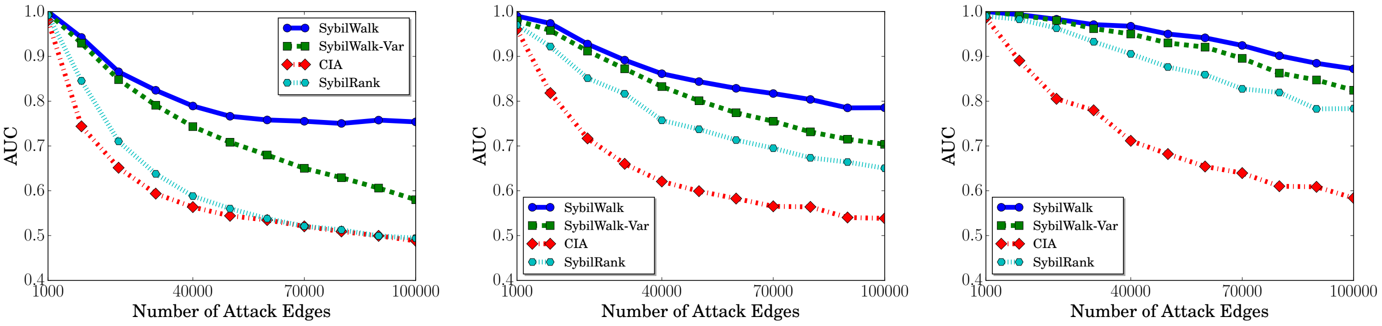
Datasets: We compare our methods with previous random walk based methods using 1) social networks with synthesized Sybils and 2) a Twitter dataset with real Sybils.

1) Social networks with synthesized Sybils: We use a real social graph as the benign region while synthesizing the Sybil region and adding attack edges between the two regions uniformly at random. There are different ways to synthesize the Sybil region. For instance, we can use a network model (e.g., Preferential Attachment model [40]) to generate a Sybil region. A Sybil region that is synthesized by a network model might be structurally very different from the benign region, e.g., although the Preferential Attachment model can generate graphs that have similar degree distribution with real social networks, the generated graphs have very small clustering coefficients, which is very different from real-world social networks. Such structural difference could bias Sybil detection results [35]. Moreover, a Sybil region synthesized by a network model like Preferential Attachment does not have community structures, making it unrealistic. Therefore, following recent studies [14, 35], we consider a Sybil attack in which the Sybil region is a replicate of the benign region. This way of synthesizing the Sybil region can avoid the structural difference between the two regions, and both Sybil region and benign region have complex community structures.

We utilize three social networks, i.e., Facebook (4,039 nodes and 88,234 edges), Enron (33,696 nodes and 180,811 edges), and Epinions (75,877 nodes and 811,478 edges), to represent different application scenarios. For each social network, we use it as the benign region; replicate it as a Sybil region; and then add attack edges uniformly at random. We obtained these datasets from SNAP (http://snap.stanford.edu/data/index.html). A node in Facebook dataset represents a user in Facebook, and two nodes are connected if they are friends. A node in Enron dataset represents an email address, and an edge between two nodes indicate at least one email was exchanged between the two corresponding email addresses. Epinions is a who-trust-whom online social network of a general consumer review site Epinions.com. The nodes in Epinions denote members of the site. And in order to maintain quality, Epinsons encourages users to specify which other users they trust, and uses the resulting web of the trust to order the product reviews seem by each person.

2) Twitter dataset with real Sybils: We obtained a directed Twitter graph from Kwak et al. [41]. In this graph, a directed edge (u, v) means that u follows v. We keep an undirected edge between two nodes if there are directed edge(s) between them. After processing, the dataset contains 41,652,230 nodes and 1,202,513,046 edges. To perform evaluation, we need the ground truth labels of the nodes. We obtained ground truth labels from Wang et al. [17]. Specifically, around 205,000 nodes were suspended by Twitter, which are treated as Sybils; around 36,157,000 nodes are still active, which are treated as benign nodes; and the remaining nodes were deleted, which are treated as unlabeled. The average number of attack edges per Sybil is 100. Therefore, this Twitter network has a very weak homophily.

Training and testing: Note that both the benign region and the Sybil region have community structures. To cope with community structure, for social networks with synthesized Sybils, we sample 100 nodes from the benign region uniformly at random and treat them as labeled benign nodes; and we sample 100 nodes from the Sybil region uniformly at random and treat them as labeled Sybil nodes. For the Twitter dataset with real Sybils, we sample 50,000 nodes from the benign region and 50,000 nodes from the Sybil region. This random sampling process is highly likely to have labeled nodes in each community. Once we have labeled nodes in each community, our methods can detect Sybils even if there are rich community structures. The training dataset consists of the randomly



(a) Facebook

(b) Enron

(c) Epinions

Fig. 3: AUCs of compared methods for different number of attack edges.

sampled nodes, and the rest of nodes are treated as testing dataset. Note that for the Twitter dataset, the unlabeled nodes are not included in the testing dataset.

Compared methods: We compare the following methods.

- **SybilRank [10]**. Given a training dataset, SybilRank only leverages labeled benign nodes to assign an initial probability distribution over the nodes of the social network. Then, SybilRank performs a random walk with the initial probability distribution. After a small number of iterations of the random walk, SybilRank normalizes probability for each node using its degree and treats the normalized probability as its benignness scores, which are used to rank test users in an increasing order. The normalization step is essential as shown by the authors of SybilRank.

- **CIA [11]**. Given a training dataset, CIA only leverages labeled Sybil nodes to assign an initial probability distribution over the nodes of the social network. Then, CIA performs a random walk with the initial probability distribution. In each step of the random walk, CIA restarts the random walk with the initial probability distribution with a certain probability, which is conventionally called *restart probability*. We set the restart probability to be 0.85 as suggested by the authors. After the random walk converges to its stationary probability distribution, the stationary probability of a node is treated as its badness score. Then, we can rank the test users decreasingly according to their badness scores. Note that SybilRank does not restart the random walk in each step.

- **SybilWalk-Var**. Variant of our SybilWalk. We set all edge weights to be 1.

- **SybilWalk**. We set weights of all edges in the constructed label-augmented social network to be 1. However, we believe learning edge weights is an interesting future work.

We do not compare with SybilGuard, SybilLimit, and SybilInfer because they are not scalable.

B. *Detection Accuracy*

AUCs on the social networks with synthesized Sybils: Each compared method produces a ranking list of test nodes, in which Sybils are supposed to rank higher than benign nodes. Area Under the Receiver Operating Characteristic Curve (AUC) is a standard metric to measure quality of a ranking method. In our case, AUC for a method is the probability that the method ranks a test Sybil node, which is sampled uniformly at random, higher than a test benign node, which is also sampled uniformly at random.

A higher AUC means a better ranking quality. AUC is 1 if all test Sybil nodes are ranked higher than all test benign nodes. AUC is 0 if all test benign nodes are ranked higher than all test Sybil nodes. A method that ranks the test nodes uniformly at random has an AUC of 0.5.

Fig. 3 shows AUCs of the compared methods as we increase the number of attack edges from 1,000 to 100,000. All methods have AUCs close to 1 when the number of attack edges is less than 1000. Therefore, we do not show those results in order to better contrast the results for large attack edges.

We observe that our random walk based methods substantially outperform previous random walk based methods when we have a large number of attack edges (i.e., the social networks have weak homophily). The improvements of our methods over previous ones are more significant as we have more attack edges. The reason is that our methods incorporate both labeled benign nodes and labeled Sybil nodes. Moreover, SybilWalk outperforms SybilWalk-Var, especially when the social networks have weak homophily. We speculate the reason is that real-world social networks often have some nodes with a large number of neighbors [42]; when such nodes are selected as training dataset, a random walk, which starts from any node, is more likely to reach such nodes than other labeled nodes; as a result, SybilWalk-Var's performance is significantly influenced by the labeled nodes with large degrees. In contrast, SybilWalk avoids the influence of labeled nodes with large node degrees via augmenting the social network with label nodes, and defining the badness score as the probability of the random walk reaching the label nodes.

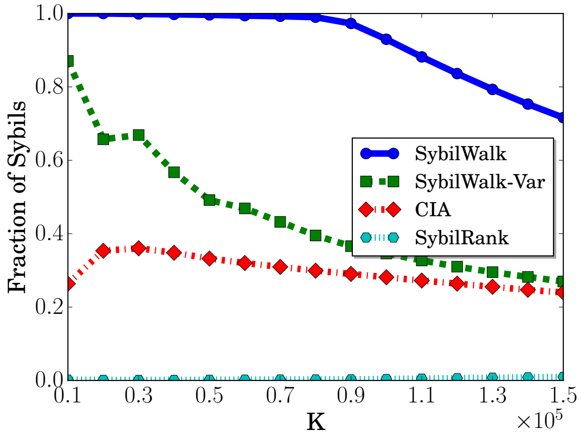


Fig. 4: Fraction of Sybils in top-K ranked nodes.

TABLE II: AUCs, FPRs, and FNRs on the Twitter dataset.

	SybilWalk	SybilWalk-Var	CIA	SybilRank
AUC	0.96	0.92	0.82	0.52
FPR	1.3%	4.8%	N/A	N/A
FNR	17.3%	31.1%	N/A	N/A

AUCs, FPRs, and FNRs on the Twitter network with real Sybils: Table II shows the results on the Twitter dataset. We run SybilWalk for two iterations. Except measuring the ranking quality, we also show the classification results. In particular, for SybilWalk and SybilWalk-Var, we classify a node to be a Sybil if and only if its badness score is larger than 0.5. False Positive Rate (FPR) is the fraction of testing benign nodes that are classified as Sybils, and False Negative Rate (FNR) is the fraction of testing Sybils that are classified as benign. Note that CIA and SybilRank are not classification methods, so they do not have FPR and FNR results.

Our results are consistent with those on the social networks with synthesized Sybils. Specifically, our methods substantially outperform CIA and SybilRank, and SybilWalk outperforms SybilWalk-Var. The reason is that the Twitter network has a weak homophily (i.e., a large number of attack edges, compared to the edges in the benign region and Sybil region), and our methods take advantage of both labeled benign nodes and labeled Sybils to tolerate a weak homophily.

To better illustrate the ranking quality, Fig. 4 shows the fraction of Sybils in top-K ranked nodes in the ranking list produced by each method, where we vary K from 10,000 to 150,000 with a step size of 10,000. We observe that our methods can accurately detect top-ranked Sybils. Specifically, 99% of the top-80,000 nodes produced by SybilWalk are Sybils. However, only 29.9% and 0.27% of the top-80,000 nodes produced by CIA and SybilRank are Sybils, respectively.

C. Robustness to Label Noise

A labeled node has a noisy label if the given label does not match its true label. Label noises often arise in practice due to human mistakes. We say the training dataset has α% of label noise if α% of labeled nodes have noisy labels. Specifically, in our experiments, we sample α% of labeled benign nodes and change their labels to be Sybil, and we sample α% of labeled Sybil nodes and change their labels to be benign. Fig. 5 shows AUCs of the compared methods on the Facebook dataset as we increase the label noises α%. Note that, in order to avoid the influence of weak homophily, we set the number of attack edges to be small (i.e., 500) such that all methods achieve AUCs close to 1 if there are no label noises.

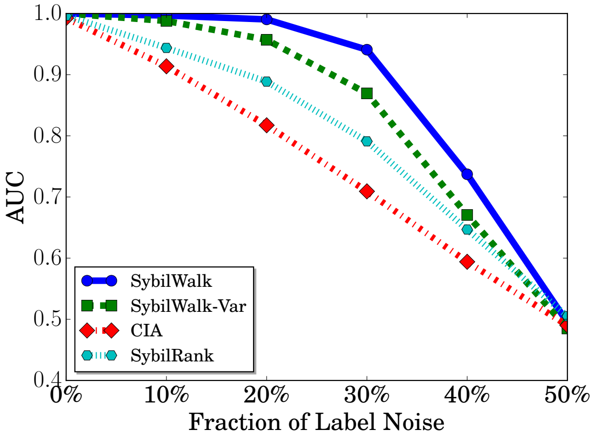


Fig. 5: AUCs of compared methods on the Facebook dataset as we increase the level of label noises.

First, our methods are more robust to label noises than CIA and SybilRank. The reason is that our methods incorporate both labels in the training dataset. Second, SybilWalk is more robust to label noise than SybilWalk-Var. Specifically, SybilWalk achieves AUCs close to 1 when fraction of label noises is upto 20%, while SybilWalk-Var can tolerate label noises upto 10%. The reason is that SybilWalk-Var fixes the badness scores of the labeled nodes, so the incorrect badness scores of the labeled nodes with noisy labels keep spreading among the social network. In contrast, SybilWalk does not fix the badness scores of labeled nodes, and the noisy labels could be corrected when iteratively computing the badness scores. Third, when 50% of labeled nodes have noisy labels, all methods achieve AUCs that are close to 0.5, i.e., all methods rank the test nodes uniformly at random. This is because 50% of label noise essentially means the training dataset is not informative.

D. Scalability

We evaluate scalability in terms of the time used by each method. Since evaluating scalability requires social networks with varying number of edges, we evaluate scalability on synthesized graphs with different number of edges. In particular, we add edges to the Facebook dataset randomly.

Fig. 6 shows the running times of the compared methods for different number of edges. Note that all these methods are iterative algorithms, so their running times highly depend on the number of iterations. To avoid bias introduced by the number of iterations, we run these methods for the same number of iterations, i.e., 20 in our experiments. All methods have linear time complexity, which is consistent with our

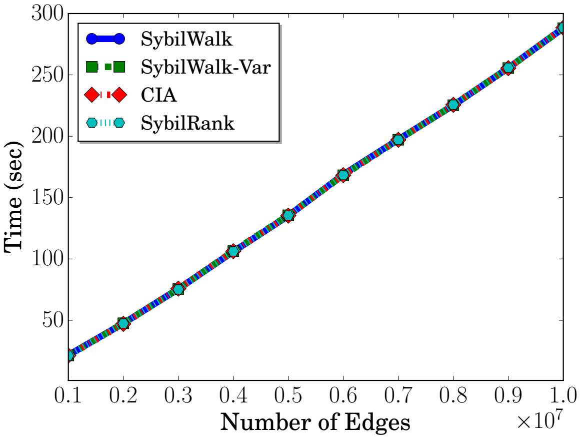


Fig. 6: Running times of compared methods on synthesized graphs as we increase the number of edges.

theoretical analysis in Section V-B. Moreover, SybilWalk is as scalable as previous random walk based methods.

E. Summary

- SybilWalk can tolerate a weaker homophily and is more robust to label noises than existing random walk based methods, while having the same scalability as existing random walk based methods.
- SybilWalk can tolerate a weaker homophily and is more robust to label noises than SybilWalk-Var. Moreover, they have the same scalability.

VII. Conclusion and Future Work

In this work, we design and evaluate SybilWalk, a new random walk based Sybil detection method. SybilWalk overcomes the limitations of existing random walk based methods while maintaining their advantages. The key technique of SybilWalk is to capture the structural gap between benign nodes and Sybil nodes through a random walk on a *label-augmented social network*. Theoretically, we demonstrate that SybilWalk achieves a tighter bound on the number of Sybils that are ranked lower than certain benign nodes than all existing random walk based methods. Empirically, we show that 1) SybilWalk can tolerate a weaker homophily than existing random walk based methods, 2) SybilWalk is more robust to label noises than existing random walk based methods, and 3) SybilWalk is as scalable as the most efficient existing random walk based methods.

Interesting future work includes 1) learning the edge weights in the label-augmented social network, 2) analyzing the bound of the number of falsely rejected benign nodes, and 3) generalizing our theoretical analysis to Markov Random Fields based methods.

References

[1] Facebook User Stat., May 2016.

[2] Facebook Popularity., May 2016.

[3] Hacking Election., May 2016.

[4] Hacking Financial Market., May 2016.

[5] Kurt Thomas, Chris Grier, Justin Ma, Vern Paxson, and Dawn Song. Design and evaluation of a real-time url spam filtering service. In *IEEE Symposium on Security and Privacy (IEEE S & P)*, 2011.

[6] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. SybilGuard: Defending against Sybil attacks via social networks. In *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM)*, 2006.

[7] 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 (IEEE S & P)*, 2008.

[8] G. Danezis and P. Mittal. SybilInfer: Detecting Sybil nodes using social networks. In *Network and Distributed System Security Symposium (NDSS)*, 2009.

[9] Abedelaziz Mohaisen, Nicholas Hopper, and Yongdae Kim. Keep your friends close: Incorporating trust into social network-based sybil defenses. In *IEEE International Conference on Computer Communications (INFOCOM)*, 2011.

[10] Qiang Cao, Michael Sirivianos, Xiaowei Yang, and Tiago Pregueiro. Aiding the detection of fake accounts in large scale social online services. In *Symposium on Network System Design and Implementation (NSDI)*, 2012.

[11] Chao Yang, Robert Harkreader, Jialong Zhang, Seungwon Shin, and Guofei Gu. Analyzing spammer's social networks for fun and profit. In *World Wide Web (WWW)*, 2012.

[12] Yazan Boshmaf, Dionysios Logothetis, Georgos Siganos, Jorge Leria, Jose Lorenzo, Matei Ripeanu, and Konstantin Beznosov. Integro: Leveraging victim prediction for robust fake account detection in osns. In *Network and Distributed System Security Symposium (NDSS)*, 2014.

[13] Yushan Liu, Shouling Ji, and Prateek Mittal. Smartwalk: Enhancing social network security via adaptive random walks. In *ACM Conference on Computer and Communications Security (CCS)*, 2016.

[14] Neil Zhenqiang Gong, Mario Frank, and Prateek Mittal. Sybilbelief: A semi-supervised learning approach for structure-based sybil detection. *IEEE TIFS*, 9(6), 2014.

[15] Peng Gao, Neil Zhenqiang Gong, Sanjeev Kulkarni, Kurt Thomas, and Prateek Mittal. Sybilframe: A defense-in-depth framework for structure-based sybil detection. *CoRR*, 2015.

[16] Hao Fu, Xing Xie, Yong Rui, Neil Zhenqiang Gong, Guangzhong Sun, and Enhong Chen. Robust spammer detection in microblogs: Leveraging user carefulness. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2017.

[17] Binghui Wang, Le Zhang, and Neil Zhenqiang Gong. Sybilscar: Sybil detection in online social networks via local rule based propagation. In *INFOCOM*, 2017.

[18] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. 1988.

[19] Alex Hai Wang. Don't follow me - spam detection in twitter. In *International Conference on Security and Cryptography (SECRYPT)*, 2010.

[20] G. Schoenebeck S. Yardi, D. Romero and D. Boyd. Detecting spam in a Twitter network. *First Monday*, 15(1), 2010.

[21] Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. Detecting spammers on social networks. In *Annual Computer Security Applications Conference (ACSAC)*, 2010.

[22] Fabrıcio Benevenuto, Gabriel Magno, Tiago Rodrigues, and Virgılio Almeida. Detecting spammers on twitter. In *Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS)*, 2010.

[23] Kurt Thomas, Chris Grier, and Vern Paxson. Adapting social spam infrastructure for political censorship. In *USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, 2012.

[24] Hongyu Gao, Yan Chen, Kathy Lee, Diana Palsetia, and Alok Choudhary. Towards online spam filtering in social networks. In *Network and Distributed System Security Symposium (NDSS)*, 2012.

[25] Kurt Thomas, Chris Grier, Vern Paxson, and Dawn Song. Suspended accounts in retrospect: An analysis of twitter spam. In *Internet Measurement Conference (IMC)*, 2011.

[26] Gang Wang, Tristan Konolige, Christo Wilson, and Xiao Wang. You are how you click: Clickstream analysis for sybil detection. In *Usenix Security*, 2013.

[27] Jonghyuk Song, Sangho Lee, and Jong Kim. Spam filtering in Twitter using sender-receiver relationship. In *International Symposium on Recent Advances in Intrusion Detection (RAID)*, 2011.

[28] Zhi Yang, Christo Wilson, Xiao Wang, Tingting Gao, Ben Y. Zhao, and Yafei Dai. Uncovering social network Sybils in the wild. In *Internet Measurement Conference (IMC)*, 2011.

[29] Gang Wang, Manish Mohanlal, Christo Wilson, Xiao Wang, Miriam Metzger, Haitao Zheng, and Ben Y. Zhao. Social turing tests: Crowdsourcing Sybil detection. In *Network and Distributed System Security Symposium (NDSS)*, 2013.

[30] G. Danezis, C. Diaz, C. Troncoso, and B. Laurie. Drac: An architecture for anonymous low-volume communications. In *Privacy Enhancing Technologies Symposium (PETS)*, 2010.

[31] Christo Wilson, Bryce Boe, Alessandra Sala, Krishna P.N. Puttaswamy, and Ben Y. Zhao. User interactions in social networks and their implications. In *Eurosys*, 2009.

[32] Eric Gilbert and Karrie Karahalios. Predicting tie strength with social media. In *ACM Conference on Human Factors in Computing Systems (CHI)*, 2009.

[33] Wei Wei, Fengyuan Xu, C.C. Tan, and Qun Li. SybilDefender: Defend against Sybil attacks in large social networks. In *IEEE International Conference on Computer Communications (INFOCOM)*, 2012.

[34] Yufeng Wang and Akihiro Nakao. Poisonedwater: An improved approach for accurate reputation ranking in p2p networks. *Future Generation Computer Systems (FGCS)*, 26(8):1317–1326, 2010.

[35] Lorenzo Alvisi, Allen Clement, Alessandro Epasto, Silvio Lattanzi, and Alessandro Panconesi. Sok: The evolution of sybil defense via social networks. In *IEEE Symposium on Security and Privacy (S & P)*, 2013.

[36] Abedelaziz Mohaisen, Aaram Yun, and Yongdae Kim. Measuring the mixing time of social graphs. In *Internet Measurement Conference (IMC)*, 2010.

[37] Neil Zhenqiang Gong, Ameet Talwalkar, Lester Mackey, Ling Huang, Eui Chul Richard Shin, Emil Stefanov, Elaine(Runting) Shi, and Dawn Song. Joint link prediction and attribute inference using a social-attribute network. *ACM TIST*, 5(2), 2014.

[38] Neil Zhenqiang Gong and Bin Liu. You are who you know and how you behave: Attribute inference attacks via users' social friends and behaviors. In *USENIX Security Symposium*, 2016.

[39] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *International Conference on Machine Learning (ICML)*, 2003.

[40] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286, 1999.

[41] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *World Wide Web*, 2010.

[42] Aaron Clauset, Cosma Rohilla Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *Society for Industrial and Applied Mathematics Review (SIAM Review)*, (51), 2009.

Appendix A
Proof of Theorem 1

We show the analysis about SybilWalk. Analysis for SybilWalk-Var is similar, so we omit it for simplicity.

Overview: Initially, Sybils have higher badness scores than benign nodes on average. In each iteration of SybilWalk, the average badness score of Sybil nodes decreases while the average badness scores of benign nodes increases. Our key idea is to derive the decrease of the average badness score of Sybil nodes and the increase of the average badness score of benign nodes in each iteration. Then, we can analyze the decrease of the average badness score of Sybil nodes and increase of the average badness score of benign nodes after a certain number of iterations. For a fast-mixing benign region, after $\log n$ iterations, benign nodes have similar badness scores. Suppose the decrease of the average badness scores of Sybil nodes all focus on a subset of Sybils. If we want this subset of Sybils to decrease badness scores to be smaller than benign nodes, then this subset of Sybils is bounded as $O(\frac{g \log n}{d(s)})$, where $d(s)$ is the average degree of Sybil nodes.

Notations: We first define some notations. $G = (V, E)$ denotes a social graph. For a node set N, we denote its *volume* as the sum of (weighted) degrees of nodes in N, i.e., $Vol(N) = \sum_{u \in N} d_u$, where d_u is the (weighted) degree of node u. Moreover, we define

$$C_b = \frac{g}{Vol(B)} \tag{3}$$

$$C_s = \frac{g}{Vol(S)}, \tag{4}$$

where B and S are the set of benign nodes and set of Sybil nodes, respectively.

We denote by $P_s^{(t)}$ as the average badness score of Sybil nodes in the tth iteration, and by $P_b^{(t)}$ the average badness score of benign nodes in the tth iteration. Initially, $P_s^{(0)}$ is larger than 0.5 while $P_b^{(0)}$ is smaller than 0.5. Furthermore, we denote by $D^{(t)}$ as the difference between the average badness score of benign nodes and that of Sybil nodes in the tth iteration. Formally, we have:

$$D^{(t)} \quad = P_b^{(t)} - P_s^{(t)}, \tag{5}$$

where $D^{(0)} < 0$ is the initial badness score difference. This difference comes from the initialized settings of badness scores for the label nodes and the labeled nodes in the training dataset. Note that, we assume there are no label noises.

Decrease of average badness score of Sybil nodes and increase of average badness score of benign nodes in the $(t+1)$th iteration: In the $(t+1)$th iteration, the expected average badness score of Sybil nodes and the expected average badness score of benign nodes can be approximated as follows:

$$P_s^{(t+1)} = \frac{g}{Vol(s)} P_b^{(t)} + \frac{Vol(s) - g}{Vol(s)} P_s^{(t)} \tag{6}$$

$$P_b^{(t+1)} = \frac{g}{Vol(b)} P_s^{(t)} + \frac{Vol(b) - g}{Vol(b)} P_b^{(t)}. \tag{7}$$

Therefore, we have:

$$\begin{align} & D^{(t+1)} \tag{8} \\ & = P_b^{(t+1)} - P_s^{(t+1)} \tag{9} \\ & = (1 - \frac{g}{Vol(b)} - \frac{g}{Vol(s)})(P_b^{(t)} - P_s^{(t)}) \tag{10} \\ & = (1 - \frac{g}{Vol(b)} - \frac{g}{Vol(s)})^{t+1} D^{(0)} \tag{11} \end{align}$$

Thus, the decrease of the average badness scores of Sybil nodes is as follows:

$$\begin{align} & P_s^{(t+1)} - P_s^{(t)} \tag{12} \\ & = \frac{g}{Vol(s)} (P_b^{(t)} - P_s^{(t)}) \tag{13} \\ & = (1 - \frac{g}{Vol(b)} - \frac{g}{Vol(s)})^t \frac{g}{Vol(s)} D^{(0)} \tag{14} \\ & = (1 - C_b - C_s)^t \times C_s D^{(0)}, \tag{15} \end{align}$$

where the above equation is negative (so we call is a decrease) because $D^{(0)}$ is negative. Therefore, we have:

$$\begin{align} & P_s^{(t)} - P_s^{(0)} \tag{16} \\ & = \sum_{i=0}^{t-1} (1 - C_b - C_s)^t \times C_s D^{(0)}, \tag{17} \end{align}$$

Similarly, the increase of the average badness scores of benign nodes is as follows:

$$\begin{align} & P_b^{(t+1)} - P_b^{(t)} \tag{18} \\ & = -\frac{g}{Vol(b)} (P_b^{(t)} - P_s^{(t)}) \tag{19} \\ & = -(1 - \frac{g}{Vol(b)} - \frac{g}{Vol(s)})^t \frac{g}{Vol(b)} D^{(0)} \tag{20} \\ & = -(1 - C_b - C_s)^t \times C_b D^{(0)}, \tag{21} \end{align}$$

where the above equation is positive (so we call it increase) because $D^{(0)}$ is negative. Furthermore, we have:

$$\begin{align} & P_b^{(t)} - P_b^{(0)} \tag{22} \\ & = -\sum_{i=0}^{t-1} (1 - C_b - C_s)^t \times C_b D^{(0)}, \tag{23} \end{align}$$

Ranking analysis: We assume after w iterations, benign nodes have similar badness scores, which are the average badness score of benign nodes. For a fast-mixing benign region, $w = O(\log n)$. Suppose we have n_s Sybils. After $w = O(\log n)$ iterations, we assume the decrease of badness score of Sybil nodes all focus on n_{ss} Sybils, which gives an upper bound of Sybils whose badness scores are smaller than benign nodes. If we want these Sybil nodes to have badness scores that are smaller than benign nodes, then we have:

$$\begin{align} & \frac{(P_s^{(w)} - P_s^{(0)}) n_s}{n_{ss}} < P_b^{(w)} - P_s^{(0)} \tag{24} \\ & \iff n_{ss} < \frac{(P_s^{(w)} - P_s^{(0)}) n_s}{P_b^{(w)} - P_s^{(0)}} \tag{25} \end{align}$$

Moreover, we have:

$$\begin{align} & \frac{(P_s^{(w)} - P_s^{(0)}) n_s}{P_b^{(w)} - P_s^{(0)}} \tag{26} \\ & = \frac{(P_s^{(w)} - P_s^{(0)}) n_s}{P_b^{(w)} - P_b^{(0)} + P_b^{(0)} - P_s^{(0)}} \tag{27} \\ & = \frac{\sum_{0 \le t \le (w-1)} (1 - C_s - C_b)^t C_s D^{(0)} n_s}{(1 - \sum_{0 \le t \le (w-1)} (1 - C_s - C_b)^t C_b) D^{(0)}} \tag{28} \\ & < \frac{\sum_{0 \le t \le (w-1)} (1 - C_s)^t C_s D^{(0)} n_s}{(1 - \sum_{0 \le t \le (w-1)} (1 - C_s - C_b)^t C_b) D^{(0)}} \tag{29} \\ & = \frac{(1 - (1 - C_s)^w) n_s}{1 - \frac{1 - (1 - C_s - C_b)^w}{C_s + C_b} C_b} \tag{30} \\ & < \frac{C_s w n_s}{1 - \frac{1 - (1 - (C_s + C_b) w + \frac{w(w-1)}{2} (C_s + C_b)^2)}{C_s + C_b} C_b} \tag{31} \\ & \approx \frac{gw}{d(s)(1 - (w - \frac{w^2}{2} C_b) C_b)} \tag{32} \\ & \approx \frac{gw}{d(s)(\frac{1}{2} + \frac{1}{2}(w C_b - 1)^2)} \tag{33} \\ & \le \frac{2gw}{d(s)}, \tag{34} \end{align}$$

where $d(s)$ is the average node degree of Sybils. Setting $w = O(\log n)$, we have:

$$n_{ss} = O(\frac{g \log n}{d(s)}). \tag{35}$$